Design of Computerized Students Grades Database Management System

Heba Mohammed Fadhil[1,] Neran Eidan[2], Zainab Ahmed[3]

[1]Assistant Professor, [2,3]Research Scholar
Department of Information and Communication
Baghdad University, Baghdad, Iraq

Abstract— The end-of-course marks assigned by lecturers are anticipated to express the level of accomplishment of each student in the class. These marks are used to make a mass of decisions. Unless an adequately accurate and efficient method is used for the grading, the grades are appropriate to convey misinformation . In itself, the handing out of results is found to be rather tedious, especially when carried out manually, and when the total number of students is large. It is time-consuming and error- prone. However, the process carried out with a computer running a suitable software application becomes a lot easier and much more accurate. A spontaneous and systematic one elucidation is essential for all universities and organizations. With that in mind, we overhauled the existing Student Database Management System and made necessary improvement to streamline the processes.

In this paper, a computer software application was developed to facilitate the automated processing of the results. The software was developed in Visual Basic programming language as an interface and in the form of a database, SQL Relational Database Management System was used. The developed software is a useful easy user interface; furthermore, it is a controlling data management, data retrieval, and data manipulation. It will grant more ease for managing the data than manually maintaining the documents. This work is useful for saving valuable time and reduces the huge paperwork.

***Index Terms*— Database, Automated Processing, SQL, Software Application.**

I. Introduction

The fast-paced growth and development of technology have brought significant changes to the mindsets, workflows, and lifestyle of society. One of the fields most impacted by this progress is education. Not only has technology become one of the most striking fields of study in tertiary education, but it's persuade and use within the educational institutions has become essential [1] [2].

Data management is the core of every organization that wishes to succeed in this information age. Educational institutions, especially education institutions, require a data management system in order to function properly since accurate information is crucial for each student's academic record [2].

Such orientation is complicated for organizations but most mid-size and large organization has made it nowadays and uses it as a competitive weapon and gain advantages from it like a relational model and data warehouse. A database management system (DBMS) is an aggregate of data, hardware, software, and users that help an enterprise manage its operational data .The main function of a DBMS is to provide efficient and reliable methods of data retrieval to many users [3][4].DBMS as software designed to assist in maintaining and utilizing large collections of data.

Several advantages of a DBMS include:

(1) Efficient data access.
(2) Data integrity and security
(3) Data administration
(4) Concurrent access and crash recovery

DBMS provides a symmetric method of creating, updating, retrieving, data in a database. It enables end user and application programmers to share data and enable data to be shared among multiple application rather than propagated and accumulate in new files for every new application [1][5].

This paper is organized as follows: **section 2** presents the principles of database systems, concentrating on data abstraction, data independence, and efficient query processing, **section 3** describes the proposed Database system user interface, main page, and other pages functionalities, Furthermore, illustrate a sketch of the entire system that helps to view that all activities performed in the system, **section 4** presents the conclusion and suggestions for future work

II. DBMS ARCHITECTURE AND DATA INDEPENDENCE

The most important characteristics of the database advance are (1) insulation of programs and data (program-data and program-operation independence); (2) support of multiple user views; and (3) use of a catalog to store the database description (schema). The architecture for database systems, called the ***three-schema architecture*** will be discussed in this section, which helps achieve and visualize these characteristics [6].

i. THE THREE-SCHEMA ARCHITECTURE

To split the user applications and the physical database is the foremost goal of the three-schema architecture as illustrated in **(Fig. 1)**. In this architecture, schemas can be defined at the following three levels [4] [7]:
1. The **internal level** has an **internal schema,** which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.
2. The **conceptual level** has a **conceptual schema,** which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. A high-level data model or an implementation data model can be used at this level.
3. The **external** or **view level** includes a number of **external schemas** or **user views.**

Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group. A high-level data model or an implementation data model can be used at this level.

