

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
Federal State Autonomous Educational Institution of Higher Education
“South Ural State University (national research university)”
School of Electrical Engineering and Computer Science
Department of Computer Science

THESIS IS CHECKED

Reviewer,
Cand. of Sci. Vice-director of IT de-
partment PAO "ChKPZ"

_____ D.S. Bogatenkov

“ ____ ” _____ 2019

ACCEPTED FOR THE DEFENSE

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

_____ L.B. Sokolinsky

“ ____ ” _____ 2019

**DEVELOPMENT OF THE INFORMATION WEBSITE
FOR UNIVERSITY OF MAZAYA**

GRADUATE QUALIFICATION WORK
SUSU-02.04.02.2019.308-640.GQW

Supervisor,
Cand. Sci., Assoc. Prof.
_____ S.A. Ivanov

Author,
student of the group CE-229
_____ I.F.S. Alasadi

Normative control
_____ O.N. Ivanova
“ ____ ” _____ 2019

MINISTRY OF SCIENCE AND HIGHER EDUCATION 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 Computer Science

APPROVED

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

_____ L.B. Sokolinsky

“ ____ ” _____ 2019

TASK

of the master graduate qualification work

for the student of the group CE-229

Alasadi Imad Fawzi Saddam

in master direction 02.04.02

“Fundamental Informatics and Information Technologies”

(master program “Database Technologies”)

1. The topic (approved by the order of the rector from 25.04.2019 No. 899)

Development of the Information Website for University of Mazaya

2. The deadline for the completion of the work: 05.06.2019.

3. The source data for the work

3.1. Tutorials point. [Electronic Resource] URL:

https://www.tutorialspoint.com/php/php_introduction.htm (the date of access: 01.01.2019).

3.2. Sodiya A.S. User interface design and ergonomics. – Nigeria: National Open University of Nigeria, 2009. – 200 p.

3.3. Official site of MySQL Server. [Electronic Resource] URL:

<http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html> (the date of access: 25.10.2018).

4. The list of the development issues

4.1. To study the problem statement and make the comparative analysis of the PHP frameworks for website development.

4.2. To develop the structure of the required database for the information website.

- 4.3. To design the web-application.
- 4.4. To implement the website.
- 4.5. To test the system.

5. Issuance date of the task: 08.02.2019.

Supervisor

Cand. Sci., Assoc. Prof.

S.A. Ivanov

The task is taken to perform

I.F.S. Alasadi

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	5
INTRODUCTION.....	6
1. THE ANALYSIS OF THE SUBJECT AREA	10
1.1. The problem statement.....	10
1.2. Comparative analysis of the best PHP frameworks for web_development..	11
1.3. Development technologies.....	14
1.3.1. Laravel.....	14
1.3.2. MySQL.....	17
1.3.3. Text editor	18
1.3.4. XAMPP SERVER.....	18
2. DEVELOPMENT OF A SITE.....	19
2.1. Functional requirements.....	19
2.2. Use case diagram.....	19
2.3. Activity diagram.....	21
2.4. Development of the database	22
2.5. Development of the interface	28
3. IMPLEMENTATION OF THE WEBSITE APPLICATION	31
3.1. Architecture of the system	31
3.2. Several fragments of PHP-code for implementing the basic_functionality .	32
4. TESTING OF THE WEB APPLICATION	37
4.1. Full implementation of the website.....	37
4.2. The used methods of testing.....	50
CONCLUSION	55
REFERENCE LIST	57

ACKNOWLEDGEMENTS

First of all thanks God to help me for do this project.

I want to thank my deepest thanks and gratitude to PhD, associate professor, S.A. Ivanov, my supervisor for his guidance, support, motivations and encouragement throughout the project.

Finally many thanks to my friends, who acknowledged me about this opportunity, and who helped me to overcome life difficulties.

Last but not least many thanks to my family who helped and supported me in my journey.

INTRODUCTION

Topicality

The World Wide Web (WWW) has become an integral part of our lives and has impacted just about everything we do. Reading the news, checking the weather, purchasing gifts, consulting movie show times at the closest theater, applying for a job, looking for new recipes, paying bills, learning new skills, and much, much more can be done entirely over the Web, with great convenience and substantial savings in time, fuel, and paper.

We all use websites and Web-based apps and services on a regular basis, but we may not know what exactly the World Wide Web is and what makes it work the way it does World Wide Web? According to the Merriam-Webster dictionary, the World Wide Web (or simply Web) is “a part of the Internet accessed through a graphical user interface and containing documents often connected by hyperlinks” [6].

The internet has grown as popular as it is easy to use and you can find virtually every piece of information on the internet just a click away.

People of every age from children to the old people understand how internet works which has found to be the main reason of its so grown popularity. It connects us to people all over the world and provides endless information and entertainment [4].

Student information systems store and track all student information, including grades, attendance records, and more. The software functions as a digital dropbox for school-related information. SIS software has become a vital tool for educational institutions as well as parents and students, who use it to gain access to student information, make payments, and communicate with school functionaries.

Student information systems products are used by teachers, students, and parents to access all relevant information pertaining to a student’s schooling. SIS software is leveraged for a few different functions; the main two being as a channel of communication and as a place to store student information.

There are a variety of benefits associated with using student information systems for parents and students as well as educational institutions and their staff.

Solutions in this category provide channels for communication between school officials and students, act as a singular source of student-related information, streamline billing processes, and more [3].

Key Benefits of Student information systems Software:

- 1) improve management of prospective and enrolled student data;
- 2) increase communication between divisions;
- 3) maintain data of stakeholders when transferring records between departments;
- 4) provide a unified resource location for relevant stakeholders, including alumni, faculty, support staff, and donors;
- 5) standardize data formats among divisions;
- 6) ease the transfer of data to external institutions;
- 7) reduce the time spent on maintaining and organizing student records.

This project of “Information Website” is intended for use by university students, Students can get all the necessary information such as: how to apply for university, university departments, schedule, and teaching staff, and study programs.

Creating this site requires sufficient knowledge of hypertext markup language (HTML), and cascading style sheets (CSS), PHP and JavaScript because the website will create by using laravel, for the database requires knowledge of the principles of the MySQL database.

Goal and objectives of the research

The goal of the project is to develop an information website for the University of Mazaya which allows users to get all necessary information.

In order to attain this goal, we must solve the following objectives:

- 1) to study the problem statement and make the Comparative analysis of the PHP frameworks for website development;

2) to develop the structure of the required database for the information website;

3) to design the web-application;

4) to implement the website;

5) to test the system.

The practical significance

In general, the websites has improved the education by using advanced techniques and tools for research and development. The websites has improved the speed of learning process through the internet.

A university website is the virtual place where the members of the university community meet.

Today, almost every university has its own website where the university administrators post important information about the university and its activities.

The practical benefit of this scientific research to develop the information website for the University of Mazaya to help the students to get all the necessary information Such as; how to apply for university, university departments, schedule, and teaching staff, and study programs.

This project can be useful because it contains good features to help the student:

1) this project is useful because it contains good properties to help the users to get all necessary information;

2) allow every student to enter to the website;

3) easily for the user to contact with teachers;

4) saving the time and efforts while working at site;

5) unlimited number of users.

Structure of the thesis

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

In the first chapter, the problem statement is given as well as the overview and comparative analysis of the best PHP frameworks for web development.

Additionally, we describe the chosen development technologies.

In chapter two, there is a description of functional requirements, use case diagram, database scheme and development of the interfaces.

In chapter three, we show architecture of the system and several fragments of PHP-code for implementing the basic functionality of the system.

Chapter four is devoted to the testing of the application and includes full implementation of the website with main interfaces and the used methods of testing.

The thesis has 58 pages; the list of references contains 22 resources.

1. THE ANALYSIS OF THE SUBJECT AREA

1.1. The problem statement

Student information system literally means the general information systems for maintaining and providing student information. It exists in all the schools, colleges, universities and any other education institutions. However, those information systems vary.

Some of them are paper based; heavily manual work is involved in managing and maintaining information such as student personal Records files.

However, recently, most schools, even down to the very smallest, utilize computers in some way or another.

The uses to which the computers are put vary enormously, ranging from word processing and spreadsheet through to worldwide on-line access, complicated user access permission system and vast functionalities.

In these days we can say that many of today's business simply could not function effectively without automated information processing systems of some form or another, so do university schools.

All of them have common tasks such as collecting, storing and processing information regarding their students, staff and the work done within the department.

With the increase of information, it is unwise to adopting the traditional paper based system which is slow to access and therefore, inefficient.

Recent years, many systems were developed either by the universities or the software companies in order to partially automate many of the processes carried out by the department.

Those developments dramatically reduce the time take in searching information and should enable the school to maintain precise and up-to-date information.

For example, previously, students would have to enquire for much of his information from the school student office, which required more time and effort, particularly from members of staff.

The new system is more efficient and also often presents more accurate information. However, those systems vary even within the same university.

I will create a university information system, its information system for education establishments to manage data for universities.

University information systems provide all necessary information for students; documenting grading, transcripts, results of student tests and other assessment scores; building student schedules; tracking student attendance; and managing many other student-related data needs in a university.

This system will help the student to get all the necessary information Such as; how to apply for university, university departments, schedules, and teaching staff, and study programs.

1.2. Comparative analysis of the best PHP frameworks for web development

During my work on this project I encountered the problem that developers always face, which framework of the PHP frameworks I chose, which offer to us the ability to build more complex, secure, and well-rounded web applications faster than ever before, so I decided to do the comparative analysis of the best PHP frameworks [8].

As known, in order to compare tools you need to compare the way most common development capabilities are implemented in each of them.

Table 1 show the comparison of PHP frameworks.

Table 1. Comparison of PHP frameworks.

No.	Function	Laravel	Yii
1.	Template Engine	One of the most amazing features of Laravel is its Blade templating engine which offers various templates that make the tasks easy and quick for the web developers [15].	By default, Yii uses PHP as its template language you can configure Yii to support other rendering engines, such as Twig or Smarty available as extensions [10].

No.	Function	Laravel	Yii
2.	Database	Microsoft BI MongoDB MySQL PostgreSQL Redis SQLite [10].	Microsoft BI MongoDB MySQL Oracle PostgreSQL SQLite [10].
3.	Security	When it comes to security, Laravel has the highly efficient security features that are applied to the web application. In fact, Laravel has its own authentication system which restricts the unauthorized access to the web application system [17].	Yii framework has great security features which the web developers always loved. But, after Laravel's inception, its authentication features beat every other PHP frameworks [17].
4.	Documentation and learning curve	One of the best documentation is available for Laravel. Web developers use Laracasts which is just like a Netflix app which can be used to learn Laravel. [17].	Yii does not come with stronger documentation when compared to Laravel as they even come up with learning as if you are watching series on Netflix [17].
5.	Resources and Support	Laravel has a large community with good support and it's growing every day, though its new framework as compared to other from 2012 [16].	Yii framework comes on 2008 and it also has a large community with good support but not like Laravel framework [16].
6.	Rapid Development	Among all the other PHP frameworks, Laravel has all the capabilities, features, integrations, and configurations that can contribute to rapid app development [17].	Both of PHP frameworks were developed in order to achieve rapid application development. But, there are templates and even libraries that allow Laravel to win over Yii framework for the rapid app development [17].

No.	Function	Laravel	Yii
7.	Routing	In Laravel, we have default route files and Need to declare route for each and every action for all controller [16].	With Yii do not need to declare route to each and every action [16].
8.	Object-Relational Mapping (ORM)	Yes	Yes
9.	Free to use	Yes	Yes
10.	MVC	Yes	Yes

We can say that Laravel is perfect for the large-sized and complex web applications as it is equipped with numerous features.

Yii is a convenient and comprehensive tool for smaller projects.

With regard to Templating engines there is no clear winner here. Both two frameworks use templating engines for better front-end coding and maintenance.

A small advantage to Yii is that the framework does not have a pre-defined templating system.

After examining different PHP frameworks and comparing their abilities at handling an MVC architecture pattern I came up with the ideal choice for a PHP MVC framework, which is Laravel.

At first, learning a new framework might seem an overwhelming task, but it was not the case with Laravel, thanks to its clear and concise documentation, and its developers that make a lively active community.

