Republic of Iraq

Ministry of Higher Education
University of Babylon

College of Science for Women Department of
Computer Science

Examination Committee System

Graduation Project Submitted to the Department of Computer Science/College of Science for Women/Babylon University as a Part of obtaining a Bachelor's Degree in Science

by

Noor Hassan Al-kniney

Supervisor by

Lecturer. Hadeel Qasem Gheni

2024 AD **1445 AH**

بِسْمِ اللَّهِ الرحمن الرَّحِيمِ

﴿ اللَّهُ نُورُ السَّمَاوَاتِ وَالْأَرْضِ مَثَلُ نُورِهِ كَمِشْكَاةٍ فِيهَا مِصْبَاحٌ الْمِصْبَاحُ فِي زُجَاجَةٍ الزُّجَاجَةُ كَأَنَّهَا كَوْكَبٌ دُرِّيٌّ يُوقَدُ مِن شَجَرَةٍ مُبَارَكَةٍ زَيْتُونَةٍ لَا شَرْقِيَّةٍ وَلَا غَرْبِيَّةٍ يَكَادُ زَيْتُهَا يُضِيءُ وَلَوْ لَمْ تَمْسَسْهُ نَارٌ نُورٌ عَلَىٰ نُورٍ يَهْدِي اللَّهُ لِنُورِهِ مَن يَشَاءُ وَيَضْرِبُ اللَّهُ الْأَمْثَالَ لِلنَّاسِ وَاللَّهُ بِكُلِّ شَيْءٍ عَلِيمٌ ﴾

صدق الله العلي العظيم

اية 35/ سورة النّور

الإهـــداء

أولاً وقبل كل شيء ، أود أن أعبر عن امتناني العميق لله تعالى الذي منحني القوة والثقة لإكمال هذا المسار التعليمي.

ثانياً ، أود أن أشكر عائلتي الرائعة والدي ووالدتي على دعمهم المستمر في كل خطوة من خطوات رحلتي.

ثالثاً ، أود أن أشكر جميع الأصدقاء الذين كانوا دائمًا موجودين لتشجيعي ودعمي

في الأوقات الصعبة.

رابعاً ، أود أن أشكر جميع الأساتذة المحترفين والمخلصين الذين ساهموا بشكل

كبير في تزويدي بالمعارف والمهارات.

ونخص بالجزيل الشكر والعرفان إلى كل من أشعل شمعة في دروب عملنا وإلى من وقف على المنابر وأعطى من حصيلة فكره لينير دربنا

إلى الأساتذة الكرام في كليتي ونتوجه بالشكر الجزيل إلى الدكتورة

هديل قاسم غني

الذي تفضلها بإشرافها على هذا البحث فجزاها الله عنها كل خير ولها منا كل التقدير والاحترام...

List of Contents

List of figures

List of Abbreviations

Abbreviations	Explain
(CLI)	Common Language Infrastructure
(VBA)	Visual Basic for Applications
(DBMS)	database management system
(**DAO**)	Data Access Objects
VB	Visual Basic

Abstract

The folds of our research, depends on the design and programming of the student grade and information system project, which contains an unlimited number of students' names and grades stored in the program's database and divided into tables. Those programs and electronic systems can make using from papers and avoid the usual routine and move to speed Keeping pace with technology and retrieving students’ grades with accuracy and high speed through programming, communication, processing stored data, matching each word in the way it fits and what is stored in it in the table that the user previously entered into the comprehensive system, saves us time and does not consume much effort in searching for grades and student names and can develop capabilities And the capabilities of the system and its use in all colleges because it is programmed in an open source way as it provides continuous updates to the system and interfaces that suit it more.

CHAPTER ONE

Chapter One

General Introduction

1.1 Introduction:

Rapid development in recent years in technology and software has facilitated programming processes, has entered into everything in terms of management and other information, and databases have become necessary in all places, especially the basic ones in universities and colleges, and we see that the old method of transactions has become boring and routine in terms of recording Students and their entry grades of study subjects and the issuance of the graduation document for students or other information related to it and print the document for each student based on the student number, which is the main key to searching for and retrieval of information.[1].

The system consists of a number of visual interfaces, and each interface has a specific role. The interface saves detailed information about each student and his records in the database. The program depends on data entry by the user, which is flexible, easy, and modifiable, and this is what makes us avoid the old way of registration, pens, and writing in records and papers and keep pace with the development taking place in the whole world [2].

1.2 Benefit from the project:

The main benefit of the project is the development of the practical reality in managing student grades as well as the time taken to complete the results and grades of students and create a database of their own.

1.3 Project Objectives

1- Create integrated databases Degree of Students.

2- The speed of displaying the degree for each student.

3- Development from the practical reality of most College

4- Teaching the student to make various programs

5- Establishing an integrated system connected to the database.

