

MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION

Federal State Autonomous Educational Institution of Higher Education  
South Ural State University (National Research University)  
School of Electrical Engineering and Computer Science  
Department of System Programming

THESIS IS CHECKED

Reviewer,  
The head of the maintenance  
group, OOO "IT Management"  
\_\_\_\_\_ A.A. Nusratullin  
" \_\_\_\_ " \_\_\_\_\_ 2017

ACCEPTED FOR THE DEFENSE

Head of the department,  
Dr. Sci., Prof.  
\_\_\_\_\_ L.B. Sokolinsky  
" \_\_\_\_ " \_\_\_\_\_ 2017

**DEVELOPMENT OF MANAGEMENT INFORMATION SYSTEM "SALARY" FOR MINISTRY OF HIGHER EDUCATION OF REPUBLIC IRAQ**

GRADUATE QUALIFICATION WORK  
SUSU-02.04.02.2017.115-153.GQW

Supervisor  
Cand. Sci., Asso  
\_\_\_\_\_ S.A. Ivanov

Author,  
the student of the group CE-216  
\_\_\_\_\_ H.A. Altuwaynee

Normative control  
\_\_\_\_\_ O.N. Ivanova  
" \_\_\_\_ " \_\_\_\_\_ 2017

Chelyabinsk-2017

## INTRODUCTION

Database system is the tool, which makes business operations much easy to deal with, the amount of data that database can process and deal with in very short time exceeds thousands time than what people can do, the accuracy of results in addition to the degree of organizing makes the using of database necessary to any business or organization.

The implementation of new techniques becomes essential in Iraq, as the country tend to apply the e-government strategy; based on this principle this project comes to computerize the action of accounting and employee salaries management and distribution.

In the republic of Iraq the ministry of education consists of several directorates separated on the governorates, in every governorate there is a general directorate of education which is in charge of managing the schools and their staff comprised of many sup-departments such as: staff department, services department and accounting department. The department of accounting is responsible for calculating salaries of all employees in the directorate as well as schools. Schools are divided into three types: elementary, secondary and high schools. The aim of this project is to develop a program to manage information and calculating employee salaries. It's necessary mentioning that the current system which they use to calculate and distribute the salaries still relies on paper-based system and they don't have any programs. This requires a long time and great efforts with the possibility of errors.

In order to perform the project we started at studying the operations of accounting department in the directorate of education, we have analyzed the follow of work, and after determining the inputs, process and outputs of accounting department, we draw up the wide lines of the main functions that the program should have which will be explained throughout the thesis.

The thesis will be conducted into introduction and two chapters. The first chapter will represent the theoretical aspect of the thesis. In this chapter C#, SQL

and database systems will be discussed. The second chapter will be the practical side of the thesis. In this chapter we shall develop our database system that we mentioned at the beginning of this introduction and then we shall end at some conclusions.

The purpose and objectives of the study:

- provide a theoretical framework about the databases systems;
- provide a theoretical framework about C# programming language;
- develop database system to manage employee information and accounts.

The importance of the study

This study will add to the body of knowledge on database system development with particular emphasis on MYSQL and C# programming language.

## **1. THEORETICAL PART**

### **1.1. MySQL**

MySQL is the most popular Open Source Relational SQL database management system. MySQL is one of the best RDBMS being used for developing web-based software applications [4]. Still you can make use of the SQL Express Edition, which is the lite weight version of the commercial Web.

Standard and Enterprise SQL server versions, the largest arena for MySQL is as the back end database for more than 12 million websites around the world. Many tycoons in the industry, including Yahoo Fi-Nance use MySQL as their back end database. Another advantage is that since MySQL is open-source [5], the source code is available under the General Public License, which means anyone can look into the source code and make changes, whereas with SQL Server it is a commercial application.

An unusual advantage that MySQL offers that you can incorporate different database storage engines with it, depending on your need. For example, you may want a database which will be mainly used for quick access, or accessing heavily portioned tables, or sometimes transactions [3].

Important to note is that you can run MySQL on almost all operating systems, including Windows. Your MySQL database can be moved across operating systems at any time. A widespread opinion is that MySQL performs best on UNIX or Linux OS. But still the performance MySQL offers is not as great as SQL Server. It is because of two reasons. One is that, since SQL Server is designed specifically for Windows unlike MySQL, it optimized to work with the OS. Second reason can be attributed to the poor design of the query optimizer in MySQL. However, again, the hope is that, since it is open-source any day, you can expect a release with a much advanced query optimizer [2].

### **1.2. C# programing language**

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications, we can use C# to create

Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language [7].

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time. C# syntax simplifies many of the complexities of C++ and provides powerful features such as nullable value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterators, which enable implementers of collection classes to define custom iteration behaviors that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct [10].

As an object-oriented language, C# supports the concepts of encapsulation, inheritance, and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it may implement any number of interfaces. Methods that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. In C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance [14].

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

- Encapsulated method signatures called delegates, which enable type-safe event notifications;

- Properties, which serve as accessors for private member variables;
- Attributes, which provide declarative metadata about types at runtime;
- Inline XML documentation comments;
- Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

If you have to interact with other Windows software such as COM objects or native Win32 DLLs, you can do this in C# through a process called "Interop". Interop enables C# programs to do almost anything that a native C++ application can do. C# even supports pointers and the concept of "unsafe" code for those cases in which direct memory access is absolutely critical [13].

The C# build process is simple compared to C and C++ and more flexible than in Java. There are no separate header files, and no requirement that methods and types be declared in a particular order. A C# source file may define any number of classes, structs, interfaces, and events [12].

## **2. SOFTWARE DESIGN**

### **2.1. Description of the work Area**

This project came to develop and manage information a large number of employees and their salaries account In order to get an idea of comprehensive and accurate work we must explain the very fine details of the problem and the goals of the project as well as computational procedures and rules according to the instructions of Education Ministry. We have previously mentioned that Ministry of high education has one directorate of education in each governorate and each directorate of education has accounts department that are responsible for management information and calculation of salaries around 25 thousand employees. It's necessary mentioning that the current system which they use to calculate and distribute the salaries is still based on the old paper method and they don't have any programs that can perform they still rely on paper-based system. This requires a long time and great efforts with the possibility of errors, to explain the employees' salaries account system according to the rules of Iraq Ministry of Education. Each employee has a so-called nominal salary, which varies from employee to another. According to the number of years the employee worked and the education that he obtained in addition to the position, another financial funds could be added to the nominal salary such as incentives which also depends on the specific mechanism and for example marital status. The married employee get increasing is estimated at 50000 IQD added to the nominal salary, if he or she has kids (the employee that have children under the age of 18 years ) will get increase around 50000IQD for each child maximum for three children. Certificate employee gets financial increase according to his certificate, whereas financial increase for the employee that have a master degree different from the financial increase for another employee holds a bachelor's degree. Career is one of the mechanisms that determine the financial increase. Place of living the employee can receive a financial increase, according to his place to live. If employee lives outside the city, the Ministry of Education will give him transport fares to and from his work location.

Risks there is a financial increase granted to employees who perform dangerous work such as Electricity staff and security guards, The scientific title, considered the scientific title of the employee is one of the mechanisms that allow employees to obtain financial increase, whereas financial increase that obtained the employee who holds the scientific title of Assistant Professor differ from financial increase that obtained the employee who holds the scientific title of Associate Professor, post of work, the employee, according to the Ministry of Education in Iraq, gets a financial increase by his post of work, There is also a financial deduction deducted from gross salary, gross salary is the summation of the nominal salary with all financial increases mentioned above.

According to instructions of the ministry of education in Iraq the financial deductions include the retirement, usually deducted a certain amount of money from the gross salary of the employee that will receives it as pension after finishing work contract with the Ministry of Education. Taxes are imposed on employees, and are usually calculated as a percentage of the salaries that Ministry of Education pays to its staff also known as withholding tax. Real estate as financial deduction is other directorate not related to the Ministry of Education only may it could give the employee of the ministry of education funding for purchase or construction some property. Therefore the ministry of education deducted from the salary of its employees for the payment of advances funding to other directorates as well as in the same sense there are other financial deductions called Housing funding and agriculture funding. In order to deal with all these details that we have mentioned previously we need the database for storing, analyzing and restoring data.



## 2.2. UML diagram

### 2.2.1. Use case diagram

As the most known diagram type of the behavioral UML diagrams, Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions are interacted [18].

Use case diagram is especially important in organizing and modeling the behavior of a system. In this use case diagram there exist ten actors, three use cases and their relationships. The use case diagram is shown in the fig.1.