The biggest benefit of Laravel is its laconic core system with highly organized components, which allows Laravel enthusiasts to write elegant code.

Laravel has inherited the best PHP practices and this fact attracts more and more programmers all around the globe.

The learning curve for mastering Laravel is very steep and not every newbie can overcome it. However, they have all the resources needed for learning and even a Laravel course called Laracast.

Early on in the development process with Laravel, one would feel at ease with its simplicity and ease of use. My own experience with another big framework, that is, .NET framework is that one ends up investing an important amount of time struggling with incomprehensible XAML configuration settings, complex syntax, unfinished documentation, and a feeling in the end that the framework's purpose of saving time and effort was not truly achieved. It is the other way around with Laravel, which is actually one of its major strengths.

My own experience with Laravel is that it made my development process a more enjoyable experience.

1.3. Development technologies

1.3.1. Laravel

Laravel is an open-source PHP framework, which is robust and easy to understand. It follows a model-view-controller design pattern. Laravel reuses the existing components of different frameworks which help in creating a web application.

Laravel offers a rich set of functionalities which incorporates the basic features of PHP frameworks like CodeIgniter, Yii and other programming languages like Ruby on Rails.

If you are familiar with Core PHP and Advanced PHP, Laravel will make your task easier. It saves a lot of time if you are planning to develop a website from scratch. Moreover, a website built in Laravel is secure and prevents several web attacks [20].

It provides a lot of new features so here are the best reasons why Laravel is the fastest growing framework nowadays.

1. Method Injection

The injection means one to one mapping. Method injection allows you to inject any class object or instance into a method as an argument. In the background, Laravel create an object when the method called by Container. A class can be anything it can be Request or it can be Validator or custom Class [19].

2. Authentication scaffolding

Scaffolding means ready boilerplate that provides the ready code you just have to adjust as per your own use. Authentication scaffolding provides ready login, register and forgets password feature default. To use it you just have to run one simple command on your Laravel application's root folder via command Panel [13].

3. Routing Request

Into a core PHP if you run any file you can see that filename with .php extension into browsers' URL box. It's not user-friendly and SEO related URL. To do that you can revamp .htaccess rule.

Laravel maintain this thing itself you do not need to write an external rule in the .htaccess file to make your URL relative.

Laravel provides routes.php file in version laravel5 and laravel5.2. In a later version of 5.2 contains the web.php file.

This file is a single source of truth that contains all URL request. You can write any URL then you have to map it with specific controller's method.

You must have to register your every URL into a routing file. Routing Request is the way that you can manage user-friendly and relative URL in your application.

The main benefit of this is you can separate request into groups by giving it prefix that can be admin or front or other. There are some methods for that we will see it later [19].

4. Eloquent ORM System

ORM stands for Object-relational mapping. ORM is programming technique used to convert compatible type in Object oriented programming.

Laravel has its own ORM system named with Eloquent ORM.

Eloquent provide very easiest to way to handle database operations such as create, update and delete. You do not have to worry about SQL Injection it prevents a default by ORM. Eloquent provide relationship concept between two or multiple tables.

There are different types of relations like One to one, one to many and many to many. If you are using eloquent you do not need to use core queries like JOIN and SELECT and any other.

Eloquent is a class defined by Laravel and you have to extend it in your model and just use it simple. The eloquent system uses subqueries.

Eloquent ORM system provider beautiful way to access your database via Model class [19].

5. Blade template engine

One of the best and beautiful syntax provided by Laravel to handle view files. The Blade is the template engine based on PHP. Template engine provides a way to write short code syntax into view or presentation file.

The template engine is a way to write or separate presentation logic. Template engine also added security for end-user customization. Ease of use for non-programmers, such as graphic artists or web designers [19].

6. Event Object

An Event means one object that broadcasting something and there are single or multiple listeners to observe that. The Same way into Laravel you can define events suppose login event after user login successful or it can be anything.

An event is a class that you can find into app/Events directory. If you want to listen to it then you have to define Listen Class into app/Listener directory.

Do not worry if your app does not have events and listeners directory now it will generate runtime. There are different steps to write event such as first you have to define Event class then you have to register it.

Every event has single or multiple listeners, of course, it can be independent. You can dispatch an event from anywhere with custom data that you have to pass to listeners [19].

7. Multiple File System

To prevent access any image or CSS or any file from URL or to prevent hacking tricks now in days developer use cloud file store to store file and make it safer. A filesystem is a way that helps you to manage file storing and that wonderful feature of Laravel.

Laravel provider Drivers to handle a different store, of course, you can store your file into multiple servers via the multiple drivers. The driver can be Amazon S3, and Rackspace Cloud Storage or local [19].

8. Artisan CLI

Laravel provides command line interface that is Artisan CLI. Using Commands you can do so many things in Laravel such as to clear cache, Generate files and to run Laravel also you need Artisan CLI [19].

9. Config files system to manage configurations

All your application related configurations file stored into config directory. Laravel provides different config files to manage configurations such database config or mail system config also you can create define config file [19].

1.3.2. MySQL

MySQL is the world's most popular open-source database. In fact, today MySQL is a viable competitor to the pricey goliaths such as Oracle and Microsoft's SQL Server (and, ironically, MySQL is owned by Oracle).

Like PHP, MySQL offers excellent performance, portability, and reliability, with a moderate learning curve and little to no cost, MySQL is an open-source application, like PHP, meaning that it is free to use or even modify [7].

MySQL is a database management system (DBMS) for relational databases. A database, in the simplest terms, is a collection of data, be it text, numbers, or binary files, stored and kept organized by the DBMS [7].

1.3.3. Text editor

For this project the text editor of choice to build our web application will be visual studio code. It is a web developer's editor that can do few useful tasks from the editor window itself.

Therefore, the developer does not have to constantly switch between windows and run tasks from other applications.

1.3.4. XAMPP SERVER

XAMPP is open source free software developed by Apache friends. XAMPP software package contains Apache distributions for Apache server, MariaDB, PHP, and Perl. And it is basically a local host or a local server.

This local server works on your own desktop or laptop computer. You can just install this software on your laptop or desktop and test the clients or your website before uploading it to the remote web server or computer.

This XAMPP server software gives you the suitable environment for testing MYSQL, PHP, Apache and Perl projects on the local computer [22].

2. DEVELOPMENT OF A SITE

2.1. Functional requirements

The features that are available for the administrator.

1. The admin must be able to create, read, update and delete pages.
2. The admin must be able to create, read, update and delete sliders.
3. The admin must be able to create, read, update and delete widgets.
4. The admin must be able to create, read, update and delete categories.
5. The admin must be able to create, read, update and delete articles.
6. The admin must be able to create, read, update and delete comments.
7. The admin must be able to create, read, update and delete users.

The features that are available for the teacher.

1. The teacher must be able to create, read, update and delete his articles.
2. The teacher must be able to create, read, update and delete comment on

articles.

The student must be able to create, read, update and delete comment.

The “Search” function must be available for all users.

2.2. Use case diagram

Use case diagram is a behavioral UML diagram type and frequently used to analyze various systems. Unified Modeling Language (UML) helps the programmers to model computer applications. This use case diagram shows nine use cases. There are four types of user: admin, teacher, student and unregistered user represented as actors.

These actors are connected with these use cases by relationships to show the ability for each user in the website.

Use case diagrams are important for imagining the practical necessities of a framework that will convert into plan decisions and improvement needs.

They give a decent abnormal state examination from outside the framework. Use case outlines determine how the framework connects with entertainers without agonizing over the subtleties of how that work is actualized.

Fig. 1 shows the Use case diagram for the information website functions.

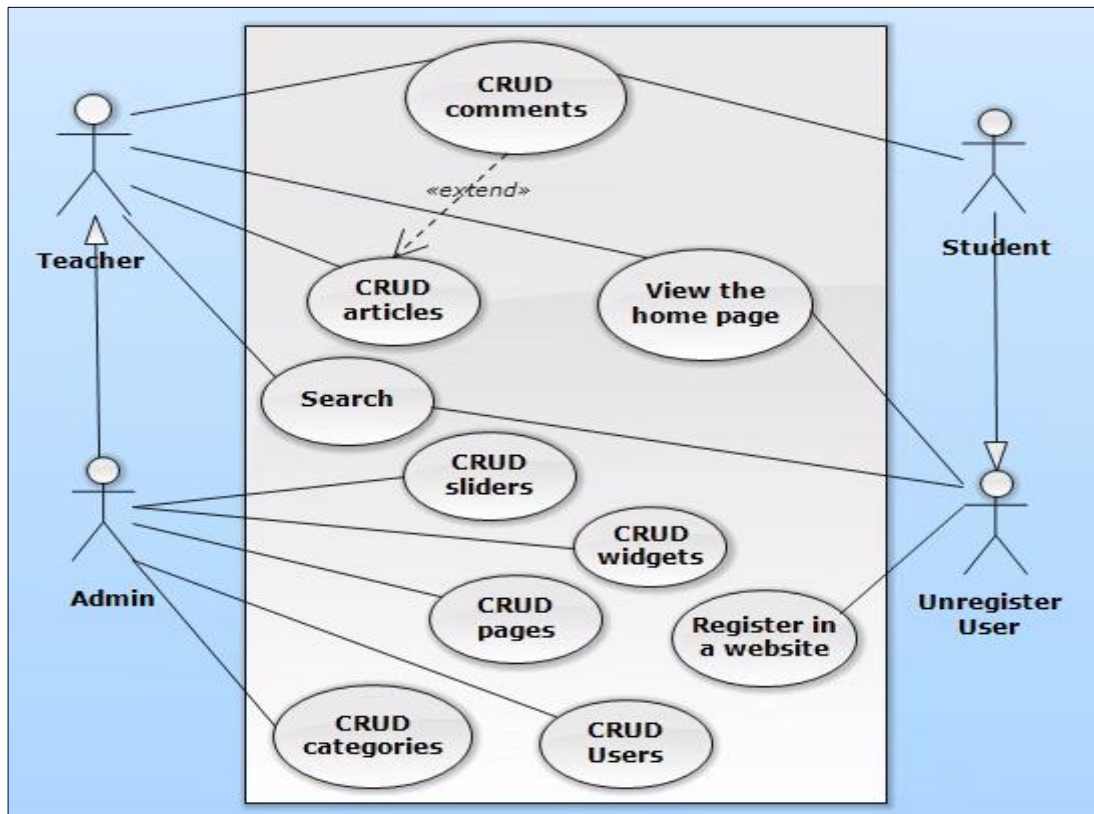


Fig. 1. Use case diagram

Use case “CRUD Users” is available for the Admin only.

Use case “CRUD pages” is available for the Admin only.

Use case “CRUD categories” is available for the Admin only.

Use case “CRUD widgets” is available for the Admin only.

Use case “CRUD sliders” is available for the Admin only.

Use case “Search” is available for the admin, teacher, student and unregis-ter user as they are connecting with generalize relationship.

Use case “CRUD comments” is available for the admin, teacher and stu- dent as they are connecting with generalize relationship.

Use case “CRUD articles” is available for the Admin and teacher as they are connecting with generalize relationship.

Use case “Register in a website” is available for the unregister user.

Use case “View the home page” is available for the Admin, teacher, stu- dent and unregis-ter user as they are connecting with generalize relationship.

2.3. Activity diagram

The Unified Modeling Language includes several subsets of diagrams, including structure diagrams, interaction diagrams, and behavior diagrams [5].

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to represent business workflow [21].

This diagram shows how the user can write comment on articles, at the first the user will open the article then write text of comment after that send comment at last show comment under article.

Fig. 2 shows the Activity diagram for writing comment.