1.4 Research problem:

The project, which was designed and programmed to solve many of the problems that the Issuing a student document suffers from in colleges and universities, as it calculates the exam scores for students as well as the problem of issuing a grade document for each student, where the system issues and prints a complete graduation document for the student It also solves the problem of searching for a student with ease, as the old work in the Issuing a student documents done through paper files, while the system, once the student's number is entered, retrieves the student's data quickly and efficiently.

CHAPTER TWO

Chapter Two

Theoretical Background

2. Introduction

Before proceeding to explain, the details of the project must be clarified the Programs and techniques used in the construction of the project, On the one hand, the database and Interfaces.

2.1 Visual Basic dot net (VB.Net)

Visual Basic .NET (**VB.NET**) is a multi-paradigm, object-oriented programming language, implemented on the .NET Framework. Microsoft launched VB.NET in 2002 as the successor to its original Visual Basic language. Although the ".NET" portion of the name was dropped in 2005, this article uses "Visual Basic [.NET]" to refer to all Visual Basic language releases since 2002, in order to distinguish between them and the classic Visual Basic. Along with Visual C#, it is one of the two main languages targeting the .NET framework. Microsoft's integrated development environment (IDE) for developing in Visual Basic .NET language is Visual Studio. Most Visual Studio editions are commercial; the only exceptions are Visual Studio Express and Visual Studio Community, which are freeware. In addition, the .NET Framework SDK includes a freeware command-line compiler called vbc.exe. Mono also includes a command-line VB.NET compiler.**[5]**

VB.NET uses statements to specify actions. The most common statement is an expression statement, consisting of an expression to be evaluated, on a single line. As part of that evaluation,

In addition, in Visual Basic .NET:

- There is no unified way of defining blocks of statements. Instead, certain keywords, such as "If … Then" or "Sub" are interpreted as starters of sub-blocks of code and have matching termination keywords such as "End If" or "End Sub".
- Statements are terminated either with a colon (":") or with the end of line. Multiple line statements in Visual Basic .NET are enabled with " _" at the end of each such line. The need for the underscore continuation character was largely removed in version 10 and later versions.
- The equals sign ("=") is used in both assigning values to variables and in comparison.[8]
- Round brackets (parentheses) are used with arrays, both to declare them and to get a value at a given index in one of them. Visual Basic .NET uses round brackets to define the parameters of subroutines or functions.
- A single quotation mark ('), placed at the beginning of a line or after any number of space or tab characters at the beginning of a line, or after other code on a line, indicates that the (remainder of the) line is a comment.[7]

2.3. MS Access:

Microsoft Access is a database management system (DBMS) from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. It is a member of the Microsoft Office suite of applications, included in the Professional and higher editions or sold separately. Microsoft Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other applications and databases. Software developers, data architects and power users can use Microsoft Access to develop application software. Like other Microsoft Office applications, Access is supported by Visual Basic for Applications (VBA), an object-based programming language that can reference a variety of objects including DAO (Data Access Objects), ActiveX

Data Objects, and many other ActiveX components. Visual objects used in forms and reports expose their methods and properties in the VBA programming environment, and VBA code modules may declare and call Windows operating system operations.

Microsoft's first attempt to sell a relational database product was during the mid-1980s, when Microsoft obtained the license to sell R: Base In the late 1980s Microsoft developed its own solution codenamed Omega. It was confirmed in 1988 that a database product for Windows and OS/2 was in development [6].

It was going to include the "EB" Embedded Basic language which was going to be the language for writing macros in all Microsoft applications but the unification of macro languages did not happen until the introduction of Visual Basic for Applications (VBA). Omega was also expected to provide a front end to the Microsoft SQL Server. The application was very resource-hungry, and there were reports that it was working slowly on the 386 processors that were available at the time. It was scheduled to be released in the 1st quarter of 1990, but in 1989 the development of the product was reset and it was rescheduled to be delivered no sooner than in January 1991. Parts of the project were later used for other Microsoft projects: Cirrus (codename for Access) and Thunder (codename for Visual Basic, where the Embedded Basic engine was used After Access's premiere, the Omega project was demonstrated in 1992 to several journalists and included features that were not available in Access.[10]

2.4 Database:

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS). Together, the data and

the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data [11].

2.4.1 Structured Query Language (SQL)

SQL is a programming language used by nearly all relational databases to query, manipulate, and define data, and to provide access control. SQL was first developed at IBM in the 1970s with Oracle as a major contributor, which led to implementation of the SQL ANSI standard, SQL has spurred many extensions from companies such as IBM, Oracle, and Microsoft. Although SQL is still widely used today, new programming languages are beginning to appear.

