MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION

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

THESIS IS CHECKED
Reviewer,
CEO of Render Farm Ltd.

_____ V.A. Dorokhov
11 June 2018

ACCEPTED FOR THE FENSE
Head of the department
Dr. Sci., Prof.

_____ L.B. Sokolinsky
“___”_____ 2018

DEVELOPMENT OF A COMPUTER-AIDED DOCUMENT MANAGEMENT SYSTEM FOR HUMAN RESOURCES DEPARTMENT OF UNIVERSITY OF KUFA

GRADUATE QUALIFICATION WORK
SUSU–02.04.02.2018.308-601.GQW

Supervisor
Cand. Sci., Assoc. Prof.

_____ P.S. Kostenetskiy

Author,
the student of the group CE-219

_____S.H.H. Al-Said Whab

Normative control

_____ O.N. Ivanova

“___”_____ 2018

Chelyabinsk–2018

CONTENTS

INTRODUCTION

Many well-known examples of using of information technology for competitive advantage include systems that can link an organization to suppliers, distribution and channels. Generally, these systems can use information or processing capabilities in one organization to improve the performance of another or to improve relationships among organizations. Declining costs of capturing and using information have joined with increasing competitive pressures to stimulate many changes in using information to create value. The ideas do not form a procedure leading inexorably to competitive advantage. Nevertheless, they have can have value when we combine them with an appreciation of the competitive dynamics of particular industries as well as grasping the information power.

To help promote employees achievement and success, university must have access to complete, accurate, and timely information about employees. One of the benefits of this system that the employee manage system will simplify retrieval of required information and is a great instrument for university improvement by taking measures from the information acquired [11]. It is also regarded as one of the simplest ways to prepare the employees' salaries' ladder in a balanced manner in companies and governmental institutions by determining a roof of specialization salary occupied by this profession and by assessing professions by their value.

Paper-based system, which is used to employees data management at the University of Kufa, requires great effort and a long time and large size to store, University of Kufa need to develop database system to manage employee data, provides access to full information, accurate, and timely about employees.

Databases (DB) are used to store collections of related data. Database management Systems (DBMS) are the underlying runtime environment for a database. A DBMS Provides a high-level language to define the structure of the data; known as the data definition language (DDL). In addition, DBMS have high-level languages to access and modify data in the database; this is the data manipulation language.

The Standard DML is the Structured Query Language (SQL), which is based on relational calculus Database access entails either: a request for data, i.e. a SQL SELECT statement or a modification of the data, i.e. SQL INSERT, UPDATE or DELETE statements. Programs that access the database are called transactions and are written in a data manipulation language such as SQL or in a procedural language with SQL extensions. Transactions are executed by the DBMS as one atomic unit.

Management system forms the backbone of every nation. Hence it is important to provide a strong educational foundation to the young generation to ensure the development of open-minded global citizens securing the future for everyone. Advanced technology available today can play a crucial role in streamlining education-related processes to promote solidarity among students, Professors and the university staff. Automation is the utilization of technology to replace human with a machine that can perform more quickly and more continuously. By automating documents that took up many large storage rooms can be stored on few disks. Transcript images can be annotated. It reduces the time to retrieve old transcripts from hours to seconds and it reduce the waste of money and effort in this area, And thus we get more accurate results, to manage our educational institutions [9].

For the management of a large database of dean office of university you must use application has a high potential in the management and analysis of these data to provide precise management and high secrecy of these data. I believe MS SQL server currently owns better specifications, because it has a large capacity and services are very good in data security and data management.

I have Developed application is provides a help to the Dean office of the College at the University of Kufa, because it has a high capacity to manage employees data by using MS SQL server, where stored the data in server, through eight tables and linked those tables by relationships (one to many), through the primary key and foreign key for each table. University of Kufa play an important role in the development of society.

Research goal and objectives

The goal of this thesis was designing of human resources management information system for University of Kufa.

To achieve this goal I have to carry out the following objectives:

- install and use the MySQL DBMS Server;
- design and create a database for storing university data;
- develop a database application to manage the data in the database.

Practical meaningfulness

1. Easy access to the data.
2. The new system is more user-friendly, reliable and flexible.
3. Data alteration is easy.
4. Maintenance of the project is easy.
5. Reduced manual work.
6. Timely Report generation.

The main goal of this system is to reduce the effort of the manager to maintain daily events such as payroll, Staff performance, study details and all student belongings.

1. DEVELOPMENT TOOLS

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 various web-based software applications. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. This tutorial will give you a quick start to MySQL and make you comfortable with MySQL programming [14].

MySQL [13]. Is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Wideness's daughter. In addition "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python [5]. Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google [6]. information Whatever the size of the organization or the type of information and take, for example an elementary school, the database to which they relate will include all the information related to the teachers and put this information in a special table where teachers This table record for each teacher, and also includes this database of all information related to the students and also where this information is classified in Tables Each student has its own record and other information vary from school to school as accomplished in the field of sports and scientific debates and cultural Facebook [10]. Twitter. Flickr. And YouTube.

1.2. PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994 [17]. the PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page [2]. but it now stands for the recursive acronym PHP: Hypertext Preprocessor.

Basically, PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "PHP: Hypertext Preprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications that can publish a blog, remotely control hardware, or run a powerful website such as Wikipedia or Wikibooks. Of course, to accomplish something such as this, you need a database application such as MySQL [16].

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

Basically, PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "PHP: Hypertext Preprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created.

During the 2010s, there have been increased efforts towards standardization and code sharing in PHP applications by projects such as PHP-FIG in the form of PSR-initiatives as well as Composer dependency manager and the Packagist repository. PHP hosts a diverse array of web frameworks requiring framework-specific knowledge, with Laravel recently emerging as a popular option by incorporating ideas made popular from other competing non-PHP web frameworks, like ruby on rails. PHP Admin is a free software tool written in PHP, intended to handle the administration of MySQL over the web. It supports a wide range of operations on [15].

MySQL and other DBMS. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc.) can be performed via the user interface, while the developer will have the ability to directly execute any SQL statement.

1.3. HTML

Hypertext Markup Language (HTML) [3]. Is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

Once you have obtained an HTML editor and are ready to begin setting up your website, think about how you want the site to look and be set up. Hyper Text Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages It is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, alt-hough some represent empty elements and so are unpaired, for example <img>. The first tag in such a pair is the start tag, and the second is the end tag [19].

HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another. HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look . HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements.

1.4. Normal Forms (DBMS)

A form in Access is a database object that you can use to create a user interface for a database application. A "bound" form is one that is directly connected to a data source such as a table or query, and can be used to enter, edit, or display data from that data source.

Codd introduced the concept of normalization and what is now known as the first normal form (1NF) in 1970 [12]. Codd went on to define the second normal form (2NF) and third normal form (3NF) in 1971. and Codd and Raymond F. Boyce defined the Boyce-Codd normal form (BCNF) in 1974.

Database normalization is process used to organize a database into tables and columns. The idea is that a table should be about a specific topic and that only those columns which support that topic are included. For example, a spreadsheet containing information about sales people and customers serves several purposes:

- identify sales people in your organization;
- list all customers your company calls upon to sell product;
- identify which sales people call on specific customers.

By limiting a table to one purpose, you reduce the number of duplicate data that is contained within your database, which helps eliminate some issues stemming from database modifications. To assist in achieving these objectives, some rules for database table organization have been developed. The stages of organization are called normal forms; there are three normal forms most databases adhere to using. As tables satisfy each successive normalization form, they become less prone to database modification anomalies and more focused toward a sole purpose or topic. Before we move on be sure you understand the definition of a database table. Here are the most commonly used normal forms:

- first normal form(1NF);
- second normal form(2NF);
- third normal form(3NF);
- Boyce & Codd normal form (BCNF).

First normal form (1NF):

Is a property of a relation in a relational database. A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain [7]. The first definition of the term, in a 1971 conference paper by Edgar Codd, defined a relation to be in first normal form when none of its domains have any sets as elements.

First normal form is an essential property of a relation in a relational database.

Database normalization is the process of representing a database in terms of relations in standard normal forms, where first normal is a minimal requirement.

