MINISTRY OF EDUCATION AND SCIENCE OF THE RUSSIAN FEDERATION

Federal 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,

Cand. Sci., Assoc. Prof. of Department

IITTMI

_____ Dmitrieva O.A.

“___”_____ 2017

ACCEPTED FOR THE DEFENSE

Head of the department, Dr. Sci., Prof.

_____ L.B. Sokolinsky

“___”_____ 2017

DEVELOPMENT OF MANAGEMENT INFORMATION SYSTEM FOR MAIN REGISTRATION DEPARTMENT IN UNIVERSITY OF TECHNOLOGY (BAGHDAD)

GRADUATE QUALIFICATION WORK
SUSU–02.04.02.2017.115-142.GQW

Supervisor

Cand. Sci., Assoc. Prof.

_____ O.N. Ivanova

Author,

the student of the group CE-216

_____ Z.A.A. Ali

Normative control

_____ O.N. Ivanova

“___”_____ 2017

Chelyabinsk–2017

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

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, mother, husband, son “Ahmed”, brothers and sisters for their continued encouragement, support and patience.

Finally I do not forget to thank friends and anyone who helped me directly and indirectly to overcome obstacles.

INTRODUCTION

Actuality

Modern information technologies are widely used in all spheres of human activities. Of course, the universities are not the exception. 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 university and continue even after graduating from it. It is often difficult for the student/graduate to keep tracking issuing their documents. On the other hand, it is difficult for the employee to answer all the students'/graduates' 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 everyone (students, graduates 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 Registration and Student Affairs by providing simple interfaces for all possible users.

Research goal and objectives

The goal of the research is the development of the management information system for the main registration department in the University of Technology (Baghdad).

For the reaching this goal we must solve the following objectives:

1) to analyze the subject area;
2) to implement the comparative analysis of analogical sites;
3) to make analysis of modern tools of web sites development and choose tools for project development;
4) to design the management information system;
5) to design database scheme;
6) to implement and test the management information system for the main registration department 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 Registration and Student Affairs department. This project has many features in order to be useful for the department staff on the one hand and for students and graduates on the other hand:

1) to provide the privacy of user information from students and graduates;

2) ease of follow-up work by the head of department for all sections;

3) easy to manage the issuance of documents by the managers sections because of the easily process of addition, modification and delete the records of the student / graduate;

4) ease of dealing with the system;

5) the number of users is unlimited;

6) 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 and deployment diagrams and database scheme.

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 interface testing with screenshots of the system.

The thesis has 47 pages; the list of references contains 20 resources.

1. THE ANALYSIS OF THE SUBJECT AREA

1.1. The problem statement

The University of Technology [12] contains 16 scientific departments. Each department has registration unit; these registration units are linked administratively with the main registration department. When a student or a graduate needs some document like a document for transferring from one college to another college or Bachelor diploma, he has to submit a request to the registration unit of his scientific department. Then, after finalize the first stage for issuing document and finish all procedures in the scientific department, the registration unit sends the document to the main registration department to complete the final stage to issue the document, this stage includes the final audit of the document, seal them and the signing of the document by the head of the department and the president of the University. The problem that this final stage takes time, so students and graduates must come again and again to the Registration Department to make sure the completion of their documents, this requires time and effort from the student or graduate, especially as there are students and graduates live in areas far from the university, and it's also requires time from employee himself to answer all people who come to his section, therefore we must design a system to manage information in this stage of preparing documents in order to save time for employees and to simplify for students and graduates to follow up their affairs in terms of issuing documents in final stage from anywhere without the need to refer to the main registration department at the university, which in turn is responsible for issuing these documents. The main registration department works only with documents of undergraduates (bachelor students).

It consists of four sections; each section has a manager:

1) "Graduates affairs". They prepare documents for students who finished two years and who finished four years. There are six types of documents:

a) diploma (for students who finished two years);

b) bachelor diploma with degrees in Arabic;

c) bachelor diploma with degrees in English;

d) graduation certificate in Arabic;

e) graduation certificate in English;

f) mural certificate;

2) “Documents Authentication”. The external company asks, if some diploma is not faked. This section confirm that this diploma is true and was given to that student;

3) “Undergraduates studies affairs”. They prepare four types of documents:

a. certificate of continuation of study;

b. certificate of ending (discontinue) of study;

c. document for transferring from one college to another college or from one department to another department inside one college;

d. postponements – special document which allow a student to stop studying for one year, after that he will be able to come back;

4) ratification. They take diplomas; send them to Ministry of Foreign Affairs. The Ministry put there stamps and return diplomas to ratification section.

The main purpose of this system is:

a) do not disturb employees and interrupt them during their work;

b) students or graduates don’t need to come to the university to ask about the readiness of their documents, except when documents are ready.

1.2. Comparative analysis of analogues

There is a comparative analysis of three analogical sites in other universities.

University of Babylon (Iraq-Babylon) [10]. In University of Babylon there is a system for communicating with students, in which they can enter their user names and passwords to login to the system. This system gives them the

possibility to send questions, inquiries and complaints. Now the system is under maintenance. It is not working now (fig. 1).

Fig. 1. University of Babylon

The Hashemite University (Jordan) [9]. They have a system for students "Student Portal", in this portal we have to enter University ID and password for logging into system and there is University logo in the top of page (fig. 2). We could not know the structure of this system because it's only for student who have the authority to enter.

Fig. 2. The Hashemite University

Zayed University (UAE) [13]. Here we have a screenshot of the registration office and its e-services. There are six services: “To Whom It May Concern” (it gives certificates of study continuation); “Graduation Certificate” for replacement of old documents by new ones; “Student ID Card” as a replacement of this card; “Academic Transcript”, “Student Course Registration” and “Student Data Modification” (fig. 3).

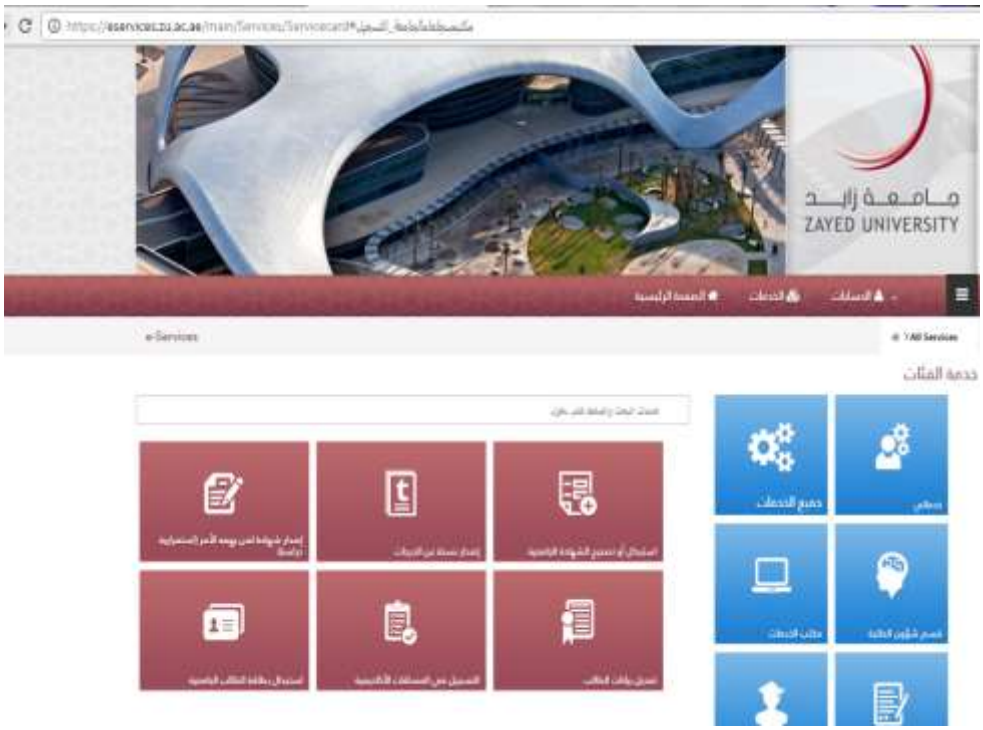


Fig. 3. Zayed University

If we press any of the services there is a page will appear with login form as shown in fig. 4.

Fig. 4. Login into e-services in Zayed University

This site shows that there are some implemented functions, which we need, but not all of them. They do not prepare certificate of ending (discontinue) of study; document for transferring from one college to another college or from one department to another department inside one college; postponements. Also, they don't work with ratification process through the site.

Analysis shows that each university has its own system for students or for graduates depending on University's structure and types of documents issued by. So, University of Technology needs a system that meets the needs of it. Moreover, in comparison with other universities, the University of Technology will have a wider range of functions for students', graduates' and undergraduates' documents then others.

1.3. Tools for development

There are a large number of students and graduates at the University of Technology (7150 Students and 1543 graduates) so we need a database to manage information about them [11].

A database management system (DBMS) is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. A general-purpose DBMS is designed to allow the definition, creation, querying, update, and administration of databases. Well-known DBMSs are MySQL, PostgreSQL, MongoDB, Microsoft SQL Server, Oracle, Sybase, SAP HANA, and IBM DB2 [5].

I choose MySQL because it is popular for use with web applications, closely associated to PHP programming language, which is often used along with MySQL. Many high-traffic web sites use MySQL as the backend for its data warehouse. MySQL is very popular with startup companies, small or medium businesses and projects because it can be easy to use at a low cost. In case when high speed applications are performed for web, gaming and medium or small data storages MySQL surpasses all the other database management systems [15, 16].

A Content Management System (CMS) is a software application used to create, customize, and manage information. Most CMS can be used alone or in conjunction (integrated) with other applications. They can be set up directly on a network, the Internet, or even to run “locally” on your own computer.

The most widespread use of CMS today involves the quick creation of powerful websites that do not require a high level of programming knowledge to set up, customize, and maintain.

CMS can be broken down into two main types: Proprietary CMS and Open Source CMS.

Proprietary CMS

Many companies sell licenses to use their own proprietary CMS. “Proprietary” generally means someone owns the rights to the CMS application and you need permission or a license to use it. Even with a license, in most cases, license holders may still be prohibited from duplicating the CMS or making alterations to the application unless they purchase a more expensive "developers" license.

Some proprietary CSM can, and are designed to work outside the environment of the creator but it is important that you understand where the CMS you choose will run properly because many types of proprietary CMS will only work when the site you build with them is hosted by the CMS owner. For example, most online "build it yourself" website services use some form of proprietary CMS. If you build a website "live" through their tools the site will only work as long as you keep it with that company's CMS.

If you try to move your domain somewhere else, the website you created in their proprietary environment may no longer work or may be converted to another format.

Two of the biggest downsides of using a proprietary CMS are the cost of licenses and, because many web host companies do not support proprietary CMS, you may be limited as to where you can host your website.