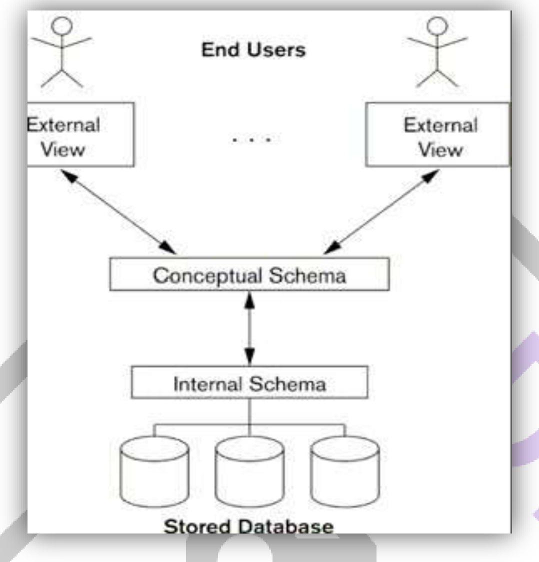


Figure 1. The Three Schema Architecture

To envisage the schema levels in a database system a convenient tool for the user is to use the three-schema architecture. Most DBMSs do not detach the three levels completely, but support the three-schema architecture to some extent. Some DBMSs may include physical-level details in the conceptual schema. In most DBMSs that support user views, external schemas are specified in the same data model that describes the conceptual-level information. Some DBMSs allow different data models to be used at the conceptual and external levels.
Notice that the three schemas are only descriptions of data; the only data that actually exists is at the physical level. In a DBMS based on the three-schema architecture, each user group refers only to its own external schema. Hence, the DBMS must transform a request specified on an external schema into a request against the conceptual schema, and then into a request on the internal schema for processing over the stored database. If the request is database retrieval, the data extracted from the stored database must be reformatted to match the user's external view. The processes of transforming requests and results between levels are called mappings**.** These mappings may be time- consuming, so some DBMSs—especially those that are meant to support small databases—do not support external views. Even in such systems, however, a certain amount of mapping is necessary to transform requests between the conceptual and internal levels [5] [6].

ii. DATA INDEPENDENCE

To explain the concept of **data independence** the three-schema architecture can be used**,** which can be defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level. We can define two types of data independence [4] [7]:
1. **Logical data independence** is the capacity to change the conceptual schema without having to change external schemas or application programs. We may change the conceptual schema to expand the database (by adding a record type or data item), or to reduce the database (by removing a record type or data item). In the latter case, external schemas that refer only to the remaining data should not be affected. Only the view definition and the mappings need be changed in a DBMS that supports logical data independence. Application programs that reference the external schema constructs must work as before, after the conceptual schema undergoes a logical reorganization. Changes to constraints can be applied also to the conceptual schema without affecting the external schemas or application programs.

2. **Physical data independence** is the capacity to change the internal schema without having to change the conceptual (or external) schemas. Changes to the internal schema may be needed because some physical files had to be reorganized—for example, by creating additional access structures—to improve the performance of retrieval or update. If the same data as before remains in the database, we should not have to change the conceptual schema.

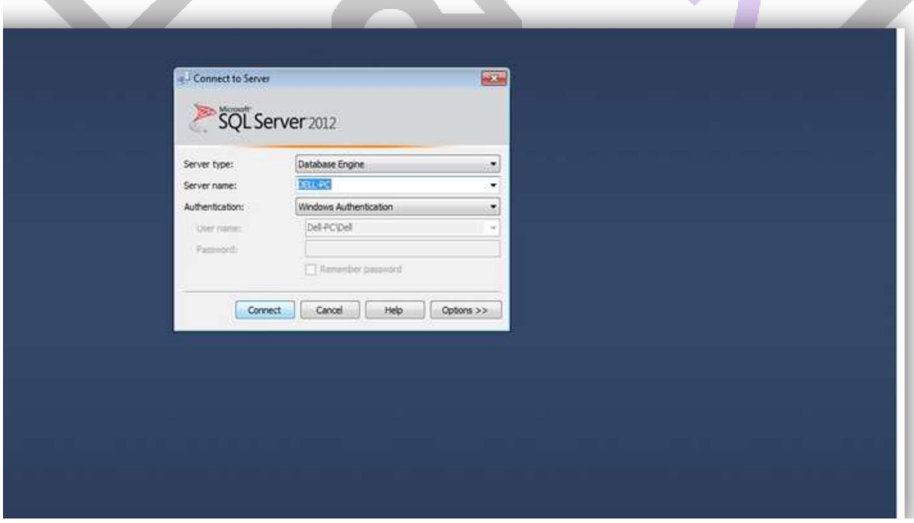
Whenever we have a multiple-level DBMS, its catalog must be expanded to include information on how to map requests and data among the various levels. The DBMS uses additional software to accomplish these mappings by referring to the mapping information in the catalog. Data independence is consummate for the reason that when the schema is altered at some level, the schema at the next higher level remains untouched; only the mapping between the two levels is changed. Hence, application programs referring to the higher-level schema need not be changed [1] [3].
True data independence can be achieved more easily by using the three-schema architecture, individually physical and logical. However, the two levels of mappings create an overhead during compilation or execution of a query or program, leading to inefficiencies in the DBMS. Because of this, few DBMSs have implemented the full three-schema architecture [2].