First normal form enforces these criteria:

- eliminate repeating groups in individual tables;
- create a separate table for each set of related data;
- identify each set of related data with a primary key.

Second normal form (2NF):

Is a normal form used in database normalization? 2NF was originally defined by E.F. Codd. In 1971. A table that is in first normal form (1NF) must meet additional criteria if it is to qualify for second normal form. Specifically: a 1NF table is in 2NF if and only if, given any candidate key and any attribute that is not a constituent of a candidate key, the non-key attribute depends upon the whole of the candidate key rather than just a part of it. In slightly more formal terms: a 1NF table is in 2NF if and only if none of its non-prime attributes are functionally dependent on a part (proper subset) of a candidate key. (A non-prime attribute is one that does not belong to any candidate key.) Note that when a 1NF table has no composite candidate keys (candidate keys consisting of more than one attribute), the table is automatically in 2NF.

After meeting the requirements of 1NF, 2NF requires the database designer to do the following:

1. Split up all data resulting in many-to-many relationships and store the data as separate tables. For example, in a database used by a school's application, two of the tables are STUDENT and SUBJECT. In real life, a student takes several subjects simultaneously while a subject is studied by several students. These are many-to-many relationships. 2NF states that this relationship must be split into more than the two tables above (STUDENT and SUBJECT). One way of splitting them is by introducing a third table, which contains the columns Student_ID, Subject_ID, Semester and Year. In this way, there is no direct relationship between STUDENT and SUBJECT because all relationships are created indirectly through the third table.

2. Create relationships between tables by use of foreign keys. For example, a bank's database contains two tables: CUSTOMER_MASTER (for storing customer details) and ACCOUNT_MASTER (for storing details about bank accounts, including which customer holds which account). There must be a way to link the two tables to know who the customer is for each account. The way to do this is via a foreign key, which is a column in the ACCOUNT_MASTER table pointing to a corresponding column in the CUSTOMER_MASTER table.

Third normal form (3NF):

Is a normal form that is used in normalizing a database design to reduce the duplication of data and ensure referential integrity by ensuring that (1) the entity is in second normal form, and (2) all the attributes in a table are determined only by the candidate keys of that relation and not by any non-prime attributes. 3NF was designed to improve database processing while minimizing storage costs. 3NF data modeling was ideal for online transaction processing (OLTP) applications with heavy order entry type of needs.

Consider a bank's database, which contains two tables: CUSTOMER_MASTER for storing customer details and ACCOUNT_MASTER for storing details about bank accounts, including which customer holds which account. In this case, there needs to be a way to link the two tables in order to tie an account to the customer who owns it. The way to do this is via a foreign key.

This is a column in the ACCOUNT_MASTER table that points to or references a corresponding column (called the primary key) in the CUSTOMER_MASTER parent table. Let's call this column CustID.

Suppose that customer Andrew Smith creates an account in the CUSTOMER_MASTER table with CustID 20454. Mr. Smith holds a savings account with the number S-200802-005, whose details are stored in the ACCOUNT_MASTER table. This means that the ACCOUNT_MASTER table will have a column called CustID, which is not an original piece of data. Instead, it also has the value 20454, which simply references the same CustID in the CUSTOMER_MASTER table.

Now, 3NF dictates that in our ACCOUNT_MASTER table, the only information we hold about the customer should be the CustID (20454) as a foreign key, and it refers to and identifies the customer who owns this same CustID in the CUSTOMER_MASTER table (Andrew Smith). No other data about our customer (such as name, date of birth, gender and so on) should be stored in the ACCOUNT_MASTER table, or indeed any other table, because all this data about him is already stored in CUSTOMER_MASTER. By doing this, the only customer data stored outside the CUSTOMER_MASTER table is the CustID. This pays handsome dividends by ensuring there is no data duplication, which in turn makes queries run much more efficiently and reduces the amount of storage required.

Boyce–Codd normal form (or **BCNF** or **3.5NF**)**:**

Is a normal form used in database normalization? It is a slightly stronger version of the third normal form (3NF). BCNF was developed in 1974 by Raymond F. Boyce and Edgar F. Codd to address certain types of anomalies not dealt with by 3NF as originally defined.

If a relational schema is in BCNF then all redundancy based on functional dependency has been removed, although other types of redundancy may still exist.

A relational schema Ris in Boyce–Codd normal form if and only if for every one of its dependencies $X \rightarrow Y$, at least one of the following conditions hold

$X \rightarrow Y$ is a trivial functional dependency ($Y \subseteq X$);

X is a superkey for schema *R*.

Chris Date has pointed out that a definition of what we now know as BCNF appeared in a paper by Ian Heath in 1971 [8].

Since that definition predated Boyce and Codd's own definition by some three years.

It seems to me that BCNF ought by rights to be called Heath normal form. But it isn't.

Edgar F. Codd released his original paper "A Relational Model of Data for Large Shared Databanks" in June 1970. This was the first time the notion of a relational database was published. All work after this, including the Boyce-Codd normal form method was based on this relational model. Normalization is the process of efficiently organizing data in a database.

There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Reaching these two goals reduces the space used by the database and ensures the data is stored Guidelines have been developed to confirm that databases are normalized.These are referred to as normal forms and are numbered from one (the lowest form of normalization, referred to as first normal form or 1NF) through six (sixth normal form or 6NF).

2. RELATED SOFTWARE

Comparative analysis of existing analog sites

These programs are widely spread in Universities and in all of Iraq's institutions, in which they offer the time and effort for both the employee and the manager in receiving the salary we will review some of them.

Page University of Kufa

This site does not contain the salary employee page, the employee cannot see his salary, and the method used is old paper way

The page contains all information about colleges, students, activities, news and etc.

Therefore, the program is not integrated because it does not contain information about employees and salaries, through this program to be considered incomplete and open source and a weak program. See fig. 1.

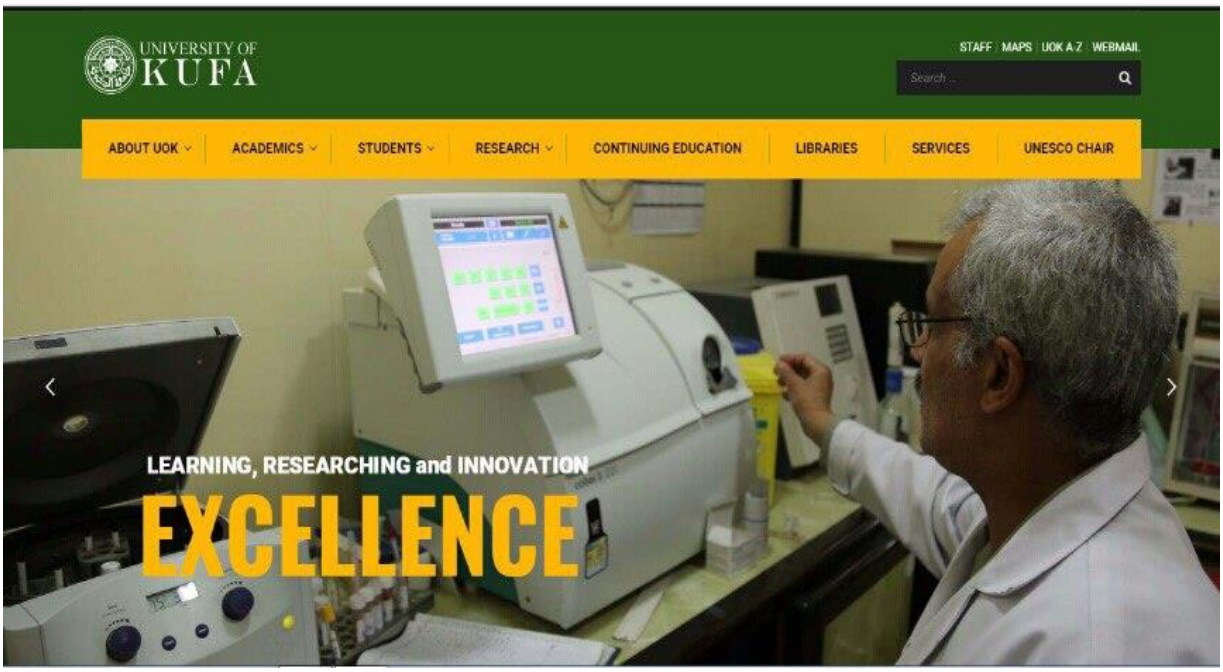

Fig. 1. Page university of Kufa

This program is designed by using the Visual Basic Dot Net programming language in addition to using another program, which is Microsoft Excel where a database can be created. It is one of the simplest and easiest programs and is considered a very slow program that does not give accurate results.