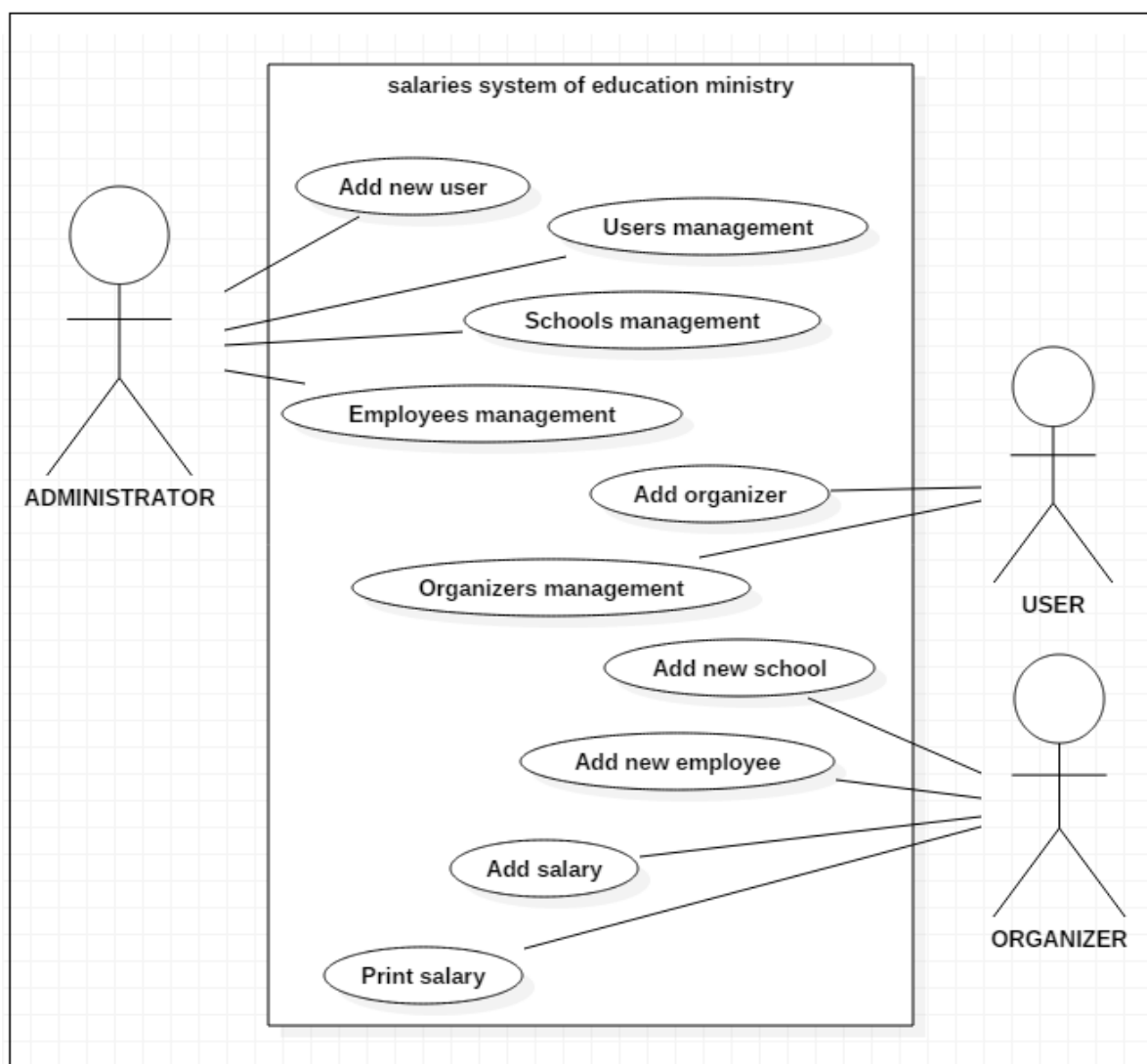


Fig. 1. Use case diagram

### 2.3. The database schema

This database scheme contains nine tables. The two most important tables in this database are the salaries and schools (see figure number 2).

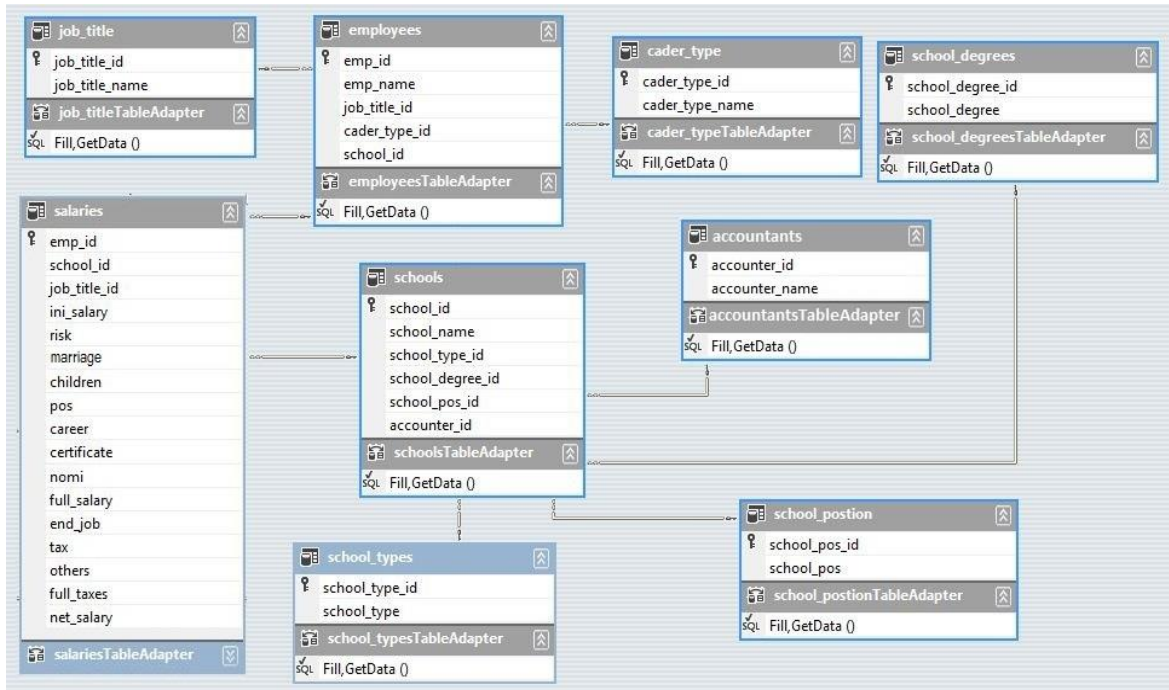


Fig. 2. Database schema

The following is details for each table.

The structure of employees table is in figure number 3.

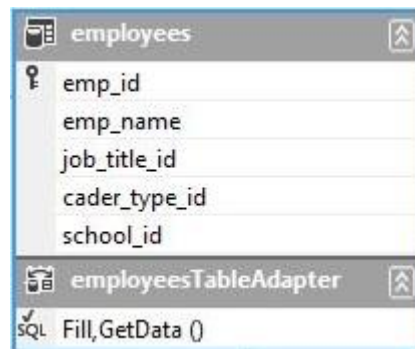


Fig. 3. Employees table

The employees' table consists five columns:

- emp\_id: the primary key for employees table from type Auto number;
- emp\_name: the name of employee who work in General Directorate of Education from type varchar;

- job\_title\_id: The foreign key of job\_title table from type int;
- Cader\_type\_id The foreign key of cader\_type table from type int;
- School\_Id: The foreign key of school table from type int.

The structure of accountant's table is in figure number 4.

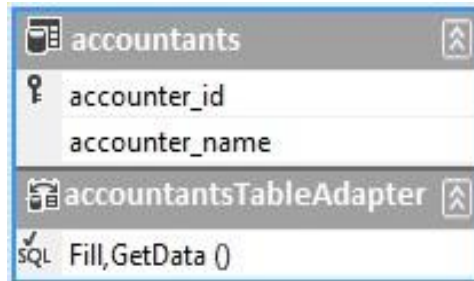


Fig. 4. Accountants table

The structure of a counter's table consists of 2 columns

- Accounter\_id: the primary key for a counter's table from type Auto number.
- Acconuter\_name: the name of Accountants from type varchar.

The structure of school\_degrees table is in figure number 5.

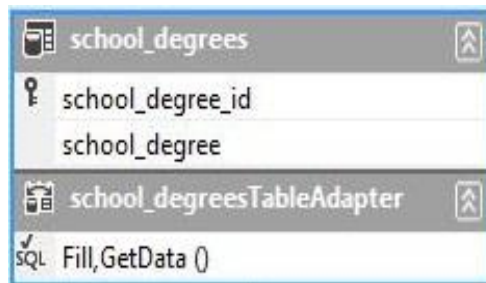


Fig. 5. School\_degrees table

The structure of a school\_degrees table consists of two columns:

- School\_degree\_id: the primary key for school\_degrees table from type utonumber;
- School\_degree: the sort of each school in general directorate of education from type varchar.

The structure of school\_ postion table is in figure number 6

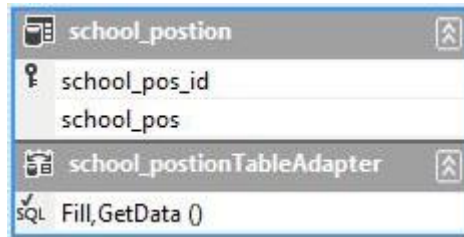


Fig. 6. School\_postion

The structure of a school\_postion table consists of two columns.

- School\_pos\_id: the primary key for school\_postion table from type Auto number;
- School\_pos: to determine the site of school from type varchar.

The structure of schools table is in figure number 7.