III. SYSTEM DESIGN

A system is a well thought-out collection of inter related subsystems with a communal responsibility of meeting a goal. A system also defined as an organized or complex unitary whole. System analysis is consequently the study of the system's processing including analysis of inputs and outputs in order to find better, more cost-effective and proficient means of processing data.
The principle of the system design is to efficiently divide the overall problems into small and more controllable problems that can be easily handled by separate program modules. The separate program modules will later be integrated forming the entire system. This design methodology is called "Top-Down-Design". The program modules that control the main logic of the system must rely on lower level modules to perform subordinate tasks reliably. In order to design a system, the relational database must be designed first.
Conceptual design can be divided into two parts: the ***data model*** and the ***process model.*** The data model focuses on what data should be stored in the database while the process model deals with how the data is processed. To put this in the context of the relational database, the data model is used to design the relational tables. The process model is used to design the queries that will access and perform operations on those tables.

i. PROGRAM DESIGN

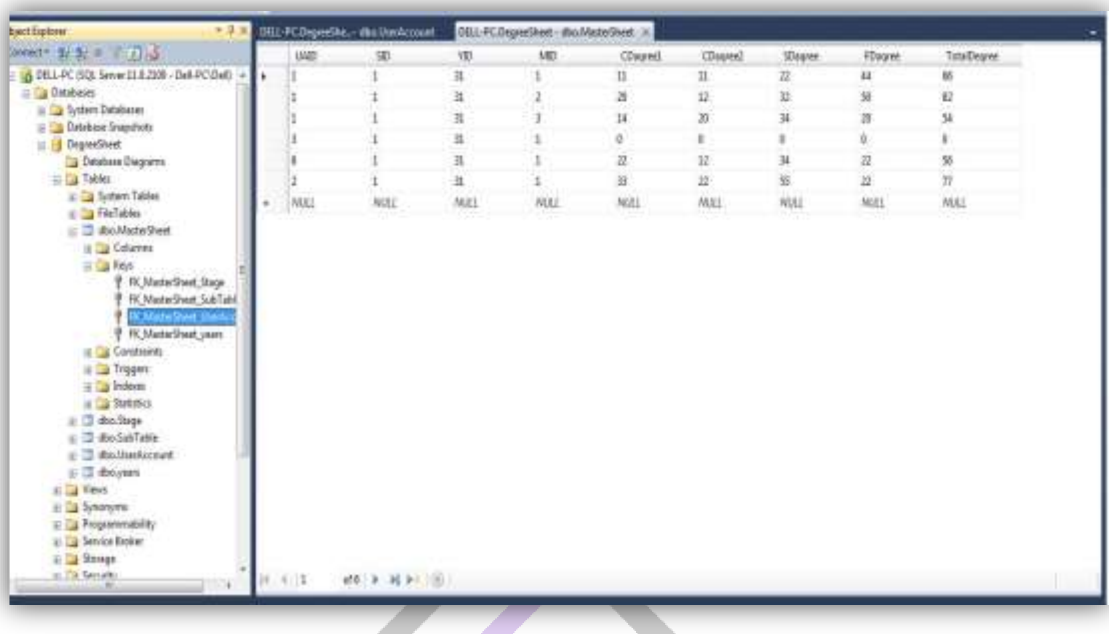
The program design does the work of identification of all modules of its software and the relationship that exists between them and also solution statement and coding. In **(Fig. 2)** below shows the front of the SQL server which is use to store data.



