DEVELOPMENT OF WEB-SITE FOR DEPARTMENT OF SCIENTIFIC AFFAIRS AND CULTURAL COMMUNICATIONS IN THE UNIVERSITY OF TECHNOLOGY (BAGHDAD)

GRADUATE QUALIFICATION WORK
SUSU–02.04.2017.115-141.GQW

Supervisor
Cand. Sci., Assoc. Prof.
__________ O.N. Ivanova

Author,
the student of the group CE-216
__________ Z.M.H. Alaqili

Normative control
__________ O.N. Ivanova
“___”___________ 2017

Chelyabinsk–2017
1. **The topic** (approved by the order of the rector from 28.04.2017 No. 835)

   Development of web-site for department of scientific affairs and cultural communications in the University of Technology (Baghdad).

2. **The deadline for the completion of the work:** 01.06.2017.

3. **The source data for the work**

   
   

4. **The list of the development issues**

   4.1. To analyze the subject area.
   
   4.2. To implement the problem statement.
   
   4.3. To implement the comparative analysis of analogues.
   
   4.4. To make analysis of modern tools of web sites development and choose tools for the project development.
4.5. To design the database.
4.6. To design the interfaces.
4.7. To implement the description of the process of web-application development in CMS Joomla!
4.8. To implement and test the Web-site for Department of Scientific Affairs and Cultural Communications in the University of Technology (Baghdad).

5. **Issuance date of the task:** 09.02.2017.

**Supervisor**

Cand. Sci., Assoc. Prof. O.N. Ivanova

**The task is taken to perform**

Z.M.H. Alaqili
TABLE OF CONTENTS

ACKNOWLEDGMENTS 5

INTRODUCTION 6

1. THE ANALYSIS OF THE SUBJECT AREA 9
   1.1. The problem statement .............................................................................. 9
   1.2. Comparative analysis of analogues .............................................................. 10
   1.3. Used development tools ............................................................................. 11

2. DESIGN OF WEB-SITE FOR DEPARTMENT OF SCIENTIFIC AFFAIRS
   AND CULTURAL COMMUNICATIONS IN THE UNIVERSITY OF
   TECHNOLOGY (BAGHDAD) 15
   2.1. Functional and non-functional requirements ............................................ 15
   2.2. Designing of structure and behavior of the web-application .................... 16
       2.2.1. Use case diagram ............................................................................... 16
       2.2.2. Deployment diagram .......................................................................... 17
       2.2.3. State machine diagram ...................................................................... 18
   2.3. Database scheme ....................................................................................... 19
   2.4. Design of interface .................................................................................... 23

3. IMPLEMENTATION OF THE WEB-APPLICATION 35
   3.1. Description of the process of web-application development in CMS Joomla! 35
   3.2. Fragments of code ..................................................................................... 36

4. TESTING OF THE WEB-APPLICATION 41
   4.1. Functional testing ...................................................................................... 41
   4.2. Usability testing ......................................................................................... 46

CONCLUSION 49

LIST OF REFERENCES 51
ACKNOWLEDGMENTS

I would like to express my thanks and gratitude to the South Ural State University and University of Technology (Baghdad), which gave me the opportunity to obtain a master’s degree from Russia, and I would like to extend my sincerest thanks and appreciation to my supervisor PHD, associate professor, O.N. Ivanova for her support and dedication to her work.

I extend a lot of thanks and grateful to the most important people in my life my father (mercy and immortality to him), mother, brothers and sisters, my best friends Zaineb and Noor for their continued encouragement, support and patience.

Finally many thanks to my friends and anyone who helped me directly or indirectly to overcome obstacles.
INTRODUCTION

Actuality

Colleges and universities used modern information technologies. They have generally been quick to adopt new technologies. One of the major responsibilities of the university towards its students is to issue their documents because the University deals with the students from the first day of the scholarship. It is often difficult for the student to keep tracking issuing their documents. On the other hand, it is difficult for the employee to answer all the students’ questions about the readiness of their documents. Additionally, the time factor is important in our lives. Therefore, we need a system that makes this process easier for students and staff. The best way to implement this system is the use of the website connecting it to a database so as to facilitate the system access by everyone from anywhere.

This system will reduce the efforts and load of the staff of the department of scientific affairs and cultural communications by providing simple interfaces for all possible users.

Research goal and objectives

The goal of the research is the Development of web-site for department of scientific affairs and cultural communications in the University of Technology (Baghdad).

For the reaching this goal we must solve the following objectives:
1) to make a problem statement;
2) to make a comparative analysis of analogues;
3) to make analysis of modern tools of web sites development and choose tools for project development;
4) to design the structure and behavior of the web-application;
5) to design database scheme;
6) to design the interfaces;
7) to implement the description of the process of web-application development in CMS Joomla!

8) to implement and test the Web-site for Department of Scientific Affairs and Cultural Communications in the University of Technology (Baghdad).

**The practical significance**

This project will be useful not only to the University of Technology but to other universities in Republic of Iraq because they have the same administrative structure for the department of scientific affairs and cultural communications. This project has many features in order to be useful for the department staff on the one hand and for students on the other hand:

1) to provide the privacy of user information from students;
2) easy to manage the issuance of documents by the manager because of the easily process of addition, modification and delete the records of the students;
3) ease of dealing with the system;
4) the number of users is unlimited;
5) possibility of future development for the site.

**Structure of the thesis**

The thesis consists of four chapters, introduction, conclusion and reference list.

In the first chapter, the problem of statement is represented. Also, there is comparative analysis of analogues and tools for development.

In chapter Two, there is a description of functional and non-functional requirements, use case, deployment and state machine diagrams, database scheme and the design of interfaces of the system.

In chapter Three, we represent the project file structure and several SQL-queries that included in fragments of PHP-code for implementing the basic functionality of the system.
Chapter Four contains the description of the functional testing of the application and the usability testing.