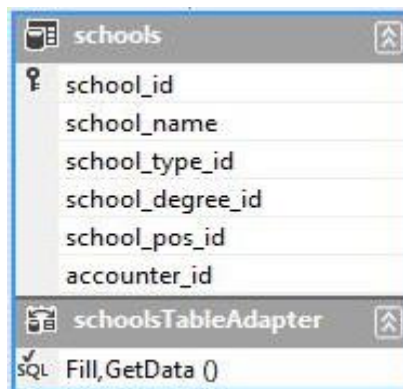


Fig. 7. Schools table

The structure of a schools table consists of six columns:

- school\_id the primary key for schools table from type Auto number;
- school\_name the name of schools that belong to the General Directorate of Education from Type varchar;
- school\_type\_id: The foreign key of school\_type table from type int;
- school\_degree\_id: The foreign key of school\_degree table from type int;
- School\_pos\_Id: The foreign key of school\_postion table from type int;

- Accounter\_id: The foreign key of accountants table from type int.

The structure of salaries table is in figure number 8

Column Name
emp_id
school_id
job_title_id
ini_salary
risk
marage
children
pos
career
certificate
nomi
full_salary
end_job
tax
others
full_taxes
net_salary

Fig. 8. Salaries table

The structure of a salaries table consists of 17 columns:

- Emp\_id the primary key for salaries table from type Auto number;
- School\_id the foreign key of school table from type int;
- Jop\_title\_id: the foreign key of Jop\_title table from type int;
- Ini\_salary the initial salary of employees from type int;
- Risk: to determine the allocations of risk whether the employee has a dangerous job or not, from type int;
  - Marriage: This column determines whether the employee is married or not, from type int;
  - Children: This table determines the allocations of children depending on whether the employee has children or not, from type int;

- Pos: contains allocations of transportation based on whether the employee lives in a remote place from the location of his work or not, from type int;
- Career: these are allocations granted to all employees unconditionally;
- Certificate: this column contains allocations of certificate, from type int;
- Nomi: these are allocations granted according to the status of the employee;
- Full salary: this column contains the total salary of the employee after the additions of all his financial dues from type int;
- End\_job: to determine the value of the discounts to be allocated to the pension of employee from type int;
- Tax: It contains the value of the taxes that must be paid by the employee from type int;
- Others: to identify any other financial deductions, from type int;
- Full taxes: To determine all the deductions, which are deducted from the employee's salary from type int;
- Net \_salary: To find out the final salary of the employee after all additions and discounts from type int.

The structure of job\_title table is in figure number 9

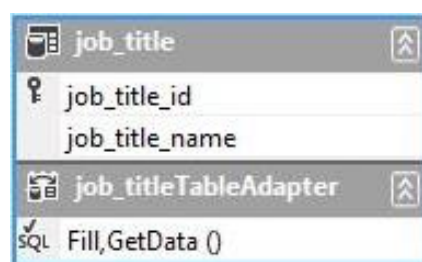


Fig. 9. Job\_title table

The structure of a job\_title table consists of 2 columns:

- Job\_title\_id the primary key for job\_table table from type Auto number.

- job\_title\_name: to determine the job title for employees from type varchar.

The structure of cader\_type table is in figure number 10.

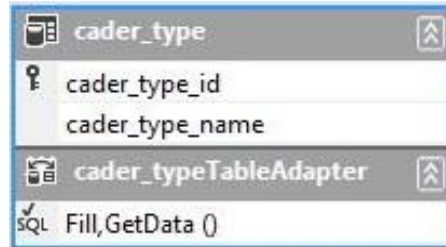


Fig. 10. Cader\_type table

The structure of a cader\_type table consists of 2 columns:

- Cader\_type\_id: the primary key for cader\_type table, from type Auto number;
- Cader\_type\_name: to describe whether the employee has a governmental contract or part contract, from type varchar.

The structure of school\_types table is in figure number 11.

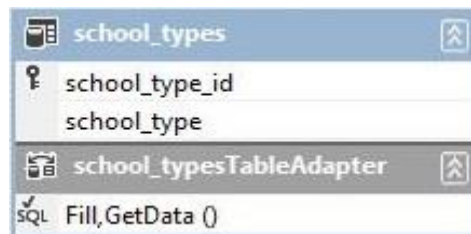


Fig. 11. School\_type table

The structure of school\_types table consists of 2 columns:

- school\_type\_id: the primary key for school\_types table, from type Auto number;
- School\_type: to classify schools basing on gender of students whether they are for male or female students, from type varchar.

### 3. SOFTWARE DEVELOPMENT

#### 3.1. Interfaces

In Fig. 12 we see Main Interface of program that contains a list of commands: file, schools, employees, organizers, salaries, search, reports and about the program. Command "file" contains a list of options. The first option is "login". As we see it is effective option due to the need. When press on this command another window named 'login window' is shown contains two fields: 'username' and 'password'. Through this window only authorized persons can access to program after typing their user name and password.

There are others options in command "file" such as "create a backup, call reserved copy, add new user, and user management" the last option is "sign out", it is effective option because the user may want to sign out from the main interface of program before login to the program.

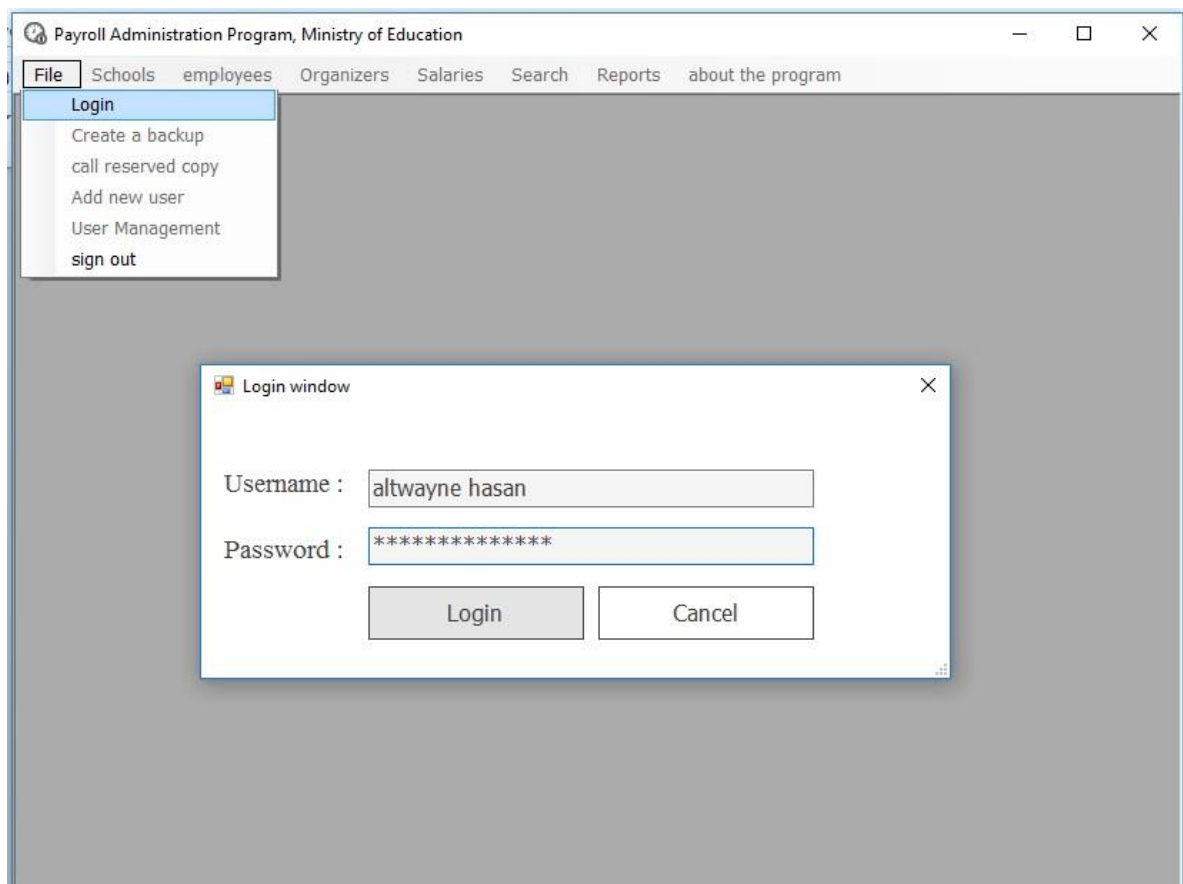


Fig. 12. Main Interface



After typing username and password and access to the program the user will see this window (fig. 13) with all commands activated.



Fig. 13. Main interface

If we come back to the command "file" again, we will find all its options are activated also, Let us explain all this options see fig. 14.