Server type: Database Engine
Server name: DELL-PC
Authentication: Windows Authentication
User name: Dell-PC\Dell
Password:
□ Remember password

Figure 2 .SQL SERVER

This application is used as a database to store the tables and this can be seen only by the designer. The database is **Degree Sheet** which include the tables we create .As for tables they are ***Master Sheet*** which are the primary table that is connected to other tables by foreign keys which are the primary keys in their tables .
In **(Fig.3)** the details of the Master Sheet are seen as columns which are (UAID ,SID ,YID ,MID ,CDagree1 ,CDagree2 ,SDagree ,FDagree , Totaldegree). All these columns are foreign key also sometimes called a referencing key.



DELL-PC.DegreeShe... - dbo.UserAccount | DELL-PC.DegreeSheet - dbo.MasterSheet

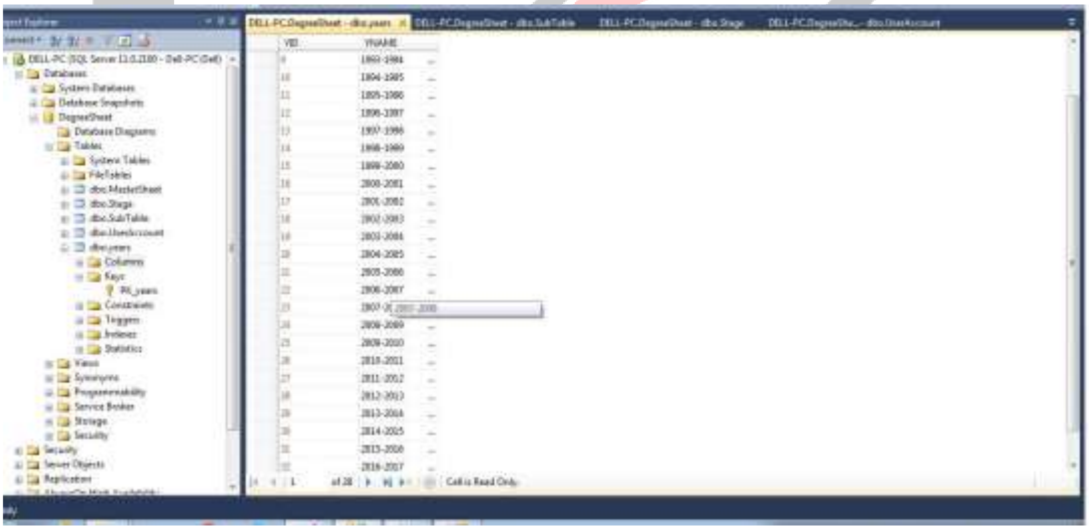
UAID	SID	YID	MID	CDegree1	CDegree2	SDegree	FDegree	TotalDegree
1	1	31	1	11	11	22	44	66
1	1	31	2	28	12	32	58	82
1	1	31	3	14	20	34	28	54
3	1	31	1	0	8	8	0	8
8	1	31	1	22	12	34	22	56
2	1	31	1	33	22	55	22	77
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 3. Master Sheet Table

The descriptions of other tables are as follows:

1. **The stage table:** which contain SID FIELD as a primary key which will be used as a foreign key in the major table and SName FIELD.
2. **The sub table:** which is the subjects table that contains the ID of the subject, the name in Arabic and English, the weight of each subject, the year and stage the subject included?

As for the years they are arranged in a year table where every year is given an ID as show in **(Fig. 4)**



DELL-PC.DegreeSheet - dbo.years | DELL-PC.DegreeSheet - dbo.SubTable | DELL-PC.DegreeSheet - dbo.Stage | DELL-PC.DegreeShe... - dbo.UserAccount

YID	YNAME
9	1993-1994
10	1994-1995
11	1995-1996
12	1996-1997
13	1997-1998
14	1998-1999
15	1999-2000
16	2000-2001
17	2001-2002
18	2002-2003
19	2003-2004
20	2004-2005
21	2005-2006
22	2006-2007
23	2007-2008
24	2008-2009
25	2009-2010
26	2010-2011
27	2011-2012
28	2012-2013
29	2013-2014
30	2014-2015
31	2015-2016
32	2016-2017

Cell is Read Only.