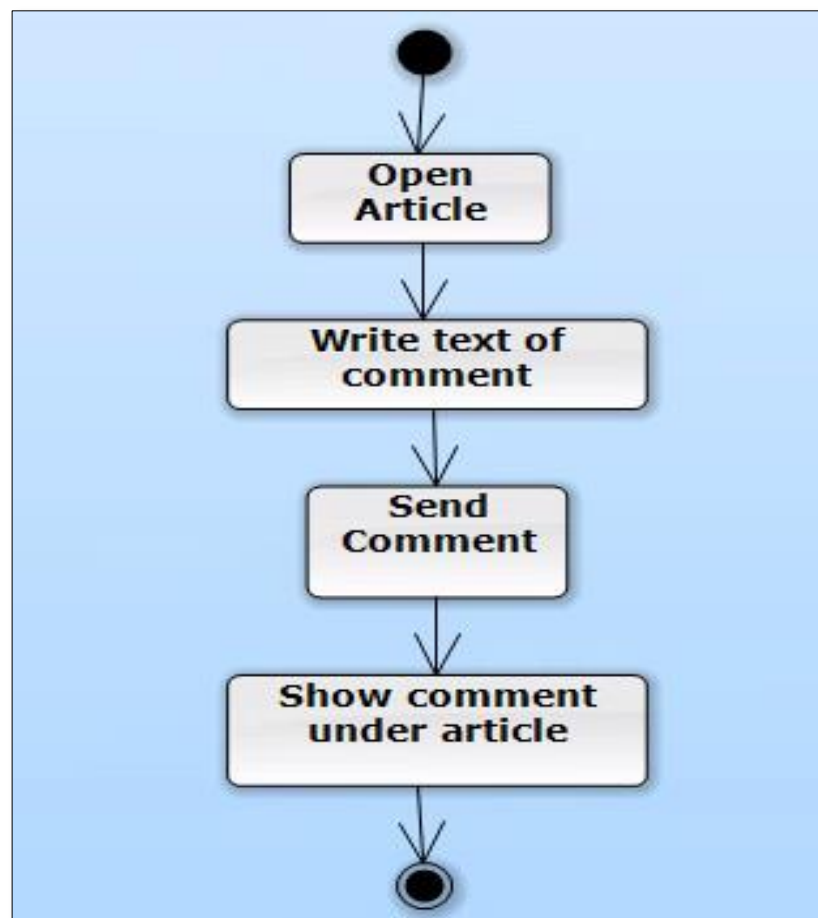


Fig. 2. Activity diagram

2.4. Development of the database

I have chosen MySQL as DBMS for the implementation of my project.

A database management system (DBMS) is system software for creating and managing databases. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data.

The DBMS manages three important things: the data, the database engine that allows data to be accessed, locked and modified and the database schema, which defines the database's logical structure. These three foundational elements help provide concurrency, security, data integrity and uniform administration procedures.

Typical database administration tasks supported by the DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated roll-backs, restarts and recovery as well as the logging and auditing of activity [11].

A university database management system is a software application for education establishments to manage university data. These systems vary in size, scope and capability, from packages that are implemented in reliably small organizations to cover student records alone, to enterprise-wide solutions that aim to cover most aspects of running large multi-campus organizations with significant local responsibility. University Database Management System deals with all kinds of student details, academics related reports, college details, course details, curriculum, batch details and other resource related details too. It tracks all the details of a student from the day one to the end of his course can be used for all reporting purposes, tracking of attendance, progress in the project or any other assignment details, final exam result exam [15].

Our design can facilitate us to explore all the activities happening in the college, even we can get to know which faculty is assigned to which course, the current status of student and upcoming requirement of the student.

MySQL, launched in 1995, has become the most popular open source DBMS.

Another root cause of MySQL's popularity has been the ongoing success of phpMyAdmin [10]. A well-established MySQL web-based interface.

Therefore, many websites in the world use MySQL as their back-end data repository [9].

We decided to specify the following tables of the database:

- 1) the “User” table;
- 2) the “Page” table;
- 3) the "Category” table;
- 4) the “Article” table;
- 5) the “Slider” table;
- 6) the “Widget” table;
- 7) the “Comment” table.

The scheme of the database consists of 7 tables as in the fig. 3.

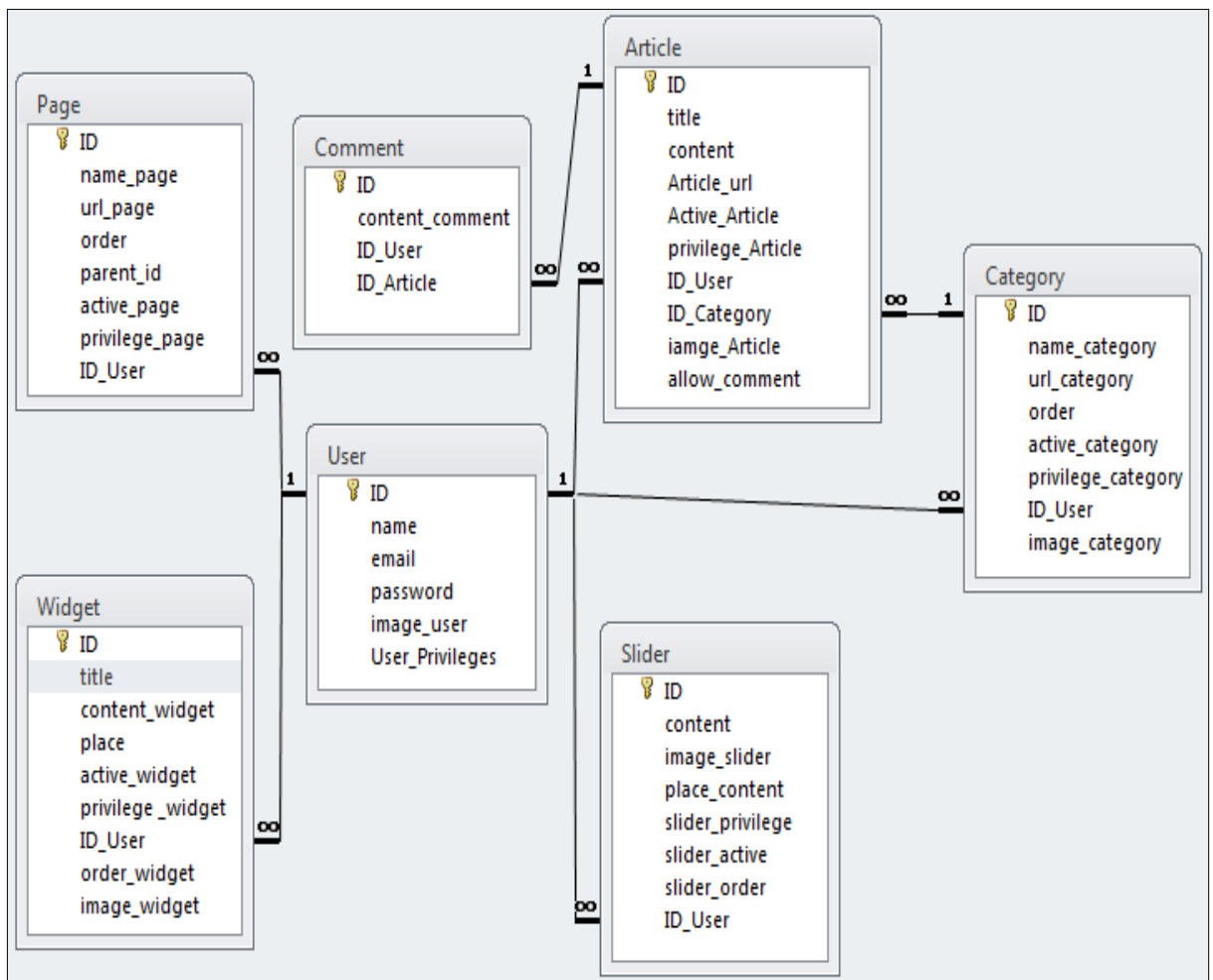


Fig. 3. The scheme of the database

The table “User” consists of 6 fields and contains the information of the users that register in the website as in the fig. 4.

It consists of the following fields:

- 1) ID: the primary key for user table and the datatype (int);
- 2) Name: the name of user and the datatype (varchar);
- 3) Email: the email of user and the datatype (varchar);
- 4) Password : the password of user and the datatype (varchar);
- 5) Image_user : the image of user and the datatype (text);
- 6) User_Privileges: privilege of user (types of user)and the datatype (int).


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/> 1	ID 	int(10)			No	None		AUTO_INCREMENT
<input type="checkbox"/> 2	name	varchar(255)	utf8_unicode_ci		No	None		
<input type="checkbox"/> 3	email	varchar(255)	utf8_unicode_ci		No	None		
<input type="checkbox"/> 4	password	varchar(255)	utf8_unicode_ci		No	None		
<input type="checkbox"/> 5	image_user	text	utf8_unicode_ci		Yes	NULL		
<input type="checkbox"/> 6	User_Privileges	int(11)			No	0		

Fig. 4 Structure of the table “User”

The table “Page” consists of 8 fields and contains the information of the Pages in the website as in the fig. 5.

It consists of the following fields:

- 1) ID: the primary key for the page table and the datatype (int);
- 2) Name_page: the name of the page and the datatype (varchar);
- 3) url_page: the link in which will be the page and the datatype (text);
- 4) order :the order of the page and the datatype(int);
- 5) parent_id: specify this page will be main of part from other page;
- 6) active_page: active or inactive pages and the datatype (int);
- 7) privilege _page: for whose will be available;
- 8) ID_User: the foreign key for the user table and the datatype (int).

#	Name	Type	Collation	Attributes	Null	Default
1	ID 🔑	int(11)			No	None
2	name_page	varchar(255)	utf8_unicode_ci		No	None
3	url_page	text	utf8_unicode_ci		No	None
4	order	int(11)			No	None
5	parent_id	int(11)			No	None
6	active_page	int(11)			No	None
7	privilege_page	int(11)			No	None
8	ID_User 🔑	int(11)			No	None

Fig. 5. Structure of the table “Page”

The table “Article” consists of 10 fields and contains the information of the articles that will post in the website as in the fig. 6.

It consists of the following fields:

- 1) ID: the primary key for the Article table and the datatype (int);
- 2) Title: the title of the article table and the datatype (varchar);
- 3) Content: the content of the Article and the datatype (text);
- 4) Article_url: the link of the article and the datatype (text);
- 5) Active_Article: active or inactive article and the datatype (int);
- 6) privilege_Article: for whose will be available;
- 7) ID_User: the foreign key for the user table and the datatype (int);
- 8) ID_Category: the foreign key for the category table and the datatype (int);
- 9) image_Article: the photo will attach with articles;
- 10) allow_comments: active or inactive comment on articles and the datatype (int).

#	Name	Type	Collation	Attributes	Null	Default
1	ID 🔑	int(11)			No	None
2	title	varchar(255)	utf8_unicode_ci		No	None
3	content	text	utf8_unicode_ci		No	None
4	Article_url	text	utf8_unicode_ci		No	None
5	Active_Article	int(11)			No	None
6	privilege_Article	int(11)			No	None
7	ID_User 🔑	int(11)			No	None
8	ID_Category 🔑	int(11)			No	None
9	iamge_Article	text	utf8_unicode_ci		Yes	NULL
10	allow_comments	int(11)			No	None

Fig. 6. Structure of the table “Article”

The table “Category” consists of 8 fields and contains the information of the Category in the website as in the fig. 7.

It consists of the following fields:

- 1) ID: the primary key for the Category table and the datatype (int);
- 2) Name_ Category: the name of the Category table and the datatype (varchar);
- 3) url_Category: the link of the Category and the datatype (text);
- 4) order: the order of the Category and the datatype (int);
- 5) Active_Category: active or inactive Category and the datatype (int);
- 6) privilege_category: for whose will be available and the datatype (int);
- 7) ID_User: the foreign key for the user table and the datatype (int);
- 8) image_Category: the photo will attach with articles and the datatype (text).



#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1 ID 	int(11)			No	None
<input type="checkbox"/>	2 name_catgroy	varchar(255)	utf8_unicode_ci		No	None
<input type="checkbox"/>	3 url_catgroy	text	utf8_unicode_ci		No	None
<input type="checkbox"/>	4 order	int(11)			No	None
<input type="checkbox"/>	5 active_catgroy	int(11)			No	None
<input type="checkbox"/>	6 privilege_category	int(11)			No	None
<input type="checkbox"/>	7 ID_User 	int(11)			No	None
<input type="checkbox"/>	8 image_Category	text	utf8_unicode_ci		Yes	NULL

Fig. 7. Structure of the table “Category”

The table “Comment” consists of 4 fields and contains the information of the Comment in the website as in the fig. 8.