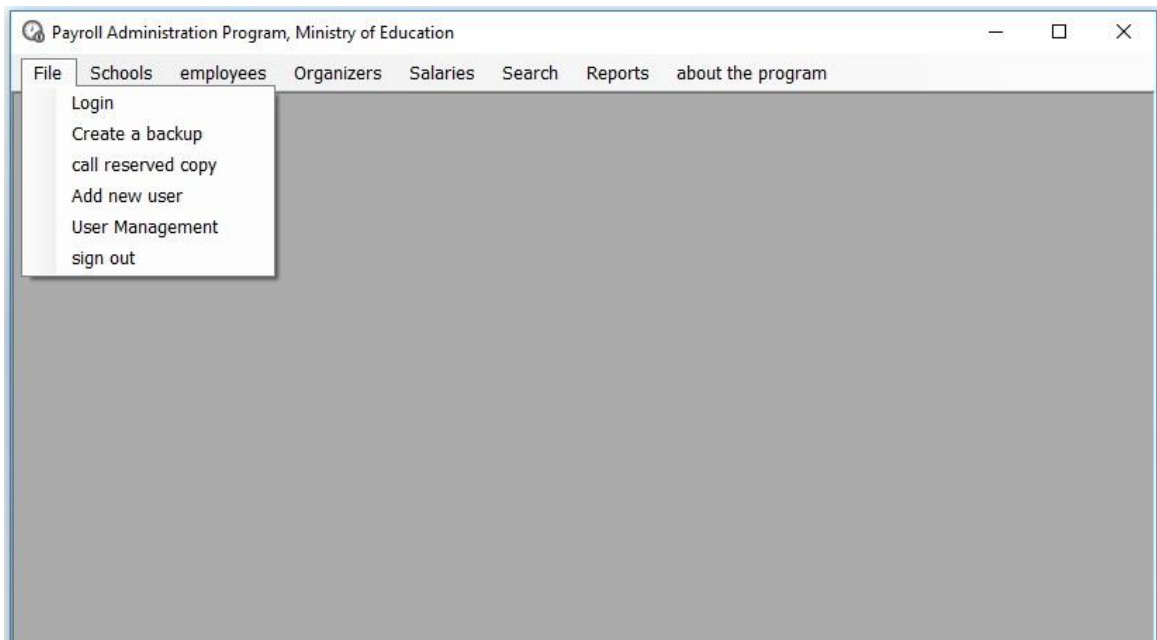


Fig. 14. File command

**Login** this option has been explained earlier.

**Create a backup:** this option allows us to create a backup copy of the program used later when needed, when press on this option will shows another window that enable us to locate where to store the backup and then storage it by press button 'create a backup', see fig. 15.

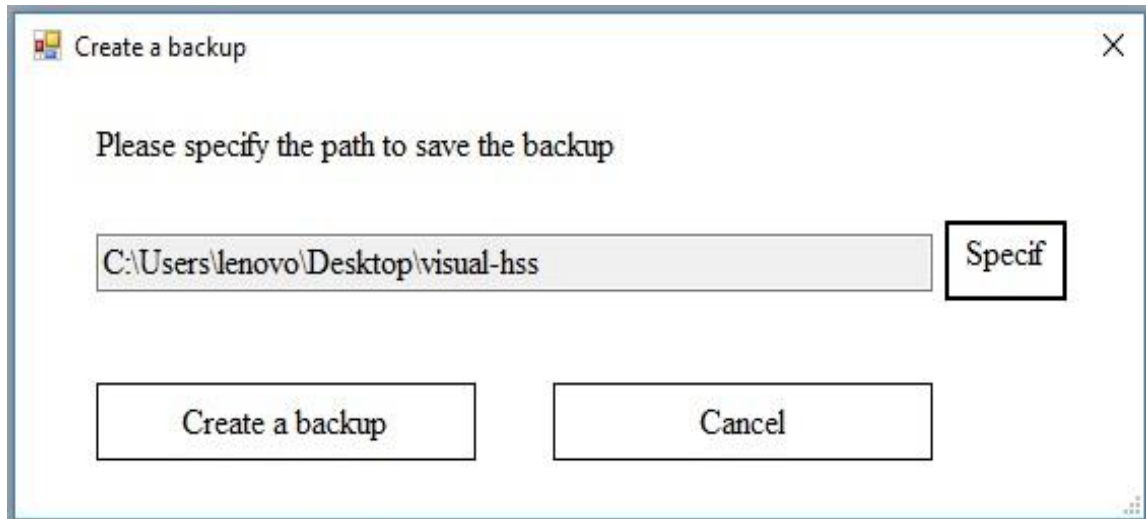


Fig. 15. Create backup window

**Call reserved copy:** when press on this option it will shows another window that enables us to call reserved copy that have been saved earlier. This window contains the button 'Select', which sets us from where we will be called the backup see Fig. 16.

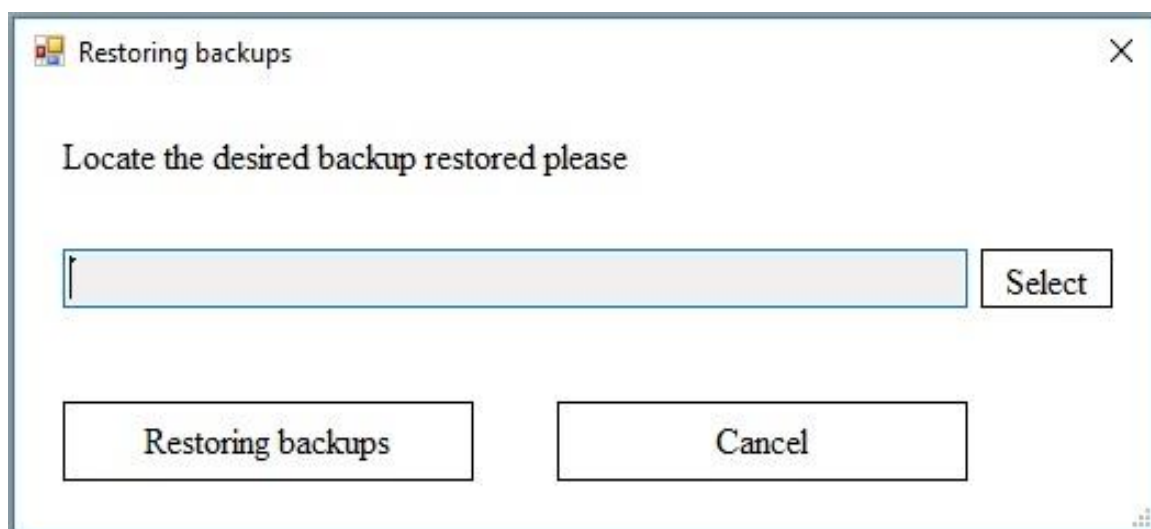
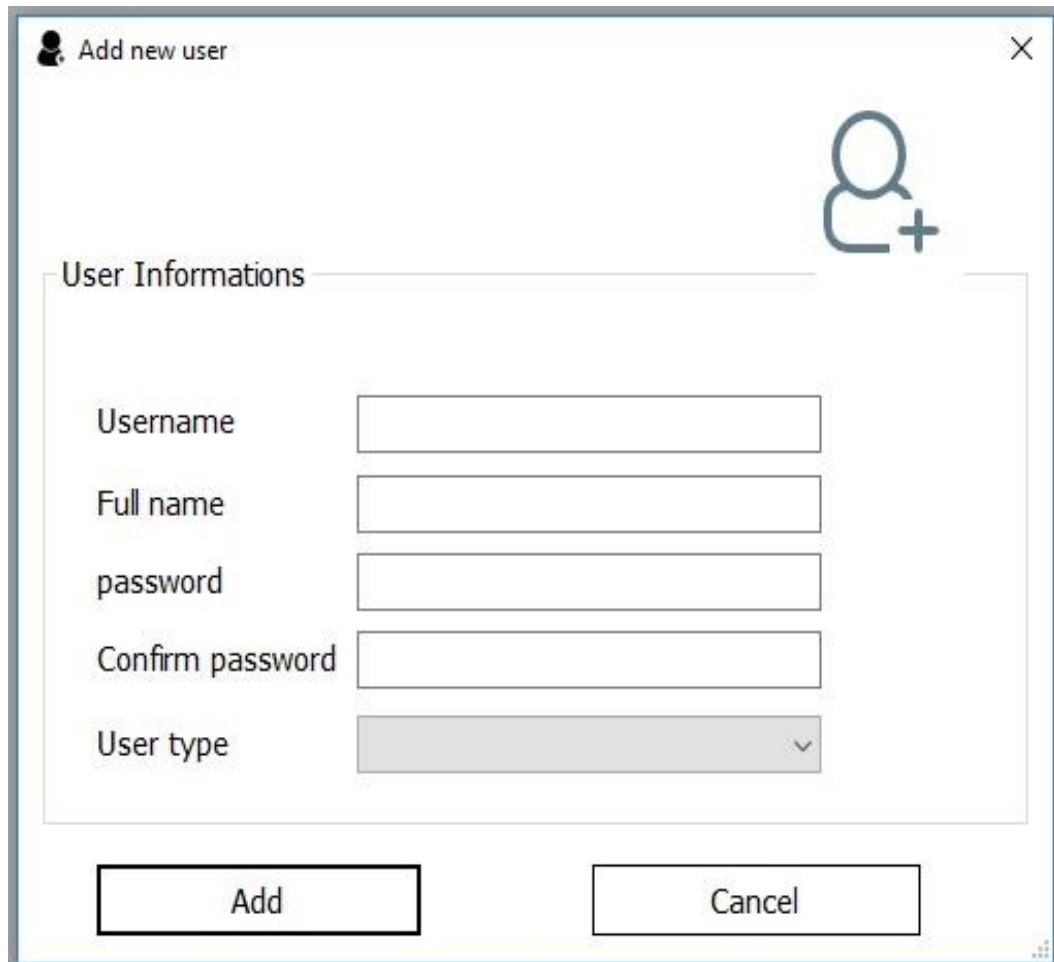


Fig. 16. Restoring backup window

**Add new user:** when press on this option will shows another window that enables us to add user. This window contains a set of fields to write new user information and then click on the button 'Add' to complete the addition process, see fig. 17.