This lack of "portability" is probably the major reason most small business owners choose to use Open Source CMS.

Open Source CMS

The most popular Open Source CMSs (OS CMS) run on PHP (a scripting language well-suit for web development that can be embedded into HTML): WordPress, Joomla, and Drupal (the USA White House website is a Drupal site). Open source (OS) programs can be used by anyone for any purpose and do not require you to purchase a license. You may also customize OS CMS without special permission.

A few of the significant benefits when using an OS CMS:

- they are cheaper; no license fees, no fees for upgrades, no contracts to sign and no long-term commitments;
- because anyone can develop OS applications there are already countless free modules, plugins, and complimentary tools so you won't need to hire a developer;
- there are hundreds of thousands of free templates (or, "themes") available for OS CMS;
- search engines easily works with OS CMS and WordPress, in particular, it is simple to optimize for search engines using plug-in tools [19].

I decide not to write PHP files directly, but to use CMS because there are seven key benefits of using a CMS [1]:

- it’s easy for the non-technically minded;
- it allows multiple users;
- it streamlines scheduling;
- it improves site maintenance;
- design changes are simple;
- it helps you manage content;
- you’re in control.

One of the most popular content management system options is WordPress, used by over 27.1 % of websites on the internet [18]. Other content management systems include Joomla, Drupal, and SharePoint.

I choose to work with WordPress because it’s easy to use and flexible enough for just about anything. That’s the main reason why WordPress has grown so much in popularity.

Due to its robust features, many of the top brands use WordPress to power their websites including but not limited to: Time Magazine, Google, Facebook, Sony, Disney, LinkedIn, The New York Times, CNN, eBay, and more. In addition [20]:

- WordPress is Free;
- easy to use and learn;
- extendable by using themes and plugins;
- search engine friendly;
- easy to manage;
- safe and secure;
- can handle different media types.

In the first chapter, there is an explanation of the reasons for which the system has been developed, as well as the description of the systems used by three Arab universities that cannot be provided free, and there is a description for the tools chosen for development of the system and the reasons for selecting them.

2. DESIGN OF MANAGEMENT INFORMATION SYSTEM FOR THE MAIN REGISTRATION DEPARTMENT IN THE UNIVERSITY OF TECHNOLOGY

2.1. Functional and non-functional requirements

One of the mandatory stage in application design is the determination of functional and non-functional requirements [8].

The problem statement defines the following functional requirements for the future system.

1. The system must work with four types of users: students, graduates, managers and head of the department.

2. The system must have four sections.

The 1st, 2nd and 4th sections of system must be available for only graduates. The third section of system must be available for only students.

Student must be able to see 1 section: “Undergraduates studies affairs”. In this section he must see tables with only information about himself.

Graduates must be able to see 3 sections: “Graduates affairs”, “Documents Authentication”, “Ratification”. In these sections he must see tables with only information about himself.

The manager must be able to work with information of only his section. The manager of any section must be able to:

a) add the new record in the table with information about students or graduates;

b) modify records in the table with information about students or graduates;

c) delete records in the table with information about students or graduates;

d) search records in the table with information about students or graduates.

The head of the department must be able to see 4 sections with all tables of information. The head of the department must be able to search records in 4 sections.

Non-functional requirements are the following.

1. The system must be connected to a database.
2. The system must be entered by only authorized users, they can access the system with user name and password.
3. The system must be accessed by web.

2.2. Use case and deployment diagrams

Unified Modeling Language (UML) 2.2 has three categories of diagrams:

1) structure diagrams;
2) behavior diagrams;
3) interaction diagrams.

A use case diagram describes the functionality provided by a system in terms of actors, their goals represented as use cases, and any dependencies among those use cases [14, 4].

According to functional requirements I developed the use case diagram (fig. 5). Use case diagram consists of three basic users (“student/graduate”, “manager” and “head of department”). “Manager” inherits the user “Head of department”.

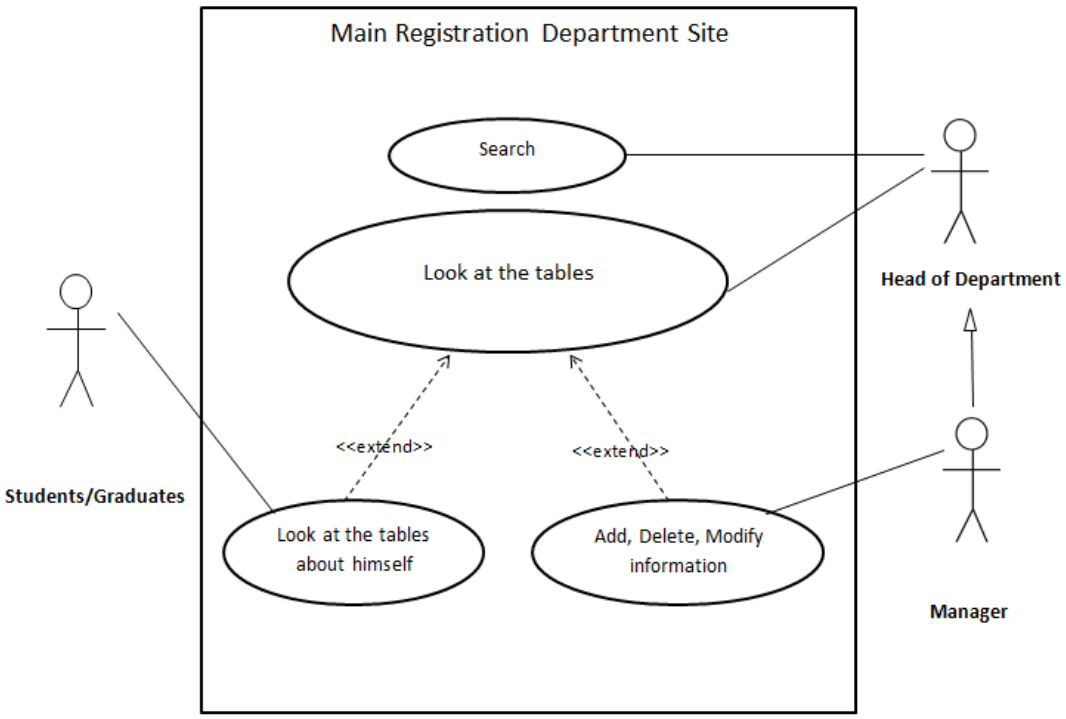


Fig. 5. Use case diagram

"Head of Department" can look at all tables; "Manager" can add, delete and modify data in those tables. "Student/graduate" can look at the table with information only about himself.

Use cases "Look at the tables about himself" and "Add, delete, modify information" are extending the base use case "Look at the tables".

A deployment diagram describes the hardware used in system implementations and the execution environments and artifacts deployed on the hardware [14]. Deployment diagram for this system deploys at two nodes: "User computer" and "Server" (fig. 6). Many user computers can connect to the Server. Server contains Database component, component "PHP scripts" and "Website". Component "Database" is connected to "Website". Component "Website" depends on "PHP scripts".

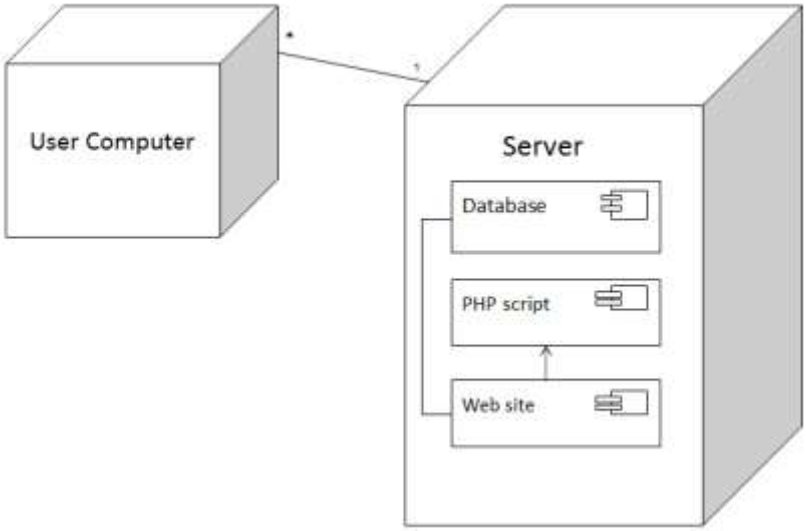


Fig. 6. Deployment diagram

2.3. Database scheme

This database schema of system consists of 12 tables (fig. 7). They are connected with each other with relationship of kind one to many.

This system has 4 type of users: student, graduate, manager and the head of the department, all of them are listed in a table “Type” (fig. 8). The database has a table “Sections” which contains 5 sections: undergraduates affairs section, graduates affaires section, documents authentication section, ratification section and head of department section (fig. 9). Each section has several managers but there is only one head of department. They are all saved in a table “Manager” (fig. 10). Other users (students and graduates) are listed in a table “Student_graduate” which contains their full names and the department that they belong to.