It consists of the following fields:

- 1) ID: the primary key for the Comment table and the datatype (int);
- 2) Comment_content: the content of the Comment and the datatype (text);
- 3) ID_User: the foreign key for the user table and the datatype (int);
- 4) ID_Article: the foreign key for the articles table and the datatype (int).



#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1 ID 	int(11)			No	None
<input type="checkbox"/>	2 content_comment	text	utf8_unicode_ci		No	None
<input type="checkbox"/>	3 ID_User	int(11)			No	None
<input type="checkbox"/>	4 ID_Article 	int(11)			No	None

Fig. 8. Structure of the table “Comment”

The table “Slider” consists of 8 fields and contains the information of the sliders in the website as in the fig. 9.

It consists of the following fields:

- 1) ID: the primary key for the Slider table and the datatype (int);
- 2) Content: the content of the Slider and the datatype (text);
- 3) image_Slider: the attached photo and the datatype (text);
- 4) place_content: in which place will the content and the datatype (int);
- 5) slider_privilege: for whose will the Slider be available and the datatype(int);
- 6) Slider_active: active or inactive Slider and the datatype (int);
- 7) Slider_order: the order of the Slider and the datatype (int);
- 8) ID_User: the foreign key for the user table and the datatype (int).



#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1 ID 	int(11)			No	None
<input type="checkbox"/>	2 content	text	utf8_unicode_ci		No	None
<input type="checkbox"/>	3 image_slider	text	utf8_unicode_ci		No	None
<input type="checkbox"/>	4 place_content	int(11)			No	None
<input type="checkbox"/>	5 slider_privilege	int(11)			No	None
<input type="checkbox"/>	6 slider_active	int(11)			No	None
<input type="checkbox"/>	7 slider_order	int(11)			No	None
<input type="checkbox"/>	8 ID_User 	int(11)			No	None

Fig. 9. Structure of the table “Slider”

The table “Widget” consists of 9 and contains the information of the widget in the website as in the fig. 10.

It consists of the following fields:

- 1) ID: the primary key for the Widget table and the datatype (int);
- 2) Title: the title of the Widget table and the datatype (varchar);
- 3) Content_ widget: the content of the Widget and the datatype (text);
- 4) place: in which place will the Widget and the datatype (int);
- 5) active_ widget: active or inactive Widget and the datatype(int);
- 6) privilege_ widget: for whose will Widget be available and the datatype(int);
- 7) ID_User: the foreign key for the user table and the datatype (int);
- 8) Order_ widget: the order of the Widget and the datatype (int);
- 9) image_ widget: the attached photo and the datatype (text).



#	Name	Type	Collation	Attributes	Null	Default
<input type="checkbox"/>	1 ID 	int(11)			No	None
<input type="checkbox"/>	2 title	varchar(255)	utf8_unicode_ci		No	None
<input type="checkbox"/>	3 content_widget	text	utf8_unicode_ci		No	None
<input type="checkbox"/>	4 place	int(11)			No	None
<input type="checkbox"/>	5 active_widget	int(11)			No	None
<input type="checkbox"/>	6 privilege_widget	int(11)			No	None
<input type="checkbox"/>	7 ID_User 	int(11)			No	None
<input type="checkbox"/>	8 order_widget	int(11)			No	None
<input type="checkbox"/>	9 image_wedg	text	utf8_unicode_ci		Yes	NULL

Fig. 10. Structure of the table “Widget”

2.5. Development of the interface

User Interface is the point at which a user or a user department or organization interacts with a computer system.

The goal of user interface design is to make the user's interaction as simple and efficient as possible [13].

We will implement future views of a website. Fig. 10 shows the main page for the web site of a system for information website interface.

The users who visit the website can read and see all articles, sliders and widgets in the website.

The users can login to the website by enter his username and password if he had already registered before, otherwise he can create a new user account in "Create an account" section. It will lead him to the register page.

The main page contain many sections in the top of the page there logo and the name of the website and besides the logo there pages which created by the admin.

Below the header there are sliders, and in the right side and bottom of website there are widgets and there are articles in the body which also created by admin.

In the footer there are logout and dashboard bottoms, which are shown below fig. 11.

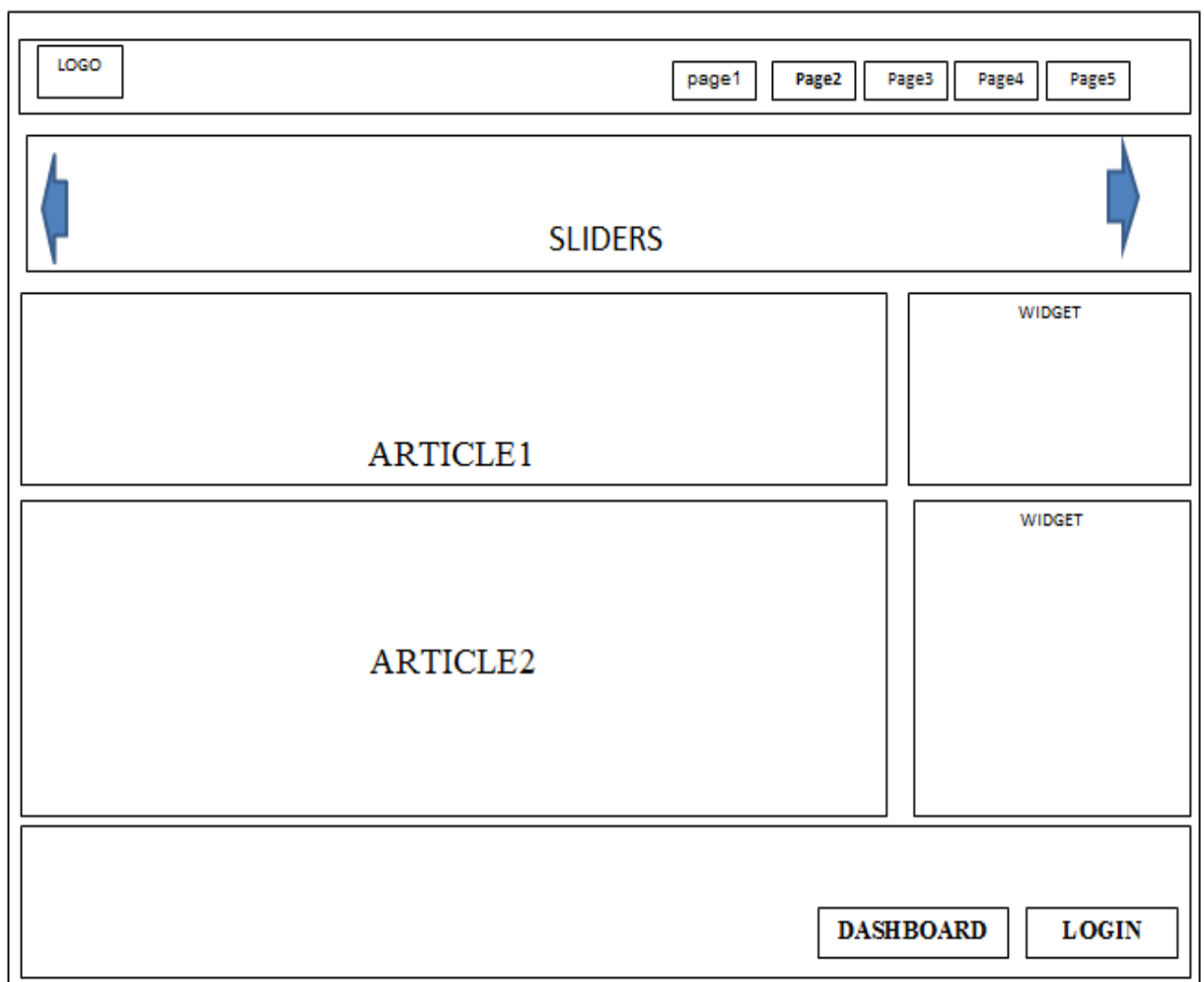


Fig. 11. The main page

Each website contains many interfaces and allows the user to move between the interfaces and to summarize the interfaces in our website this tree and it's a Schema of available interfaces for admin, teacher and student and it contains all the elements in the website and also explains the process of moving from one to another shown below fig. 12 and fig. 13.

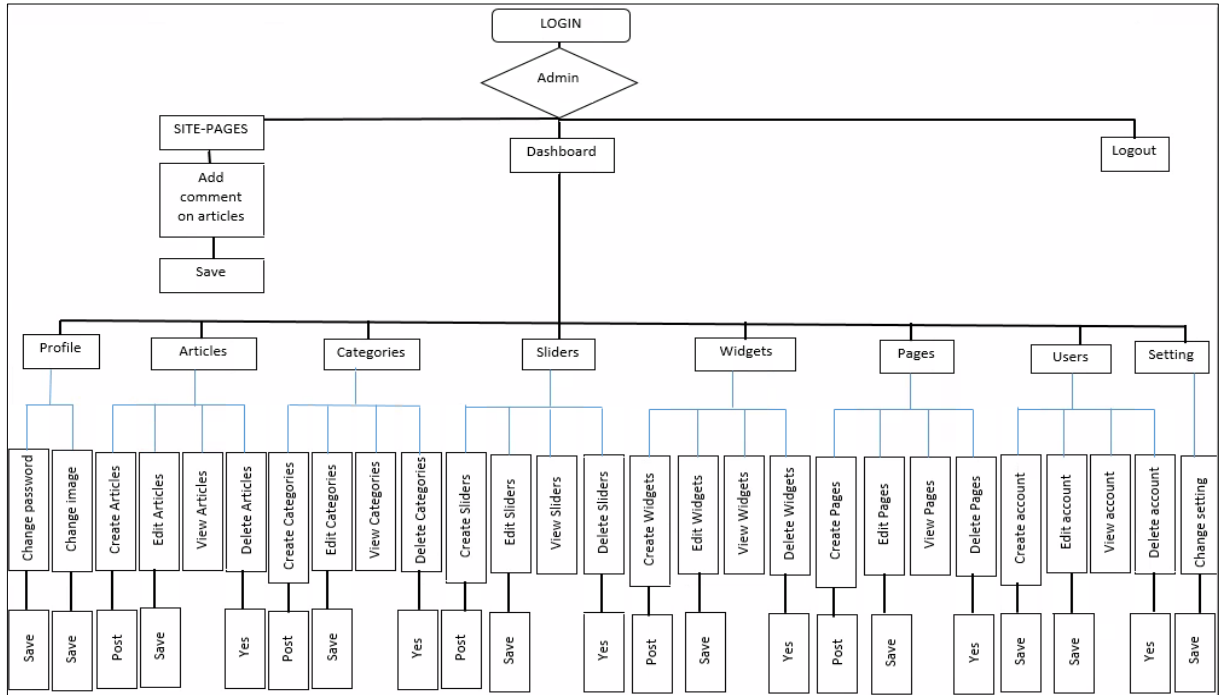


Fig. 12. Schema of available interfaces for administrator

Fig. 13 shows the schema of available interfaces for teacher and student.