2.4.2 Evolution of the database

Databases have evolved dramatically since their inception in the early 1960s. Navigational databases such as the hierarchical database (which relied on a tree-like model and allowed only a one-to-many relationship), and the network database (a more flexible model that allowed multiple relationships), were the original systems used to store and manipulate data. Although simple, these early systems were inflexible. In the 1980s, relational databases became popular, followed by object-oriented databases in the 1990s. More recently, NoSQL databases came about as a response to the growth of the internet and the need for faster speed and processing of unstructured data. Today, cloud

databases and self-driving databases are breaking new ground when it comes to how data is collected, stored, managed, and utilized [12].

2.4.3 Types of Databases

There are many different types of databases. The best database for a specific organization depends on how the organization intends to use the data.

Relational Databases

Relational databases became dominant in the 1980s. Items in a relational database are organized as a set of tables with columns and rows. Relational database technology provides the most efficient and flexible way to access structured information.

Object-Oriented Databases

Information in an object-oriented database is represented in the form of objects, as in object-oriented programming.

Distributed Databases

A distributed database consists of two or more files located on different sites. The database may be stored on multiple computers, located in the same physical location, or scattered over different networks [12].

Data Warehouses

A central repository for data, a data warehouse is a type of database specifically designed for fast query and analysis.

NoSQL Databases

A NoSQL, or non_relational database, allows unstructured and semi_structured data to be stored and manipulated (in contrast to a relational database, which defines how all data inserted into the database must be composed). NoSQL databases grew popular as web applications became more common and more complex.

Graph Databases

- A graph database stores data in terms of entities and the relationships between entities.
- **OLTP databases.** An OLTP database is a speedy, analytic database designed for large numbers of transactions performed by multiple users.

These are only a few of the several dozen types of databases in use today. Other, less common databases are tailored to very specific scientific, financial, or other functions. In addition to the different database types, changes in technology development approaches and dramatic advances such as the cloud and automation are propelling databases in entirely new directions. Some of the latest databases include [11].

Open Source Databases

An open source database system is one whose source code is open source; such databases could be SQL or NoSQL databases.

Cloud Databases

A cloud database is a collection of data, either structured or unstructured, that resides on a private, public, or hybrid cloud computing platform. There are two types of cloud database models: traditional and database as a service (DBaaS). With DBaaS, administrative tasks and maintenance are performed by a service provider.

Multimodal Database

Multimodal databases combine different types of database models into a single, integrated back end. This means they can accommodate various data types.

Document/JSON Database

Designed for storing, retrieving, and managing document-oriented information, document databases are a modern way to store data in JSON format rather than rows and columns.

Self-Driving Databases

The newest and most groundbreaking type of database, self-driving databases (also known as autonomous databases) are cloud-based and use machine learning to automate database tuning, security, backups, updates, and other routine management tasks traditionally performed by database administrators.

2.4.5 Relational Database

A **relational database** is a digital database based on the relational model of data, as proposed by E. F. Codd in 1970.A system used to maintain relational databases is a **relational database management system** (**RDBMS**). Many

relational database systems have an option of using the SQL (Structured Query Language) for querying and maintaining the database [12].

In 1974, IBM began developing System R, a research project to develop a prototype RDBMS. The first system sold as an RDBMS was Multics Relational Data Store (June 1976). Oracle was released in 1979 by Relational Software, now Oracle Corporation. Ingres and IBM BS12 followed. Other examples of an RDBMS include DB2, SAP Sybase ASE, and Informix. In 1984, the first RDBMS for Macintosh began being developed, code-named Silver Surfer, and was released in 1987 as 4th Dimension and known today as 4D.

The first systems that were relatively faithful implementations of the relational model were from:

- University of Michigan – Micro DBMS (1969)
- Massachusetts Institute of Technology (1971)
- IBM UK Scientific Centre at Peterlee – IS1 (1970–72) and its successor, PRTV (1973–79)

The most common definition of an RDBMS is a product that presents a view of data as a collection of rows and columns, even if it is not based strictly upon relational theory. By this definition, RDBMS products typically implement some but not all of Codd's 12 rules [13].

A second school of thought argues that if a database does not implement all of Codd's rules (or the current understanding on the relational model, as expressed by Christopher J. Date, Hugh Darwen and others), it is not relational. This view, shared by many theorists and other strict adherents to Codd's principles, would disqualify most DBMSs as not relational. For clarification, they often refer to some RDBMSs as *truly-relational database management systems*

(TRDBMS), naming others *pseudo-relational database management systems* (PRDBMS).

As of 2009, most commercial relational DBMSs employ SQL as their query language.

Alternative query languages have been proposed and implemented, notably the pre-1996 implementation of Ingres QUEL [13].

Relational model

This model organizes data into one or more tables (or "relations") of columns and rows, with a unique key identifying each row. Rows are also called records or tuples. Columns are also called attributes. Generally, each table/relation represents one "entity type" (such as customer or product). The rows represent instances of that type of entity (such as "Lee" or "chair") and the columns represent values attributed to that instance (such as address or price).

For example, each row of a class table corresponds to a class, and a class corresponds to multiple students, so the relationship between the class table and the student table is "one to many.