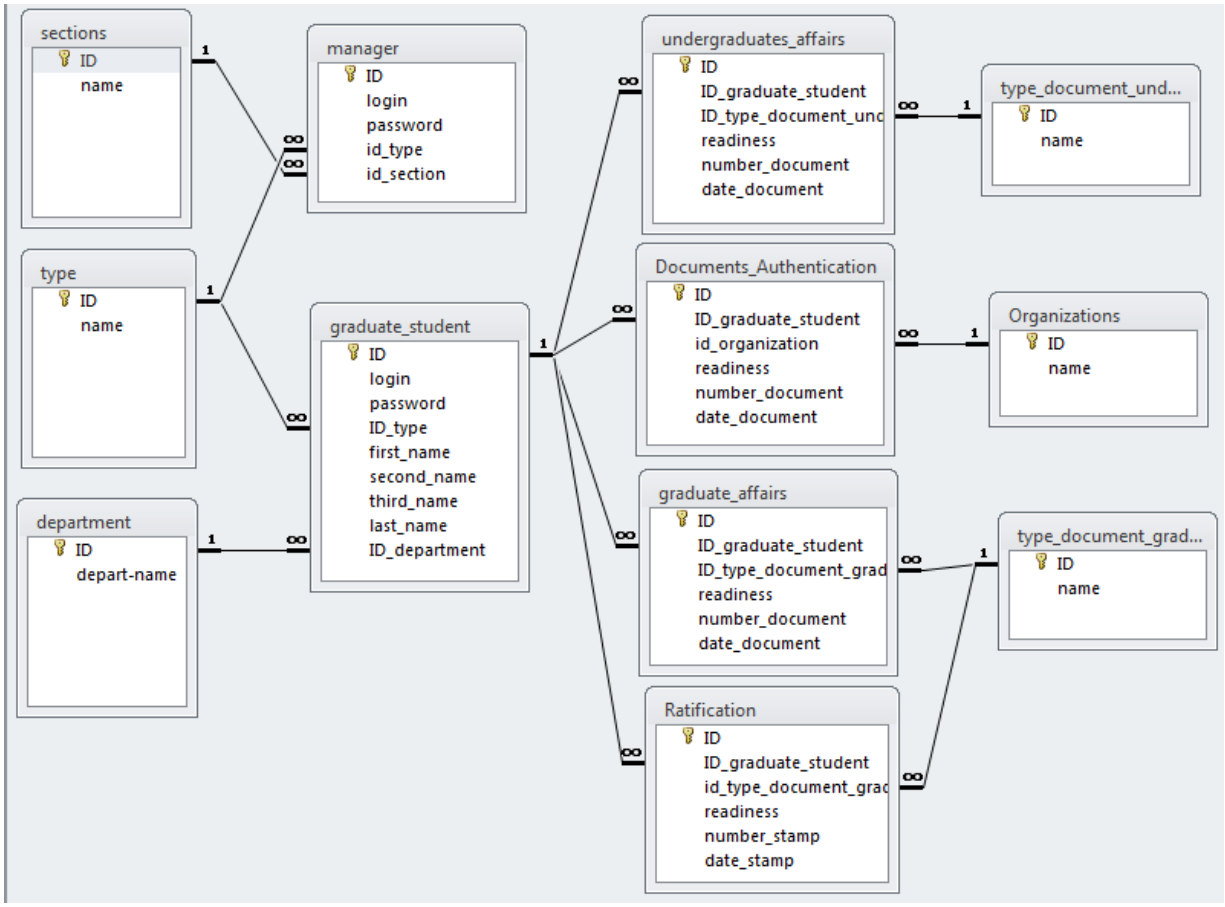


Fig. 7. Database scheme

There are 16 departments. All departments' names are listed in a table "Department" (fig. 11).

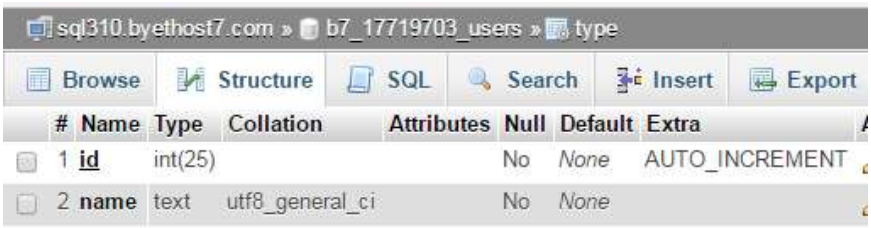


Fig. 8. Structure of the table "type"

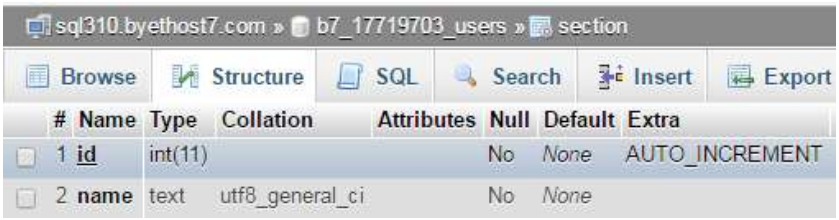


Fig. 9. Structure of the table "sections"

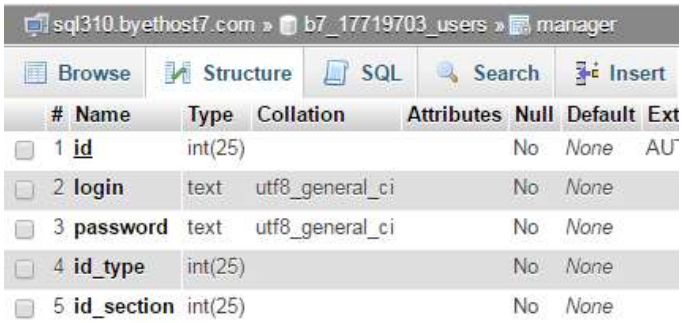


Fig. 10. Structure of the table “manager”

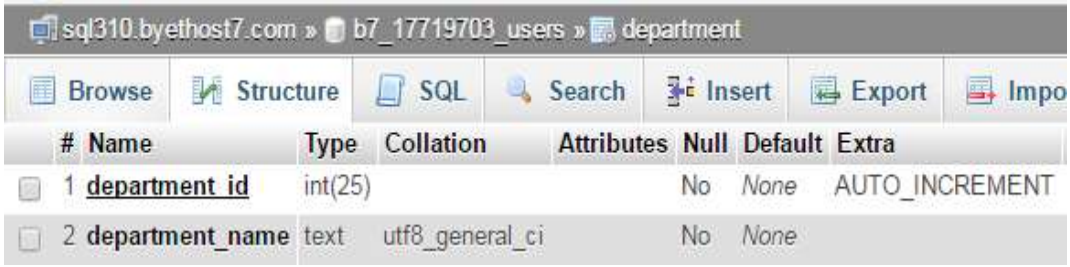
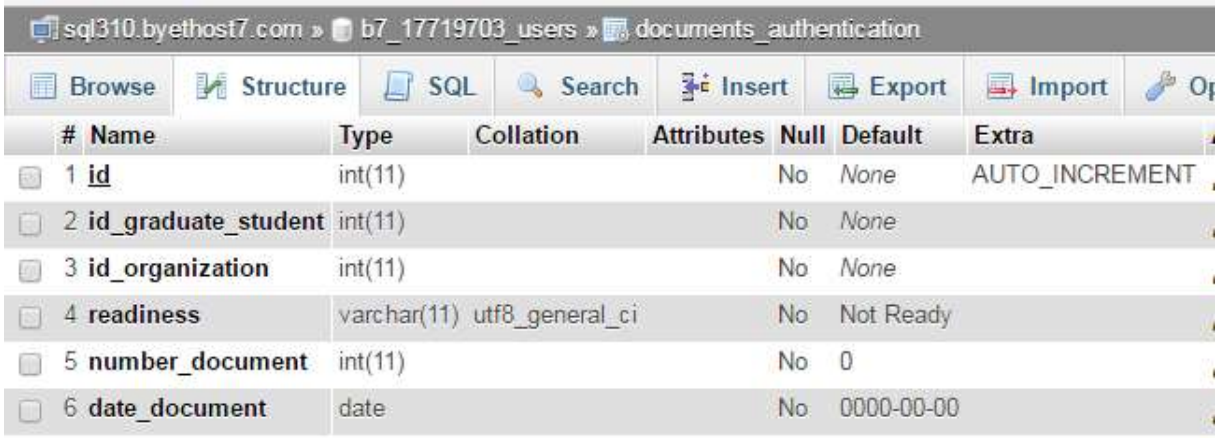


Fig. 11. Structure of the table “department”

The table “Documents_authentication” contains information about the documents that have been sent by the “Organizations” in order to confirm graduate’s document for employment purposes or for the purposes of the study. The table “Documents_authentication” consists of 6 fields as shown in (fig. 12). These fields have different types of data: int, varchar and date. The primary key is the field “id”, it is automatically incremented. Names of organizations are listed in a table “Organizations” (fig. 13).

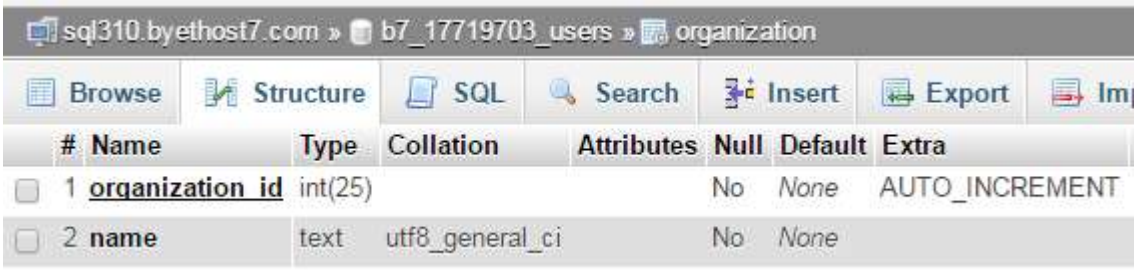
The table “Ratification” contains information about documents that need to be sent to the Ministry of Foreign Affairs in order to be stamped there and then be brought back. The table “Ratification” consists of 6 fields as shown in (fig. 14). These fields have different types of data: int, varchar and date. The primary key is the field “id”, it is automatically incremented. The third field is for type of document for graduates because there are 6 types of graduate’s documents: Diploma, Bachelor diploma with degrees in Arabic, Bachelor diploma with degrees in English, Graduation Certificate in Arabic, Graduation

Certificate in English and Mural certificate. They are all listed in a table "Type_document_graduate" (fig. 15).



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	id_graduate_student	int(11)			No	None	
3	id_organization	int(11)			No	None	
4	readiness	varchar(11)	utf8_general_ci		No	Not Ready	
5	number_document	int(11)			No	0	
6	date_document	date			No	0000-00-00	

Fig. 12. Structure of the table "Documents_authentication"



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	organization_id	int(25)			No	None	AUTO_INCREMENT
2	name	text	utf8_general_ci		No	None	

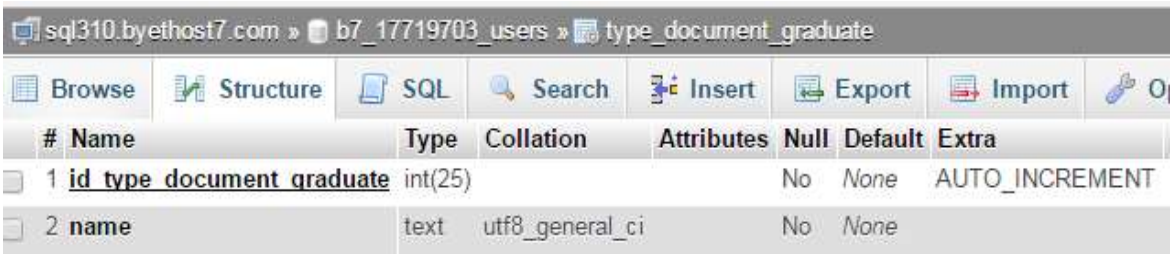
Fig. 13. Structure of the table "organization"

The table "undergraduates_ affairs" consists of 6 fields as shown in (fig. 16). These fields have different types of data: int, varchar and date. The third field is for type of document for undergraduates because there are 5 types of undergraduate's documents: Certificate of continuation of study, Certificate of ending of study, Transition between Universities, Transition between Department and Postponements. They are all listed in a table "type_document_undergraduate" (fig. 17).

The table "graduates_affairs" contains 6 fields as shown in (fig. 18). These fields about graduate's documents type and their number and date and the readiness of them.