The image shows a software dialog box titled "Add new user". It features a close button (X) in the top right corner. In the upper right area, there is a large icon representing a person with a plus sign. Below this icon, the text "User Informations" is displayed. Underneath, there are five input fields: "Username", "Full name", "password", "Confirm password", and "User type". The "User type" field is a dropdown menu. At the bottom of the dialog, there are two buttons: "Add" and "Cancel".

Fig. 17. Add new user

**User management:** the purpose of this option to view and access to Specific information of each user, such as the full name, username, password and user type, and allow us to delete, add and modify each user, as well as change the user type if he is administrator or just as a user, when press on this option will shows another window which enables us to do what we have mentioned above, see fig. 18.

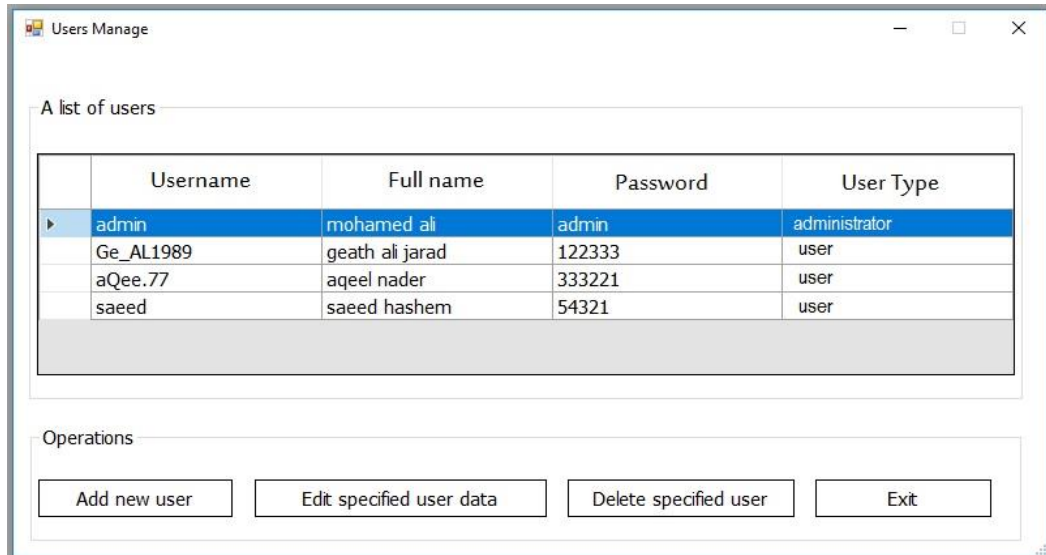


Fig. 18. User management

**Sign out:** when press this button you will leaving the program.

The second command in the main interface of the program is "schools" When we press this command will shows us another window containing the following options.

**Add new school data:** This option allows us to add a new school with all it is own information, such as: gender of the school, name of the school, stage of The school and its location and the name of organizer that responsible for this school, see fig. 19.

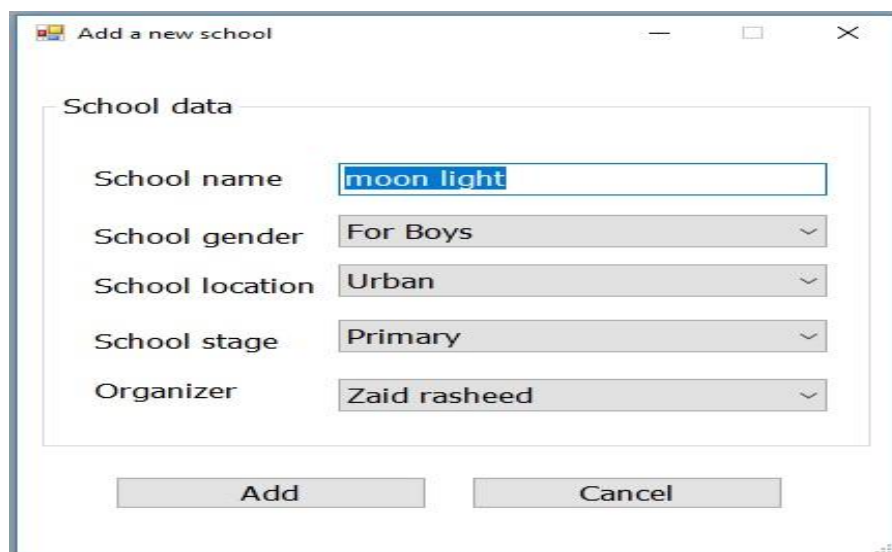


Fig. 19. Add new school

**Management of schools data:** This option gives us access to all the data for each school, as well as allows us to delete and modify the data for each school in addition to that this option allows us to add new school data, see fig. 20.

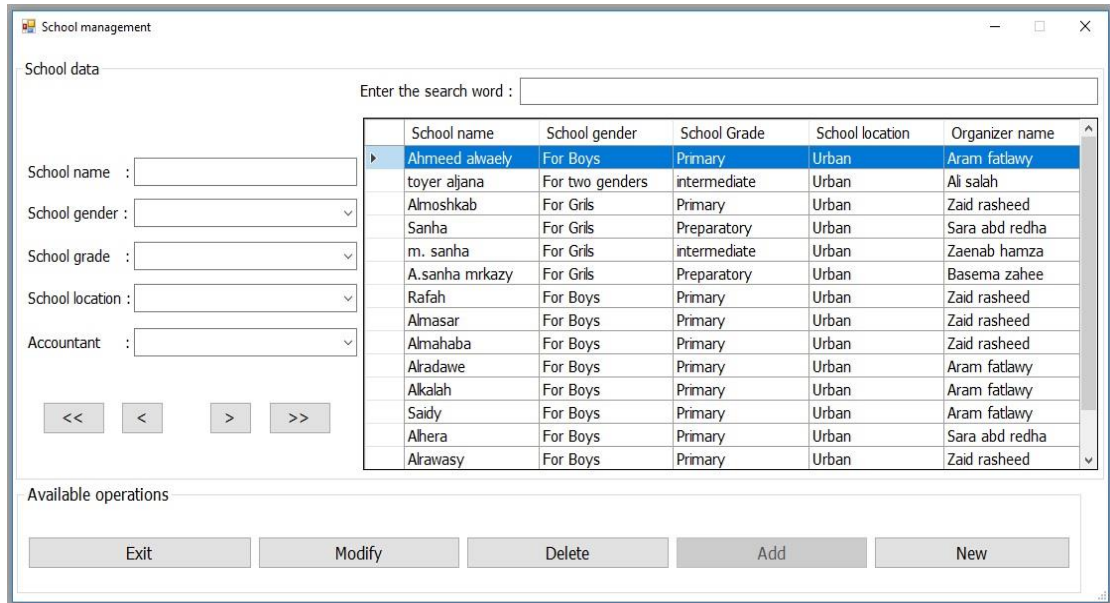


Fig. 20. School management

The next command in the main interface of program is "Employees" this command containing the following options:

**Add data of new employees:** is option allows us to add a new employee with his information: employee name, career title and staff type, see fig. 21.

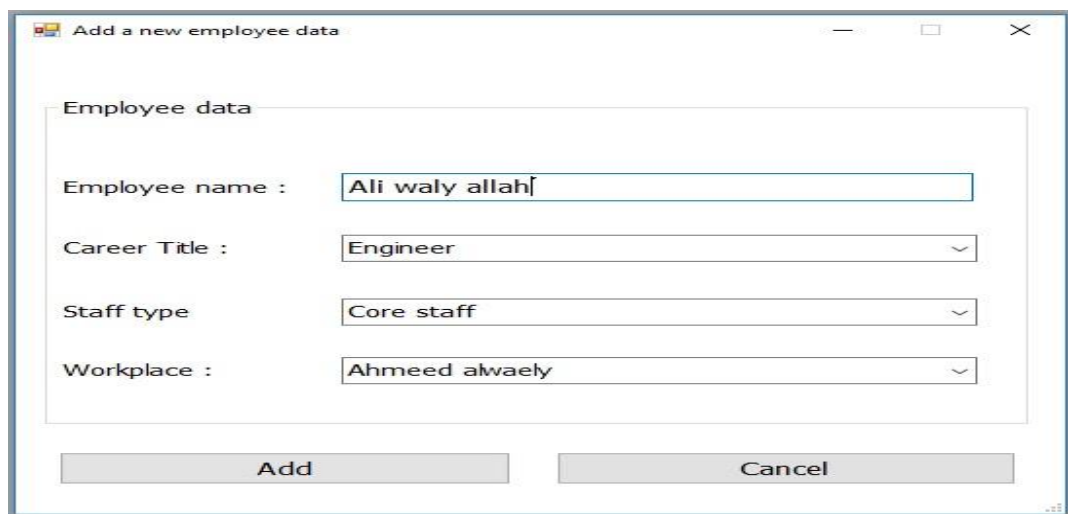


Fig. 21. Add new employee

**Management of employee's data:** This option gives us access to all the data for each employee, as well as delete and modify the data for each employee. In addition to that this option allows us to add new employee data, See fig. 22.