3. DESIGH SOFTWARE

3.1. Use Case Diagram

Use Case Diagram [1]. In software and systems engineering can be done by using case as a list of actions or event steps that are typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal. The actor either a human or external system. In systems engineering the use of cases are considered a high level than within software engineering often representing tasks or stakeholder goals. The detailed requirements may then be captured in the Systems Modeling Language (SysML) or as contractual statements.

For represent system functionality, the requirements of the system from the user's perspective we use Case model. Use Case is a sequence of steps describing an interaction between a user and a system. Use cases describe the way the user will interact with the system and how the system will respond will often be accompanied by other types of diagrams as well. While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that "Use case diagrams are the blueprints for your system [18]. They provide the simplified and graphical representation of what the system must actually do. Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed. Every software construction needs a plan, for this reason, it was necessary to develop a plan of action [20].

The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed. Every software construction needs a plan, for this reason, it was necessary to develop a plan of action, in order that the work is correct and free of problems and programmatic iterations, and for this reason were used UML modeling language. UML is called a modeling language, not a method. Most methods consist, at least in principle, of

both a modeling language and a process .For represent system functionality, the requirements of the system from the user's perspective we use Case model [4]. Use Case is a sequence of steps describing an interaction between a user and a system. Use cases describe the way the user will interact with the system and how the system will respond. Fig. 2 shows the use case diagram.

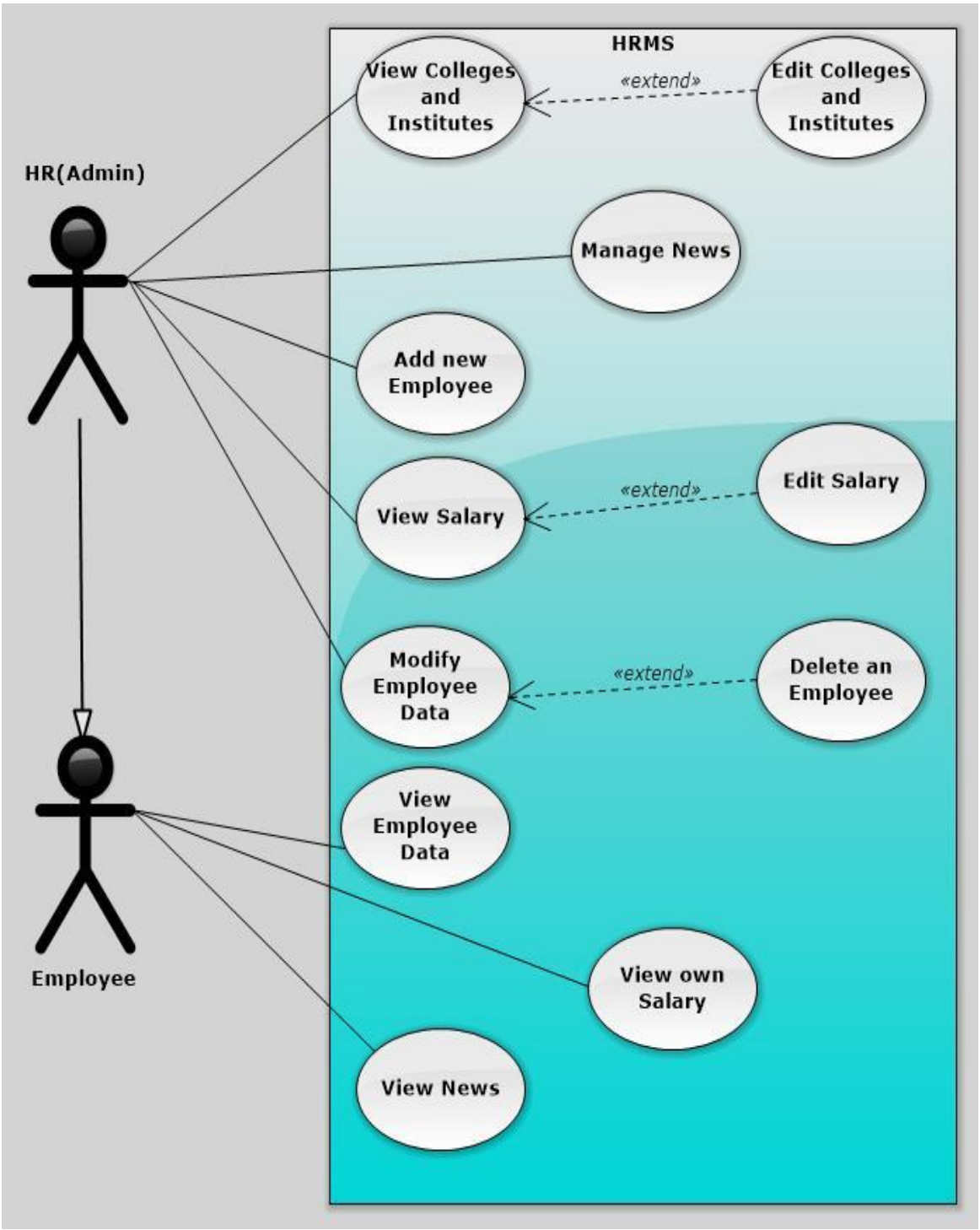

Fig. 2. Use case diagram

I have two actors of users in my program (HR admin, employee), each one has specific powers.

For instance, in this entity relationship diagram there is a connection among admin (manager) and discipline, employee, salary. While binary relationships, those between two entity sets, are by far the most common type of relationship.

The use case diagram for employees consists of the following entities:

Admin stands for employees who work in the university, they have Properties, which are:

- View colleges and institutes;

Through this section, the Admin can see the information of the colleges and institutions entered into the database.

- Edit colleges and institutes;

Through this section, the administrator can enter the information of colleges and institutions into the database.

- Manage news;

Through this section, the administrator can enter news from the university to the database.

- Add new Employee;

Through this section, the administrator can insert a new employee into the database.

- View salary;

Through this section, the administrator can view salary of all employees.

- Edit salary;

Through this section, the administrator can edit salary of employee.

- View employee data;

Through this section, the administrator view employee data of any employees.

- Modify employee data;

Through this section, the administrator modifies personal data of all employees.

- Delete an employee;

Through this section, the administrator can delete employee after discharge.

- View employee data;

Through this section, the administrator view data of all employees.

Employee can see his a career and a salary, it has some Properties:

- View employee data;

Through this section, the employee views his own personal data

- View news;

Through this section, the employee sees news of a university.

- View own salary;

Through this section, the employee sees his own salary.

3.2. User Interface of the system

The user interface or human–machine interface is considered a part of the machine that handles the interaction of the human-machine. Membrane switches, rubber keypads and touchscreens are examples of the physical part of the Human Machine Interface, which we can see, and touch.

In complex systems, the human–machine interface is typically computerized. The term human–computer interface refers to this kind of system.

In the context of computing, the term typically extends as well to the software dedicated to control the physical elements used for human-computer interaction.

The user interface of system is a vehicle or an industrial installation, which is sometimes called to as the human–machine interface (HMI).

HMI is a modification of the original term MMI (man-machine interface) in practice, the abbreviation MMI is still frequently used although some may argue that MMI represents something different now. Another abbreviation is HCI, but it is genrally used for human–computer interaction Other terms are used as operator interface console (OIC) and operator interface terminal (OIT) Nevertheless, it is abbreviated, the terms indicate the 'layer' that can split a human

that is operating a machine from the machine itself. Without a clean and usable interface, humans would not be able to interact with information systems.