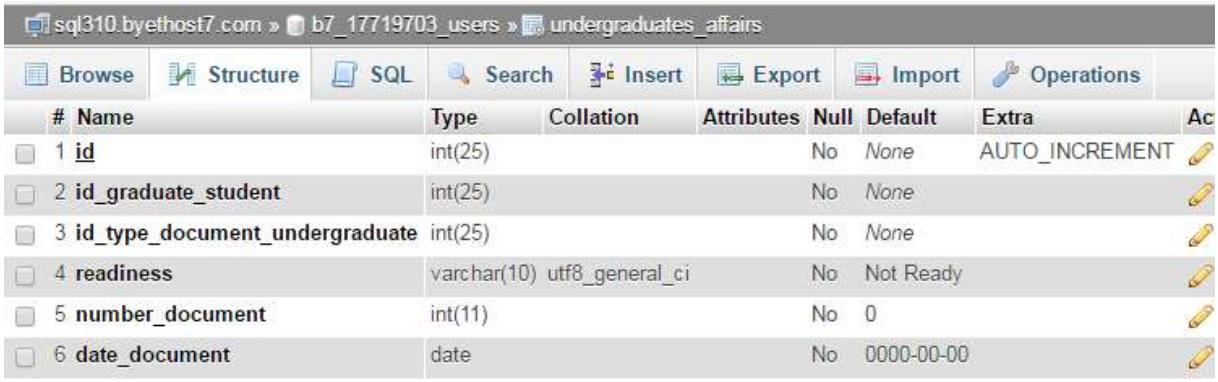
#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(25)			No	None	AUTO_INCREMENT
2	id_graduate_student	int(25)			No	None	
3	id_type_document_graduate	int(25)			No	None	
4	readiness	varchar(10)	utf8_general_ci		No	Not Ready	
5	number_stamp	int(11)			No	0	
6	date_stamp	date			No	0000-00-00	

Fig. 14. Structure of the table “Ratification”



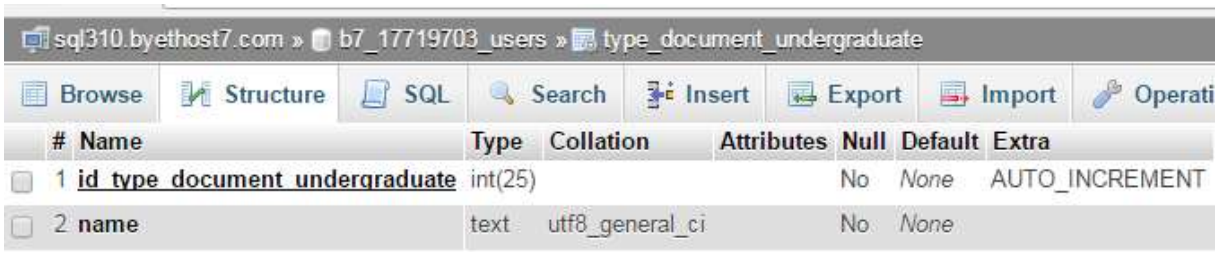
#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id_type_document_graduate	int(25)			No	None	AUTO_INCREMENT
2	name	text	utf8_general_ci		No	None	

Fig. 15. Structure of the table “type_document_graduate”



#	Name	Type	Collation	Attributes	Null	Default	Extra	Ac
1	id	int(25)			No	None	AUTO_INCREMENT	
2	id_graduate_student	int(25)			No	None		
3	id_type_document_undergraduate	int(25)			No	None		
4	readiness	varchar(10)	utf8_general_ci		No	Not Ready		
5	number_document	int(11)			No	0		
6	date_document	date			No	0000-00-00		

Fig. 16. Structure of the table “undergraduates_affairs”



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id_type_document_undergraduate	int(25)			No	None	AUTO_INCREMENT
2	name	text	utf8_general_ci		No	None	

Fig. 17. Structure of the table “type_document_undergraduate”

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(25)			No	None	AUTO_INCREMENT
2	id_graduate_student	int(25)			No	None	
3	id_type_document_graduate	int(25)			No	None	
4	readiness	varchar(10)	utf8_general_ci		No	Not Ready	
5	number_document	int(11)			No	0	
6	date_document	date			No	0000-00-00	

Fig. 18. Structure of the table “graduates_affairs”

3. THE IMPLEMENTATION OF THE WEB-APPLICATION

3.1. File structure

File structure contains 36 files with the extension “php” in addition to WordPress files. Files of web-application are located inside the folder of theme “Responsive”. Each file is defined as a template inside the theme to be more flexible for coding. There are 3 main folders in WordPress: “wp-admin”, “wp-content” and “wp-include”. “Responsive” folder is located in folder “wp-content”. In table 1 there is the description of PHP-files of the project and their functions.

Table 1. File structure and functions

No.	File	Functions
1.	login	To login users to their pages
2.	graduate affairs	To view table for graduates and their documents and the possibility of deleting records for manager of graduates affairs section
3.	addgrad	To add records to the graduates affairs section table. It’s done only by manager
4.	modifygrad	To modify records in the graduates affairs section table. It’s done only by manager
5.	searchgrad	To search records by name in the graduates affairs section table. It’s done only by manager
6.	ratificationmanager	To view table for graduates and their documents and the possibility of deleting records for manager of ratification section
7.	addratification	To add records to the ratification section table. It’s done only by manager
8.	modifyratif	To modify records in the ratification section table. It’s done only by manager
9.	searchratif	To search records by name in the ratification section table. It’s done only by manager
10.	documents_authentication	To view table for graduates and their documents and the possibility of deleting records for manager of documents authentication section
11.	addrowdoc	To add records to the documents authentication section table. It’s done only by manager
12.	modifydoc	To modify records in the documents authentication section table. It’s done only by manager
13.	searchdoc	To search records by name in the documents authentication section table. It’s done only by manager

Continuation of tab. 1

14.	getdata	To bring all surnames that belong to the chosen department at pages: addgrad, addratification, addrowdoc, modifygrad, modifyratif and modifydoc
15.	getnamedata	To bring all names that belong to the chosen surname at pages: addgrad, addratification, addrowdoc, modifygrad, modifyratif and modifydoc
16.	getfathernamegrad	To bring all father names that belong to the chosen name at pages: addgrad, addratification, addrowdoc, modifygrad, modifyratif and modifydoc
17.	getgrandfathernamegrad	To bring all grandfather names that belong to the chosen name at pages: addgrad, addratification, addrowdoc, modifygrad, modifyratif and modifydoc
18.	undergraduates affairs	To view table for students and their documents and the possibility of deleting records for manager of undergraduates affairs section
19.	addrowunder	To add records to the undergraduates affairs section table. It's done only by manager
20.	modifyundergrad	To modify records in the undergraduates affairs section table. It's done only by manager
21.	searchunder	To search records by name in the undergraduates affairs section table. It's done only by manager
22.	getsurnameunderdata	To bring all surnames that belong to the chosen department at pages: addrowunder and modifyundergrad
23.	getnameunderdata	To bring all names that belong to the chosen surname at pages: addrowunder and modifyundergrad
24.	getfathernameunderdata	To bring all father names that belong to the chosen name at pages: addrowunder and modifyundergrad
25.	getgrandfathernameunderdata	To bring all grandfather names that belong to the chosen father name at pages: addrowunder and modifyundergrad
26.	head_of_department	To view 4 section for the head of department
27.	head_graduates affairs	To show all records in graduate affairs section to the head of department
28.	head_documents authentication	To show all records in documents authentication section to the head of department
29.	head_ratification	To show all records in ratification section to the head of department

End of tab. 1

30.	head_undergraduates affairs	To show all records in undergraduates affairs section to the head of department
31.	headgradsearch	To search records by name in the graduates affairs section table. It's done by the head of the department at page "head_graduates affairs"
32.	headsearchdoc	To search records by name in the documents authentication section table. It's done by the head of the department at page "head_documents authentication"
33.	headsearchratif	To search records by name in the ratification section table. It's done by the head of the department at page "head_ratification"
34.	headsearchunder	To search records by name in the undergraduates affairs section table. It's done by the head of the department at page "head_undergraduates affairs"
35.	graduate	To show to the graduate his own records from 3 sections tables: graduates affairs, ratification and documents authentication
36.	student	To show to the student his own records from undergraduates affairs section table

3.2. SQL-queries

I have written 26 different SQL-queries in my project [3]. They work with different tables in database. There are queries for selection, updating, removing and adding information to the tables written in PHP script [6].

Fig. 19 shows the fragments of code of selection all graduates from the table "graduates_affairs" and representation them on manager's page.

```
$sql= "SELECT * FROM department,graduate_student,graduates_affairs,type_document_graduate
        WHERE department. department_id=graduate_student.id_department AND
        graduate_student.id=graduates_affairs.id_graduate_student AND
        graduates_affairs.id_type_document_graduate=type_document_graduate.id_type_document_graduate";
$records= mysqli_query($db,$sql) or die('Error querying database');
$count=mysqli_num_rows($records);
```

Fig. 19. Show all graduates in "graduates affairs" section

At the interface of manager, the user can select department. The other dropdown lists will contain information about only those surnames and names of graduates, which belong to the chosen department. Fig. 20 shows the fragment of code for selecting department and sends the value of department to java script

function “getsurname”. This function in turn sends this value to a file “getdata” in order to use this value in the query and get surnames related to the value of department.

```
Department:<select name="department" id="selectdepartment" onchange="getsurname(this.value)" >
  <option value=""> Choose department</option>
  <?php

    $sql2="SELECT * FROM department   ORDER BY department_name ASC";
    $record2=mysqli_query($db,$sql2) or die('Error querying database');

    while ( $result2= mysqli_fetch_assoc($record2)) {
     echo "<option value='". $result2['department_id']."'>" . $result2['department_name']."</option>";
    }
  ?>
        </select>

         // dropdown list for surnames will receive surnames values from function "getsurname"
Surname: <select name='last_name' id="lastnamelist" onclick="getnameId(this.value)">

          <option value="" ></option>
        </select>
```

Fig. 20. Selecting department and sending its value to JS function

Fig. 21 shows java script function “getsurname” that sends department’s value to file “getdata”. This file includes a query to get all surnames.

```
<script>
    function getsurname(val){
        $.ajax({
            type:"POST",
            //sending value to file "getdata"
            url:"getdata.php",
            data:"depart="+val,
            success: function(data) {
                    // returned values will be send to lastnamelist
                        $("#lastnamelist").html(data);
                 }
            })
    }
</script>
```

Fig. 21. Java script function “getsurname”