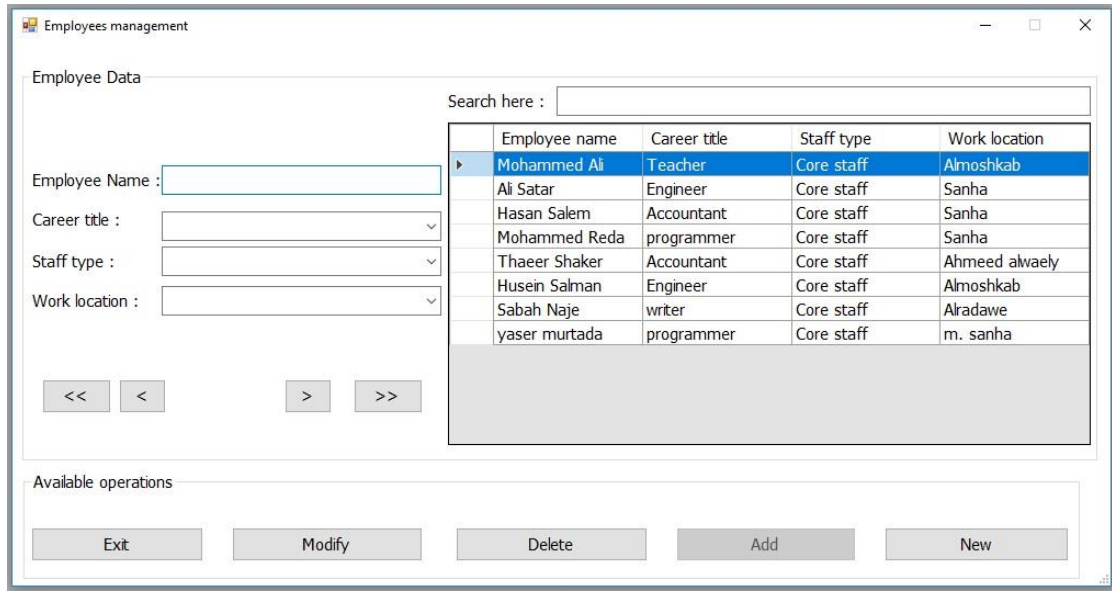


Fig. 22. Employee's management

Command "organizer" is the fourth command in main interface of program it is contains the following options:

**Add new organizer:** this option for add new organizer, when press it will shows us new window that contains field 'organizer name' after write the name of organizer press 'add' button for additional, see fig. 23.

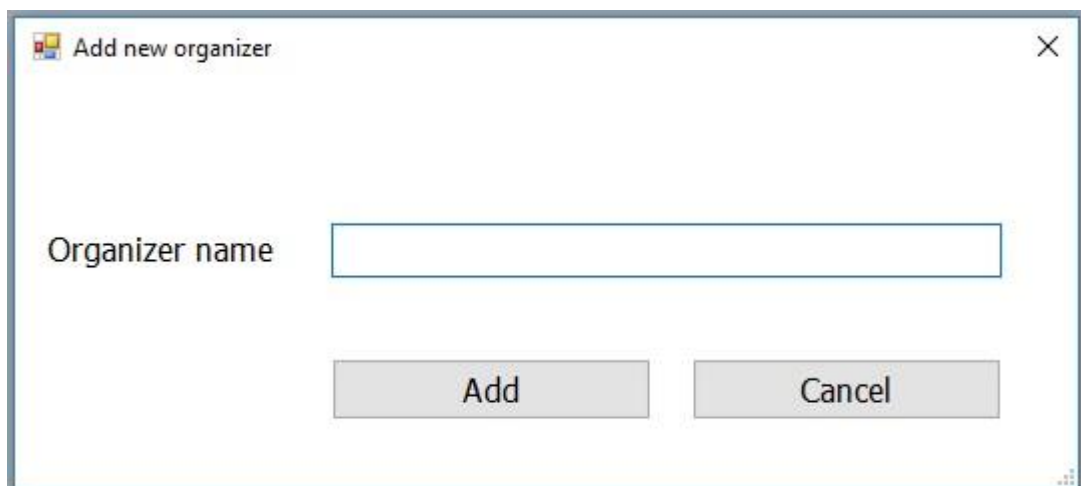


Fig. 23. Add new organizer

**Management of organizer data:** This option allows us to modify, delete and add organizers see fig. 24.

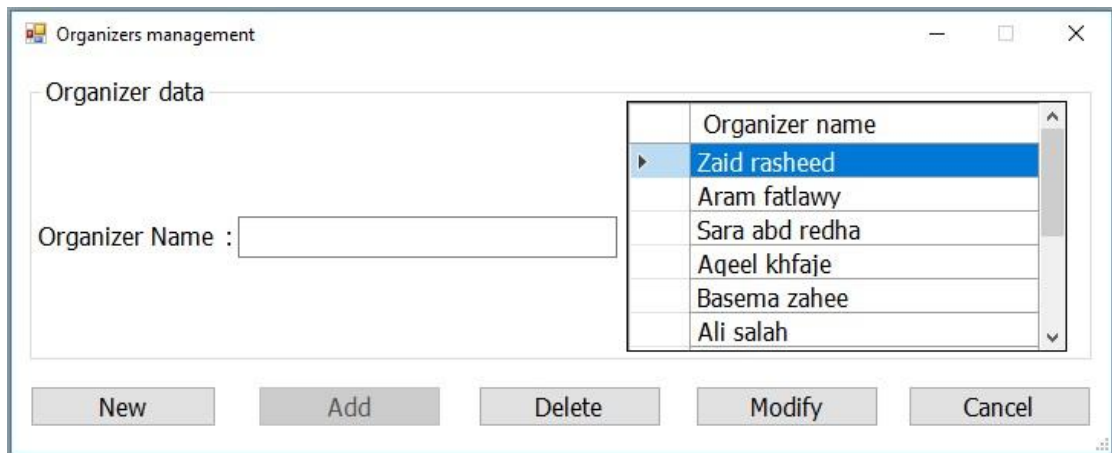


Fig. 24. Organizer management

The next command in is the "salaries" It contains option 'add salary'. When we click on this option it will take us to another window called 'add and modify employee's salary'. This interface allows to organizers typing all financial allocations and deductions of employee. It is contains buttons save, modify and cancel to enable the organizer make some modifications and then saving, see fig. 25.

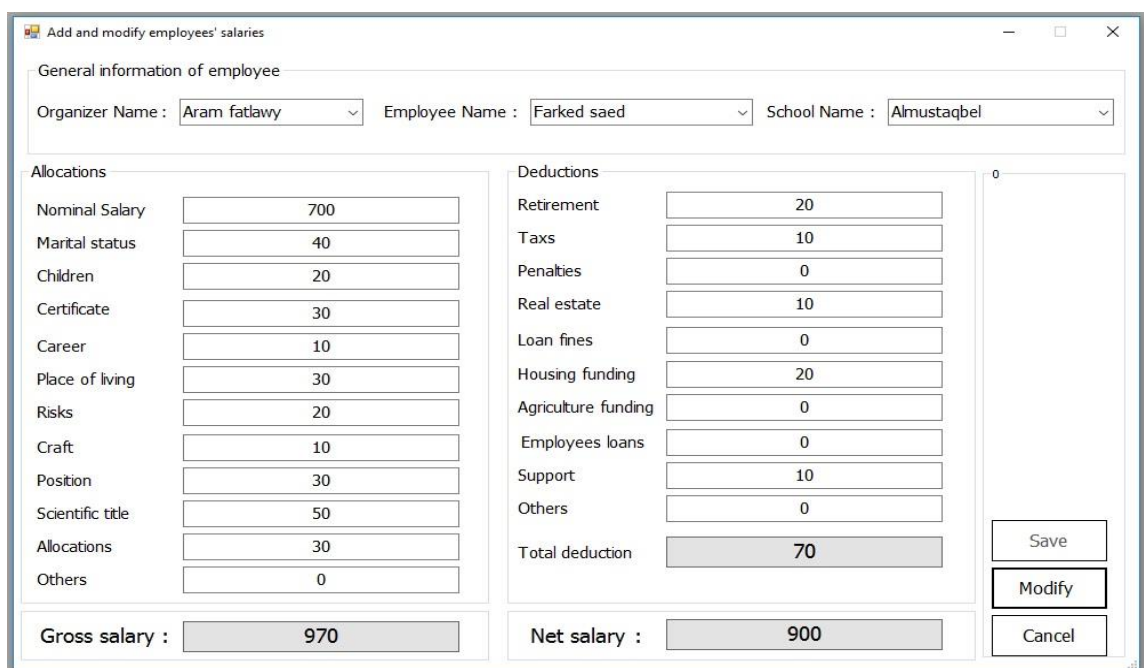


Fig. 25. Add & modify employee's salary

Command "search". This command allows us to search for employee data by employee name, staff type, career title or organizer name. When the user click "search" command the system shows us search window button "export to excel file". This button is used to save search results within an excel file see fig.26.