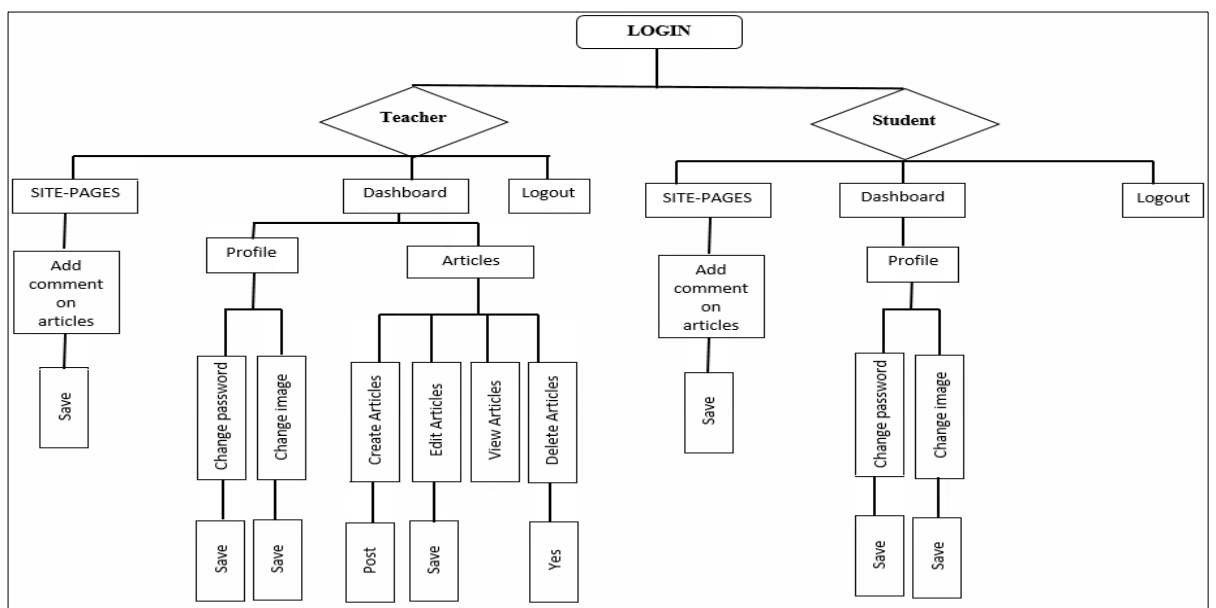


Fig. 13. Schema of available interfaces for teacher and student

3. IMPLEMENTATION OF THE WEBSITE APPLICATION

3.1. Architecture of the system

Component diagram is a special kind of diagram in UML. It does not describe the functionality of the system but it describes the components used to make those functionalities [12].

Component diagrams are used to describe the physical artifacts of a system [12]. This artifact includes files, executables, libraries, etc.

Laravel applications follow the traditional Model-View-Controller design pattern, where we use:

- 1) Controllers to handle user requests and retrieve data, by leveraging Models;
- 2) Models to interact with your database and retrieve your objects' information;
- 3) Views to render pages.

Additionally, routes are used to map URLs to designated controller actions.

Fig. 14 below shows the structural relations between components in the website.

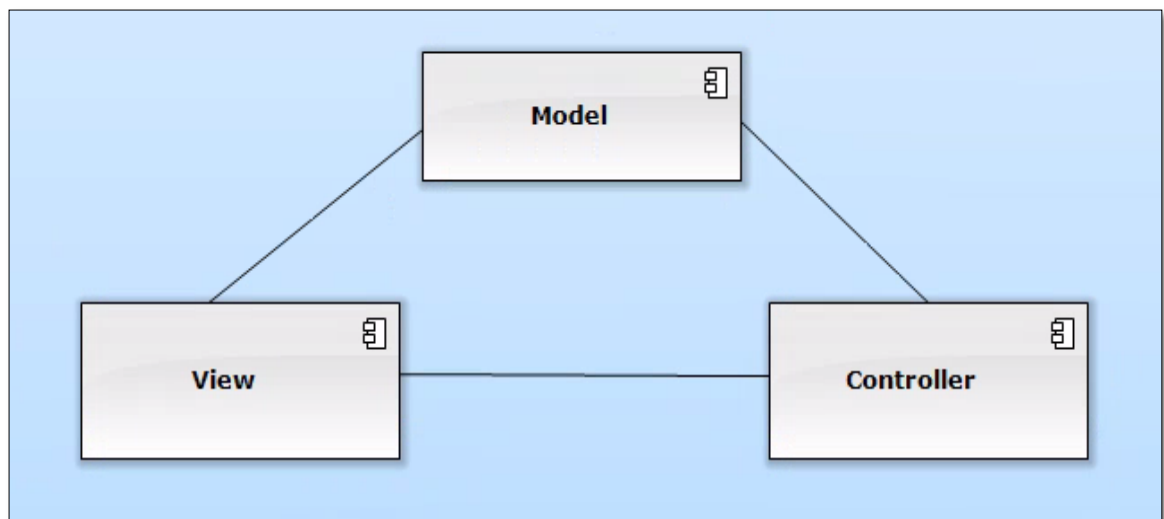


Fig. 14. Component diagram

The component “controller” contains the php-files responsible for user actions and the managing the business logic of our applications [19].

Like user login, register admin, register user, manages category, page, articles, widgets and sliders.

The component “model” mostly contains all files that used to interact with the database using Eloquent ORM [2] like table-articles, table-sliders, table-user and table-widget.

The component “view” contains the php-files for the website interface like homepage, dashboard and login.

3.2. Several fragments of PHP-code for implementing the basic functionality

In this part I will view several fragments of PHP-code for implementing that includes the main functions.

Register in the system. Fig. 15 shows the function for the user to register in the system by entering his information like his “Name”, “Email”, “and “password”.

```
class RegisterController extends Controller
{
    use RegistersUsers;
    protected $redirectTo = '/';
    public function __construct()
    {
        $this->middleware('guest');
    }
    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => 'required|string|max:255',
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|min:6|confirmed',
        ]);
    }
    protected function create(array $data)
    {
        return User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => bcrypt($data['password']),
        ]);
    }
}
```

Fig. 15. Registration code

Login form (fig. 16) shows the function for the already registered the user to login in the system by enter his user name and password.

```
class LoginController extends Controller
{
    use AuthenticatesUsers;

    /**
     * Where to redirect users after login.
     *
     * @var string
     */
    protected $redirectTo = '/';

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest')->except('logout');
    }
}
```

Fig. 16. Login code

Recovering for the password (fig. 17). Each user who has lost his password can put his email address to receive his login information as a message sent to his email.

```
use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\ResetsPasswords;

class ResetPasswordController extends Controller
{
    use ResetsPasswords;
    protected $redirectTo = '/';

    public function __construct()
    {
        $this->middleware('guest');
    }
}
```

Fig. 17. Recovering code

Create page code (fig. 18) shows the functions that explain how the admin add pages to the website.

```
public function store(request $request){
$privs=new priv_controller();
if($privs->full_priv()!=1)
    return redirect('/');
$request->url_page= str_replace(' ', '-', $request->url_page);
$validatedData = $request->validate(['name_page'=>'required', 'url_page'=>'required|unique:tbl_pags',
'order_page'=>'required|numeric', 'page_priv'=>'required', 'parent'=>'required', ]);
$page=new Tbl_pag();
$page->name_page=$request->name_page;
$page->id_users=Auth::user()->id;
$page->url_page=$request->url_page;
$page->order=$request->order_page;
$page->priv_page=$request->page_priv;
$page->parent_id=$request->parent;
$page->active_page=$request->active==true?1:0;
$page->save();
    return redirect('/pages/create')->with('success','successfully Post');
}
```

Fig. 18. Create page code

Create category code (fig. 19) shows the functions that explain how the admin add categories to the website.

```
public function store(request $request){
//check privilgs
$privs=new priv_controller();

if($privs->poster_priv()==1||$privs->full_priv()==1)
    return redirect('/');
$request->url_catgroy= str_replace(' ', '-', $request->url_catgroy);
$validatedData = $request->validate([
    'name_catgroy'=>'required',
    'url_catgroy'=>'required|unique:tbl_cats',
    'order_catgroy'=>'required|numeric',
    'catgroy_priv'=>'required',]);
$catgroy=new Tbl_cat();
$catgroy->name_catgroy=$request->name_catgroy;
$catgroy->id_users=Auth::user()->id;
$catgroy->url_catgroy=$request->url_catgroy;
$catgroy->order=$request->order_catgroy;
$catgroy->priv_catgroy=$request->catgroy_priv;
$catgroy->active_catgroy=$request->active==true?1:0;
$catgroy->save();
    return redirect('/categories/create')->with('success','successfully Post');
}
```

Fig. 19. Create category code

Create article code (fig. 20) shows the functions that explain how the admin and teacher add articles to the website.

```

public function store(request $request)
{
    $privs=new priv_controller();
    if($privs->poster_priv()==1||$privs->full_priv()==1)
        return redirect('/');
    $validatedData = $request->validate(['name_article'=>'required','content'=>'required',
    'image_article' => 'image',
    'article_priv'=>'required',
    'catgroys'=>'required',
    'article_url'=>'required|unique:tbl_arts',]);
    if($request->hasFile('image_article')){

        $filenameWithExtention = $request->file('image_article')->getClientOriginalName();
        $fileName = pathinfo($filenameWithExtention,PATHINFO_FILENAME);
        $extension = $request->file('image_article')->getClientOriginalExtension();
        $fileNameStore = 'img' . '_' .time() . '.' . $extension;

        $path = $request->file('image_article')->move(base_path() . '/public/images/', $fileNameStore);
    }
    $article=new Tbl_art();
    $article->title=$request->name_article;
    $article->id_users=Auth::user()->id;
    $article->content=$request->content;
    $article->allow_comments=$request->allow_comments==true?1:0;
    $article->priv_article=$request->article_priv;
    $article->id_catgroys=$request->catgroys;
    $article->article_url=$request->article_url;
    $article->active_article=$request->active==true?1:0;
}

```

Fig. 20. Create article code

Create slider code (fig. 21) shows the functions that explain how the admin add sliders to the website.

```

public function store(request $request){
    $privs=new priv_controller();
    if($privs->poster_priv()==1||$privs->full_priv()==1)
return redirect('/');
    $validatedData = $request->validate(['content'=>'required','image_slider' => 'required|image',
    'slider_priv'=>'required','place'=>'required','order_sliders'=>'required|numeric',]);
    if($request->hasFile('image_slider')){
    $filenameWithExtention = $request->file('image_slider')->getClientOriginalName();
    $fileName = pathinfo($filenameWithExtention,PATHINFO_FILENAME);
    $extension = $request->file('image_slider')->getClientOriginalExtension();
    $fileNameStore = 'img' . '_' .time() . '.' . $extension;
    $path = $request->file('image_slider')->move(base_path() . '/public/images/', $fileNameStore);
    }
    $slider=new Tbl_sdl();
    $slider->id_users=Auth::user()->id;
    $slider->content=$request->content;
    $slider->slider_order=$request->order_sliders;
    $slider->place_content=$request->place;
    $slider->slider_priv=$request->slider_priv;
    $slider->slider_active=$request->active==true?1:0;
    if($request->hasFile('image_slider'))
    $slider->image_slider= $fileNameStore;
    $slider->save();
    return redirect('/sliders/create')->with('success','successfully Post');
}

```

Fig. 21. Create slider code

Create widget code (fig. 22) shows the functions that explain how the admin add widget to the website.

```

public function store(request $request){
    //check privilgs
    $privs=new priv_controller();
    if($privs->full_priv()!=1)
        return redirect('/');
    //end check
    $validatedData = $request->validate(['name_wedg'=>'required','content'=>'required',
    'image_wedg' => 'image','wedg_priv'=>'required','place'=>'required',
    'order_wedgs'=>'required|numeric',]);
    if($request->hasFile('image_wedg'))
    {
        $filenameWithExtention = $request->file('image_wedg')->getClientOriginalName();
        $fileName = pathinfo($filenameWithExtention,PATHINFO_FILENAME);
        $extension = $request->file('image_wedg')->getClientOriginalExtension();
        $fileNameStore = 'img' . '_' . time() . '.' . $extension;
        $path = $request->file('image_wedg')->move(base_path() . '/public/images/', $fileNameStore);
    }
    $wedg=new Tbl_wedg();
    $wedg->title=$request->name_wedg;
    $wedg->id_users=Auth::user()->id;
    $wedg->content_wedg=$request->content;
    $wedg->order_wedgs=$request->order_wedgs;
    $wedg->place=$request->place;
    $wedg->priv_wedg=$request->wedg_priv;
    $wedg->active_wedg=$request->active==true?1:0;
    if($request->hasFile('image_wedg'))
    $wedg->image_wedg= $fileNameStore;
    $wedg->save();
    return redirect('/wedgs/create')->with('success','successfully Post');
}

```

Fig. 22. Create widget code

Functions for CRUD comments. This code explain how the admin, teacher, student create their comment on articles and the admin, teacher, student also can edit, delete their comments as shown in fig. 23.

```

class comment_controller extends Controller{
    public function get_comments($id){
        $comments = Tbl_com::select('*', 'tbl_com.created_at as create_comment', 'users.id as users_id', 'tbl_com.id as comment_id')
        ->orderBy('tbl_com.created_at', 'ASC')
        ->join('users', 'users.id', '=', 'tbl_com.id_users')
        ->join('tbl_arts', 'tbl_arts.id', '=', 'tbl_com.id_article')
        ->where('tbl_arts.id', '=', $id)->get();
        return $comments;}
    public function insert(request $request){
        $comment=new Tbl_com();
        $comment->content_comment=$request->comment_content;
        if (Auth::check())
        $comment->id_users=Auth::user()->id;
        else $comment->id_users=1;
        $comment->id_article=$request->id_article;
        $comment->save();// save comment
        $comments=comment_controller::get_comments($request->id_article);
        return view('article.components.ajax.result')->with('comments',$comments);}
    // delete comment
    public function delete_comment(request $request){
        $comment=Tbl_com::find($request->id);
        $comment->delete();
        $comments=comment_controller::get_comments($request->id_article);
        return view('article.components.ajax.result')->with('comments',$comments);}
    // edit comment
    public function edit_comment(request $request){
        $comment=Tbl_com::find($request->id);
        $comment->content_comment=$request->content_comment;
        $comment->save();
        $comments=comment_controller::get_comments($request->id_article);
        return view('article.components.ajax.result')->with('comments',$comments); }
}

```

Fig. 23. Functions for CRUD comments

4. TESTING OF THE WEB APPLICATION

4.1. Full implementation of the website

Any visitor can see the homepage which contains several functions (“Dashboard”, “Search”, “login”, “Contact Us”) and many sections (“pages”, “sliders”, “widgets”, “articles”, “category”) as shown in the fig. 24.