Fig. 22 shows the file “getdata” which receives the department value from function ”getsurname” and uses this value in query to obtain all surnames that belong to the related department.

```php
<?php
 if (isset($_POST["depart"])){

  $depart=$_POST['depart'];
  $query="SELECT DISTINCT last_name FROM graduate_student WHERE
          graduate_student.id_type='3'AND graduate_student.id_department=$depart ";
  $result= mysqli_query($db,$query);
   foreach ($result as $graduate){
?>
   <option value="<?php echo $graduate['last_name'];?>"><?php echo $graduate['last_name']; ?> </option>
<?php
   }
 }
?>
```

Fig. 22. File “getdata”

On the manager’s page, the user can delete student’s information from the relative table. Fig. 23 shows this case for “undergraduates affairs” section.

```php
if(isset($_POST['delete'])){
    $impid=implode(",",$id);
    $sqld="DELETE FROM undergraduates_affairs WHERE id IN (".$impid.") "or DIE("Error In Delete");
    $Qdelete= mysqli_query($db,$sqld);
  }
```

Fig. 23. Removing student’s information

Fig. 24 and (fig. 25) show how to put all selected values in variables and then to get “id” for the student according to his full name and his department from the tables “graduate_student” and “department” to insert then all values to the table “undergraduates_affairs”

```php
//put all selected values in variables to add them to table "undergraduates_affairs"
   $firstname=mysqli_real_escape_string($db,$_POST['firstname']);
   $secondname=mysqli_real_escape_string($db,$_POST['second_name']);
   $thirdname=mysqli_real_escape_string($db,$_POST['third_name']);
   $lastname=mysqli_real_escape_string($db,$_POST['last_name']);
   $department=mysqli_real_escape_string($db,$_POST['department']);
   $typeofdocument=mysqli_real_escape_string($db,$_POST['document']);
   $noofdocument=mysqli_real_escape_string($db,$_POST['number_document']);
   $readiness=mysqli_real_escape_string($db,$_POST['readiness']);
   $dateofdocument=mysqli_real_escape_string($db,$_POST['date_document']);
```

Fig. 24. Putting student’s information in variables

```
$sql1= "SELECT id from graduate_student,department WHERE graduate_student.first_name='$firstname'
        graduate_student.second_name= '$secondname' AND  graduate_student.third_name='$thirdname'
        graduate_student.id_department = department.department_id AND
        department.department_id='$department'";
$records1= mysqli_query($db,$sql1) or die('Error querying database1');
$row1 = mysqli_fetch_array($records1,MYSQLI_ASSOC);

 // inserting all values to table "undergraduates_affairs"
$sql="INSERT INTO undergraduates_affairs (id,id_graduate_student,id_type_document_undergraduate,
      readiness,number_document,date_document) VALUES
      ('','$row1[id]','$typeofdocument',' $readiness','$noofdocument','$dateofdocument')";
$records= mysqli_query($db,$sql) or die('Error querying database');
```

Fig. 25. Inserting obtained values to table “undergraduates_affairs”

On the manager’s page, user can select the type of document from dropdown list. Fig. 25 shows how this done for undergraduates affairs section.

```
Type of Document<select name="document">
  <option value="none"></option>
  <?php
  $sql2="SELECT name FROM type_document_undergraduate";
  $record2=mysqli_query($db,$sql2) or die('Error querying database');

  while ( $result2= mysqli_fetch_assoc($record2)) {
      echo "<option value='".$result2['id_type_document_undergraduate']."'>". $result2['name'].
       } ?>
  </select>
```

Fig. 25. Selecting type of document

Fig. 26 and (fig. 27) show fragment of code for modifying the graduate information by the manager of the “documents authentication” section. This is implemented by putting all modified values in variables and getting “id” for the graduate by his full name and his department and then updating graduate’s information with new values in the table “documents_authentication”

```
$name=mysqli_real_escape_string($db,$_POST['name']);
$secondname=mysqli_real_escape_string($db,$_POST['second_name']);
$thirdname=mysqli_real_escape_string($db,$_POST['third_name']);
$lastname=mysqli_real_escape_string($db,$_POST['last_name']);
$department=mysqli_real_escape_string($db,$_POST['department']);
$organization=mysqli_real_escape_string($db,$_POST['organization']);
$noofdocument=mysqli_real_escape_string($db,$_POST['number_document']);
$readiness=mysqli_real_escape_string($db,$_POST['readiness']);
$dateofdocument=mysqli_real_escape_string($db,$_POST['date_document']);
```

Fig. 26. Putting modifying values in variables

```
$sql3="SELECT id from graduate_student,department WHERE graduate_student.first_name='$name' AND
       graduate_student.second_name='$secondname'AND graduate_student.third_name='$thirdname'AND
       graduate_student.id_department= department.department_id AND
       department.department_id='$department'";
$records1= mysqli_query($db,$sql3) or die('Error querying database1');
$row1 = mysqli_fetch_array($records1,MYSQLI_ASSOC);

   //updating values in table "documents_authentication" with new values
$sql4="UPDATE documents_authentication SET id_graduate_student='$row1[id]',
id_organization='$organization',readiness='$readiness',number_document='$noofdocument',
date_document='$dateofdocument' WHERE documents_authentication.id='$id'";
$recordsnew= mysqli_query($db,$sql4) or die('Error querying database3');
```

Fig. 27. Modifying graduate's information

At interfaces of the head of department and manager there is the ability to search students/graduates by name. The fragment of code for this ability shown in (fig. 28.) It is implemented by searching for a given name in table "graduate_student" in the first query and selecting all matching names to show them at interface in the second query.

```
// searching for matching names of students with the given name
$sql= "SELECT * FROM graduate_student WHERE graduate_student.first_name LIKE '$search'
       OR second_name LIKE '$search' OR third_name LIKE '$search' OR last_name LIKE '$search'
       AND id_type='2' ";
 $raw_results = mysqli_query($db,$sql) or die('Error querying database1');
if(mysqli_num_rows($raw_results) > 0){

while($results = mysqli_fetch_array($raw_results)){

   //selecting all other informations that belong to the matching names
 $sqlname="SELECT * FROM department,graduate_student,undergraduates_affairs,type_document_undergraduate WHERE
           department. department_id=graduate_student.id_department AND
           graduate_student.id=undergraduates_affairs.id_graduate_student AND
           graduate_student.id= '$results[id]' AND
           undergraduates_affairs.id_type_document_undergraduate=type_document_undergraduate.id_type_document_undergraduate";
  $raw_resultsname = mysqli_query($db,$sqlname) or die('Error querying database2');
```

Fig. 28. Searching students by name

Fig. 26 and (fig. 27) show the fragment of code for modifying type of document and department at interface of manger for graduates' affairs section.

```
Type of Document<select name="document">
   <option value="<?php echo $result1['name']; ?>"> <?php echo $result1['name']; ?> </option>
 <?php
          $seldoc= $result1['name'];
          $sql2="SELECT * FROM type_document_graduate WHERE type_document_graduate.name !='$seldoc'";
          $record2=mysqli_query($db,$sql2) or die('Error querying database');

          while ( $result2= mysqli_fetch_assoc($record2)) {
            echo  "<option value='". $result2['name']."'>" . $result2['name']. "</option>";
           } ?>

   </select>
```

Fig. 29. Modifying type of document

```
Department:<select name="department" id="selectdepartment" onchange="getId(this.value);" >
          <option value="<?php echo $result1['department_id']; ?>"> <?php echo $result1['department_name']; ?> </option>
          <?php
          $seldepart= $result1['department_name'];
          $sql2="SELECT * FROM department  WHERE  department.department_name !='$seldepart' ORDER BY department_name ASC";
          $record2=mysqli_query($db,$sql2) or die('Error querying database');

          while ( $result2= mysqli_fetch_assoc($record2)) {
            echo  "<option value='". $result2['department_id']."'>" . $result2['department_name']. "</option>";
           } ?>
          </select>
```

Fig. 30. Modifying the department

4. THE TESTING OF THE WEB-APPLICATION

4.1. Testing methods

There are several methods of testing the software [17]. One of them is the functional testing.

In terms of development of the web-application the functional testing means to test for all the links in the web pages, database connections, forms used in the web pages for submitting or getting information from user, and testing of cookies.

Check of the links means to test the following [7]:

- test the outgoing links from all the pages from specific domain;
- test all internal links;
- test links jumping on the same pages;
- test to check if there are any orphan pages;
- lastly, to check for broken links in all above mentioned links.

Interface testing means to test the interfaces of the application. The main interfaces are: web server and application server interface; application server and database server interface. We understand the following under the interface testing:

- check if all the interactions between these servers are executed properly. Errors are handled properly. If database or web server returns any error message for any query by application server then application server should catch and display these error messages appropriately to users;
- check what happens if user interrupts any transaction in-between;
- check what happens if connection to web server is reset in-between.

4.2. Functional testing

At first the functional requirements were taken as the base for the functional testing. Function: Student must be able to see one section: “Undergraduates studies affairs”. In this section student must see tables with only information about himself.

Result of testing: passed

Function: Graduates must be able to see 3 sections: “Graduates affairs”, “Documents Authentication”, “Ratification”. In these sections he must see tables with only information about himself.

Result of testing: passed

Function: The manager must be able to work with information of only his section.

Result of testing: passed

Function: The manager of any section must be able to modify records in the table with information about students or graduates.

Result of testing: passed

Function: The manager of any section must be able to add the new record in the table with information about students or graduates.

Result of testing: passed

Function: The manager of any section must be able to delete records in the table with information about students or graduates.

Result of testing: passed

Function: The manager of any section must be able to search for records in the table with information about students or graduates.

Result of testing: passed

Function: The head of the department must be able to see 4 sections with all tables of information.

Result of testing: passed

Function: The head of the department must be able to search for records in 4 sections.

Result of testing: passed

Additionally we performed the functional testing of all internal links between pages and links jumping on the same pages. The result was successful.

There are no orphan pages at the site and no broken links [7, 2]. Orphan pages mean pages with lost or unused pages.

4.3. Interface testing

Any visitor can see the main page. The main page contains function for login users to the system as in the (fig. 31).