The thesis has 52 pages; the list of references contains 20 resources.
1. THE ANALYSIS OF THE SUBJECT AREA

1.1. The problem statement

The Department of Scientific Affairs and Cultural Communications in the University of Technology needs an electronic system to manage data about scholarship students (study leave, scholarship and fellowship). This Department has 150 students yearly, to make the work inside the Department of Scientific Affairs and Cultural Communications in the University of Technology easier, to save time and effort on employees and students and to facilitate whether the needed documents are completed or are still in progress. These documents concern questions of the study start date, confirmation for the external companies about the legitimacy of diploma, exchange of financial receivables and other documents of the student’s interest.

Therefore, we must design a system to manage information about all these students.

It must be easy to edit, delete, update, and search any information about students.

Students must see the type of their scholarship, greeting message, information about themselves, their work place, studying country/city, degree of city, date of administrative order, date of leaving work, date of starting study, fees for foreign course, number of studying years, salary, their documents, fees of traveling tickets, their financial dues for each year, marital status, number of family members.

They can check their information, print it or send an e-mail if there is something wrong.

The head of the department must see the same information about all students.

The manager must see the same information, in addition, he can add, delete, made reports on some statistics, received E-mails from students, answers on them and correct information about all students.
1.2. Comparative analysis of analogues

I want to develop a web-site for scholarship students in the department of Scientific Affairs and Cultural Communications in the University of Technology. There are many sites containing systems for scholarship students.

For example fig. 1 shows the system in the King Saud University for the scholarship students [9]. The student can ask about anything he wants and after that waits for the answer (not in real time).

Fig. 1. King Saud University

Fig. 2 shows the University of Babylon, also with a system for scholarship students [16]. This system now has some problems and it’s in maintenance.

Fig. 2. University of Babylon
There are some mobile applications available for students in some Arabic countries like Kingdom of Saudi Arabia and United Arab Emirates. Fig. 3 shows Umm Al-Qura University Mecca [14], [15]. The access for this mobile application is forbidden to me because I am not registered in that University.

All these sites are self-done. I need an authorization to enter, therefore I design my own system which we need in the University of Technology.

1.3. Used development tools

There are two types of web-development systems: CMS and CMF.

CMS is software that keeps track of every piece of content on your web site, much like your local public library keeps track of books and stores them. Content can be simple text, photos, music, video, documents, or just about anything you can think of. A major advantage of using a CMS is that it requires almost no technical skill or knowledge to manage. Since the CMS manages all your content, you don't have to.

The examples of CMS are Joomla!PHP-Nuke, WordPress, Drupal, Expression Engine, TextPattern, Radiant, Cushy, SilverStripe, etc.
Advantages of CMS [10]:

1) it’s easy for the non-technically minded, anyone who can use word-processing software can use a CMS for the basic functions so we don’t even have to spend much time on training;

2) it allows multiple users, in a business, there are many people who can have input into your website, from those who add product pages to those who produce blog posts for your content marketing efforts. A CMS makes it easy to manage roles and publishing permissions for all these users so that only those you allow can publish content only goes live when you’re ready;

3) it streamlines scheduling, related to that any decent CMS will give you an at a glance view of the status of all content, whether it’s live, being reviewed or a draft. That doesn’t just apply to blog posts but to product pages and other website pages. It allows you to assign tasks and check that they have been done. And it’s easy to integrate planned content with your marketing plan so that everyone knows what’s happening when;

4) it improves site maintenance, need to change something on your site? Without a CMS that can mean having to trawl through hundreds of pages, making changes on each one. With a CMS, the underlying architecture is the same so you can make maintenance changes, update the CMS software and add functionality without breaking the site. In fact, with the right CMS, it may keep itself up to date automatically;

5) design changes are simple, speaking of making changes, let’s talk about the look of the site. If you want to change the site design, a CMS makes the process easy. That’s because the content and design are in separate virtual boxes, so you can make design changes while keeping the site functional. Another advantage is the ability to make a change in your administrative dashboard and have it automatically propagate to the entire site. This gives your site a consistent appearance and is great for branding. It also makes it easy to add a mobile interface to your site;
6) it helps you manage content, this may seem obvious, but for some businesses, content management is not just about publishing content but about being able to remove it when it’s out of date. With a CMS this is as simple as unpublishing the content all menus and links update automatically, so that your customers continue to have a good experience of the site. And if you want to include customized content such as countdown calendars and lists, the CMS can be configured to allow this. And a good CMS includes SEO (search engine optimization) too, making sure your content ranks well in the search engines;

7) you’re in control, all of this adds up to the biggest advantage of a CMS. Instead of being reliant on an external vendor, with a CMS you are in control, with the ability to assign tasks and roles and to check progress at any time. That puts you in the driving seat when it comes to this important business tools.

CMF is a system that facilitates the use of reusable components or customized software for managing web content.

The examples of several CMF: BansheePHP, XOOPS, WordPress, TYPO3, Sitecore, SilverStrip, processWire, Microsoft Share Point Foundation, etc.

The difference between CMS and CMF is that CMS is easier to use, allows publishing, editing and modifying content on a website as well as maintenance from a central page, it is used to create typical sites with ordinal structure. CMF more complex, it is devoted to creating of large multipart structural projects with a lot of non-typical, unique algorithms and handlers written specially for these projects.

So, I decided to write my web-site in CMS Joomla! I decided to do that because:

- Joomla! is a free and open-source content management system (CMS) for publishing web content;
- is written in PHP, uses object-oriented programming (OOP) techniques and software design patterns;
- stores data in a MySQL, MS SQL, or PostgreSQL database;
- includes features such as page caching, RSS feeds, news flashes, blogs, search, and support for language internationalization.
- can be run on many platforms, including Windows, Linux, Mac and Unix it’s easy for users to find hosting service providers.
- Many companies are designed there web-sites by used Joomla! CMS like GE Distributed Power Russia, Fratteli.rs, Titan Garages and Sheds, MTV (Music Television), Mitsubishi, Porsche, Burger King, Harvard, Lipton Ice Tea, E-Style hirdresser, Spread the Joomla! Love, Lasarex Cancer Foundation, NOW architecture and other.