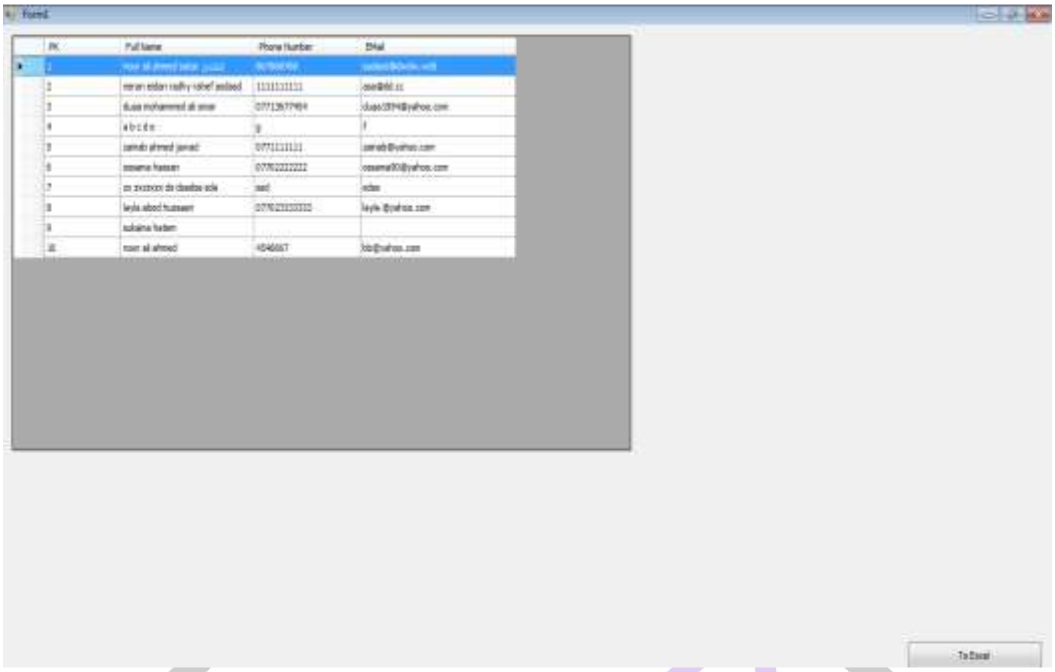
Figure 4. Year's table

ii. SQL TABLES

All these tables work together and perform their functions by what is called **stored procedure.** A stored procedure is a prepared SQL code that can be saved and reused over and over again. So if we think about a query that is written over and over again, instead of having to write that query each time it could be saved as a stored procedure and then just call the stored procedure to execute the SQL code that is saved as part of the stored procedure.
In addition to running the same SQL code over and over again it also have the ability to pass parameters to the stored procedure, so depending on what the need is the stored procedure can act accordingly based on the parameter values that were passed.

iii. USER INTERFACE

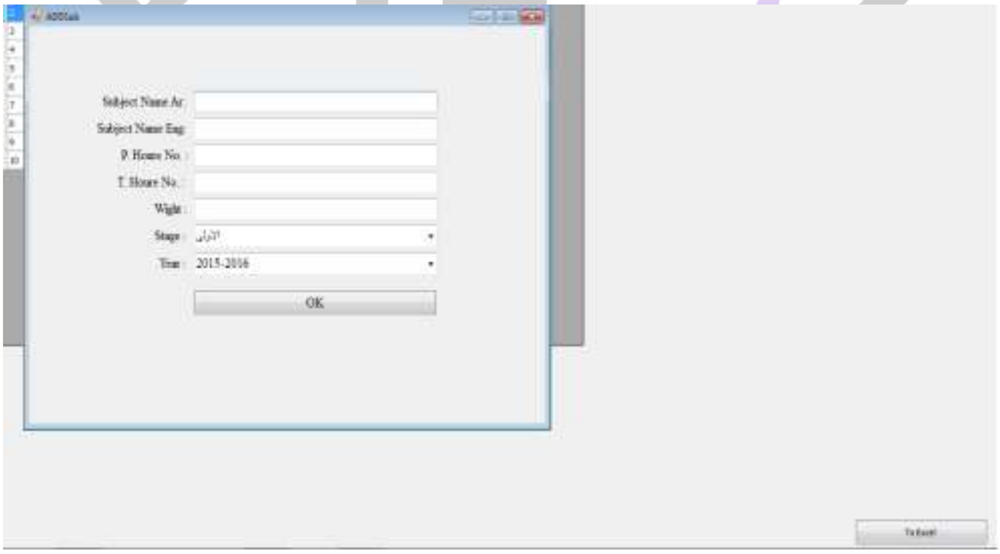
This system provides a simple interface for the maintenance of student information. The first thing to do is to connect the visual to the SQL so that the visual will make use of everything in the SQL by the connection code in main class.
The main from will contain the student information (student name, student id, student phone, student email) as shown below (Fig. 5):