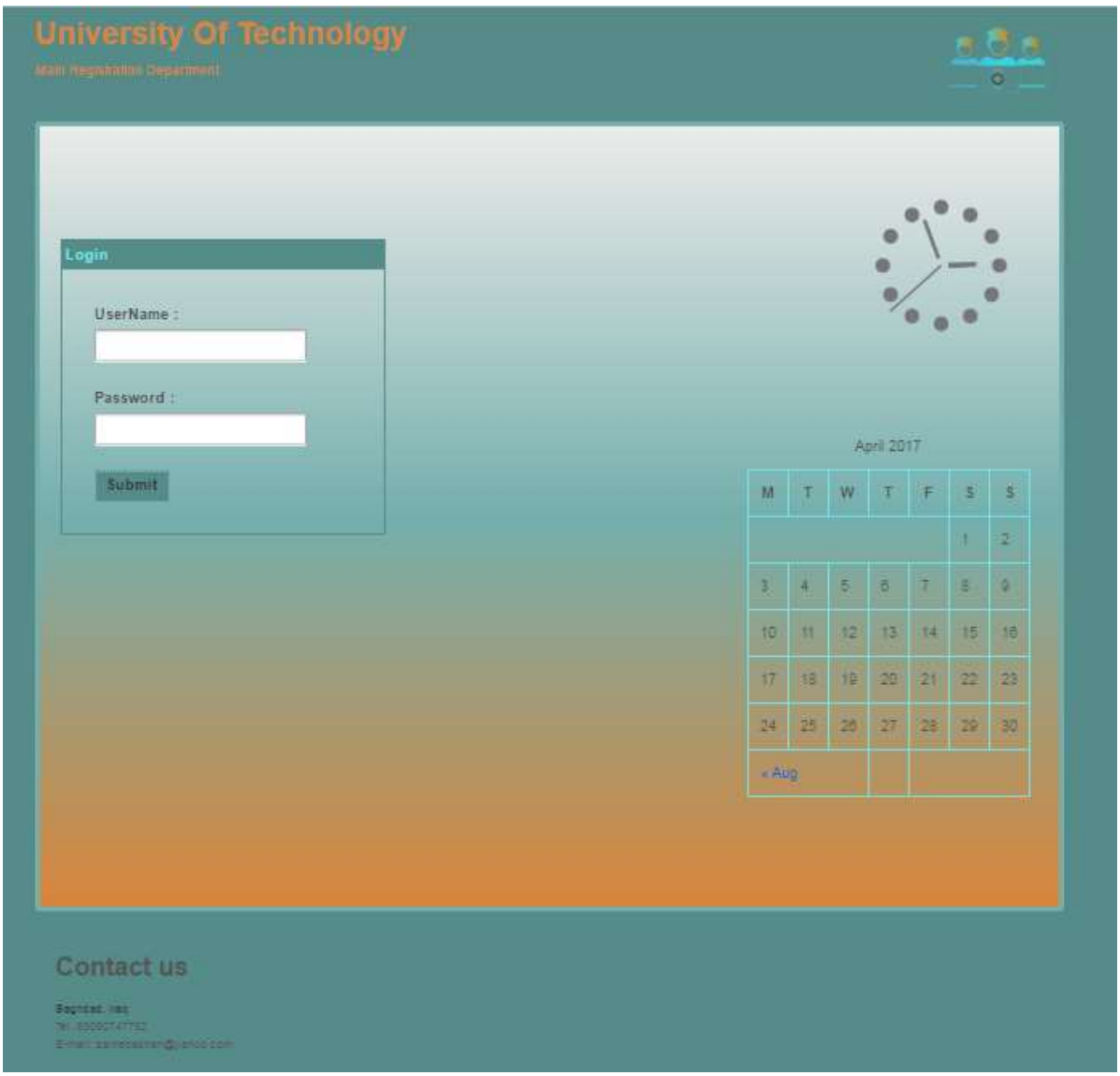


Fig. 31. The main page

After entering the right “username” and “password” the system will redirect each user to his own interface. (Fig. 32) shows the interface for manager of “Graduates affairs” section. The page “Graduates affaires” contains 4 functions (“Add”, “Delete”, “Modify” and “Search”). Fig. 33 shows the “Add graduate page” shown when Manager clicks button “Add” at the page “Graduates affairs”. Fig. 34 shows the page “Modify graduate’s data” shown when manager clicks on “Modify” link for specific graduate.

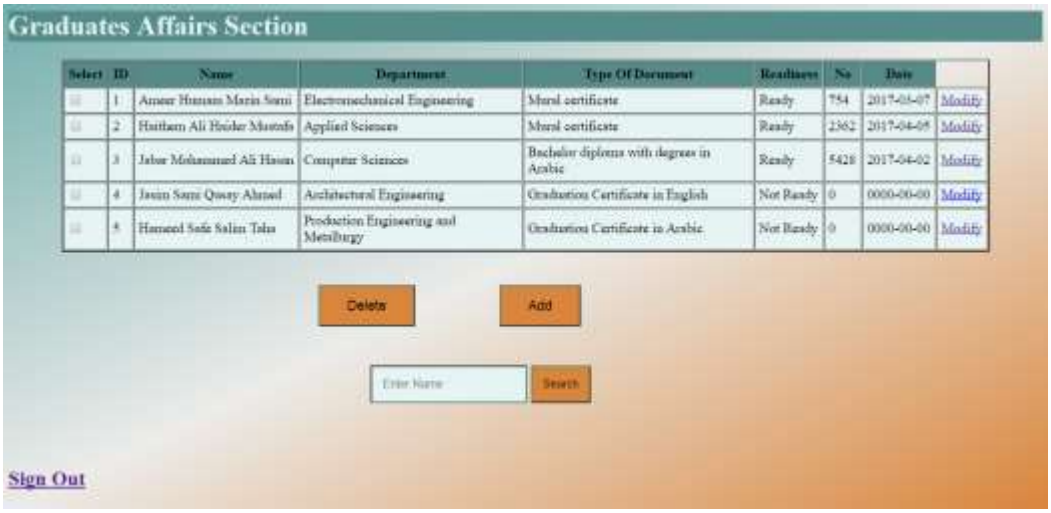


Fig. 32. Interface for Manager of “Graduates affairs” section

Fig. 33. “Add graduate page”

Fig. 34. “Modify graduate’s data page”

Fig. 35 shows page “Search Graduate” shown when manager clicks on button “Search” after entering a name in input box.

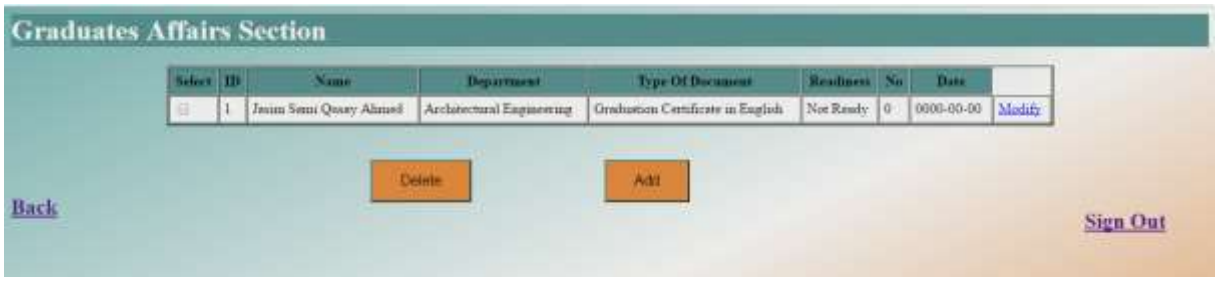


Fig. 35. “Search graduate page” for manager of
“Graduates affairs” section

The function “Delete” working at the same page “Graduates affairs” with showing confirmation message at the same page as shown in (fig. 36).

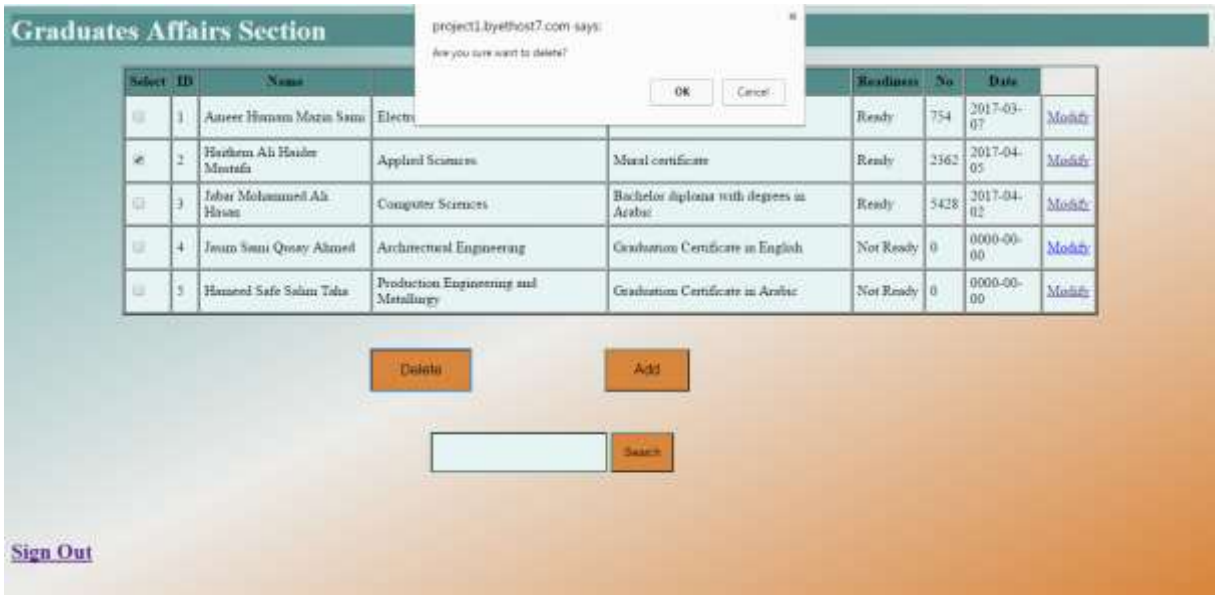


Fig. 36. Confirmation message for “Delete” function

“Undergraduates affairs page” will be shown for manager of “Undergraduates affairs section” as shown in (fig. 37). It contains 4 functions (“Add”, “Delete”, “Modify” and “Search”).

Fig. 38 shows the “Add student page” shown when Manager clicks button “Add” at the page “Undergraduates affairs”. Fig. 39 shows the page “Modify student’s data” shown when manager clicks on “Modify” link for specific student.