Fig. 24. The home page

I have in my website three modules the first one is administrator, the second one is a teacher and third one is student every one of them have function which available for his.

The admin, teacher and student can login to the website by entering them special username and password to login in the website as shown in the fig. 25.

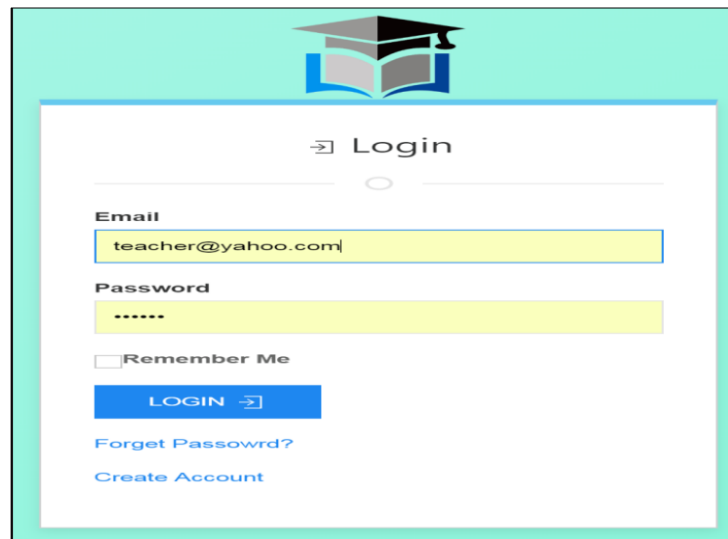


Fig. 25. The login page

The admin, teacher and student can recover his password if they forgot it at any time by using them special email address and they will receive on them email the requested information as in the fig. 26.

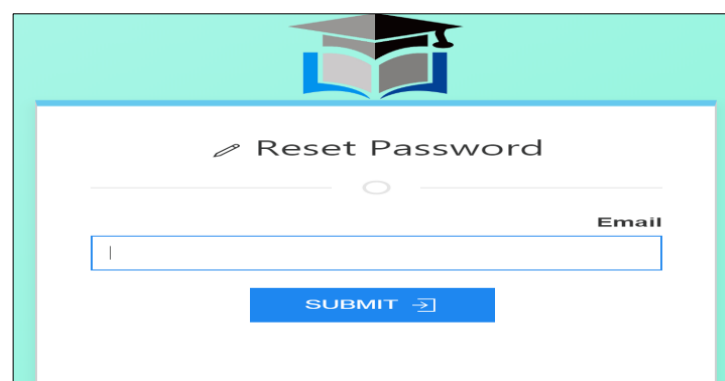


Fig. 26. The reset password page

After the admin, teacher and student login in to the website they will see the homepage and view everything in the website and also they can search about anything by writing the name and search about it as shown in the fig. 27.

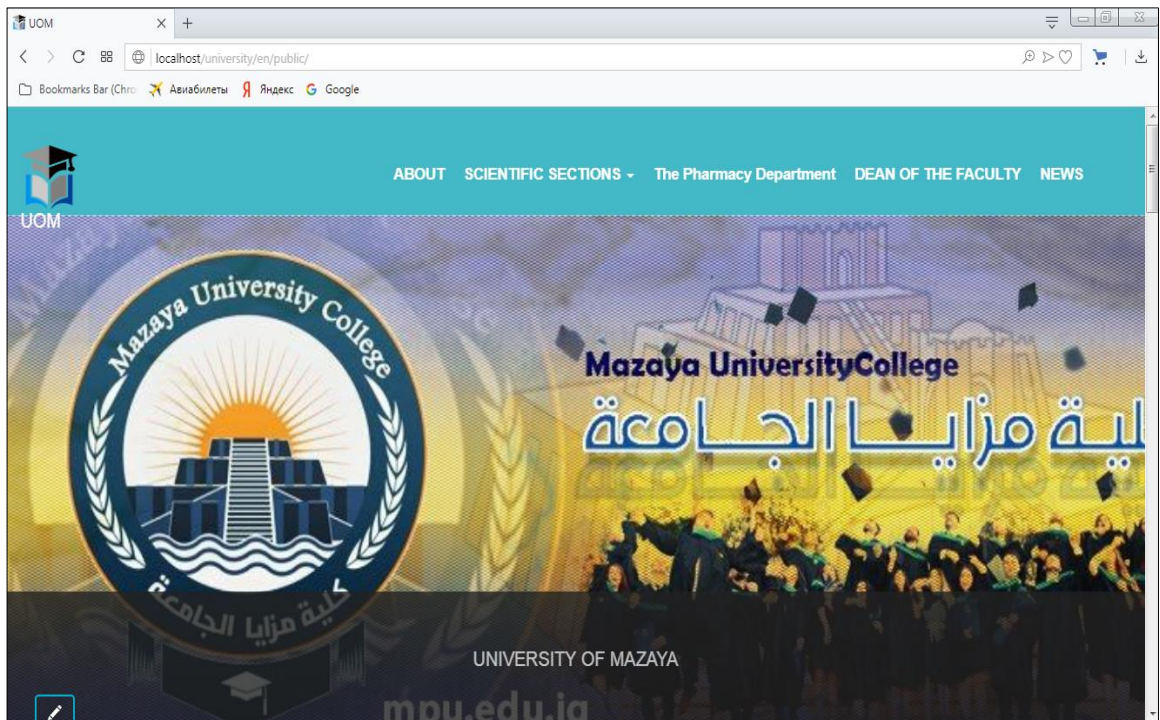


Fig. 27. The home page

After the admin, teacher and student login in to the website and entered to the homepage they can go to the dashboard, we will see in dashboard page many section absolutely all sections will available for admin but not all will available for teacher and student, as shown in the fig. 28.

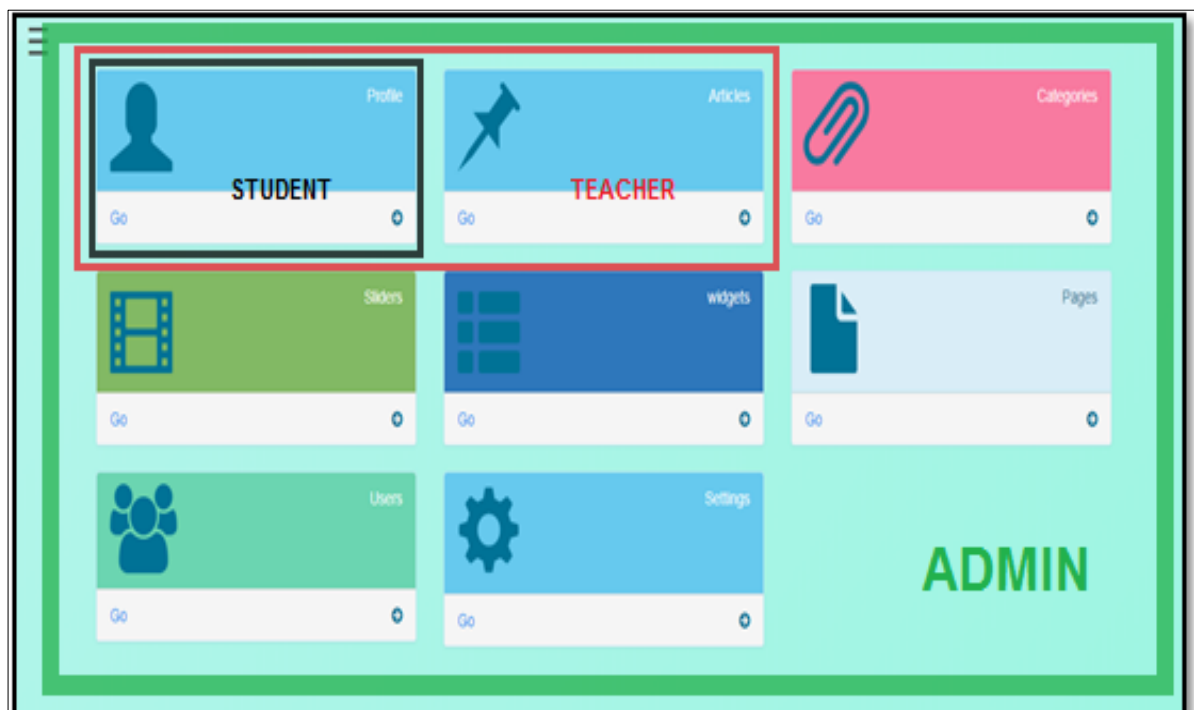


Fig. 28. The Dashboard page

The admin, teacher and student can see his profile by clicking on the button “profile” in the dashboard page and now the admin, teacher and student can add photo to them account and change the password as shown in the fig. 29.

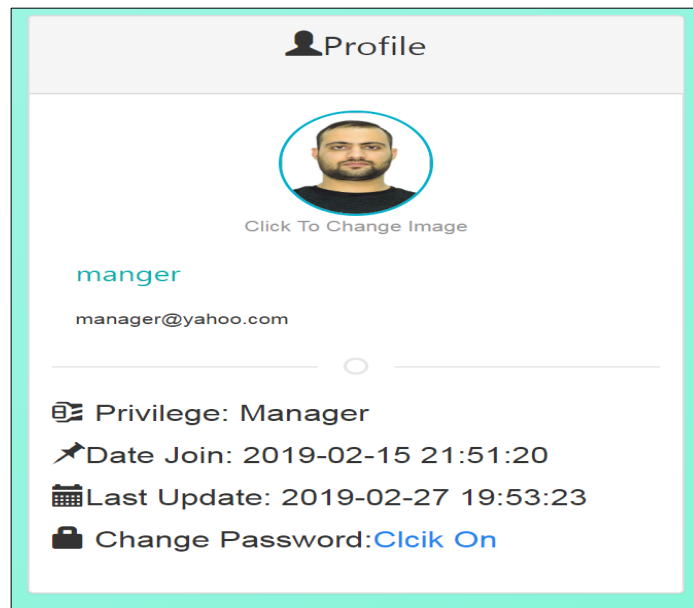


Fig. 29. The profile page

The admin, teacher and student can change them password by clicking on the button “Click On” after that enter the last password and enter new password then confirm it as shown below in the fig. 30.

Fig. 30. The change password page

The admin can also view all pages which already added in the website by clicking on the button “pages” in the dashboard page and also can edit, view and delete any pages as shown below in the fig. 31.