Keys

Each row in a table has its own unique key. Rows in a table can be linked to rows in other tables by adding a column for the unique key of the linked row (such columns are known as foreign keys). Codd showed that data relationships of arbitrary complexity can be represented by a simple set of concepts.

Part of this processing involves consistently being able to select or modify one and only one row in a table. Therefore, most physical implementations have a

unique primary key (PK) for each row in a table. When a new row is written to the table, a new unique value for the primary key is generated; this is the key that the system uses primarily for accessing the table. System performance is optimized for PKs. Other, more natural keys may also be identified and defined as alternate keys (AK). Often several columns are needed to form an AK (this is one reason why a single integer column is usually made the PK). Both PKs and AKs have the ability to uniquely identify a row within a table. Additional technology may be applied to ensure a unique ID across the world, a globally unique identifier, when there are broader system requirements.

The primary keys within a database are used to define the relationships among the tables. When a PK migrates to another table, it becomes a foreign key in the other table. When each cell can contain only one value and the PK migrates into a regular entity table, this design pattern can represent either a one-to-one or one-to-many relationship. Most relational database designs resolve manyto-many relationships by creating an additional table that contains the PKs from both of the other entity tables – the relationship becomes an entity; the resolution table is then named appropriately and the two FKs are combined to form a PK. The migration of PKs to other tables is the second major reason why system-assigned integers are used normally as PKs; there is usually neither efficiency nor clarity in migrating a bunch of other types of columns [13].

CHAPTER THREE

Chapter Three
Methodology

3.1 Methodology

The project of generating a graduation document for students needs a database consisting of seven tables, the first for users who work on the system, the second table for adding students, while the third table is for saving grades for each student that is entered into the database that is entered through the data entry interface, and the fourth table It is for the notes that are written on the student, while the other tables are concerned with classifying and dividing students according to the academic stage.

The interfaces are also designed using Visual Basic language, where each interface performs the tasks for which it was programmed, such as the data display interface by guiding the user to retrieve and display data to manage the student document generation system for graduation, the search interface also for information that has been associated with the student's name so that it retrieves All grades for all stages of the student.

We also won't need to establish relational relationships between the table because we don't need them. The programming language was sufficient for the system to function accurately, and the link was programmatically via the primary key of each table.

In the following figure, the relationships between the tables and the names of the fields for each table are represented, as they are related to a one-to-one relationship

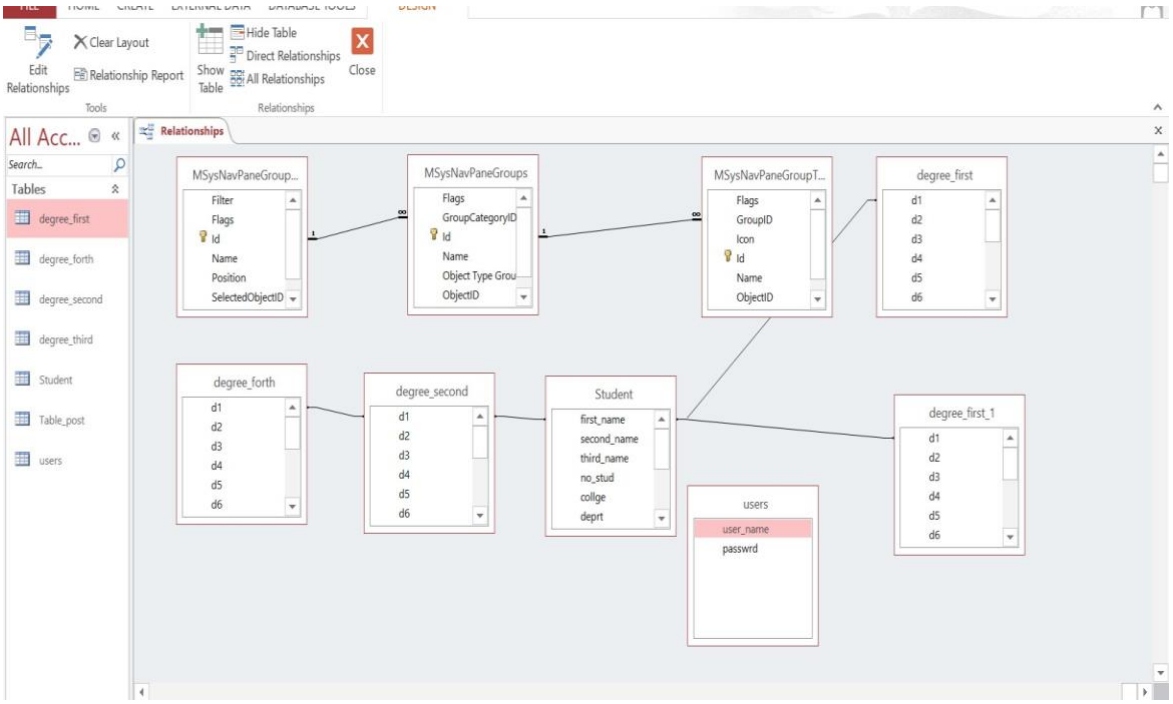

Figure (3-1) Relation Database

CHAPTER

FOUR

Chapter Four

Result and Discuses

4.1 Result

1. Start project when Run Project show form following:

Figure (3-1) main form

2. When press Login show the form following:

Figure (3-2) main menu

3. When you click on the Add Students button, the following interface appears, through which we enter the student's information that is stored in the table assigned to it within the database, as shown in the figure:

Figure (3-3) Add Student

4. When press on Add Course of information to the database as the form follows.

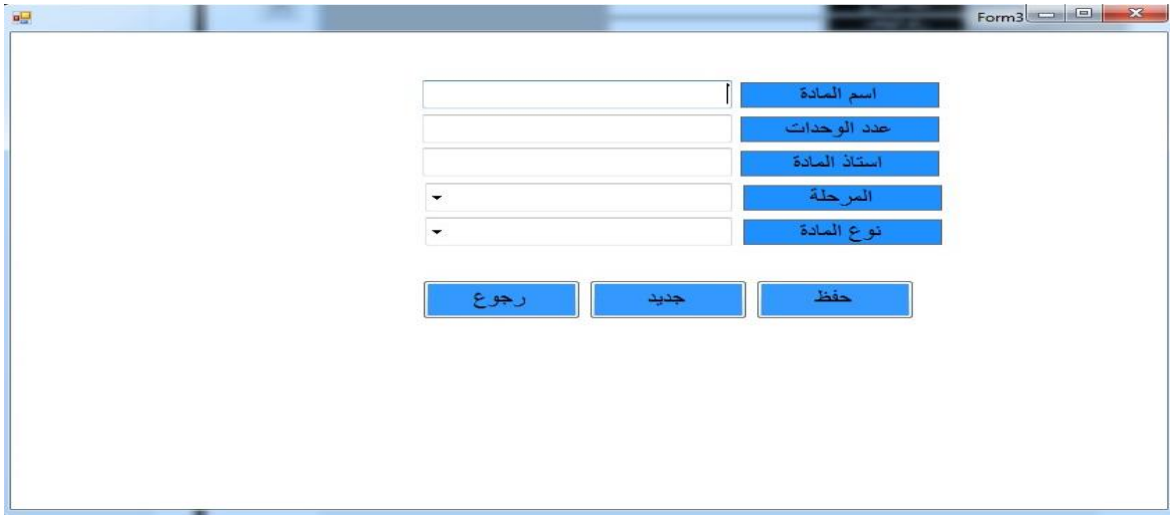

Figure (3-4) Add Note

5. When selecting the button for the grades, you will see the following interface through which we can add students ’grades for the first and second round as in the following figure:

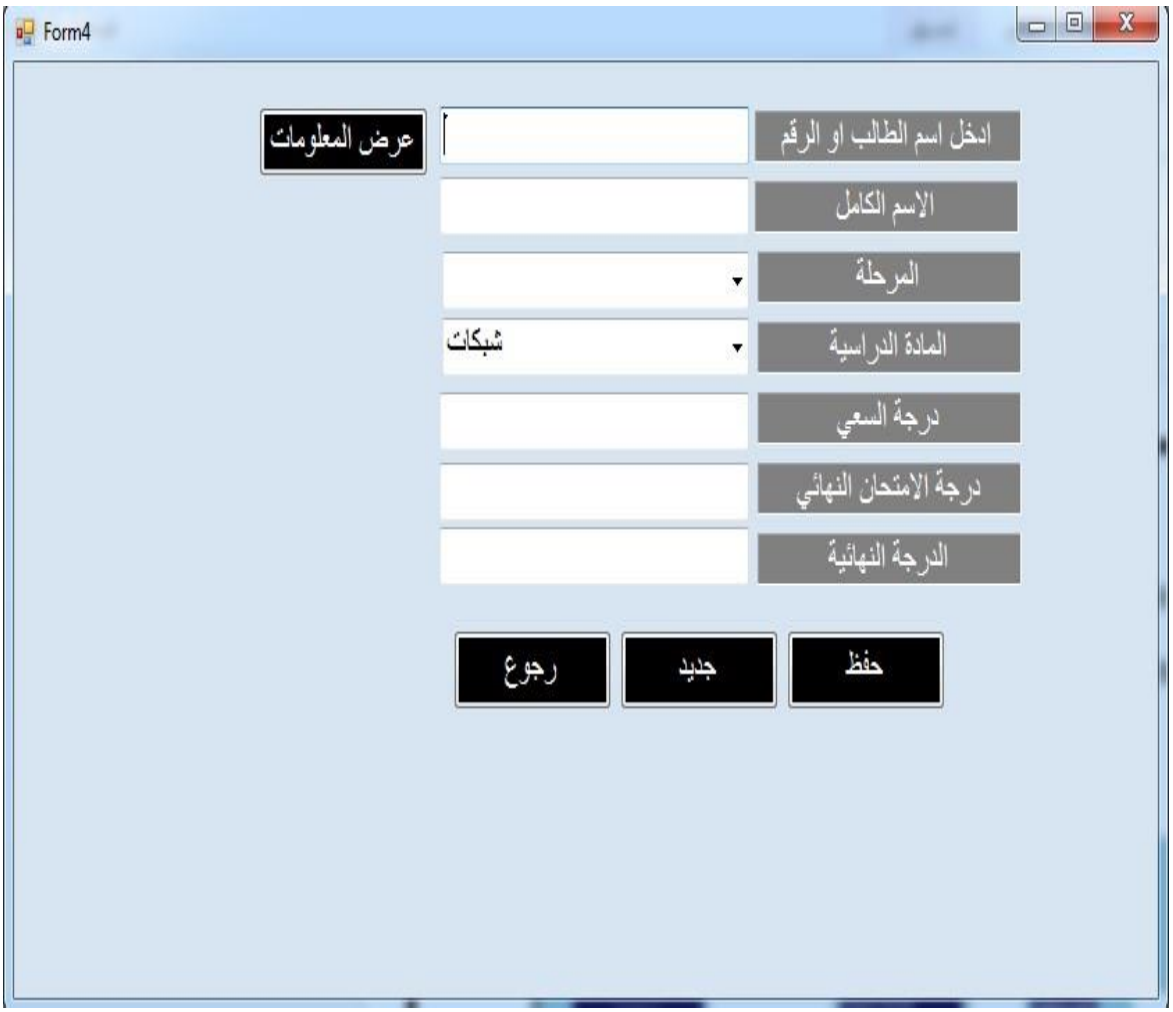

Figure (3-5) Add Degree

6. The issuance of Master Shit for the student's graduation is done by issuing the name of the student and after entering the student's university number, the details of graduation from the grades for four years appear, as in the following picture.