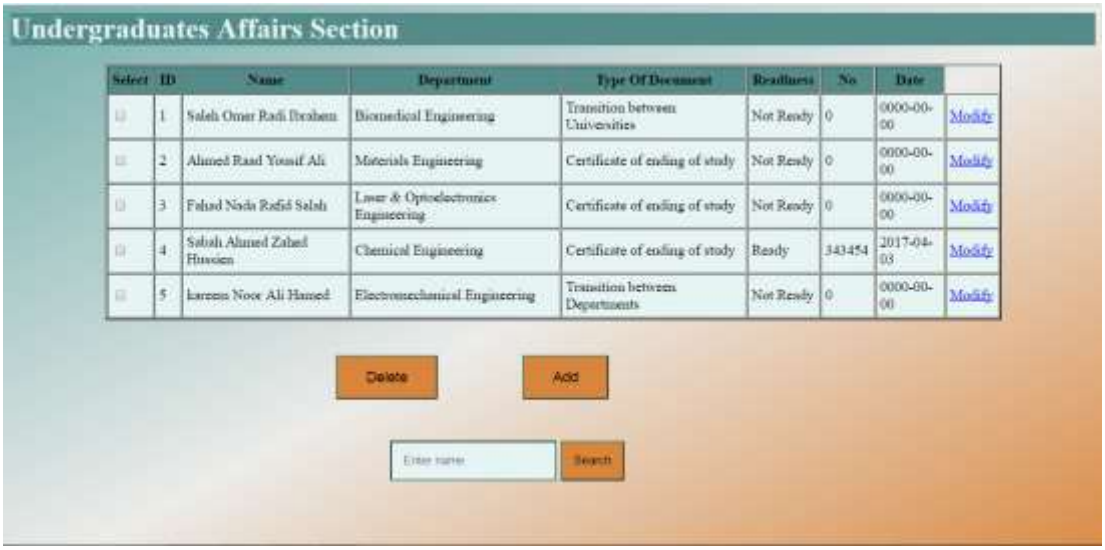


Fig. 37. Interface for Manager of “Undergraduates affairs” section

Fig. 38. “Add student page”

Fig. 39. Page “Modify student’s data”

Fig. 40. shows page “Search Student” shown when manager clicks on button “Search” after entering a name in input box.

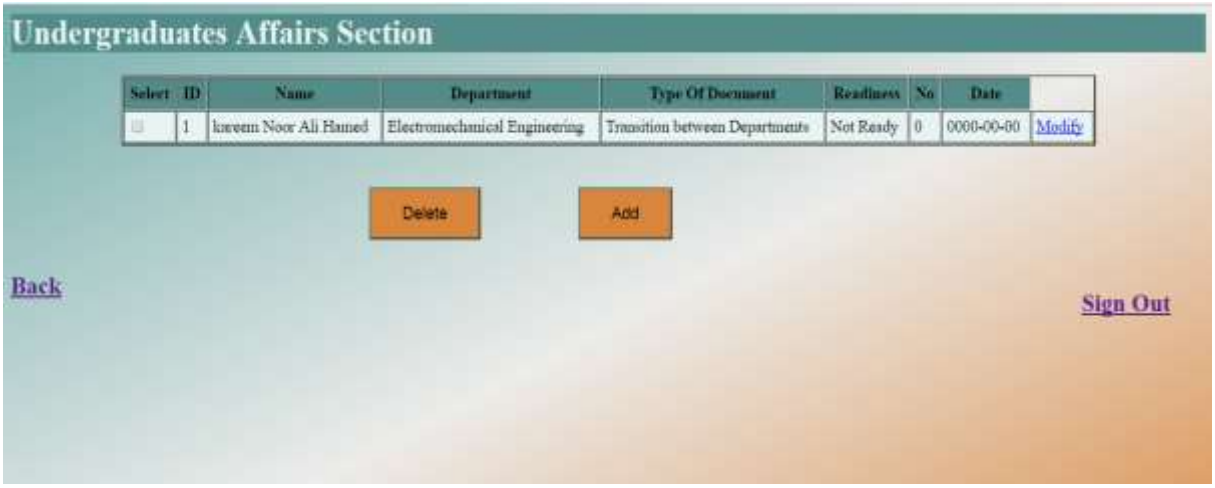


Fig. 40. “Search student page” for manager of
“Undergraduates affairs” section

The function “Delete” at page “Undergraduates affairs” like in the page “Graduates affairs” working at the same page “Undergraduates affairs” with showing confirmation message at the same page as shown in (fig. 36).

Fig. 41 shows the interface for manager of “Documents authentication” section. It contains 4 functions (“Add”, “Delete”, “Modify” and “Search”).

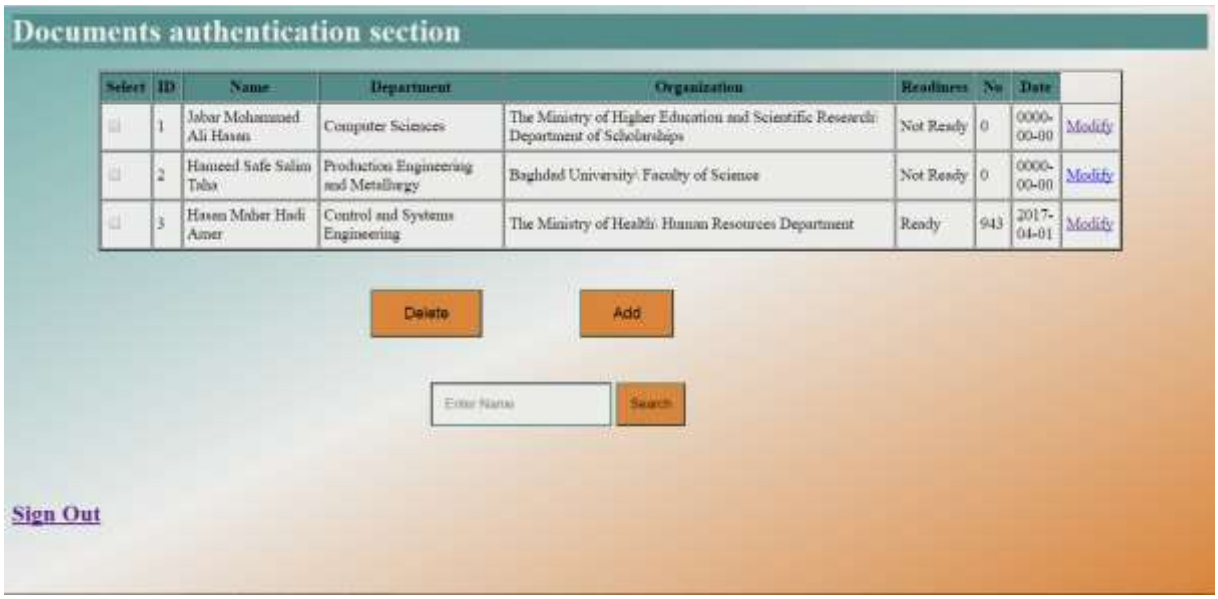


Fig. 41. Interface for Manager of “Documents authentication” section

Fig. 42 shows the "Add graduate page" shown when Manager clicks button "Add" at the page "Documents authentication".

Fig. 42. "Add graduate page" for "Documents authentication" section

Fig. 43 shows the page "Modify graduate's data" shown when manager clicks on "Modify" link for specific graduate at page "Documents authentication".

Fig. 43. Page "Modify graduate's data" for "Documents authentication" section

Fig. 44. shows page "Search graduate" shown when manager clicks on button "Search" after entering a name in input box.

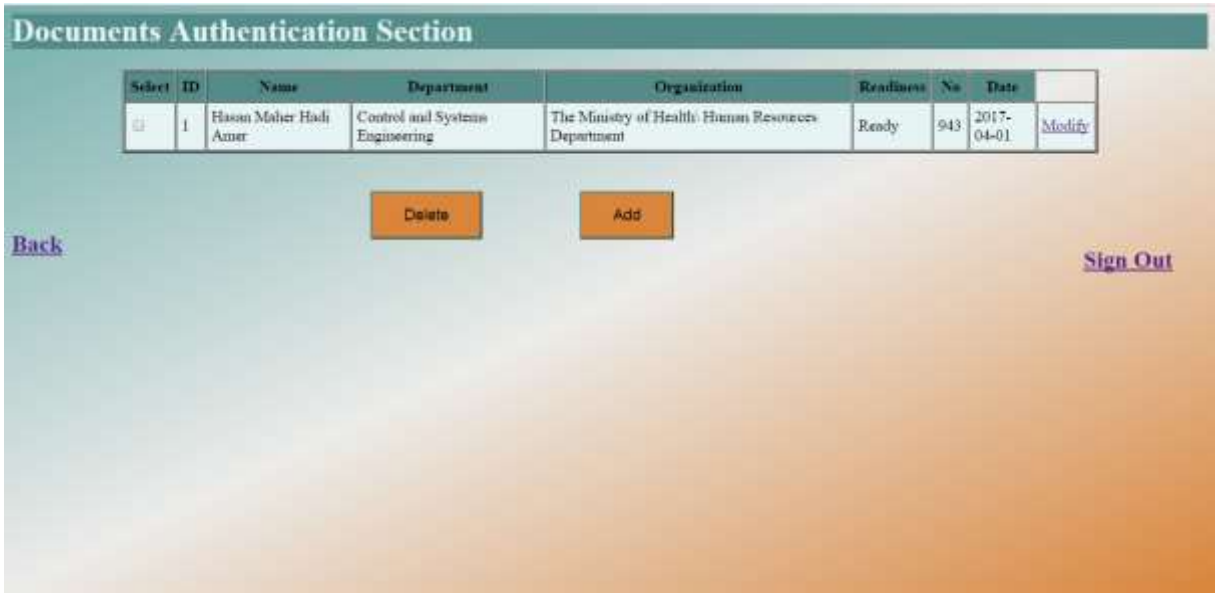


Fig. 44. “Search graduate page” for manager of
“Documents authentication” section

The function “Delete” at page “Documents authentication” like in the page “Graduates affairs” working at the same page “Documents authentication” with showing confirmation message at the same page as shown in (fig. 36).

Fig. 45 shows the interface for manager of “Ratification” section. It contains 4 functions (“Add”, “Delete”, “Modify” and “Search”).

Fig. 46 shows the “Add graduate page” shown when Manager clicks button “Add” at the page “Ratification”.