Generally, the goal of user interface design is to produce a user interface which makes it easy (self-explanatory), efficient, and enjoyable (user-friendly) to operate a machine in the way which produces the desired result.

This generally means that the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human.

User interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing the user experience.

The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals (user-centered design).

A form is a database object that we can use to create a user interface for a database application. A "bound" form is one that is directly connected to a data source such as a table or query, and can be used to enter, edit, or display data from that data source. Alternatively, we can create an "unbound" form that does not link directly to a data source, but which still contains command buttons, labels, or other controls that you need to operate our application.

With the increased use of personal computers and the relative decline in societal awareness of heavy machinery, the term user interface is generally assumed to mean the graphical user interface, while industrial control panel and machinery control design discussions more commonly refer to human-machine interfaces.

Other terms for user interface are man–machine interface (MMI) and when the machine in question is a computer human–computer interface.

In my system there are 10 forms, each form has a certain function, such as it is shown in fig. 3.

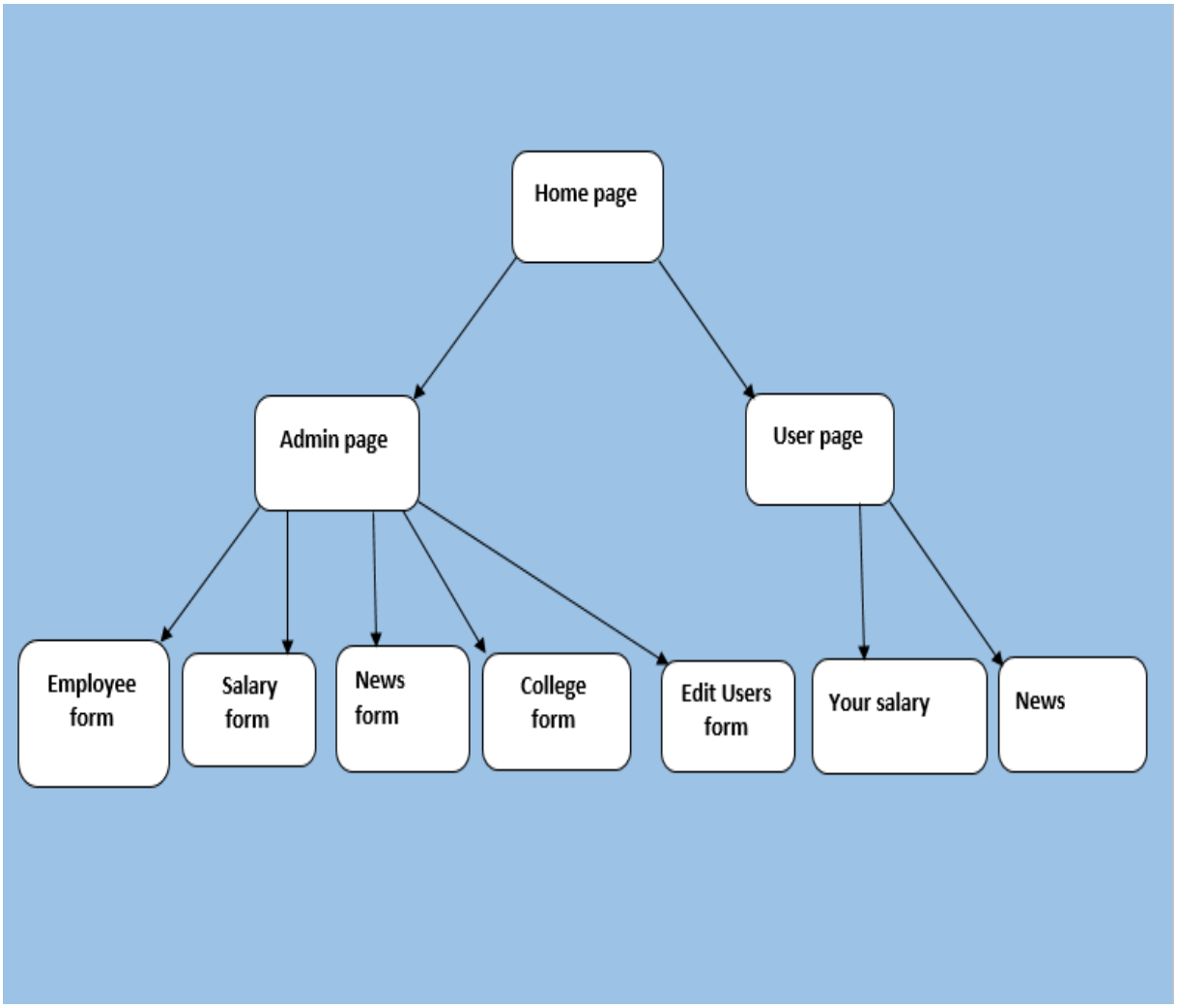

Fig. 3. User interface of the system

3.3. Description of the Database

A database is an organized collection of data A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements. Database designers typically organize the data to model aspects of reality in a way that supports processes requiring information, such as (for example) modeling the availability of rooms in hotels in a way that supports finding a hotel with vacancies.

A database-management system (DBMS) is a computer-software application that interacts with end-users, other applications, and the database itself to capture and analyze data.

A general-purpose DBMS allows the definition, creation, querying, update, and administration of databases.

A database is not generally portable across different DBMSs, but different DBMSs can interoperate by using standards such as SQL and ODBC or JDBC to allow a single application to work with more than one DBMS. Computer scientists may classify database-management systems according to the database models that they support; the most popular database systems since the 1980s have all supported the relational model - generally associated with the SQL language. Sometimes a DBMS is loosely referred to as a "database".

3.4. The Database Scheme

The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language.The states of a created conceptual schema are transformed into an explicit mapping, the database schema. A database schema specifies, based on the database administrator's knowledge of possible applications, the facts that can enter the database, or those of interest to the possible end-users. The notion of a database schema plays the same role as the notion of theory in predicate calculus. A model of this "theory" closely corresponds to a database, which can be seen at any instant of time as a mathematical object. Thus a schema can contain formulas representing integrity constraints specifically for an application and the constraints specifically for a type of database, all expressed in the same database language In a relational database, the schema defines the tables, fields, relationships, views, indexes, packages, procedures, functions, queues, triggers, types, sequences, materialized views, synonyms, database links, directories, XML schemas, and other elements.

. In other words, schema is the structure of the database that defines the objects in the database.

A model of this “theory” closely corresponds to a database, which can be seen at any instant of time as a mathematical object. In my database scheme, I have eight tables and each table contains detailed information. Fig. 4 shows my database.