All information must be kept on database. Of course, we need to use database management system. Joomla! easily connects with DBMS MySQL.

MySQL is an open-source relational database management system (RDBMS), is ideal for both small and large applications.

MySQL is very fast, reliable, and easy to use, free to download and use, it uses standard SQL.

MySQL is developed, distributed, and supported by Oracle Corporation.

MySQL is the de-facto standard database system for web sites with huge volumes of both data and end-users (like Facebook, Twitter, and Wikipedia).

In the first chapter, there is the description of the problem area, comparison of other existing analogues and description of chosen development tools. We decided to use Joomla! as CMS, PHP as a basic programming language for the site design and development and MySQL as a data store system.
2. DESIGN OF WEB-SITE FOR DEPARTMENT OF SCIENTIFIC AFFAIRS AND CULTURAL COMMUNICATIONS IN THE UNIVERSITY OF TECHNOLOGY (BAGHDAD)

2.1. Functional and non-functional requirements

The problem statement defined the following functional requirements.

The system must do the following:

- work with three types of authorized users: student, manager and the head of the department;
- calculate operations by using addition mathematical function like final salary, marital and accompanying children for all students.

The manager must be able to:

- see information about all students;
- add information about all students;
- delete information from students;
- search any information about students;
- receive reports on (get amount of students, get gender of students, get amount of students who gets master degree, get amount of students who gets PHD degree, get amount of students who gets extension period) statistics;
- correct information about all students;
- print information about all students.

The head of the department must be able to:

- see information about all students;
- search any information about students;
- print information about all students.

The student must be able to:

- see the information about himself;
- print his information;
• write an e-mail to the department if there is something wrong.

The non-functional requirements are the following.

• users enter the system only by internet;
• system is working at least in browsers Google Chrome, Mozilla Firefox and Internet Explorer of the last versions.

2.2. Designing of structure and behavior of the web-application

2.2.1. Use case diagram

Use case diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of systems (activity diagrams, state chart diagrams, sequence diagrams, and collaboration diagrams are four other kinds of diagrams in the UML for modeling the dynamic aspects of systems).

Use case diagrams are central to modeling the behavior of a system, a subsystem, or a class. Each one shows a set of use cases and actors and their relationships [1].

The use case diagram of the system “Scholarships Affairs” consists of three basic users (“student”, “manager”, “head”), each of them can “enter site” with login and password. “Manager” inherits the user “Head of department” (fig. 4).

“Head of Department” can look at all tables and search information about students; “Manager” can add, delete, search, receive reports, print information about all students and modify data in those tables. “Student” can look at the table with information only about himself and print them or send an e-mail if his information is wrong.

Use cases “Look at the tables about himself, print information”, “Send an e-mail” and “Add, delete, modify, print” are extending the use case “Search, Check and look at the tables”. The first two use cases are available to the student. The last one is available to the manager.
2.2.2. Deployment diagram

A deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams are one of the two kinds of diagrams used in modeling the physical aspects of an object-oriented system.

It is used to model the static deployment view of a system (topology of the hardware).

A deployment diagram is just a special kind of class diagram, which focuses on a system’s nodes. Graphically, a deployment diagram is a collection of vertices and arcs. Deployment diagrams commonly contain Nodes and Dependency and association relationships. It may also contain notes and constraints. Deployment diagrams are important for visualizing, specifying, and documenting embedded, client/server, and distributed systems and also for managing executable systems through forward and reverse engineering [1].

This system is deployed at two nodes: “Client” and “Server” (fig. 5). Many user computers can connect to the Server. Server contains Database component, and “Web-application”. Component “Database” is connected to “Web-application”.

Fig. 4. Use case diagram
2.2.3. State machine diagram

A state machine diagram is a diagram that describes the dynamic behavior of objects over time by modeling the lifecycles of objects of each class. Each object is treated as an isolated entity that communicates with the rest of world by detecting events and responding to them. Events represent the kinds of changes that an object can detect the receipt of calls or explicit signals from one object to another, a change in certain values, or the passage of time. Anything that can affect an object can be characterized as an event. Real-world happenings are modeled as signals from the outside to the system [19].

This system consists of the following states “login page”, “Main page”, “Viewing information” and “Checking information” (fig. 6).

When user “write username and password correct”, then the system will check this user.

1. If “user = Student” then the system will calculate the salary of this student and view only information about this student and he will able to “Checking information” and doing actions “send an e-mail”, “print” and he will “log out” and finish his state.
2. If “user = Head” then the system will go to the state “Main Page”, the system will go to the state “Viewing information” when the “Head” “Enter category” view the information about all students and he can do actions “Search”, “Print”, “Statistics” and he will “logout” and finish his state.

3. If “user = Manager” then the system will go to the state “Main Page”, the system will go to the state “Viewing information” when the “Manager” “Enter category” view the information about all students and the “Manager” will able to do this actions “Add”, “Delete”, “Search”, “Modify”, “Save”, “Statistics” and “Print” and he will “logout” and finish his state.

Fig. 6. State machine diagram

2.3. Database scheme

A database is a data structure that stores organized information. Most databases contain multiple tables, which may each include several different fields. Simply it’s a collection of organized information, usually as a set of related lists of similar entries. The data is often organized so that it is easily accessible. The following are examples of database that we use often (address book, dictionary and telephone book) [11].
In this system I used DBMS MySQL because it’s very fast, reliable, and easy to use, free to download and use, it uses standard SQL and Joomla! CMS easily connects with DBMS MySQL, so that I used this kind of database (fig. 7) shows the database schema.