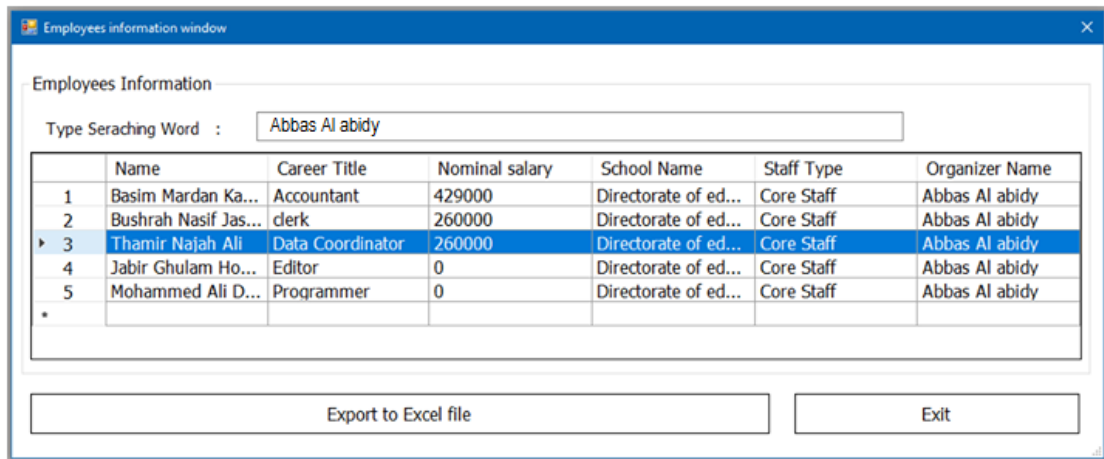


Fig. 26. Employee's information window

The next command called "reports" contents the option 'print payroll'. When pressed it will take us to a new window (see figure number 27 and 28). Which through this window, the organizer can display employee payrolls for any school from the schools that he is responsible

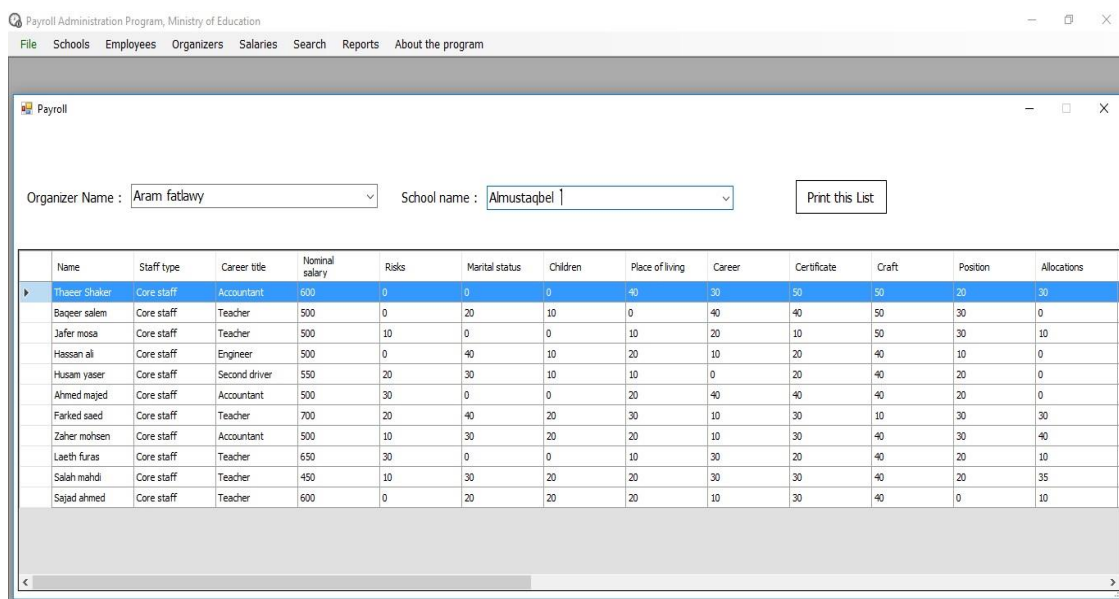


Fig. 27. Payroll window



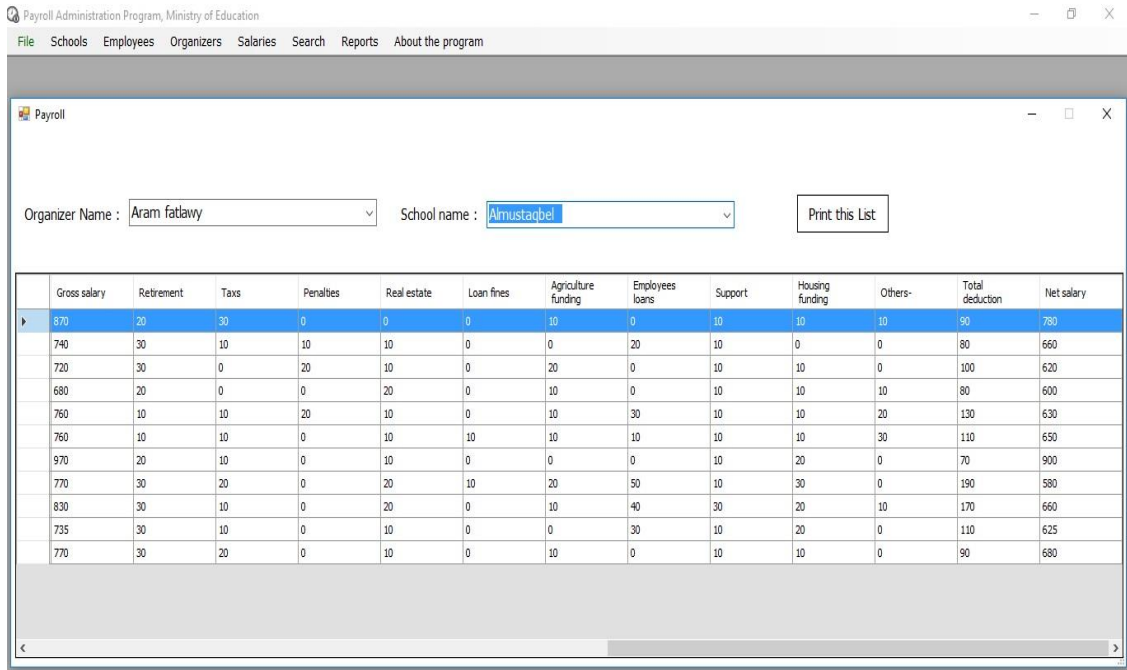


Fig. 28. Payroll window

When the organizer presses the "print this list" button in the window above, the system takes us to another window, called the payroll window, which represents the final outputs of the program, see figure number 29.

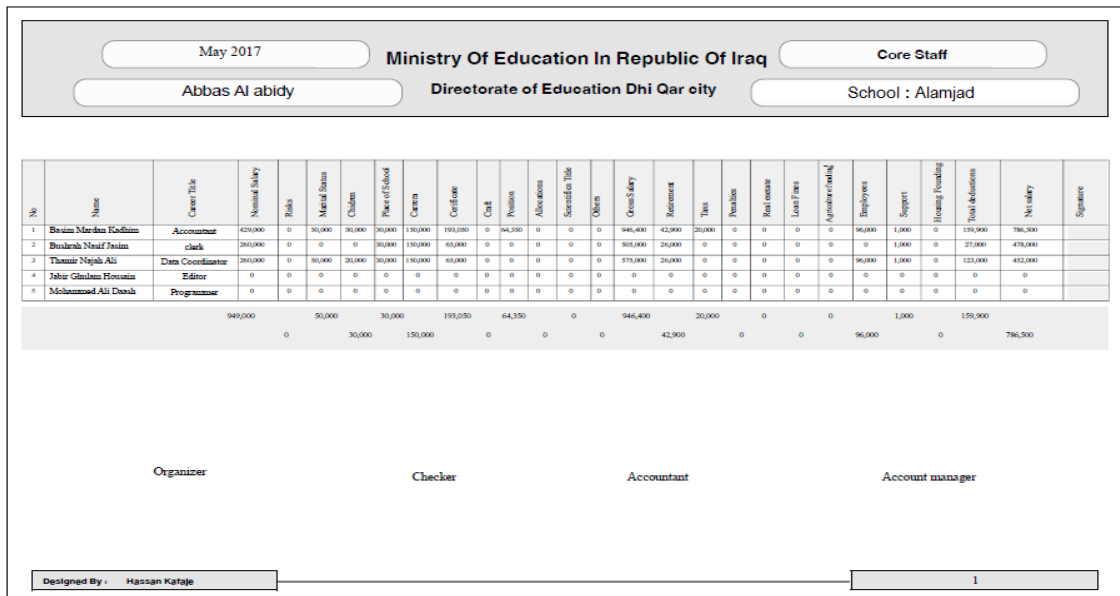


Fig. 29. Salaries list

## 4. TESTING

Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use, software testing involves the execution of a software component or system component to evaluate one or more properties of interest.