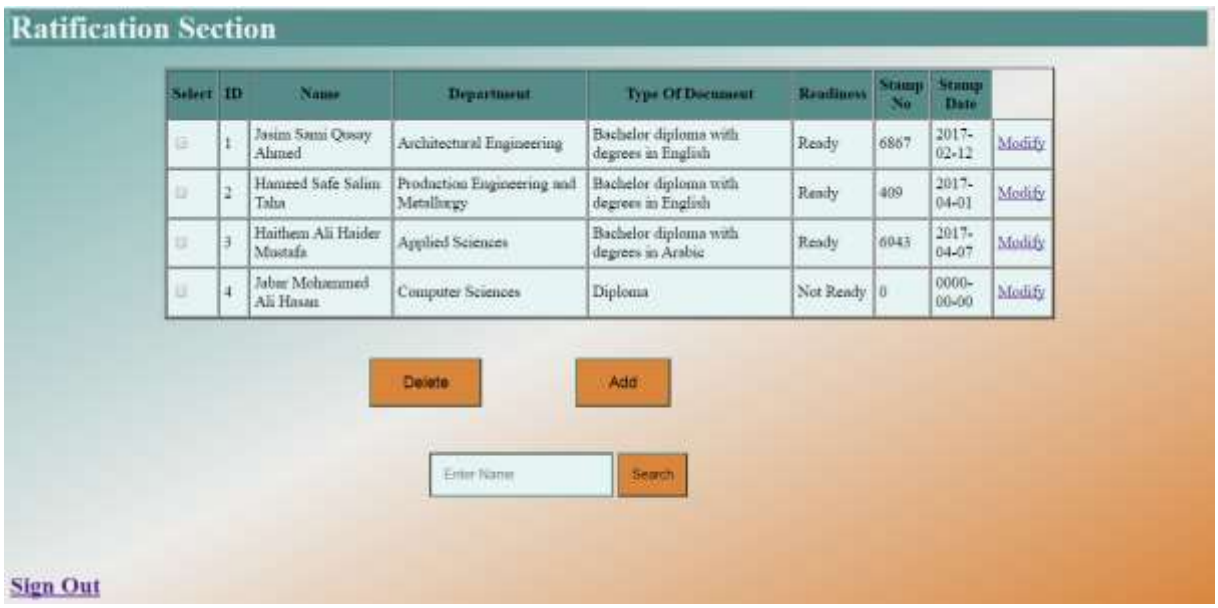


Fig. 45. Interface for Manager of “Ratification” section

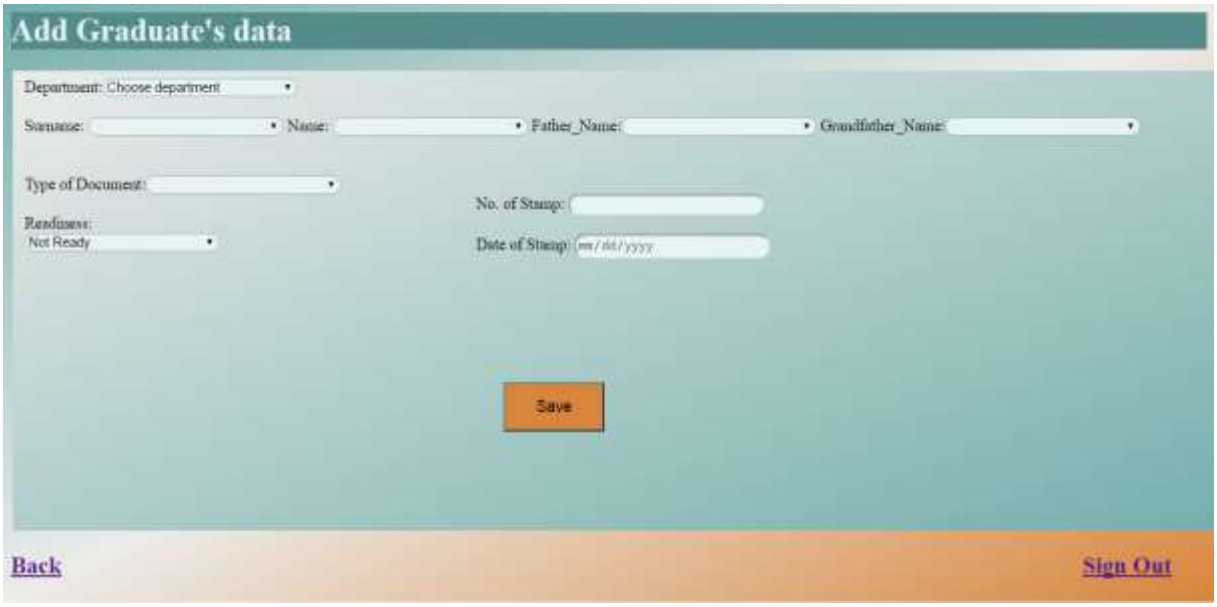


Fig. 46. “Add graduate page” for “Ratification” section

Fig. 47 shows the page “Modify graduate’s data” shown when manager clicks on “Modify” link for specific graduate at page “Ratification”.

Fig. 47. Page “Modify graduate’s data” for “Ratification” section

Fig. 48. shows page “Search graduate” shown when manager clicks on button “Search” after entering a name in input box at page “Ratification”.

The function “Delete” at page “Ratification” like in the page “Graduates affairs” working at the same page “Ratification” with showing confirmation message at the same page as shown in (fig. 36).

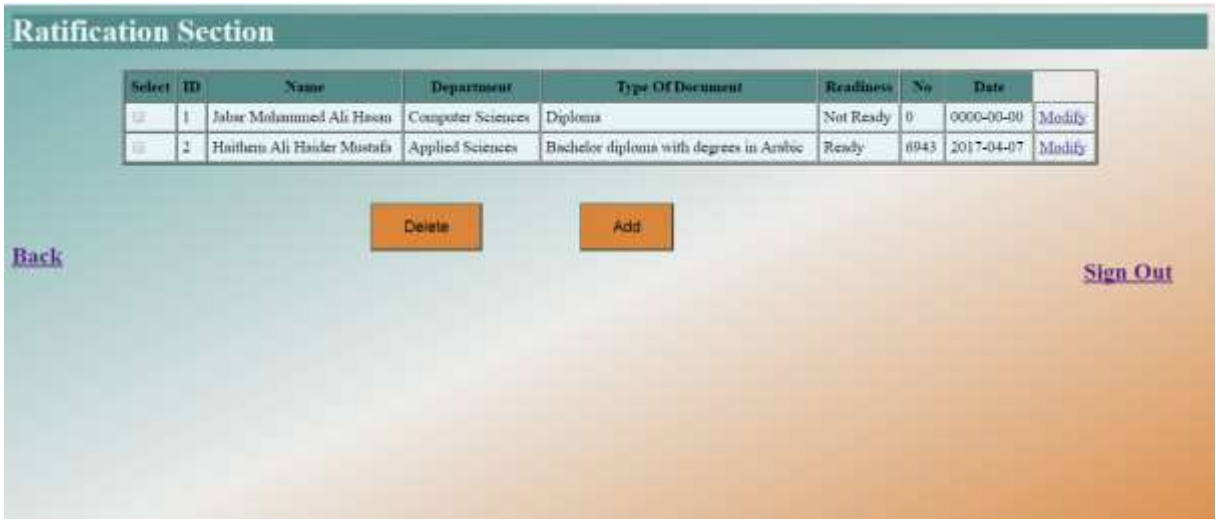


Fig. 48. “Search graduate page” for manager of
“Ratification” section

The head of the department can see the same information, which the manager can without the ability to modify, add, and delete anything from the database. As well as the student can see only his own information in section “Undergraduates affairs” represented in table, whereas the graduates can see their own information in sections “Graduates affairs”, “Ratification” and “Documents authentication” arranged in tables.

CONCLUSION

There is a problem of a lack of such web-applications, which help students and graduates to follow up their documents without reference to the department of Registration and Student Affairs in the University. This process must ensure the privacy of the student or the graduate and must not waste the employee's time to answer questions about students' documents. Therefore, we need such a system that solve this problem providing the easy interface for both employee to view, add, update, delete and search information and for students and graduates to view information about their documents.

The goal of the system is to implement the development of the management information system for the main registration department in the University of Technology (Baghdad).

The web site is divided into 5 main parts: the first one is the login form. The second part is the view of tables for four sections to the head of department. The third part is devoted to view, update, delete or search information by each manager for relative section. The forth part is for a student to view his own documents' information in a table. The fifth part is for a graduate to view his own documents' information in tables for relative sections.

During the developing of the web application, we solved the following tasks:

1) to analyze the subject area;
2) to implement the comparative analysis of analogical sites;
3) to make analysis of modern tools of web sites development and choose tools for project development;
4) to design the management information system;
5) to design database scheme;
6) to implement and test the management information system for the main registration department in the University of Technology (Baghdad).

For the following opportunities of the application development we can define the following points:

1) include the date of receipt of the document and the problems or reasons for the delay in issuing the document;

2) include the interaction between a manager and a student or a graduate;

3) change the function of search to the live search to accelerate and expand the scope of the search;

4) send the notification to the student and the graduate in case of the readiness of their documents.

REFERENCES

1. 7 Advantages of Using a CMS to Run Your Site. [Electronic resource] URL: http://moveableonline.com/blog/2013/10/29/7-advantages-using-cms-run-site/ (date of access: 20.11.2016).

2. A guide to Web application testing. [Electronic resource] URL: www.pushtotest.com/guide-to-web-applications-testing#functionaltesting/ (date of access: 12.04.2017).

3. Beaulieu A. Learning SQL. 2nd ed. – USA, Sebastopol: O'Reilly Media, Inc., 2009. – 337 p.

4. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language: Users Guide. – USA: Addison-Wesley, 1998. – 512 p.

5. DBMS-Database management system. [Electronic resource] URL: http://www.javavillage.in/dbms.php (date of access: 12.04.2017)

6. Gilmore W.J. Beginning PHP and MySQL: From Novice to Professional. 4th ed. – USA: APress, 2010. – 825 p.

7. Kundu S. Web Testing: Tool, Challenges and Methods. // International Journal of Computer Science Issues, March 2012. – Vol. 9. – Issue 2. – No. 3. – P. 481–486.

8. Marsic I. Software Engineering. – USA, New Jersey: Rutgers University, 2012. – 627 p.

9. Official site of The Hashemite University. [Electronic resource] URL: https://reg1.hu.edu.jo/ (date of access: 11.11.2016)

10. Official site of university of Babylon. [Electronic resource] URL: http://students.uobabylon.edu.iq/students.aspx (date of access: 11.11.2016)

11. Official site of University of Technology. [Electronic resource] URL: http://www.uotechnology.edu.iq/statistics-dep/index.htm (date of access: 20.11.2016).

12. Official site of university of Technology-Iraq. [Electronic resource] URL: http://www.uotechnology.edu.iq (date of access: 11.11.2016).

13. Official site of Zayed University. [Electronic resource] URL: https://eservices.zu.ac.ae/main/Services/Servicecard#OfficeofUniversityRegistrar_Registration (date of access: 11.11.2016)

14. Padmanabhan B. Principles of Software Engineering: Unified Modeling Language (UML) Overview. // EECS810, 2/21/2012. – P. 10.

15. Popular Database Management Systems Overview. [Electronic resource] URL: http://support.dbconvert.com/hc/en-us/articles/203189021-Popular-Database-Management-Systems-Overview (date of access: 20.11.2016)

16. Seyed M.M., Hugh E. Williams. Learning MySQL. – USA, Sebastopol: O'Reilly Media, Inc., 2007. – 622 p.

17. Software Testing Class. [Electronic resource] URL: http://www.softwaretestingclass.com/web-application-testing/ (date of access: 12.04.2017).

18. Usage of content management systems for websites. [Electronic resource] URL: https://w3techs.com/technologies/overview/content_management/all (date of access: 20.11.2016).

19. What is a Content Management System? [Electronic resource] URL: www.thebalance.com/an-overview-of-types-of-content-management-systems-3515920 (date of access: 20.11.2016).

20. Why you should use WordPress. [Electronic resource] URL: www.wpbeginner.com/ (date of access: 20.11.2016).