This database schema of system consists of 11 tables. Table “Student” contains information about all students. It is connected to the table “Work place” which they belong to. Each record in table “Student” connects to tables: “Degree Study”, “Documents”, “Financial dues” and “City”, in which their documents issued. Table “Documents” connects to the table “Type of document”. Table “City” connects to the tables “Degree City” and “Country” for each student, table “Degree Study” connects to the table “Category student”, all of these tables are related by one to many relationship.

The table “Student” contains information about all students. It consists of 23 fields. These fields are: Id_Student, Id_degree_study, Surname, First_Name, Second_Name, Third_Name, Age, Gender, Id_WorkPlace, Id_City, ExtensionPeriodStart, ExtensionPeriodEnd, Num-
ber_Of_Family_members, Fees_For_Foreign_Course, Date_of_the_administrative_order, Date_of_leaving_work, Number_of_studying_years, fees_for_leaving_country_ticket, fees_for_return_to_country_ticket, Username, Password. These fields have different types of data: int, varchar and date. The primary key for this table is “Id_Student” which is automatically incremented (fig. 8).

Fig. 8. Structure of the table “Student”

The table “City” consists of 4 fields, these fields are: Id_City, City_name, Id_Degree_city, Id_Country. The primary key for this table is “Id_City” which is automatically incremented (fig. 9).

Fig. 9. Structure of the table “City”
The table “Degree_Study” contains the degrees of studies and the basic salary for each category. It consists of 4 fields, these fields are: Id_degree_study, Degree_Study_name, Id_Category, Basic_salary. The primary key for this table is “Id_degree_study” which is automatically incremented (fig. 10).

![Fig. 10. Structure of the table “Degree_Study”](image)

The table “Documents” contains the information about all documents. It consists of 5 fields, these fields are: Id_Document, Id_Student, Id_Type, Document_Number, Document_Date. The primary key for this table is “Id_Document” which is automatically incremented (fig. 11).

![Fig. 11. Structure of the table “Documents”](image)

The table “Financial_dues” contains the information about all tuition fees in year. It consists of 4 fields, these fields are: Id_Financial_dues, Id_Student, Tuition_fees, Year. The primary key for this table is “Id_Financial_dues” which is automatically incremented (fig. 12).

![Fig. 12. Structure of the table “Financial_dues”](image)
Fig. 12. Structure of the table “Financial_dues”

The table “Staff” contains the information about the manager and the head of the department. It consists of 6 fields, these fields are: Id, Surname, Name, Username, Password, Role. The primary key for this table is “Id” which is automatically incremented (fig. 13).

Fig. 13. Structure of the table “Staff”

2.4. Design of interface

Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand and use to facilitate those actions [4].

Here I will implement the interfaces of my application. Fig. 14 shows the main page for the site of a system (Scholarship). It contains the main menu in the top of the page: “Home”, “Contact Us” and “Location” which is the basic elements for each page.

Fig. 15 shows the page for “Student page” it appears when the student entered correct username and password, in this page the student can see all infor-
mation about himself he can print it, send an email to the department if there is something wrong in his information or sign-out from the system.

![Main page diagram](image)

**Fig. 14. “Main page”**

![Student page diagram](image)

**Fig. 15. “Student page”**

Fig. 16 shows the page for “The head of the department page”. It appears when the head entered correct username and password, in this page the head of the department can see information about all students, he can print it, view student’s details or sign-out from the system.

![Head of the department page diagram](image)

**Fig. 16. “Head of the department page”**
Fig. 17 shows the page “Details”. It appears when the head of the department clicks the link “Details” for the chosen student in the table of student. In this page the head can see information about the chosen student. He can print it, go back to his page or sign-out from the system.

![Student details page](image)

Fig. 17. “Student details page”

Fig. 18 shows the page “Manager”. It appears when the manager entered the correct username and password, in this page the manager can see information about all students, add information about all students, delete information from students, search any information about students, receive reports on the amount of students, gender of students, amount of students who gets master degree, amount of students who gets PHD degree, amount of students who gets extension period. He also can print information about all students and sign-out from the system.

![Manager page](image)

Fig. 18. “Manager page”
Fig. 19 shows the page “Delete student”. It appears when the manager clicks the button “Delete” after the manager selects a student to delete, in this page the manager can delete or cancel deleting a student from the system and go back to his page.

![Confirmation dialog]

Fig. 19. “Delete student page”

Fig. 20 shows the page “Add student”. It appears when the manager clicks the button “Add Student”, in this page the manager can add a new student to the system, go back to his page or sign-out from the system.

![Add student page]

Fig. 20. “Add Student page”

The page “Modify Student” is similar to the page “Add Student”. The difference between the two pages is when the manager clicks the button “Modify” another...
er page will appear and the manager will see the edit information about the chosen student, he can update (correct) information for the student, save it, go back to his page or sign-out from the system.

Fig. 21 shows the page “Search student”. It appears when the manager clicks the button “Search Student”, in this page the manager can search about any information of the student in the system, go back to his page, print information or sign-out from the system.

![Search Student page](image1)

Fig. 21. “Search Student page”

Fig. 22 shows the results of search; it appears when the manager clicks the button “Search”, in this page the manager can print the results, modify information, view student details, go back to his page, or sign-out from the system.

![Results of Search page](image2)

Fig. 22. “Results of Search page”
Fig. 23 shows the page “Student Details”; it appears when the manager clicks the button “Details”, in this page the manager can see information about the chosen student he can print it, go back to his page, delete information from the chosen student, add new information or sign-out from the system.

![Student Details page](image)

Fig. 23. “Student Details page”

Fig. 24 shows the page “Add New Document”; it appears when the manager clicks the button “Add Document”, in this page the manager can add new documents for the chosen student he can save it or go back to his page.