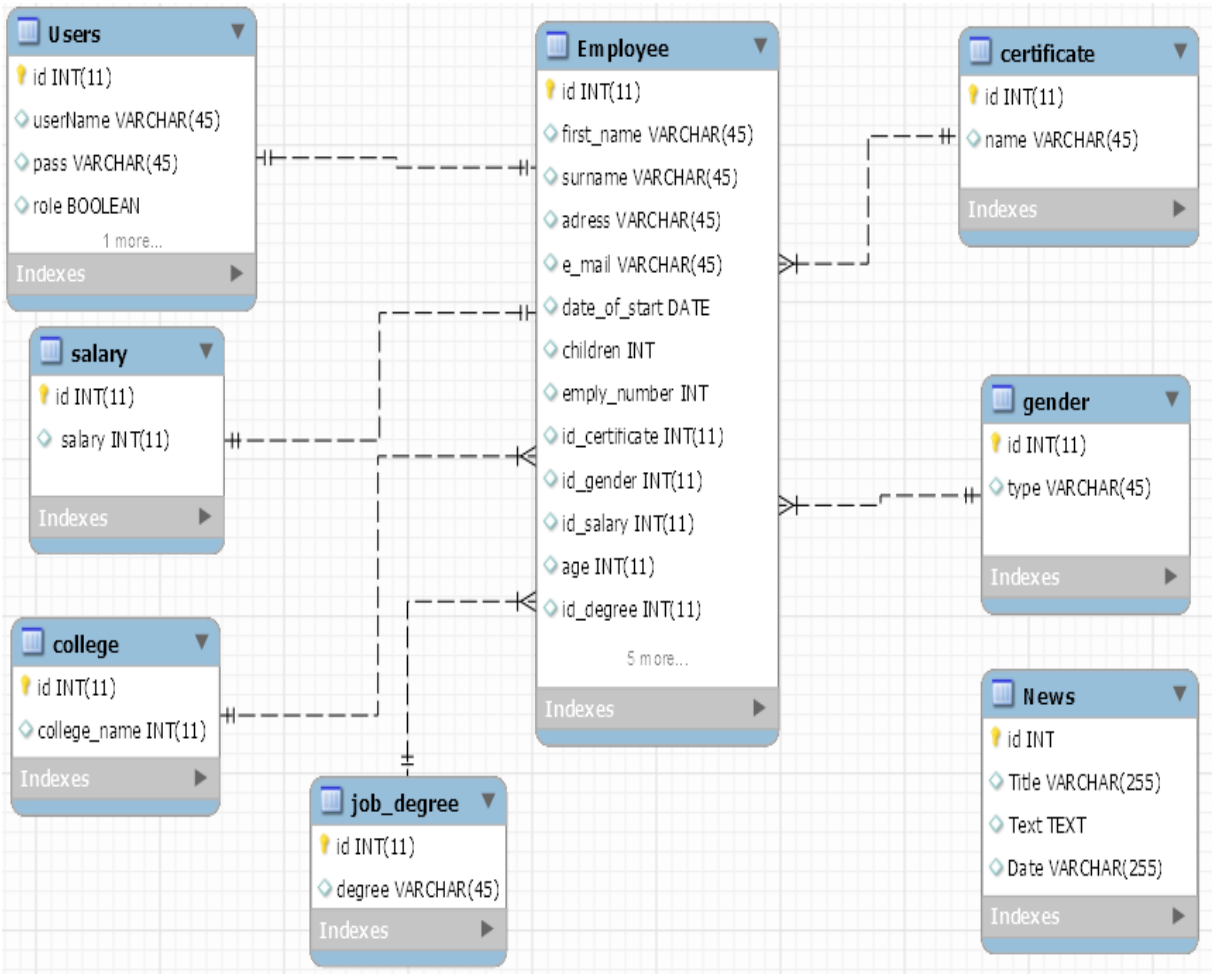

Fig. 4. Database schema

In my database I have 8 tables (User, Employee, Certificate, salary, college, Job degree, gender and news) each table contains the following.

The first table: Users tables contains the following.

1. Id the primary key for employee.
2. Username for user.

3. Password for user.

4. Role. The user password table is shown in fig. 5.

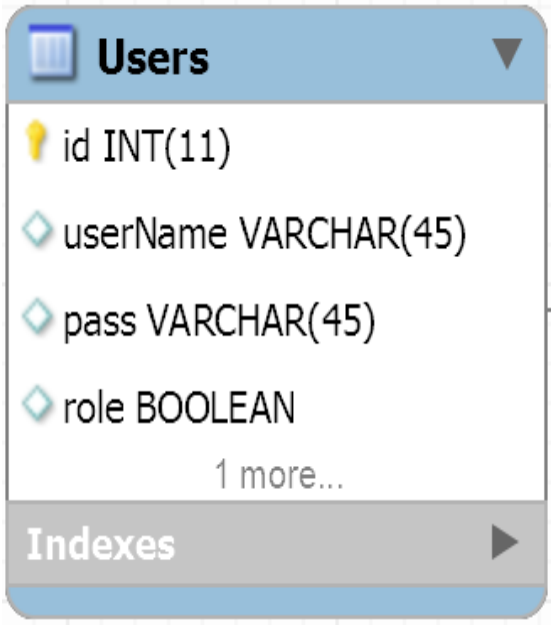

Fig. 5. User's password table

The second table: Employee tables contains the following.

1. Id the primary key for employee table.
2. First name the first name for employee who works in university.
3. Surname the second name father name for the employee.
4. Address the address for employee.
5. E-mail the email for employee.
6. Id_degree the foreign key for job degree table.
7. Data_of_star the family name for employee.
8. Id_Salary the foreign key for salary table.
9. Id_gender the foreign key for gender table.
10. Children the number of children of employee.
11. Emply_number the number of employee.
12. Id_ certificate the foreign key for certificate table.
13. Age the old of employee.

The employees table is shown in fig. 6.

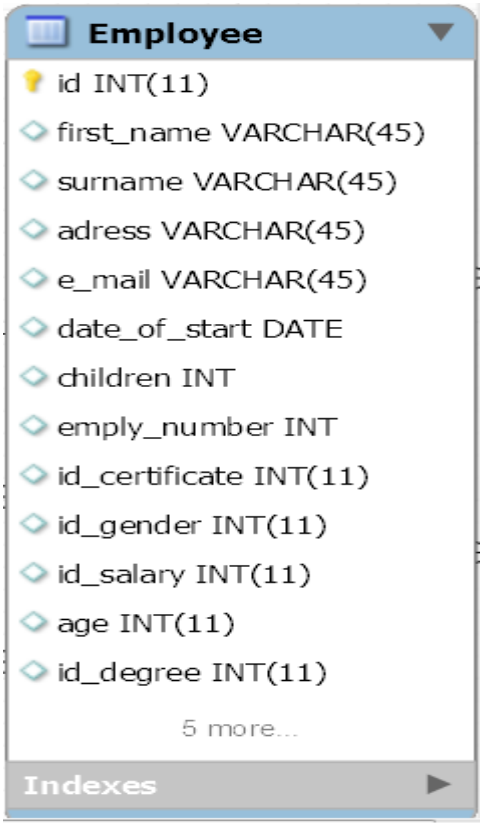

Fig. 6. Employee table

The third table: The salary tables contains the following.

1. Id the primary key for salary table.
2. Salary the basic salary for employee, shown in fig. 7.

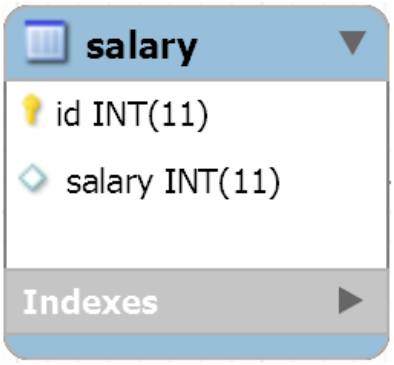

Fig. 7. Salary table

The fourth table: The job degree tables contains the following.

1. Id the primary key for table job degree.
2. Degree the type of job degree for every employee. Table is shown in fig. 8.

Fig. 8. Job degree table

The Fifth table: the certificate tables contains the following.

1. Id the primary key for certificate.
2. Certificate of employee. This is shown in fig. 9.

Fig. 9. Certificate

The table 6: The gender tables contains the following.

1. Id the primary key for gender;
2. Type the type gender of employee. This is shown in fig. 10.

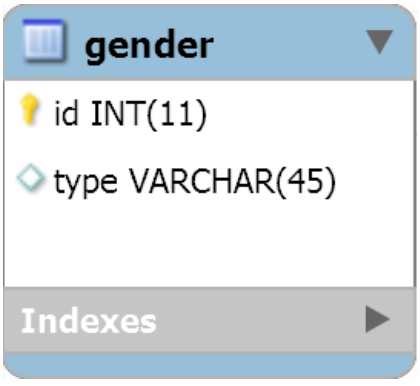

Fig. 10. Gender table

The table 7: the college tables contains the following.

1. Id the primary key for college.
2. College_name the name of college. This is shown in fig. 11.

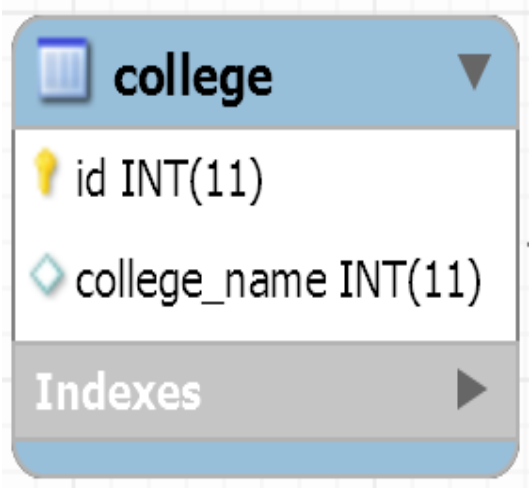

Fig. 11. College table

The table 8: the news tables contains the following.