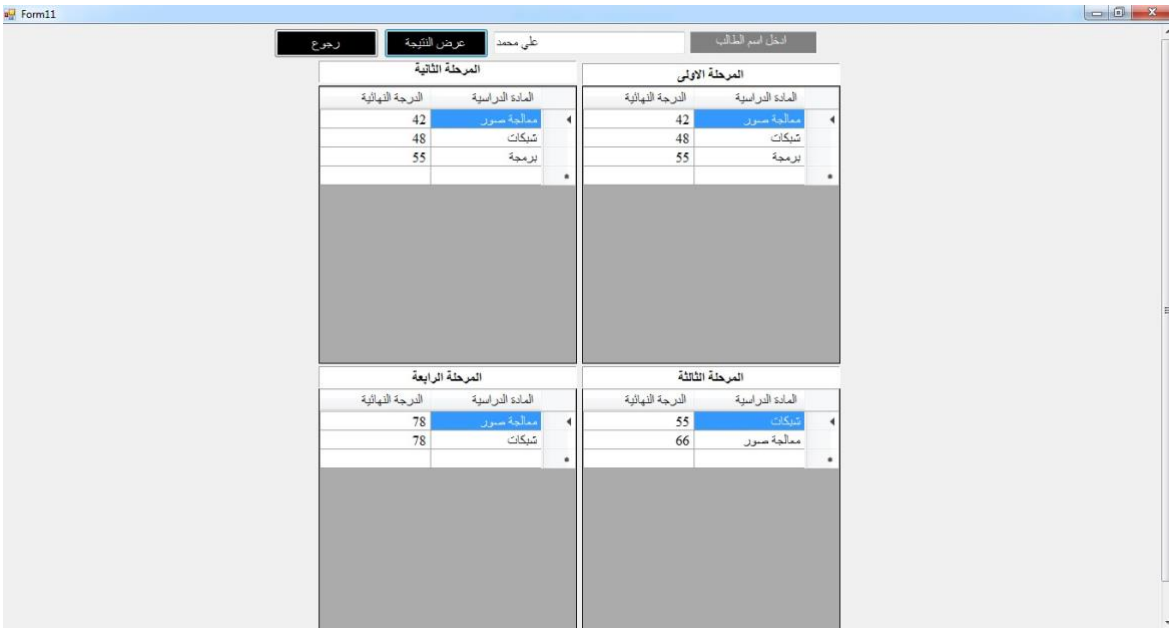

Figure (3-6) Document Result

4.2 Code project:

-code form 3 : add New

```
Imports System.Data.OleDb

Public Class Form3

    Private Sub Button3_Click(sender As Object, e As
EventArgs) Handles Button3.Click
        Form2.Show()
        Me.Hide()

    End Sub

    Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
TextBox8.Text = ""
    End Sub
Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
Try
Using conn As New
```

```
    OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Dat
    a Source=|DataDirectory|\database.mdb;")
    conn.Open()
    Dim command As New OleDbCommand("insert into
    patient(name_Patient,No_File,Date_first,Name_Doctor,Date
    _File,Age_patient,Adrss,phone,Gender)
    values(@d1,@d2,@d3,@d4,@d5,@d6,@d7,@d8,@d9)", conn)
With command.Parameters.AddWithValue("@d1",TextBox1.Text)
.AddWithValue("@d2", TextBox2.Text)
.AddWithValue("@d3", TextBox3.Text)
.AddWithValue("@d4", TextBox4.Text)
.AddWithValue("@d5", TextBox5.Text)
.AddWithValue("@d6", TextBox6.Text)
.AddWithValue("@d7", TextBox7.Text)
.AddWithValue("@d8", TextBox8.Text)
.AddWithValue("@d9", ComboBox1.Text)
```

```
End With
command.ExecuteNonQuery()
End Using
MessageBox.Show("Data Input Succssfull",
"INFO", MessageBoxButtons.OK,
MessageBoxIcon.Information)
Catch ex As Exception
End Try
End Sub
Private Sub DateTimePicker1_ValueChanged(sender As
Object, e As EventArgs) Handles
DateTimePicker1.ValueChanged
        TextBox3.Text = DateTimePicker1.Value
End Sub
Private Sub DateTimePicker2_ValueChanged(sender As
Object, e As EventArgs) Handles
DateTimePicker2.ValueChanged
        TextBox5.Text = DateTimePicker2.Value
End Sub
End Class

Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
Try
Using conn As New
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=|DataDirectory|\database.mdb;")
conn.Open()
Dim command As New OleDbCommand("insert into
```

```
patient(d1,d2,d3,d4,d5,d6,d7,d8,d9,10,d11,d12,d13,d14,d1
5,d16,d17,d18,d19,d20,d21)
values(@d1,@d2,@d3,@d4,@d5,@d6,@d7,@d8,@d9,@d10,@d11,@d1
2,@d13,@d14,@d15,@d16,@d17,@d18,@d19,@d20,@d21)", conn)
With command.Parameters
.AddWithValue("@d1", TextBox1.Text)
.AddWithValue("@d2", TextBox2.Text)
.AddWithValue("@d3", TextBox3.Text)
.AddWithValue("@d4", TextBox4.Text)
.AddWithValue("@d5", TextBox5.Text)
.AddWithValue("@d6", TextBox6.Text)
.AddWithValue("@d7", TextBox7.Text)
.AddWithValue("@d8", TextBox8.Text)
.AddWithValue("@d9", TextBox8.Text)
.AddWithValue("@d10", TextBox8.Text)
.AddWithValue("@d11", TextBox8.Text)
.AddWithValue("@d12", TextBox8.Text)
.AddWithValue("@d13", TextBox8.Text)
.AddWithValue("@d14", TextBox8.Text)
.AddWithValue("@d15", TextBox8.Text)
.AddWithValue("@d16", TextBox8.Text)
.AddWithValue("@d17", TextBox8.Text)
.AddWithValue("@d18", TextBox8.Text)
.AddWithValue("@d19", TextBox8.Text)
.AddWithValue("@d20", TextBox8.Text)
.AddWithValue("@d21", TextBox8.Text)
End With
command.ExecuteNonQuery()
```

```
End Using
MessageBox.Show("Data Input Succssfull",
"INFO", MessageBoxButtons.OK,
MessageBoxIcon.Information)
        Catch ex As Exception
        End Try
    End Sub

    Private Sub Button4_Click(sender As Object, e As
EventArgs) Handles Button4.Click
        Using conn As New
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=|DataDirectory|\database.mdb;")
conn.Open()

            Dim CoM As New OleDbCommand(Nothing, conn) '
            Dim dr As New OleDbDataAdapter(CoM)
            Dim ds As New DataSet
            Dim acscmd As New OleDbCommand
            CoM.CommandText = "select * FROM [patient]
WHERE [No_File]= " & "'" & TextBox21.Text & "'"
acscmd.Connection = conn
 'acscmd.CommandText = CommandType.Text
 acscmd.CommandText = CoM.CommandText
 Dim dr1 As OleDbDataReader =
 acscmd.ExecuteReader()
```

```
            If dr1.Read = True Then
ds.Clear()
                dr.Fill(ds) 'dataset تعبئة

                TextBox20.Text =
(ds.Tables(0).Rows(0)("name_Patient"))



            Else
                MessageBox.Show("not found", "تنبيه",
MessageBoxButtons.OK, MessageBoxIcon.Information)

            End If
            Err.Clear()
Exit Sub
conn.Close()         End
Using
    End Sub

    Private Sub Button3_Click(sender As Object, e As
EventArgs) Handles Button3.Click
        Form2.Show()
        Me.Hide()
    End Sub
End Class
```