Fig. 25 shows the page “Add New Financial dues”; it appears when the manager clicks the button “Add Financial dues”, in this page the manager can add new tuition for the chosen student he can save it or go back to his page.

![Add New Document page](image)

Fig. 24. “Add New Document page”
Fig. 25. “Add New Financial Dues”

Fig. 26 shows the page “Reports”; it appears when the manager clicks the button “Reports”, in this page the manager can choose different kinds of reports.

Fig. 26. “Reports page”

Fig. 27 shows the page “Total amount of students”; it appears when the manager clicks the button “Get amount of students”, in this page the manager can get a report on amount of students in the system in general or by select one category, go back to his page or sign-out from the system.

Fig. 27 shows the page “Total amount of students”; it appears when the manager clicks the button “Get amount of students”, in this page the manager can get a report on amount of students in the system in general or by select one category, go back to his page or sign-out from the system.
Fig. 27. “Total amount of students”

Fig. 28 shows the page “Results of report Total amount of students”; it appears when the manager clicks the button “OK”, in this page the manager gets report on amount of students in the system in general or by select one category, the manager can print results, go back to his page or sign-out from the system.

Fig. 28. “Results of report total amount of students”

Fig. 29 shows the page “Report Gender”; it appears when the manager clicks the button “Get report in Gender of students”, in this page the manager can get a report on amount of students in their gender in general or by select one category, go back to his page or sign-out from the system.

Fig. 29 shows the page “Report Gender”; it appears when the manager clicks the button “Get report in Gender of students”, in this page the manager can get a report on amount of students in their gender in general or by select one category, go back to his page or sign-out from the system.

Fig. 30 shows the page “Results of report Total amount gender of students”; it appears when the manager clicks the button “OK”, in this page the manager gets
report on amount gender of students in the system in general or by select one category, the manager can print results, go back to his page or sign-out from the system.

![Fig. 29. “Report Gender”](image)

Fig. 29. “Report Gender”

![Fig. 30. “Results of report Total amount gender of students”](image)

Fig. 30. “Results of report Total amount gender of students”

Fig. 31 shows the page “Report PHD degree”; it appears when the manager clicks the button “Get amount of students who will get PHD degree”, in this page the manager can get a report on amount of students who will get PHD degree in general or by select one category, go back to his page or sign-out from the system.
Fig. 31. “Report PHD degree”

Fig. 32 shows the page “Report amount of students who will get PHD degree”; it appears when the manager clicks the button “OK”, in this page the manager gets report on amount of students who will get PHD degree in the system in general or by select one category, the manager can print results, go back to his page or sign-out from the system.

Fig. 32. “Report amount of students who will get PHD degree”

Fig. 33 shows the page “Report Master degree”; it appears when the manager clicks the button “Get amount of students who will get Master degree”, in this page the manager can get a report on amount of students who will get master degree in general or by select one category, go back to his page or sign-out from the system.
Fig. 33. “Report Master degree”

Fig. 34 shows the page “Report amount of students who will get Master degree” it appears when the manager clicks the button “OK”, in this page the manager gets report on amount of students who will get Master degree in the system in general or by select one category, the manager can print results, go back to his page or sign-out from the system.

Fig. 34. “Report amount of students who will get Master degree”

Fig. 35 shows the page “Report Extension period”; it appears when the manager clicks the button “Get amount of students who get Extension period”, in this page the manager can get a report on amount of students who gets extension period in general or by select one category, go back to his page or sign-out from the system.
Fig. 35. “Report Extension period”

Fig. 36 shows the page “Report amount of students who gets Extension period”; it appears when the manager clicks the button “OK”, in this page the manager gets report on amount of students who gets extension period in the system in general or by select one category, the manager can print results, go back to his page or sign-out from the system.

Fig. 36. “Report amount of students who gets Extension period”
3. IMPLEMENTATION OF THE WEB-APPLICATION

3.1. Description of the process of web-application development in CMS Joomla!

Joomla! is an award-winning content management system (CMS), which enables to build web-sites and powerful online applications. Many aspects including its ease-of-use and extensibility have made Joomla! the most popular web-site software available. Best of all, Joomla! is an open source solution that is freely available to everyone [8].

If you need to access tables within the database in Joomla! you can simply use the JFactory->getDbo method. This uses the already established connection that Joomla! uses to connect to the database (fig. 37) [2].

$$db = JFactory::getDbo();$$
$$query = $db->getQuery(true);$$

Fig. 37. “Connect to the database”

In Joomla! we can’t directly write code without installation of extensions for writing in PHP and JavaScript [5]. In my system I used the “Sourcerer Extension” which helped me to write my code (fig. 38).

Fig. 38. “Sourcerer Extension”
All files are kept in folder “jooml30”. It contains the necessary files which they are direct in the folder with the web-site. I have 18 articles in my application. They belong to the template “ice_future” in the web-site (fig 39). The total amount of code lines is approximately 4500 lines.

![Fig. 39. “Articles in web-site”](image)

### 3.2. Fragments of code

Fig. 40 shows the SQL-query adding a new student to the system [20].

Fig. 41 shows SQL-query deleting student from the system [18].

Fig. 42 shows SQL-query getting information for student according to different parameters [3].

Fig. 43 shows SQL-query modifying student’s information.