1. Id the primary key for news.
2. Title the name of title.
3. Text the text of news.
4. Data the data of news. This is shown in fig. 12.

Fig. 12. News table

3.5. Backend Implementation

The backend is implemented as files in PHP language, they implement these main function:

1. User authentication;
2. Database connection;
3. Getting and updating information of employee;
4. Show the salary SQL-query.

User authentication

User authentication is implemented by the SQL-query, which connects to the tables, checks the user, and determines the next page fig. 13.

```
$myusername = $_POST['user'];
$mypassword = $_POST['pass'];
$myusername = stripslashes($myusername);
$mypassword = stripslashes($mypassword);
$query = "SELECT * FROM users WHERE Username='$myusername' and Password='$mypassword'";
$result = mysql_query($query);
if (!$result) { die('invalid query: ' . mysql_error()); } $count = mysql_num_rows($result);
mysql_close();
if($count > 0) { if (mysql_fetch_array($result)['idusers'] == 1){ header ("location:idex.php?login=errorpass");}
else { header ("location:student1.php");
// this is a student}}else{ echo 'Incorrect username or password';
header("location:index.php?login=success"); }
```

Fig. 13. Check of user and powers

Database connection

The backend gets it database connection using SQL queries, such as it is shown in fig. 14.

```
$dbhandle = @mysql_connect($hostname, $username, $password) or die ("could not connect to database: " . mysql_error());
if (!$dbhandle){ die('connection to database failed: ' . mysql_error());} $selected = mysql_select_db("kufa");
if (!$selected) { die('failed to select db: ' . mysql_error());}
```

Fig. 14. Fetch code connect to the database

Getting and updating information of employee

The data is retrieved and updated using SQL queries, such as it are shown in fig. 15.

```
$getemp = mysql_query("SELECT * FROM users WHERE type=0");
if(isset($_POST['submit'])){
$query = mysql_query("INSERT INTO employee (`firstname`, `lastname`, `adress`, `email`, `date_of_start`, `photo`, `childrens`, `employ_number`, `gender`, `id_user`) VALUES
('$postfirstname','$postlastname','$postadress','$postemail','$postdate_of_start','$postphoto','$postchildrens','$postemploy_number','$postgender','$postid_user')");
if($query){
    echo "Employee Added In Database";
     }else{
     echo mysql_error();}
```

Fig. 15. Insert data into the database

Show the salary SQL-query

The data is select and show salary using SQL queries, such as it are shown in fig. 16.

```
$iduser = $_GET['id'];
$getsalary= mysql_query("SELECT * FROM salary WHERE id_user = $iduser");
$row = mysql_fetch_array($getsalary);
$selectemployee = mysql_query("SELECT * FROM employee WHERE id_user = $iduser");
$rowemp = mysql_fetch_assoc($selectemployee);
```

Fig. 16. Get the salary from database

4. WEB APPLICATION AND TESTING

4.1. Web Application

User interface design is the design of websites and software applications with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals. The University page is the login page as shown in the fig. 17. This enables the manager and employee to sign in to the program. Depending on the type of user you will be going to the desired page and all user can see the main page (login page); the main page has a password and username. The user who can enter to the main page must be admin, employee, and they must enter a correct password and username.

Fig. 17. The home page

When the user is an admin will be going to the Settings page, and if either an employee will be moving to a page employee. On the Control Panel or Settings page, which contains the following options (Home page, Add Employee, Edit Salary, Add News, Add Colleges, Edit Users, Personal Information, Your salary and Colleges), such as it is shown in fig. 18.

Fig. 18. Control panel page

On the Settings page, the admin can add an employee, add or update a salary to the employee, add or update news about the university or colleges, add colleges, edit users, site setting and personal information.

When "Admin" login on the home page it will go to the settings page, we will find the second option "add employees".

When you click this option, the following window appears called "Add New Employee" which contains a number of fields and buttons: (Username, Password, First Name, last name, Gender, Children's, Employee Number,

Data of start, Address, Email, salary, Job degree, Certificate, Type). It enables the supervisor to add new employee information, see fig. 19.

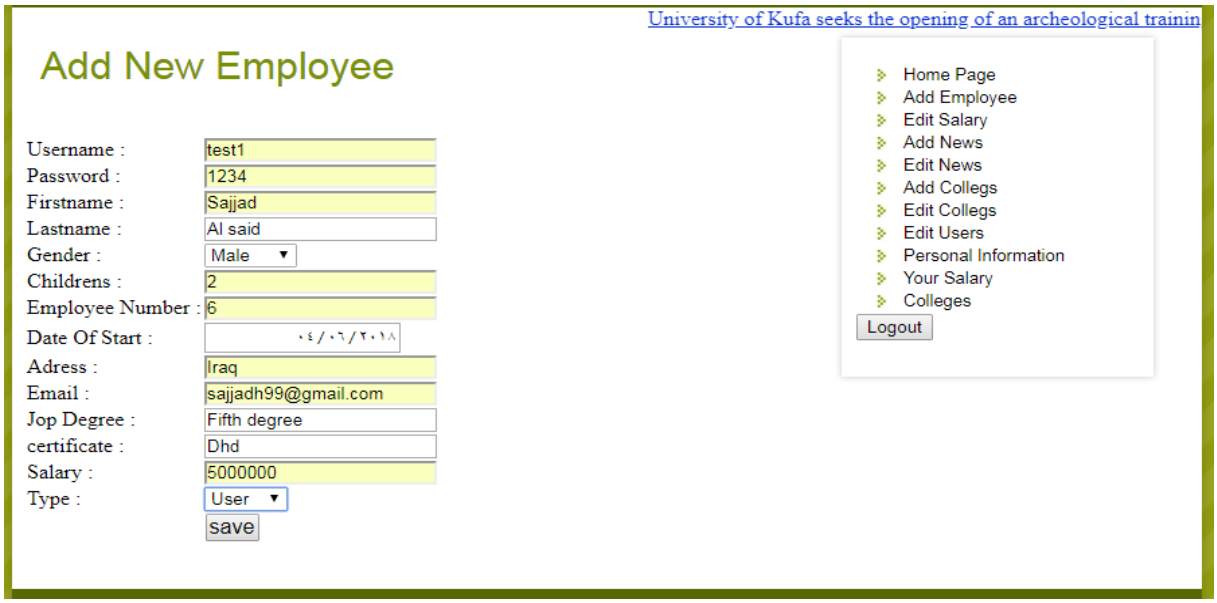

Fig. 19. Add new Employee

When "Admin" login on the home page it will go to the settings page, we will find the third option "Edit salary". This page shows a list of employees, “ADD” button next to each employee and button «select the user» and two forms «salary, Adds», and button "Save" see fig. 20.

Fig. 20. Add salary

The fourth options on the Settings page to add news about the university and contain (News title, News body and Data news) see fig. 21.