PK	Full Name	Phone Number	EMail
1	[illegible]	[illegible]	[illegible]
2	mirran eidan radhy rahef asslaad	1111111111	ase@dd.cc
3	duaa mohammed ali omar	07713677494	duaa0394@yahoo.com
4	a b c d e	g	f
5	zainab ahmed jawad	0771111111	zainab@yahoo.com
6	ossama hassan	07702222222	oosama90@yahoo.com
7	zz zxzxzx zx dsadsa sda	sad	sdas
8	layla abed hussain	07702222222	layle @yahoo.com
9	sukaina hatem		
10	noor al ahmed	4546667	bb@yahoo.com

Figure 5. Front interface

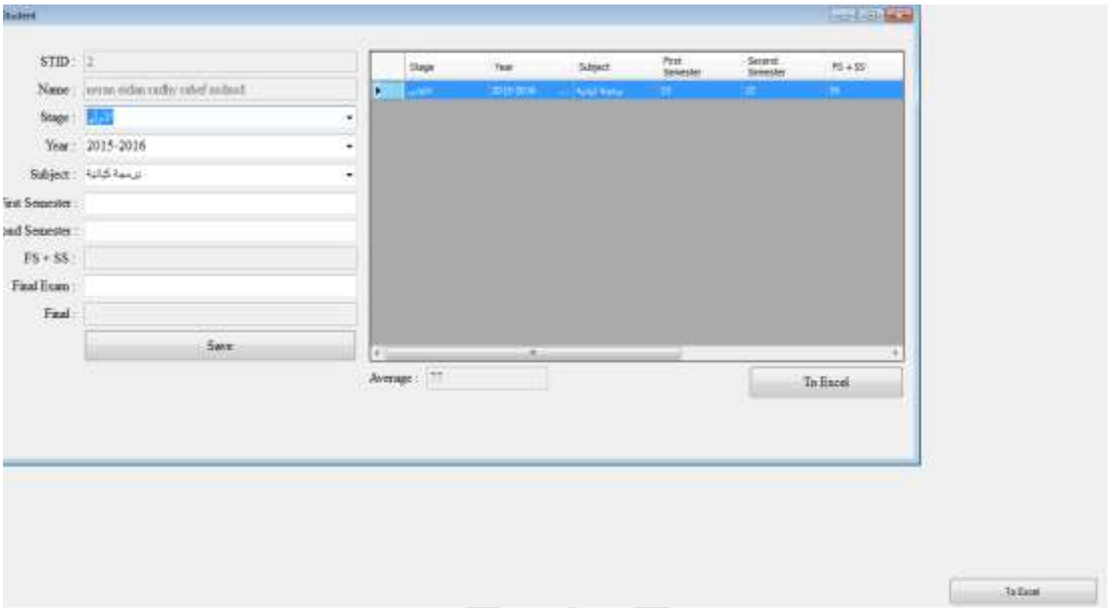
The first function is add name which allow the designer to add new students , The second function is subject which allow the designer to add new subjects and classified the division of the weight of the subject as shown in **(Fig. 6)**



Subject Name Ar:
Subject Name Eng:
P. Hours No. :
T. Hours No. :
Wight :
Stage : الاولى
Year : 2015-2016

Figure 6. Add Subject and Name interface

As for the third function is the student degree which include the first semester degree and the second and the final for each subject as shown in **(Fig. 7)**.