- Code search

```
Imports System.Data.OleDb
```

```
Public Class Form5

    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        Using conn As New
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=|DataDirectory|\database.mdb;")
conn.Open()
```

```
            Dim CoM As New OleDbCommand(Nothing, conn) '
            Dim dr As New OleDbDataAdapter(CoM)
            Dim ds As New DataSet
            Dim acscmd As New OleDbCommand
            CoM.CommandText = "select * FROM [patient]
WHERE [No_File]= " & "'" & TextBox1.Text & "'"
acscmd.Connection = conn
            'acscmd.CommandText = CommandType.Text
acscmd.CommandText = CoM.CommandText             Dim
dr1 As OleDbDataReader = acscmd.ExecuteReader()

            If dr1.Read = True Then
ds.Clear()
                dr.Fill(ds) 'dataset تعبئة

                TextBox2.Text =
(ds.Tables(0).Rows(0)("name_Patient"))



            Else
                MessageBox.Show("not found", "تنبيه",
MessageBoxButtons.OK, MessageBoxIcon.Information)

            End If
            Err.Clear()

            Exit Sub
conn.Close()         End
Using
    End Sub
```

```
    Private Sub Button3_Click(sender As Object, e As
EventArgs) Handles Button3.Click
        Form2.Show()
        Me.Hide()

    End Sub

    Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
TextBox4.Text = ""

    End Sub
End Class
```

CHAPTER FIVE

Chapter Five

Conclusion and Future work

5.1 Conclusion

We conclude from the folds of our research that it is possible to make software systems for managing student registration, issuing their graduation document, processing their detailed data, displaying or searching for these data, and modifying and deleting them, as well as the system can be connected to a number of devices on a local network. Thus, it will be an integrated system that can be worked on by the registration of students or examination committees and others, and the data is stored in a database of unlimited capacity and thus is stored in a safe and not subject to damage or loss, as is the case in student paper files that are vulnerable to damage, electronic data is stored for periods Long and in a way that can be preserved.

5.2 Future Work:

We recommend that the system be further developed to include all faculties in the university.

1. Add student data encryption algorithms to protect it.
2. Developing the system to become online, and the student can submit a request to withdraw a graduation document.
3. Add the feature of converting the document language to other languages

Reference:

1- Bachman, Charles W. *(2019).* "The Programmer as Navigator".

*Communications of the ACM. **16** (11): 653–658.* doi:10.1145/355611.362534.

2- *Beynon-Davies, Paul (2018). Database Systems (3rd ed.). Palgrave Macmillan.* ISBN 978-1403916013.

3- *Chapple, Mike (2017).* "SQL Fundamentals". *Databases. About.com.* Archived *from the original on 22 February 2019*. Retrieved 28 January 2019.

4- Childs, David L. *(2020).* "Description of a set-theoretic data structure" (PDF). *CONCOMP (Research in Conversational Use of Computers) Project. Technical Report 3. University of Michigan.* 5- Childs, David L. *(2019).* "Feasibility of a set-theoretic data structure: a general structure based on a reconstituted definition" (PDF). *CONCOMP (Research in Conversational Use of Computers) Project. Technical Report 6. University of Michigan.*

6- *Chong, Raul F.; Wang, Xiaomei; Dang, Michael; Snow, Dwaine R. (2007).* "Introduction to DB2". *Understanding DB2: Learning Visually with Examples (2nd ed.).* ISBN 978-0131580183. Retrieved 17 March 2013.

7- *Krill, Paul* "Microsoft converging programming languages | Developer World". *InfoWorld*. Retrieved 2018-08-18.

8- *Sherriff, Lucy (22 February 2015).* "Real Software slams MS IsNot patent application". *The Register*.

9- *Connolly, Thomas M.; Begg, Carolyn E. (2014). Database Systems – A Practical Approach to Design Implementation and Management (6th ed.). Pearson.* ISBN 978-1292061184*.*

10- *Bebbington, Shaun (2016).* "What is coding"*. Tumblr.* Archived *from the original on April 29, 2020*. Retrieved March 3, 2014*.*

11- *Bebbington, Shaun (2015).* "What is programming"*. Tumblr.* Archived *from the original on April 29, 2020*. Retrieved March 3, 2014*.*

12- *Eilam, Eldad (2015). Reversing: secrets of reverseengineering. John Wiley & Sons.* ISBN 978-0-7645-7481-8*.*

13- *Koetsier, Teun (2017), "On the prehistory of programmable machines: musical automata, looms, calculators", Mechanism and Machine Theory, Elsevier,* ***36*** *(5): 589–603,* doi*:*10.1016/S0094114X(01)00005-2*.*