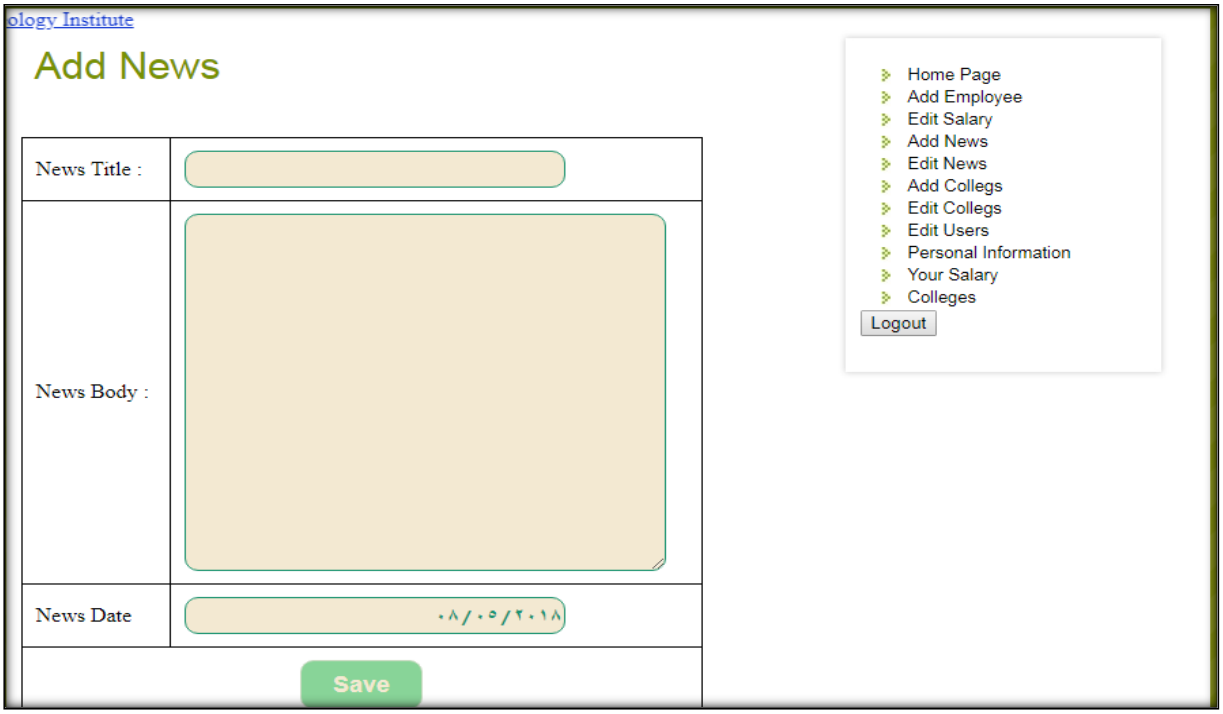

Fig. 21. Add news

The fifth option in the settings page "Add College" and contains (College name and Description) see fig. 22.

Fig. 22. Add College

The sixth options in the settings page "Edit user" This page is to modify user information and contains (First name, last name and e-mail) see fig. 23.

Fig. 23. Edit user

The seventh option on "personal Information" settings page contains (first name, last name, e-mail,certificate and children) see fig. 24.

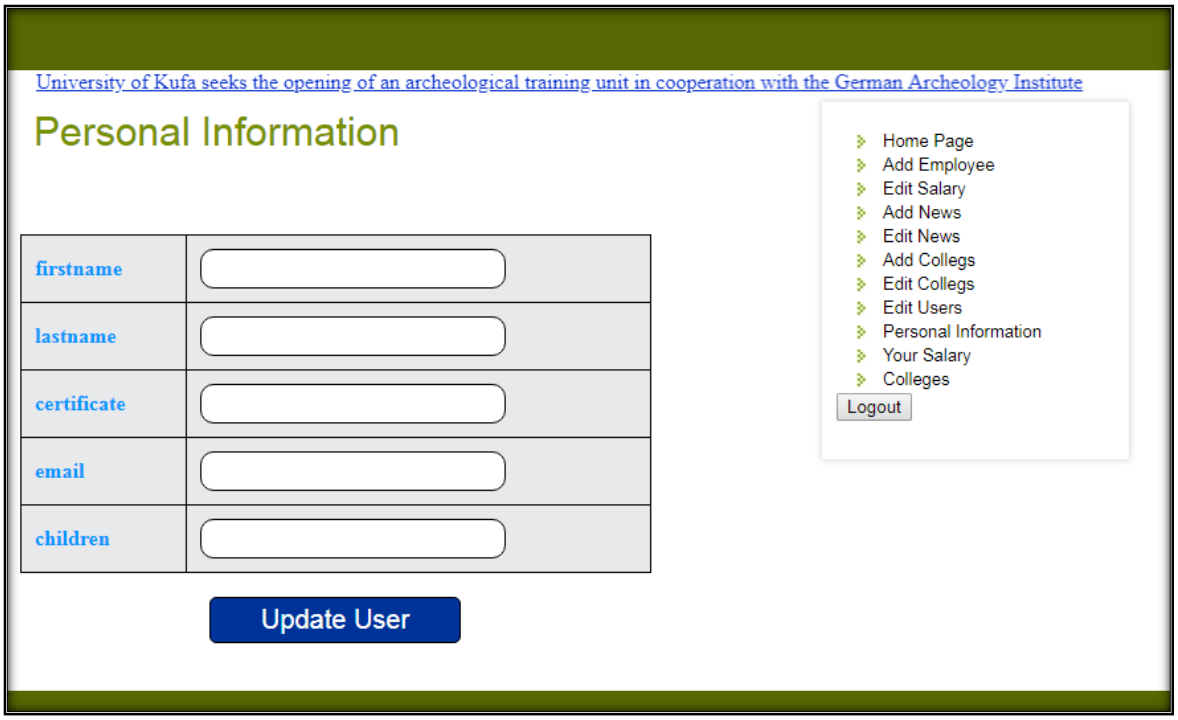

Fig. 24. Personal information

On the employee page, the employee can see his salary, see fig. 25.

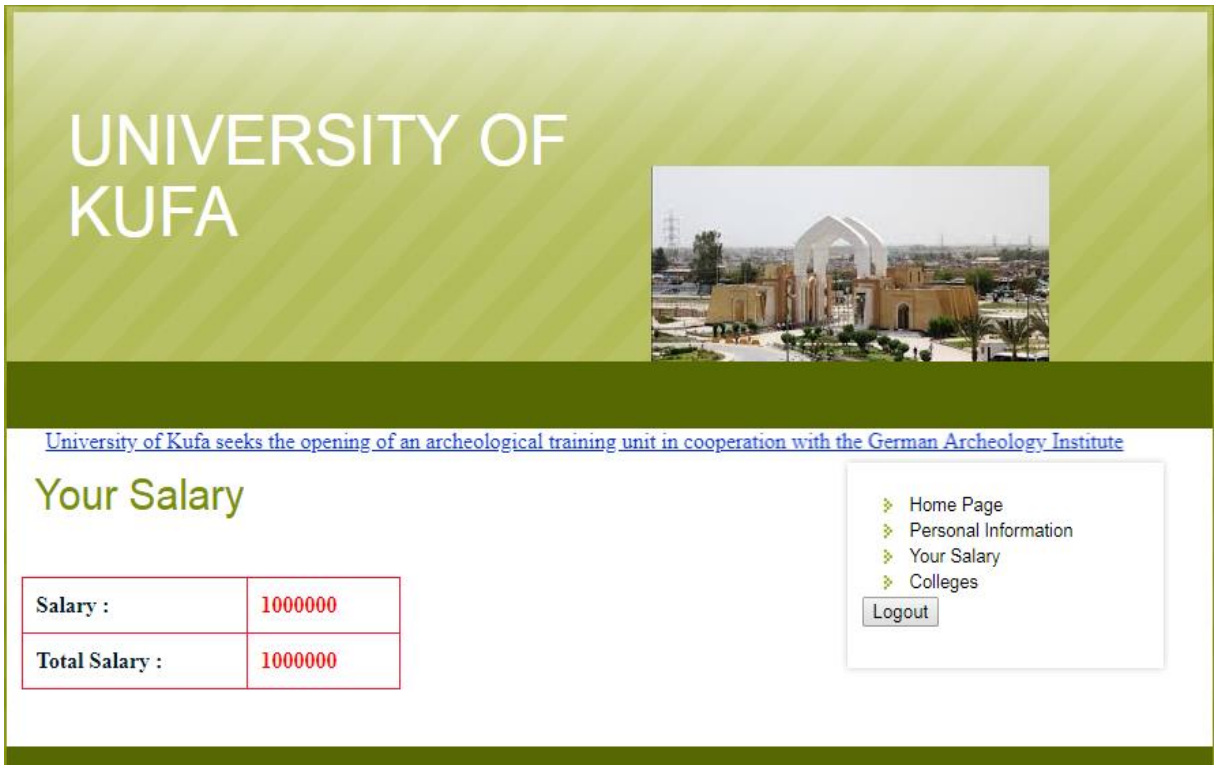

Fig. 25. Page of employee

College's page: everyone can see this page, and contain the names and information of the colleges, see fig. 26.