Fig. 44 shows SQL-query selecting information from one table in database to make drop down list.
Fig. 40. “Add Student”

```php
//Sql-query for Adding new Student.
$sqladd = "INSERT INTO Student (Surname, First_Name, Second_Name, Third_Name, Age, Gender, Id_Workplace, Id_City, Id_degree_study, ExtensionPeriodStart, ExtensionPeriodEnd, Number_Of_family_members, marital_status, Date_of_the_administrative_order, Date_of_starting_study, Date_of_leaving_work, Number_of_studying_years, fees_for_leaving_country_ticket, fees_for_return_to_country_ticket, Fees_For_Foreign_Course, Username, Password) VALUES ('$surname', '$name', '$secondname', '$thirdname', '$age', '$gender', '$workplace', '$city', '$study', '$extensionstart', '$extensionend', '$members', '$martial', '$administrativeorder', '$startstudy', '$leavingwork', '$studyingyears', '$leavefees', '$returnfees', '$languagefees', '$username', '$password');"
$db->setQuery($sqladd);
$db->execute();
```

Fig. 41. “Delete Student”

```php
//Sql query for deleting Student
$sqlid="DELETE FROM Student WHERE Id_student IN (".$idrows." ) or die ("Error in delete");
$db->setQuery($sqlid);
$sqlid = $db->execute();
$sqlid="DELETE FROM Documents WHERE Id_student IN (".$idrows." ) or die ("Error in delete");
$db->setQuery($sqlid);
$sqlid = $db->execute();
$sqlid="DELETE FROM Financial_dues WHERE Id_student IN (".$idrows." ) or die ("Error in delete");
$db->setQuery($sqlid);
$sqlid = $db->execute();
echo "Delete successfully";
```

Fig. 42. “Get information for student”

```php
//sql query for getting information about student according to different parameters
$sqlloc="SELECT * FROM Student, Degree_Study, Category_student WHERE Student.Id_degree_study=Degree_Study.Id_degree_study AND Degree_Study.Id_category='Bachelor' AND Category_student.Id_category='Bachelor' AND Degree_Study.Degree_name='Master'
$db->setQuery($sqlloc);
$loc_count = $db->query();
$loc_count = $db->getRows();
$result= $db->loadAssocList();
```
Fig. 43. “Modify information”

```php
// sql query modifying students information

$sqlmo = 'UPDATE Student SET Surname='Surname', First_Name='Name', Second_Name='SecondName', Third_Name='ThirdName', Age='Age',
Gender='Gender', Id_WorkPlace='Workplace', Id_City='City', Id_degree_study='Degree', ExtensionPeriodStart='ExtensionStart',
ExtensionPeriodEnd='ExtensionEnd', Number_of_family_members='Members', marital_status='Marital',
Date_of_the_administrative_order='AdministrativeOrder', Date_of_starting_study='Start', Date_of_leaving_work='Leave',
Number_of_studying_years='StudyingYears', fees_for_leaving_country_ticket='LeaveFees', fees_for_return_to_country_ticket='ReturnFees',
Fee_For_Foreign_Courses='LanguageFees', Username='Username', Password='Password' WHERE Id_Student='Id';

$db->setQuery($sqlmo);
$db->execute();

echo "Updating Student successfully";
```

Fig. 44. “Select information from one table and make drop down list”

```php
// sql query getting information from one table in database to make a drop down list

$ssqlst = "SELECT * FROM Category_Student";
$db->setQuery($ssqlst);
$mmyr_count = $db->query();
$mmyr_count = $db->getNumRows();
$results = $db->loadAssocList();

<form action="<?php $_php_self ?>" method="POST">
Select Category : <select name="category" id="selectcategory">
<option value="none">Select Category</option>
</select>
<?php
for ($i=0; $i< $mmyr_count-1; $i++){
    echo "<option value=".".$results[$i]["Id_Category"]. ""> ".$results[$i]["Name_Category"]."</option>
;
}
?></form>
```

Fig. 45 shows SQL-query calculating salary according to the number of family members.

Fig. 46 shows SQL-query selecting data without repetitions.

Fig. 47 shows JavaScript returning value to radio box, which is used to search information about students [6].

Fig. 48 shows alert massage before deleting student from the system.
<?php
// SQL query calculating salary according to number of family members

$stmt = "SELECT * FROM Student, Degree_Study, Category_student, Financial_dues WHERE Student.Id_student='$id_st'
AND Student.Id_degree_study=Degree_Study.Id_degree_study
AND Degree_Study.Id_Category=Category_student.Id_Category
AND Financial_dues.Id_Student='$id_st'";
$db->setQuery($stmt);
$my_count = $db->query();
$my_count = $db->getNumRows();
$row=$db->loadAssocList();
if ($member >= 1){
    $basic_salary=$row['0']['Basic_salary'];
    $salary=$basic_salary+$800;
} else
    $salary=$row['0']['Basic_salary'];
?>

---

// SQL query selecting data from one table without repetitions

$stmt = "SELECT DISTINCT Degree_Study_Name FROM Degree_Study";
$db->setQuery($sql);
$my_count = $db->query();
$my_count = $db->getNumRows();
$results= $db->loadAssocList();

---

// JavaScript returning value to the radio box

<script language="javascript" type="text/javascript">

function myFunction(select) {
    document.getElementById("result").value = select;
}
</script>

<form action="<?php $_FILE." method="POST">
    <input type="radio" name="select" onclick="myFunction(this.value)" value="Category" /> Category<br/>
    <input type="radio" name="select" onclick="myFunction(this.value)" value="Name" /> Name<br/>
    <input type="radio" name="select" onclick="myFunction(this.value)" value="City" /> City<br/>
    <input type="radio" name="select" onclick="myFunction(this.value)" value="Work place" /> Work place<br/>
    <input type="radio" name="select" onclick="myFunction(this.value)" value="Degree of the city" /> Degree of the City<br/>
    <input type="hidden" id="result" value="" name="select" />
    <input type="submit" value="OK" name="ok" />
</form>

---

Fig. 45. “Calculate salary”

Fig. 46. “Select without repetitions”

Fig. 47. “Radio box”
Fig. 48. “Alert massage”

```javascript
<!-- JavaScript showing alert massage before delete student -->

<script language="javascript" type="text/javascript">
    function Confirm()
    {
        var x = confirm("Are you sure you want to delete information about this student? You can't be able to recover it if you click OK.");
        if (x)
            return true;
        else
            return false;
    }
</script>
```

Fig. 49 shows SQL-query selecting single value results.

```
// SQL-query selecting single value result

$sqlstudy = "SELECT Id_degree_study FROM Degree_Study WHERE Degree_Study_name='$degreestudy' AND Degree_Study.Td_Category='$category'";

$db->setQuery($sqlstudy);

$study = $db->loadResult();
```

Fig. 49. “Single value results”
4. TESTING OF THE WEB-APPLICATION

4.1. Functional testing

Functional testing is used to testing the functionality of the software application under test. Basically it is to check the basic functionality mentioned in the functional specification document. Also check whether software application is meeting the user expectations. In another words it used to checking the behavior of the software application against test specification [12].