#	Name Page	Date Create	User	Order	Available	Privilege	
1	ABOUT	2019-02-15 22:38:06	manger	1	Active	All	[Edit] [View] [Delete]
2	SCIENTIFIC SECTIONS	2019-02-16 21:28:10	manger	2	Active	All	[Edit] [View] [Delete]
2.1	The Department of Computer Engineering	2019-02-16 19:32:48	manger	1	Active	All	[Edit] [View] [Delete]
2.2	The Department of Civil Engineering	2019-02-16 19:35:17	manger	2	Active	All	[Edit] [View] [Delete]


Fig. 31. The view page


The admin can create new pages in the website by clicking on the button “create” in the page “The view page” the admin can set for every page the following requirements:


- 1) name of page;
- 2) url of the page;
- 3) order of the page;
- 4) privilege means for whose the page will be available for all or just for members;
- 5) parent, in this property we can make the page part of some page;
- 6) active, in this property we can make the page active or we can use this page in another time.


After the admin did this requirement he can create the page by clicking the bottom “post” as shown below in the fig. 32.


Add New Page


 **Create Page**


Name Page 

Url Page 

Order 

Privilege 

Parent 

Active 



POST 
BACK 

Fig. 32. The add new page

The admin can also view all categories which already added in the website by clicking on the button “categories” in the dashboard page and also can edit, view and delete any category as shown below in the fig. 33.

View All Categories

Categories
Create




#	Name Category	Date Create	User	Order	Active	Privilege	
1	ABOUT	2019-02-16 22:53:57	manger	1	Active	All	  
2	The Department of Computer Engineering	2019-02-15 22:52:23	manger	1	Active	All	  
3	The Pharmacy Department	2019-02-15 22:55:00	manger	1	Active	All	  
4	NEWS	2019-02-15 22:55:28	manger	1	Active	All	  
5	DEAN OF THE FACULTY	2019-02-15 22:55:55	manger	1	Active	All	  
6	UNIVERSITYOFMAZAYA	2019-02-16 01:03:46	manger	1	Active	All	  

Fig. 33. The view categories page

The admin can create new category in the website by clicking on the button “create” in the page “view categories”, the admin can set for every category the following requirements:

- 1) name of category;
- 2) url of the category;
- 3) order of the category;
- 4) privilege means for whose the category will be available for all or just for members;
- 5) active, in this property we can make the category active or we can use this category in another time.

After the admin did this requirement he can create the page by clicking the bottom “post” as shown below in the fig. 34.

Add New Category

Create Category

Name Category

Url Category

Order

Privilege

Active

POST

BACK

Fig. 34. The add category page

Both admin and teacher can also view all articles which already added in the website by clicking on the button “articles” in the dashboard page and also can edit, view and delete any articles as shown below in the fig. 35.

View All Articles WebSite							
Article Create							
Title Article	Date Create	User	Category	Available	Privilege	Comments	
1 Al Mazaya University	2019-02-16 22:43:48	manger	ABOUT	Active	All	Not Available	
2 THE DEPARTMENT OF CIVIL ENGINEERING HOLDS A BOOK FAIR	2019-02-16 22:35:35	manger	NEWS	Active	All	Available	
3 THE DEAN	2019-02-16 22:34:47	manger	DEAN OF THE FACULTY	Active	All	Not Available	
4 Pharmacy	2019-02-16 20:47:00	manger	The Pharmacy Department	Active	All	Available	
5 STUDENT NOTIFICATIONS	2019-02-16 20:39:02	manger	NEWS	Active	All	Available	

Fig. 35. The view articles page

The admin and teacher can create new articles in the website by clicking on the button “create” in the page “view articles”.

The admin and teacher can set for every article the following requirements:

- 1) title of article;
- 2) body of the article means what will write inside the articles;
- 3) privilege means for whose the article will be available for all or just for members;
- 4) category of the article means in which the category will appear;
- 5) active, in this property we can make the article active or we can use this category in another time;
- 6) from property “available comments” the admin can make comment available to the article or not;
- 7) also the admin can attach photo by uploading some photo.

After the admin and teacher did this requirement, they can create the articles by clicking the bottom “post” as shown below in the fig. 36.

Fig. 36. The add article page

The admin can also view all sliders which already added in the website by clicking on the button “sliders” in the dashboard page and also can edit, view and delete every slider as shown below in the fig. 37.

#	Date Create	User	Order	Available	Privilege	Place	⚙️
1	2019-02-16 00:25:59	manger	3	Active	All	Footer	✎️ 👁️ ⚙️
2	2019-02-16 00:15:31	manger	1	Active	All	Footer	✎️ 👁️ ⚙️
3	2019-02-15 22:23:28	manger	2	Active	All	Footer	✎️ 👁️ ⚙️

Fig. 37. The view sliders page

The admin can create new slider in the website by clicking on the button “create” in the page “view sliders”.

The admin can set for every slider the following requirements:

- 1) order of the slider;
- 2) content of the slider;
- 3) place of the slider on side or on footer part;
- 4) active, in this property we can make the slider active or we can use this slider in another time;
- 5) privilege means for whose the slider will be available for all or just for members;
- 6) also the admin can attach photo by uploading some photo.

After the admin did this requirement, he can create the slider by clicking the bottom “post” as shown below in the fig. 38.

The screenshot shows a web form titled "Add New Slider". At the top, there is a sub-header "Add Slider" with a small icon. Below this, there is a "Order" field with a dropdown menu. The "Content Slider" section features a rich text editor with a toolbar containing icons for bold, italic, underline, font color, background color, list, and link. Below the editor is a "Privilege" dropdown menu set to "All". The "Place Content" dropdown menu is set to "Side". There is a checked "Active" checkbox. At the bottom, there are three buttons: a red "CLICK TO UPLOAD IMAGE" button, a blue "POST" button, and a red "BACK" button.

Fig. 38. The add slider page

The admin can also view all widgets which already added in the website by clicking on the button “widgets” in the dashboard page and also can edit, view and delete every widget as shown below in the fig. 39.

The screenshot shows a web interface titled "View All Widgets". It features a table with the following data:

#	Title Widget	Date Create	User	Order	Active	Privilege	Place	
1	CONTACT US	2019-02-16 00:08:45	manger	1	Active	All	Footer	[Edit] [Toggle] [Delete]
2	environment conductive	2019-02-15 22:12:57	manger	2	Active	All	Side	[Edit] [Toggle] [Delete]
3	THE MESSAGE	2019-02-15 22:10:15	manger	1	Active	All	Side	[Edit] [Toggle] [Delete]

Fig. 39. The view widgets page

The admin can create new widgets in the website by clicking on the button “create” in the page “view widgets”

The admin can set for every widget the following requirements:

- 1) title of the widget;
- 2) order of the widget;
- 3) content of the widget means what will write inside this widget;
- 4) place of the widget on side part or on footer part;
- 5) privilege means for whose the widget will be available for all or just for members;
- 6) active, in this property we can make the widget active or we can use this widget in another time ;
- 7) also the admin can attach photo by uploading some photo.

After the admin did this requirement, he can create the widget by clicking the bottom “post” as shown below in the fig. 40.

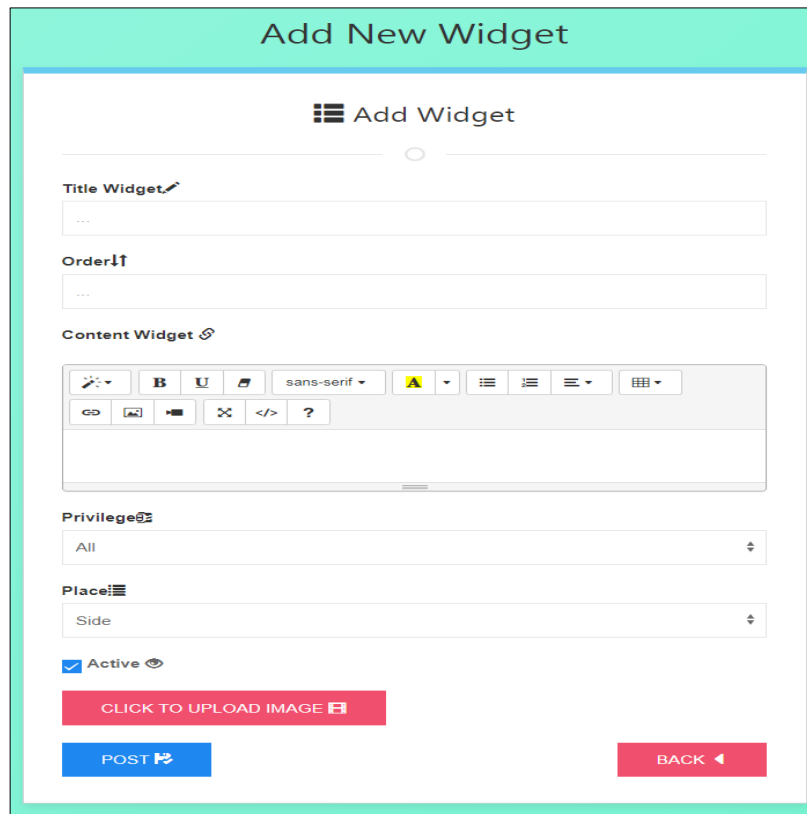


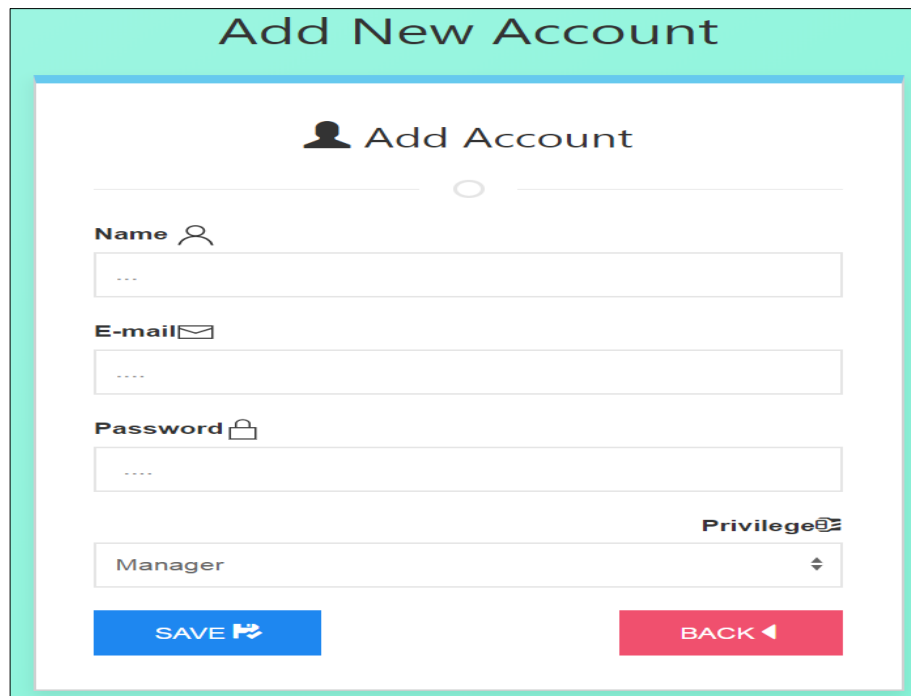
Fig. 40. The add widgets page

The admin can also view account of all users which already in the website by clicking on the button “users” in the dashboard page and also can edit ,view and delete any user as shown below in the fig. 41.

#	User	E-mail	Date Join	Privilege	
1	A.IT Admin	alhamits@gmail.com	2018-03-07 07:43:10	Student	[edit] [view] [delete]
2	ahmed	a@gmail.com	2018-03-03 05:09:15	Student	[edit] [view] [delete]
3	Guest		2018-02-23 01:01:34	Student	[edit] [view] [delete]
4	manger	manager@yahoo.com	2019-02-15 21:51:20	Manager	[edit] [view] [delete]
5	mohammed akeel	mohammed.marco19971997@gmail.com	2018-03-01 01:50:23	Teacher	[edit] [view] [delete]
6	mohammed akeel	mohammed.marco19971997rr@gmail.com	2018-03-01 03:01:24	Student	[edit] [view] [delete]
7	mohammed akeel	mohammed@gmail.com	2018-03-01 11:37:07	Manager	[edit] [view] [delete]
8	student	student@yahoo.com	2019-02-15 21:51:44	Student	[edit] [view] [delete]
9	student	student1@yahoo.com	2019-02-27 22:47:54	Student	[edit] [view] [delete]
10	teacher	teacher@yahoo.com	2019-02-15 21:51:33	Teacher	[edit] [view] [delete]

Fig. 41. The view users

The admin can create new account for the users by entering the name, email and the password of the user in the website by clicking on the button “create” in the page “view users”, the admin can make this user as manager or teacher or student from drop list “privilege” as shown below in the fig. 42.