Fig. 26. The college page

4.2. Testing

Functional system tests should be based around coverage of the functionality described in the requirements, but it is common for the design document to be used as the baseline for testing because the requirements cannot be related to the product.

Each test of my system contains input and output information. Therefore, we compare the actual results and the expected results. Show in to table 1.

Table 1. Admin interface testing

No	Test case	Test steps	Expected Result	Actual results
1.	When the admin enters the user name and Password properly	1. Enter "Admin" in name 2. Enter password	Open the Admin page	The function works correctly
2.	When the admin Enters username and password improperly	1. Enter "Admin" in name 2. Enter not correct password	System should prompt the user to enter valid values.	The function works correctly
3.	If the admin enters incorrect username and password	1. Enter not Correct Name 2. Enter password	System should prompt the user to enter valid values.	The function works correctly

Continuation of table 1

No	Test case	Test steps	Expected Result	Actual results
4.	If the admin enters incorrect username and password	1. Enter not Correct Name 2. Enter password	System should prompt the user to enter valid values	The function works correctly
5.	When admin add salary	The user "Admin" add salary	System should prompt the user add salary	The function works correctly
6.	When admin add Employee	The user" Admin" add Employee	System should prompt the user add Employee	The function works correctly
7.	When admin add News	The user" Admin" add News	System should prompt the user add News	The function works correctly
8.	When admin add College	The user "Admin" add College	System should prompt the user add College	The function works correctly

Continuation of table 1

No	Test case	Test steps	Expected Result	Actual results
9.	When admin Edit Users	The user "Admin" add Edit Users	System should prompt the user Edit Users	The function works correctly
10.	When admin enter numbers in salary filled	1.Entered number 100 2.Entered number 1000 3.Entered number 10000	System should prompt the user correct	The function works correctly
11.	When admin enter wrong numbers in Salary filled	1.Entered error number "1f00ff" 2.Entered error number "Test" 3.Entered error number "1A"	System should prompt the user correct the error	The function works correctly
12.	When admin enter number in children filled	1.Entered 1 2.Entered 3	System should prompt the user correct	The function works correctly
13.	When admin enter wrong number in children filled	1. Entered -1 2. Entered 3a2	System should prompt the user incorrect	The function works correctly

End of table 1

No	Test case	Test steps	Expected Result	Actual results
14.	When admin enter in Email address filled	1. Entered Sshh123@Gmail.com 2. Entered Sajjad32@Yahoo.com	System should prompt the user incorrect	The function works correctly
15.	When admin enter in Email address filled	1. Entered sshh@Yahoo 2. Entered nn	System should prompt the user incorrect	The function works correctly

Table 2. The user interface testing

No	Test case	Test steps	Expected result	Actual results
1.	IF the user enters valid username and valid password	1.Enter username: user 2. Enter password: user	Open the User page	The function works correctly
2.	When the user enters password improperly	1.Enter username: user 2. Enter password: 1234	System should prompt the user to enter valid values	The function works correctly

End of the table 2

No	Test case	Test steps	Expected Result	Actual results
	When the user enters username improperly	1.Enter username: 1234 2. Enter password: user	System should prompt the user to enter valid values.	The function works correctly
4.	When user wants to go to personal information page	The user press “personal information” button Enter	He will see the personal information for employee	The function works correctly
5.	When user wants to go to your salary page	The user press “your salary”button. Enter	He will see the salary	The function works correctly

CONCLUSIONS

In the Republic of Iraq, there are systems of calculating salaries. However, these systems are old and outdated as they cause many problems when working on them. To solve these problems, it is an important task to design a new program "Computer-Aided Document Management System for Human Resources Department of University of Kufa" to calculate salaries and make this process much easier. This program is different from the older ones in which it presents new options such as calculating the salaries of the employees, their days of absence and a lot of other information about them. This process will help others interested in the work of the University of Kufa itself such as accountants, directors, legal experts and governmental inspectors. This is very useful for these people mentioned above to know for example how much money every employee gets at the end of every month.

During the process of developing this program, I managed to solve the following tasks:

- install and use the MySQL DBMS Server;
- design and create a database for storing university data;
- develop a database application to manage the data in the database.

The system "Computer-Aided Document Management System for Human Resources Department of University of Kufa" has been implemented and tested.

Therefore, based on this assumption, I believe that this program can be developed in the future to cover other governmental offices and establishments making the process of calculating salaries easier and more practical. It could also be developed to perform other useful tasks in the near future.

REFERENCES

1. Alshuhai F., Siewe F. An extension of the use case diagram to model context-aware applications., An IntelliSys 2015 - Proceedings of 2015 SAI Intelligent Systems Conference. 2015. – P. 884–888.

2. Arhaei A. Evolution of the World Wide Web : From Web 1.0 to Web 4.0. Int. J. Web Semant. Technol .2012. –Vol. 3. – No. 1. – P. 1–10.

3. Benson E.O., Karger D.R. Cascading tree sheets and recombinant HTML in Proceedings of the 22nd international conference on World Wide Web – WWW., 2013. – P. 107–118.

4. Bittner K. Spence I. Use Case Modeling. – USA: Pearson Education 2003. – 326 p.

5. Delisle M. Creating your MySQL Database: Practical Design Tips and Techniques. – UK: Packt Publishing Ltd. 2006. – 105 p.

6. Filaretti D. Maffeis S. An executable formal semantics of PHP. // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)., 2014. – Vol. 8586. – P. 567–592.

7. Geim A. K. Novoselov K.S. The rise of graphene .2007. Nat. Mater. – Vol. 6. – No. 3. – P. 183–191.

8. Graulis S. Crystallography Open Database - An open-access collection of crystal structures J. Appl. Crystallogr .2009. – Vol. 42. – No. 4. – P. 726–729.

9. He L. Chen Y. Design and implementation of Web Content Management System by J2EE-based three-tier architecture -Applying in maritime and shipping business. // ICIME 2010 - 2010 2nd IEEE International Conference on Information Management and Engineering., 2010. – Vol. 2. – P. 513–517.

10. Hew K.F. Students and teachers use of Facebook. Computers in Human Behavior .2011. – Vol. 27. – No. 2. – P. 662–676.

11. Kirschner P.A., van Merriënboer, Do Learners Really Know Best? Urban Legends in Education // Educ. Psychol., 2013. – Vol. 48. – No. 3. – P 169-183.

12. Nambiar K.K. Gopinath B. Nagaraj T., S. Manjunath, Boyce-Codd normal form decomposition, Comput. Math. with Appl .1997. – Vol. 33, – No. 4. – P. 1–3.

13. Official site of MySQL Server. [Electronic Resource] URL: http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html (the date of access: 15.10.2018).

14. Oracle Corporation. MySQL :: MySQL Community Edition. mysql.com, 2017. [Electronic Resource]: URL https://www.mysql.com/products/community/. (the date of access: 10.05.2018).

15. Osman R.I.M. Performance Modeling of Database Designs using a Queuing Networks Approach. Thesis PhD. – UK: University of Bradford., 2010. – 208 p.

16. Philip G.C. Teaching Database Modeling and Design : Areas of Confusion and Helpful Hints Practical Significance of Normalization. Inf. Technol. Educ .2007. Vol. 6. P. 481–497.

17. phpMyAdmin. [Electronic Resource] URL: www.phpmyadmin.net (the date of access: 10.05.2018).

18. Rosenberg D. Stephens M. Use Case Driven Object Modeling with UML. Theory and Practice. – USA: Doug Rosenberg and Matt Stephens., 2007. – 425 p.

19. W3C: HTML. [Electronic recourse] URL: http://www.w3.org/html/ (the date of access: 10.04.2017).

20. Yue T., Briand L.C. Labiche Y. Facilitating the Transition from Use Case Models to Analysis Models: Approach and Experiments. ACM Trans. Softw. Eng. Methodol .2013. – Vol. 22. – No. 1. – P. 1–38.