### Main interface testing

Table 1. Main interface testing

NO	Test Case	Test Steps	Expected Result	Actual Result
1.	The user enters the username and password properly.	1. Enter "Aram_ali" in username. 2. Enter "*****" in password.	Should access to the program without any objection.	The function works correctly.
2.	When the user enters username and password improperly.	1. Enter "Aram_Ali" in username. 2. Enter not correct password	The system should reject these values	The function works correctly.
3.	The user enters incorrect username and correct password.	1. Enter name does not exist in database. 2. Enter "*****" in password.	System should prompt the user to enter valid values.	The function works correctly.
4.	When the user enters incorrect username and incorrect password.	1. Enter not correct username 2. Enter not correct password.	System should prompt the user to enter valid values.	The function works correctly.

### File command testing

Table 2. File command testing

NO	Test Case	Test Steps	Expected Result	Actual Result
1.	When the user	1. The user clicks on	1. The system show	The function

NO	Test Case	Test Steps	Expected Result	Actual Result
	wants to create a backup	command "file" and then button "backup". 2. The user click button "Specify" to select the storage path. 3. The user determines where to store the backup, and presses the "save"	message "please selects the path to save the backup". 2. The system responds to orders. 3. The user sees "the backup was successfully created".	works correctly.
2.	The user wants to restoring backup that stored earlier.	1. The user click on command "file" and then "call reserved copy" 2. The user selects the storage path. 3. The user click "restoring backup" button.	1. The system show message "Locate the desired backup" 2. The system responds to orders. 3. The user sees "Backup has been successfully restored".	The function works correctly.
3.	The user wants to add new users	1. The user click on "file" button and then "Add new user" button. 2. The user fills in all fields then user press the "Add" button.	The system responds to orders, and show message "The user has been successfully added".	The function works correctly.
4.	When the user wants to manage users information.	1. The user click on "file" button and then "user's management". 2. The user makes some changes, add, edit or delete.	The user sees 'operation accomplished successfully'	The function works correctly.
5.	If the user wants to sign out.	The user press "file button" and then "sign out" button.	The system will view the homepage before login.	The function works correctly.

## Schools command testing

Table. 3 Schools command testing

NO	Test Case	Test Steps	Expected Result	Actual Result
1.	When user wants add new school.	1. The user click "schools" command and then "Add school". 2. The user fills in all fields and press "add" button.	The system shows message 'the school has been successfully added'.	The function works correctly.
2.	When user wants add new school but not fills in all fields.	1. The user click "schools" command and then "Add school". 2. The user fills in some fields and not writes the school name.	The user sees "please, fill in all fields properly ".	The function works correctly.
3.	When the user wants to manage schools information.	1. The user clicks "schools" button and then "management of schools data". 2. The user makes some change, add, modify or delete.	The user sees 'operation accomplished successfully'	The function works correctly.

## Employees command testing

Table 4. Employees command testing

NO	Test Case	Test Steps	Expected Result	Actual Result
1.	When the user want to add new employee.	1. The user click "employees" button and then "add new	The system shows message 'the employee has been	The function works correctly.

NO	Test Case	Test Steps	Expected Result	Actual Result
		employee" 2. The user must fill in all fields and press "Add" button.	successfully added'.	
2.	If the user wants to delete or modify employee.	1. The user clicks "employees" button and then "management employee's data". 2. The user specifies the employee and delete or edit his information.	The system responds to orders and the user sees 'operation accomplished successfully'.	The function works correctly.

### **Organizer command testing**

Table 5. Organizer command testing

NO	Test Case	Test Steps	Expected Result	Actual Result
1.	The user wants to add new organizer.	1. The user clicks "organizers" and then "add organizer". 2. The user press "add" button without write organizer name.	The user sees this message "please inter the name of employee".	The function works correctly.
2.	The user wants to add new organizer.	1. The user clicks "organizers" and then "add organizer". 2. The user write organizer name and press "add" button.	The system shows message 'the organizer has been successfully added'.	The function works correctly.
3.	The user wants manage organizers information	1. The user clicks "organizer" and then "management of	The system responds to orders and the user sees	The function works correctly

NO	Test Case	Test Steps	Expected Result	Actual Result
		organizer data". 2. The user makes some changes and press "ok"	'operation accomplished successfully'.	

### Salaries command testing

Table 6. Salaries command testing

NO	Test Case	Test Steps	Expected Result	Actual Result
1.	When the user wants add a salary to the employee	1. The user click on "salaries" command and then "add salary" button. 2. The user fill in fields and press "save" button	The user sees 'Salary added successfully'.	The function works correctly.

### Search command testing

Table 7. Search command testing

NO	Test Case	Test Steps	Expected Result	Actual Result
1.	When the user wants to find the information of an employee.	1. The user click "search" command. 2. The user searches for the employee by the employee name or school name to which the employee belongs or the organizer name that responsible for the school.	The system will display the search results, which are all the employee information that was queried.	The function works correctly.
2.	When the user wants to find the	1. The user click "search" command.	The system displays the following	The function works

NO	Test Case	Test Steps	Expected Result	Actual Result
	information of an employee.	2. The user searches for the employee by the improperly employee name or improperly school name.	message, "You have entered invalid values".	correctly.

### **Reports command testing**

Table 8. Reports command testing

NO	Test Case	Test Steps	Expected Result	Actual Result
1.	The user wants to print report of employee's salary for some school.	1. The user click "reports" command and then "print report" button. 2. The user chooses the school name and the organizer name which responsible for the school. 3. The user press "print payroll".	The system should print payroll.	The function works correctly.

### **About the program command testing**

Table 9. Program testing

NO	Test Case	Test Steps	Expected Result	Actual Result
1.	When the user wants to know something about the program's work	The user clicks on "about the program" command and then "help" button.	The system should access to help page.	The function works correctly.

## CONCLUSION

I have designed a program to develop information and calculate the salaries of employees in the ministry of education in the republic of Iraq, which uses a paper-based system for analyzing and processing data and calculating salaries.

I have used the MySQL to create a database and C# programming language to design the program, When replace a paper-based system with a computer-based system, we notice and conclude the following:

- reduce time and effort;
- reduced data redundancy;
- greater data integrity and improved data security;
- reduced updating errors and increased consistency;
- improved data access to users through use of host and query languages.



## REFERENCES

1. Bell C., Kindahl M., Thalmann L. MySQL High Availability 2<sup>nd</sup> Edition Tools for Building Robust Data Centers. – O'Reilly Media, 2014. – 762 p.
2. Cleary S. Concurrency in C# Cookbook Asynchronous, Parallel, and Multithreaded Programming. – O'Reilly Media, 2014. – 208 p.
3. Dubois P. MySQL Cookbook. Solutions & Examples for Database Developers and DBAs. 2<sup>nd</sup> ed. – O'Reilly Media, 2006. – 980 p.
4. Greene J., Stellman A. Head First C#, 3rd Edition A Learner's Guide to Real-World Programming with C#, XAML, and .NET. – O'Reilly Media, 2013. – 1100 p.
5. Griffiths I. Programming C# 5.0 Building Windows 8, Web, and Desktop Applications for the .NET 4.5 Framework. – O'Reilly Media, 2012. – 886 p.
6. Griffiths I., Adams M., Liberty J. Programming C# 4.0 Building Windows, Web, and RIA Applications for the .NET 4.0 Framework. – O'Reilly Media, 2010. – 858 p.
7. Harrison G., Feuerstein S. MySQL Stored Procedure Programming. – O'Reilly Media, 2006. – 640 p.
8. Liberty J., Xie D. Programming C# 3.0. Best-Selling Guide to Building Windows and Web Applications with C# 3.0. 5<sup>th</sup> ed. – O'Reilly Media, 2007. – 608 p.
9. Miles R., Hamilton K. Learning UML 2.0 A Pragmatic Introduction to UML. – O'Reilly Media, 2006. – 290 p.
10. Schwartz B., Zaitsev P., Tkachenko V. High Performance MySQL. Optimization, Backups, and Replication. 3<sup>rd</sup> ed. – O'Reilly Media, 2012. – 826 p.
11. Si Alhir S. UML in a Nutshell. – O'Reilly Media, 1998. – 296 p.
12. Stephens R. C# 24-Hour Trainer. 2<sup>nd</sup> ed. – O'Reilly Media, 2015. – 600 p.

13. Web site code project. [Electronic resource] URL:  
<http://www.codeproject.com/Tips/696864/Working-with-Excel-Using-Csharp>  
(the date of access: 23.02.2017).
14. Web site coderwall. [Electronic resource] URL:  
<https://coderwall.com/p/app3ya/read-excel-file-in-c> (the date of access:  
17.02.2017).
15. Web site creately. [Electronic resource] URL:  
<http://creately.com/blog/diagrams/uml-diagram-types-examples/> (the date of access:  
06.05.2016).
16. Web site Microsoft.com. [Electronic resource] URL:  
<https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx> (the date of access:  
11.04.2016).
17. Web site Net-informations.com. [Electronic resource] URL:  
<http://csharp.net-informations.com/excel/csharp-excel-tutorial.htm> (the date of  
access: 02.03.2017).
18. Web tutorial about MySQL. [Electronic resource] URL:  
<http://www.tutorialspoint.com/mysql> (the date of access: 12.01.2017).