Types of software testing [13]: black box testing, white box testing, unit testing, incremental integration testing, integration testing, functional testing, system testing, end-to-end testing, sanity testing, regression testing, acceptance testing, load testing, stress testing, performance testing, usability testing, install-uninstall testing, recovery testing, security testing, compatibility testing, comparison testing, alpha testing, beta testing.

Functional testing is a type of testing which verifies that each function of the software application operates in conformance with the requirement specification. This testing mainly involves black box testing and it is not concerned about the source code of the application. Each functionality of the system is tested by providing appropriate input, verifying the output and comparing the actual results with the expected results. This testing involved checking of User Interface, APIs, Database, security, client/server applications and functionality of the Application Under Test. The testing can be done either manually or using automation [7].

Table 1 contains results of testing of all functional requirements defined in paragraph 2.1.

<table>
<thead>
<tr>
<th>No</th>
<th>Function</th>
<th>Procedure</th>
<th>Expected result</th>
<th>Obtained result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>To show the main page with navigation bar and login form</td>
<td>The visitor enters the site</td>
<td>Any visitor can see this page</td>
<td>Any visitor can see this page</td>
</tr>
<tr>
<td>No</td>
<td>Function</td>
<td>Procedure</td>
<td>Expected result</td>
<td>Obtained result</td>
</tr>
<tr>
<td>----</td>
<td>----------</td>
<td>-----------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>2.</td>
<td>Work with three types of authorized users: student, manager and the head of the department</td>
<td>When the user entered corrected username and password he will see his own page</td>
<td>Only authorized users can entered to the system</td>
<td>Only authorized users can entered to the system</td>
</tr>
<tr>
<td>3.</td>
<td>Calculate final-salary by using addition mathematical function</td>
<td>When student enters to the system his finalsalary will be calculated according to his family member. Also it appears to the manager and the head of the department when they click button “Details”</td>
<td>Each student can see his own final-salary, the manager and the head of the department can see the finalsalary for all students</td>
<td>Each student can see his own final-salary, the manager and the head of the department can see the finalsalary for all students</td>
</tr>
<tr>
<td>4.</td>
<td>To see information about all students</td>
<td>When the manager and the head of the department entered corrected username and password each of them will see his own page which contains information about all students</td>
<td>Only the manager and the head of the department will see this information</td>
<td>Only the manager and the head of the department will see this information</td>
</tr>
<tr>
<td>5.</td>
<td>Add new student to the system</td>
<td>When the manager entered the system and clicks the button “Add Student” this page will appear and he can add a new student and save student’s information</td>
<td>Only the manager can add a new student to the system</td>
<td>Only the manager can add a new student to the system</td>
</tr>
<tr>
<td>6.</td>
<td>Add new documents for chosen student</td>
<td>When the manager entered to the system and clicked the button “Details” this page will appear with the details of the chosen student. The manager can add a new document when he clicks the button “Add Document”. This page will appear and he can add new documents to the chosen student, save it and return to the details of the chosen student or log-out from the system</td>
<td>Only the manager can add new documents to the chosen student</td>
<td>Only the manager can add new documents to the chosen student</td>
</tr>
<tr>
<td>No</td>
<td>Function</td>
<td>Procedure</td>
<td>Expected result</td>
<td>Obtained result</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>7</td>
<td>Add financial dues for chosen student</td>
<td>When the manager entered to the system and clicked the button “Details” this page will appear with the details of the chosen student. The manager can add financial dues when he clicks the button “Add Financial due”. This page will appear and he can add new financial due to the chosen student, save it and return to the details of the chosen student or log-out from the system</td>
<td>Only the manager can add new financial dues to the chosen student</td>
<td>Only the manager can add new financial dues to the chosen student</td>
</tr>
<tr>
<td>8</td>
<td>Delete financial due from chosen student</td>
<td>When the manager entered to the system and clicked the button “Details” this page will appear with the details of the chosen student. The manager can delete financial dues from this student. When he selects the financial due and clicks the button “Delete Financial due” the alert message will appear and he can delete this financial due or keep it to the chosen student. He can return to his page or log-out from the system</td>
<td>Only the manager can delete financial dues to the chosen student</td>
<td>Only the manager can delete financial dues to the chosen student</td>
</tr>
<tr>
<td>9</td>
<td>Delete Documents from chosen student</td>
<td>When the manager entered to the system and clicked the button “Details” this page will appear with the details of the chosen student. The manager can delete documents from this student. When he selects the document and clicks the button “Delete Document” the alert message will appear and he can delete this document or keep it to the chosen student. He can return to his page or log-out from the system</td>
<td>Only the manager can delete documents to the chosen student</td>
<td>Only the manager can delete documents to the chosen student</td>
</tr>
<tr>
<td>No</td>
<td>Function</td>
<td>Procedure</td>
<td>Expected result</td>
<td>Obtained result</td>
</tr>
<tr>
<td>----</td>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>10</td>
<td>Delete student from the system</td>
<td>When the manager entered to the system and selected one or more students from the table and he clicked the button “Delete Selected”. The alert massage will appear and he can delete the chosen students from the system or cancel the deletion.</td>
<td>Only the manager can delete students from the system</td>
<td>Only the manager can delete students from the system</td>
</tr>
<tr>
<td>11</td>
<td>Search any information about students</td>
<td>When the manager entered to the system and clicked the button “Search Student” this page will appear with the radio boxes. The manager can search “Category”, “Name”, “City”, “Work place” and “Degree of the City” and get the results of this search in table. He can print results, return to his page or log-out from the system</td>
<td>Only the manager can search by this way</td>
<td>Only the manager can search by this way</td>
</tr>
<tr>
<td>12</td>
<td>Receive report on amount of students</td>
<td>When the manager entered to the system and clicks the button “Reports” this page will appear with different kinds of reports the manager can receive report on get amount of students by click the button “Get amount of Students” and this page appeared the manager can get the report in general or by select one category, he can print results, return to reports page or log-out from the system</td>
<td>Only the manager can receive reports</td>
<td>Only the manager can receive reports</td>
</tr>
<tr>
<td>13</td>
<td>Receive report on gender students</td>
<td>When the manager entered to the system and clicks the button “Reports” this page will appear with different kinds of reports the manager can receive report on get gender of students by click the button “Get report in Gender of Students” and this page appeared the manager can get the report by select the gender in general or by select one category, he can print results, return to reports page or log-out from the system</td>
<td>Only the manager can receive reports</td>
<td>Only the manager can receive reports</td>
</tr>
</tbody>
</table>