STID : 2

Name : seran eidan radhy rahef asdasd

Stage : الاولى

Year : 2015-2016

Subject : ترجمة كتابية

First Semester :

Second Semester :

FS + SS :

Final Exam :

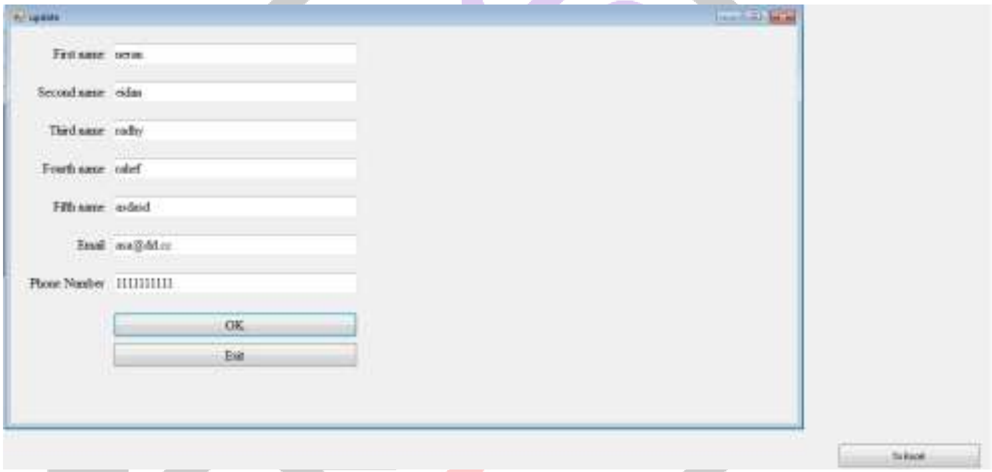
Final :

Stage	Year	Subject	First Semester	Second Semester	FS + SS
[illegible]	2015-2016	[illegible]	[illegible]	[illegible]	[illegible]

Average : 77

Figure 7. Degree interface

The forth function is to update the designer to make changes on existing data as shown in **(Fig. 8)**.



First name seran

Second name eidan

Third name radhy

Fourth name rahef

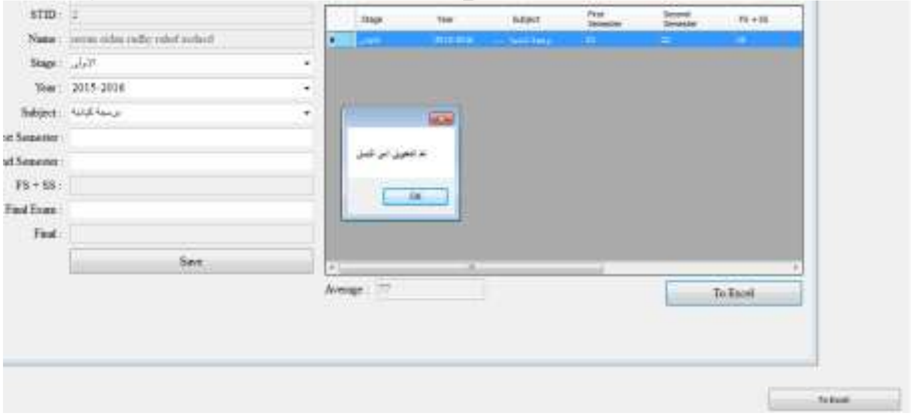
Fifth name asdasd

Email asa@dd.cc

Phone Number 1111111111

Figure 8. Change interface

At last after finishing the student grade processing now it is ready to print the degrees of the students this is done by using Microsoft Excel that is connected to the visual so that every table we need to print will be send to Excel and in excel we will print the data as shown in **(Fig. 9 and 10)**



STID : 2

Name : seran eidan radhy rahef asdasd

Stage : الاولى

Year : 2015-2016

Subject : ترجمة كتابية

First Semester :

Second Semester :

FS + SS :

Final Exam :