The screenshot shows a web form titled "Add New Account". At the top, there is a sub-header "Add Account" with a person icon. Below this, there are three input fields: "Name" with a person icon, "E-mail" with an envelope icon, and "Password" with a lock icon. Each field contains three dots indicating it is empty. Below the password field is a "Privilege" dropdown menu with a list icon, currently showing "Manager". At the bottom of the form, there are two buttons: a blue "SAVE" button with a checkmark icon and a red "BACK" button with a left arrow icon.

Fig. 42. The add user page

The admin, teacher and student can comment on the articles on the website and they also can edit and delete the comments as shown in the fig. 43.

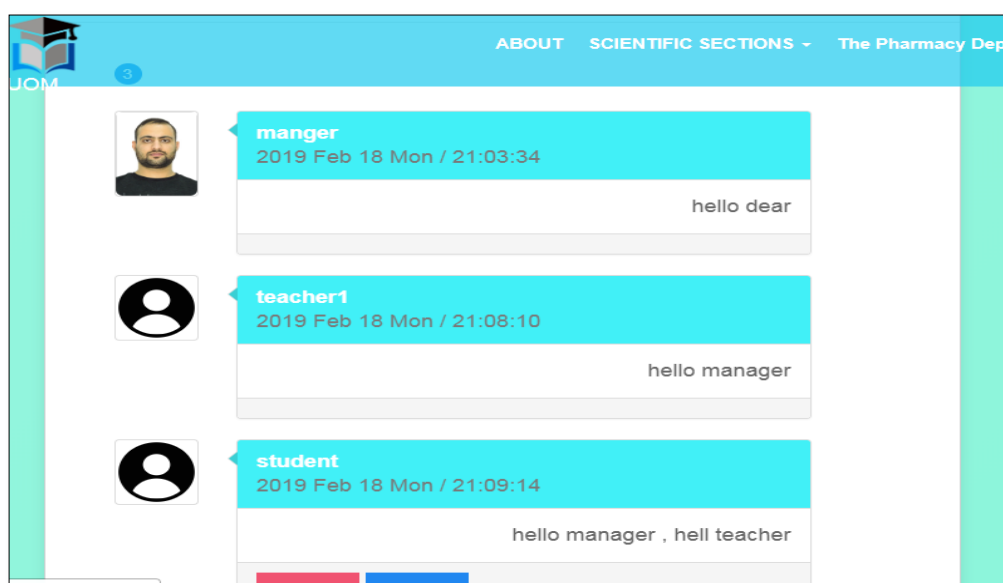


Fig. 43. The add comment page

4.2. The used methods of testing

Functional testing aims to figure out whether given functionality works as specified.

So in functional testing you test that given part of the whole system functions in a specified way. And in system testing you test the system as a whole fulfills the requirements given to it.

Therefore, we compare the actual results and the expected results for three modules this first will be for administrator function, the second will be for teacher and the third will be for student.

Table 2 show the functional testing.

Table 2. The functional testing.

No.	Function	Expected result	Obtained result	Result
1.	To give the admin the permission to login to the website.	The admin can login to the website	The admin can login to the website	work
2.	To give the admin who lost his password the permission to recover of his password.	The admin can recover his password	The admin can recover his password	work
3.	To show the home page to the admin.	The admin can see the home page after login by entering his name and password	The admin can see the home page after login by entering his name and password	work
4.	To give the admin the permission to enter to the dashboard page.	The admin can enter to the dashboard page	The admin can enter to the dashboard page	work

No.	Function	Expected result	Obtained result	Result
5.	The admin must be able to add new pages.	The admin can create new page	The admin can create new page	work
6.	The admin must be able to manage the pages.	The admin can create, view, edit and remove the pages in the website	The admin can create, view, edit and removes the pages in the website	work
7.	The admin must be able to manage categories.	The admin can create, view, edit and remove the categories in the website	The admin create, view, edit and remove the categories in the website	work
8.	The admin must be able to manage the articles.	The admin can create, view, edit and remove the articles in the website	The admin can create, view, edit and remove the articles in the website	work
9.	The admin must be able to manage the sliders.	The admin can create, view, edit and remove the sliders in the website	The admin can create, view, edit and remove the sliders in the website	work
10.	The admin must be able to manage the widgets.	The admin can create, view, edit and remove the widgets in the website	The admin can create, view, edit and remove the widgets in the website	work
11.	The admin must be able to manage the users.	The admin can add, view, edit and remove the users and all them information	The admin can add, view, edit and remove the users and all them information	work

No.	Function	Expected result	Obtained result	Result
12.	The admin must able to manage all comments.	The admin must able to create and edit, read, and remove the comment of all users	The admin must able to create and edit, read, and remove the comment of all users	work
13.	The admin can see his profile.	The admin can view all information about his account and the add photo and change the password	The admin can view all information about his account and the add photo and change the password	work
14.	To search by admin.	The admin can search about something	The admin can search about something	work
15.	To give the teacher the permission to login to the website.	The teacher can login to the website	The teacher can login to the website	work
16.	To give the teacher who lost his password the permission to recover his password.	The teacher can recover his password	The teacher can recover his password	work
17.	To show the home page to the teacher.	The teacher can see the home page after login	The teacher can see the home page after login	work
18.	To give the teacher the permission to enter to the dashboard page.	The teacher can enter to the dashboard page	The teacher can enter to the dashboard page	work

No.	Function	Expected result	Obtained result	Result
19.	The teacher must be able to manage the articles.	The teacher can create, view, edit and remove the articles in the website	The teacher can create, view, edit and remove the articles in the website	work
20.	The teacher can see his profile.	The teacher can view all information about his account and add photo and change the password	The teacher can view all information about his account and add photo and change the password	work
21.	To give the teacher the permission to manage his comment.	The teacher must be able to create, edit, read and remove his comment	The teacher must be able to create, edit, read and remove his comment	work
22.	To search by teacher.	The teacher can search about something by writing this thing in the search field	The teacher can search about something by writing this thing in the search field	work
23.	Register in the website as student.	The user can create account as student by insert his email, username and password after he registered, he can login.	The user can create account as student by insert his email, username and password after he registered, he can login.	work
24.	To give the student the permission to login to the website.	The student can login to the website by entering his name and password.	The student can login to the website by entering his name and password.	work

No.	Function	Expected result	Obtained result	Result
25.	To give the student who lost his password the permission to recover his password.	The student can recover his password	The student can recover his password	work
26.	To show the home page to student.	The student can see the home page after login by entering his name and password	The student can see the home page after login by entering his name and password	work
27.	To give the student the permission to enter to the dashboard page.	The student can enter to the dashboard page	The student can enter to the dashboard page	work
28.	The student can see his profile.	The student can view all information about his account and add photo and change the password	The student can view all information about his account and add photo and change the password	work
29.	To give the student the permission to manage his comment.	The student can create, edit, read and remove his comment	The student can create, edit, read and remove his comment	work
30.	To search by student.	The student can search about something by writing this thing in the search field	The student can search about something by writing this thing in the search field	work

CONCLUSION

The system implemented the basic functions of information website by using laravel, we achieved all the goals and working with Larval was a great.

We have achieved all the main functions required; now the admin can do the following functions:

- 1) CRUD pages;
- 2) CRUD sliders;
- 3) CRUD widgets;
- 4) CRUD categories;
- 5) CRUD articles;
- 6) CRUD users;
- 7) CRUD comments.

The teacher can CRUD his articles and his comment whereas the student can register in the website and also can CRUD comments on articles.

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

- 1) the problem statement and make the comparative analysis of the PHP frameworks for website development are studied ;
- 2) the structure of the required database for the information website are developed;
- 3) the web-application was designed;
- 4) implemented the website;
- 5) the system were tested .

The perspectives for the developed application. Always we can think out some function, which we can implement in future:

- 1) to include an integrated electronic library;
- 2) to develop the site by increasing the information and options in proportion to educational institution and used it as an ideal website for a college or university.

From this system development process, I learned a lot. Previously, I was mostly learning conceptual information, and also had done some piecemeal exercises, which were very different compared to this system design.

This project gave me a great opportunity, and finally joined together conceptual expertise with practice. Through design of this system, I summed up the following experience: First of all, I recognized the importance of the code standard. Especially in some large-scale projects, there would be hundreds or thousands of functions. If codes were written unstandardized, a series of troubles would appear in the debugging and checking stage.

Finally, I recognized the importance of the debugging method. System debugging process is more complex than the development process. In fact, each operation and each submitting a request to the server includes a lot of small pieces of code execution. If the programmer do not use an appropriate debugging method, he/she will be difficult to identify errors in the program.

REFERENCE LIST

1. Blog Pusher. [Electronic Resource] URL:
<https://blog.pusher.com/laravel-mvc-use/> (the date of access: 16.04.2019).
2. Cloud ways. [Electronic Resource] URL:
<https://www.cloudways.com/blog/models-views-laravel/> (the date of access: 16.04.2019).
3. G2crowd. [Electronic Resource]
URL:<https://www.g2crowd.com/categories/student-information-systems-sis> (the date of access: 31.12.2018).
4. Isadora Duncan. [Electronic Resource]
URL:http://www.answers.com/Q/Why_is_the_internet_so_popular (the date of access: 23.12.2018).
5. Lucid chart. [Electronic Resource]
<https://www.lucidchart.com/pages/uml-activity-diagram> (the date of access: 17.04.2019).
6. Merriam Webster. [Electronic Resource] URL:<http://www.merriam-webster.com/dictionary/world> (the date of access: 23.12.2018).
7. Official site of MySQL Server. [Electronic Resource]
URL:<http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html> (the date of access: 06.01.2019).
8. Official site of MySQL. [Electronic Resource]
URL:<https://www.mysql.com/about> (the date of access: 1.1.2019).
9. Open source. [Electronic Resource]
URL:<https://opensource.com/business/16/6/which-php-framework-right-you> (the date of access: 24.02.2019).
10. PhpMyAdmin. [Electronic Resource] URL: www.phpmyadmin.net (the date of access: 06.01.2019).
11. Search sqlserver Techtarget. [Electronic Resource] URL:
<https://searchsqlserver.techtarget.com/definition/database-management-system> (the date of access: 23.12.2018).

12. Self taught coders. [Electronic Resource] URL:
<https://selftaughtcoders.com/from-idea-to-launch/lesson-17/laravel-5-mvc-application-in-10-minutes/> (the date of access: 16.04.2019).
13. Sodiya A.S. User interface design and ergonomics. – Nigeria: National Open University of Nigeria, 2009. – 200 p.
14. Stack overflow. [Electronic Resource]
URL:<https://stackoverflow.com/questions/20239550/difference-between-functional-testing-and-system-testing> (the date of access: 05.02.2019).
15. Study Mode. [Electronic Resource]
URL:<https://www.studymode.com/course-notes/Student-Database-Management-System-51984092.html> (the date of access: 06.01.2019).
16. Toptal. [Electronic Resource]
URL:<https://www.toptal.com/php/choosing-between-symfony-and-laravel-frameworks> (the date of access: 24.02.2019).
17. Tristate technology. [Electronic Resource]
URL:<https://www.tristatetechnology.com/blog/best-php-framework-web-development/> (the date of access: 24.02.2019).
18. Tudip. [Electronic Resource] URL:<https://tudip.com/blog-post/laravel-vs-php-frameworks/> (the date of access: 24.02.2019).
19. Tutorial laravel. [Electronic Resource] URL:
<http://www.tutoriallaravel.com/laravel/laravel-features/> (the date of access: 07.02.2019).
20. Tutorials point. [Electronic Resource] URL:
https://www.tutorialspoint.com/laravel/laravel_overview.htm (the date of access: 06.01.2019).
21. Visual-paradigm. [Electronic Resource] URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/> (the date of access: 17.04.2019).
22. Wpblogx. [Electronic Resource] URL:
<https://www.wpblogx.com/what-is-xampp/> (the date of access: 07.02.2019).