Continuation of tab. 1
<table>
<thead>
<tr>
<th>No</th>
<th>Function</th>
<th>Procedure</th>
<th>Expected result</th>
<th>Obtained result</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.</td>
<td>Receive report on amount of students who will get master degree</td>
<td>When the manager entered to the system and clicks the button “Reports” this page will appear with different kinds of reports the manager can receive report on get amount of students will get master degree by click the button “Get amount of students who will get Master degree” and this page appeared the manager can get the report in general or by select one category, he can print results, return to reports page or log-out from the system</td>
<td>Only the manager can receive reports</td>
<td>Only the manager can receive reports</td>
</tr>
<tr>
<td>15.</td>
<td>Receive report on amount of students who will get PHD degree</td>
<td>When the manager entered to the system and clicks the button “Reports” this page will appear with different kinds of reports the manager can receive report on get amount of students who will get PHD degree by click the button “Get amount of students who will get PHD degree” and this page appeared the manager can get the report in general or by select one category, he can print results, return to reports page or log-out from the system</td>
<td>Only the manager can receive reports</td>
<td>Only the manager can receive reports</td>
</tr>
<tr>
<td>16.</td>
<td>Receive report on amount of students who gets extension period</td>
<td>When the manager entered to the system and clicks the button “Reports” this page will appear with different kinds of reports the manager can receive report on get amount of students who gets extension period by click the button “Get amount of students who gets Extension period” and this page appeared the manager can get the report in general or by select one category, he can print results, return to reports page or log-out from the system</td>
<td>Only the manager can receive reports</td>
<td>Only the manager can receive reports</td>
</tr>
</tbody>
</table>

End of tab. 1
According to the table, the testing of all functional requirements has been performed.

System is working at least in browsers Google Chrome, Mozilla Firefox and Internet Explorer in last versions.

All mentioned functions (both functional and non-functional) work correctly.

### 4.2. Usability testing
Usability testing is a way for a website manager to gather useful feedback on their website, from a select sample of typical users. The tests are run with a user using the website with the aim to gather as much information about how the user interacts with the website as possible. This will then provide sufficient evidence and cause, to then make changes to a website in order to make it more usable. It is the only way to test whether or not your website is actually of any use to a user in the real world [17].

List of tasks which given to real persons and which goals must they achieve:

1. Tasks for the manager:
   - add two new students to the system;
   - delete one document from “Ali Zainab”;
   - print the results of the report “amount of students which will get Master degree”;
   - find how many students are there in “Chelyabinsk city”;
   - modify the second name of the student “Boyle Susan Arther Paul” from “Arther” to “Arthur” and view her details.

2. Tasks for the head of the department:
   - find how many students are there in “Scholarship category” and print the result;
   - view the details of “Alhassani Muhammed Fadhel Abbas” and sign-out from the system.

3. Tasks for the student:
   - view your information, check it and print it;
   - send an e-mail to the department.

These tasks were given to three users. Analysis of the result of usability testing.

Tasks were easy to the student and the head of the department but for the manager they were not so easy to achieve the goals, he had these problems:
• in adding a new student to the system the manager has a mistake in selecting the right choice for the degree of city which belongs to the concrete city;
• the manager needs help in the beginning to view the details for the student “Ali Zainab” to delete her documents.
CONCLUSION

The goal of the system is to implement the development of web-site for department of scientific affairs and cultural communications in the University of Technology (Baghdad).

The web site is divided into 4 main parts: the first one is the login form. The second part is the view of table for all students to the head of department. The third part is devoted to view, add student, modify, delete, reports, print or search information by the manager. The forth part is for a student to view his own documents’ information in a table.

During the developing of the web application, we solved the following tasks:
1) analyzed the subject area;
2) implemented the comparative analysis of analogues;
3) made analysis of modern tools of web sites development and choose tools for project development;
4) designed the structure and behavior of the web-application;
5) designed database scheme;
6) designed the interfaces;
7) implemented the description of the process of the web-application development in CMS Joomla!.
8) implemented and tested the Web-site for Department of Scientific Affairs and Cultural Communications in the University of Technology (Baghdad).

For the following opportunities of the application development we can define the following points:
1) include the acceptance year for students;
2) include the year of acceptance for students in reports to accelerate the scope of statistics;
3) include reports to the head of the department;
4) change the function of search to the dependent dropdown lists search to accelerate the scope of the search;

5) change the function of add student to the dependent dropdown lists to avoid the manager make mistakes in choosing “City” and “Degree of city”.
LIST OF REFERENCES