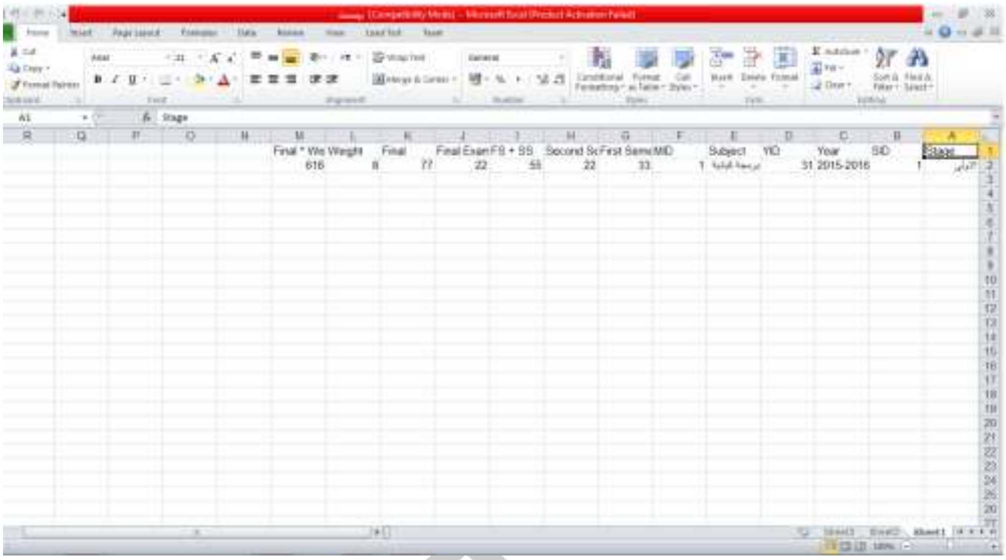
Final :

Stage	Year	Subject	First Semester	Second Semester	FS + SS
[illegible]	2015-2016	[illegible]	[illegible]	[illegible]	[illegible]

تم التحويل الى اكسل

Average : 77

Figure 9. Sending data to Excel



M	L	K	J	I	H	G	F	E	D	C	B	A
Final * We	Weight	Final	Final Exam	FS + SS	Second Se	First Seme	MID	Subject	YID	Year	SID	Stage
616	8	77	22	55	22	33	1	[illegible]	31	2015-2016	1	أولى

Figure 10. Excel interface

IV. CONCLUSIONS

A database is a far more efficient mechanism to store and organize data than spreadsheets; it allows for a centralized facility that can easily be modified and quickly shared among multiple users. It also allows the possibility of queries to obtain information for various surveys. Due to the number of users reading and modifying student data in the department the proposed system doesn't consider any back lock papers. So, the proposed system can be made more significant by adding some features to consider such back lock papers. It can also be more users friendly by replacing textboxes with drop down menu thus enabling the user to select from the drop down list instead of typing manually from the keyboard.

FOR FUTURE WORK also useful would be the ability to record which classes have been recommended by the student's advisor, as well as taken classes and grades made in each class.

REFERENCES

[1] Peter Rob and C. Coronel. (2007). Database System, Design, Implementation and Management, 7th edition.
[2] Ukem, E. O. and Ofoegbu, F. A , 2012, A Software Application for University Students Results Processing.Journal of Theoretical and Applied Information Technology. Vol. 35 No.1 .
[3] Beka, Ajayatuma. P . and Beka Francis. T. , Automated Result Processing System: A Case Study of Nigerian University , International Journal for Research in Emerging Science and Technology (IJREST) ,*Volume-2*, *Issue-9*, *Sep-2015*, **Page:** 59-68.
[4] Data Model Database Systems: Design, Implementation, and Management, 6th Edition, Rob and Coronel .
[5] A.P. de Vries, Content and multimedia database management systems, 1999.
[6] Hoffer, Jeffrey A; Prescott, Mary B; McFadden, Fred R, 2005, Modern Database Management, 7th Edition., Pearson Prentice Hall, New Jersey.
[7] Ramakrishnan, Raghu; Gehrke, Johannes, 2000, Database Management Systems, 2nd Edition., McGraw-Hill, New York.