The branch of “Nazarbayev Intellectual School of physics and mathematics in Uralsk”of АЕО “Nazarbayev Intellectual Schools”

Subject: Computer science

Course work

Personal site of Computer science teacher

**Student:**  Fazyl Yelaman,

Grade 11 «A»

**Teacher:** Zelenov Boris Alexandrovich

Uralsk, 2019

(a) Definition, investigation and analysis 2

(i) Definition - nature of the problem 2

(ii) Investigation and Analysis. 4

Investigation: 4

Analysis: 10

(b) Design. 17

(i) Nature of the solution. 17

(ii) Intended benefits 41

(iii) Limit of the scope solution 42

(c) Software development, programming, testing and installation 45

(i) Development 45

(ii) Programming. 64

(iii) Testing. 85

(iv) Installation 123

(d) Documentation. 124

(i)Systems maintenance documentation 124

(ii)User guide 124

(e) Evaluation 130

(i) Discussion of the degree of success in meeting the original objectives. 130

(ii) Evaluate the client's and user's response to the system 131

# Definition, investigation and analysis

## Definition - nature of the problem

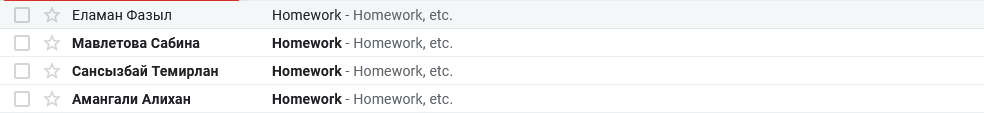
**Description of the organization**

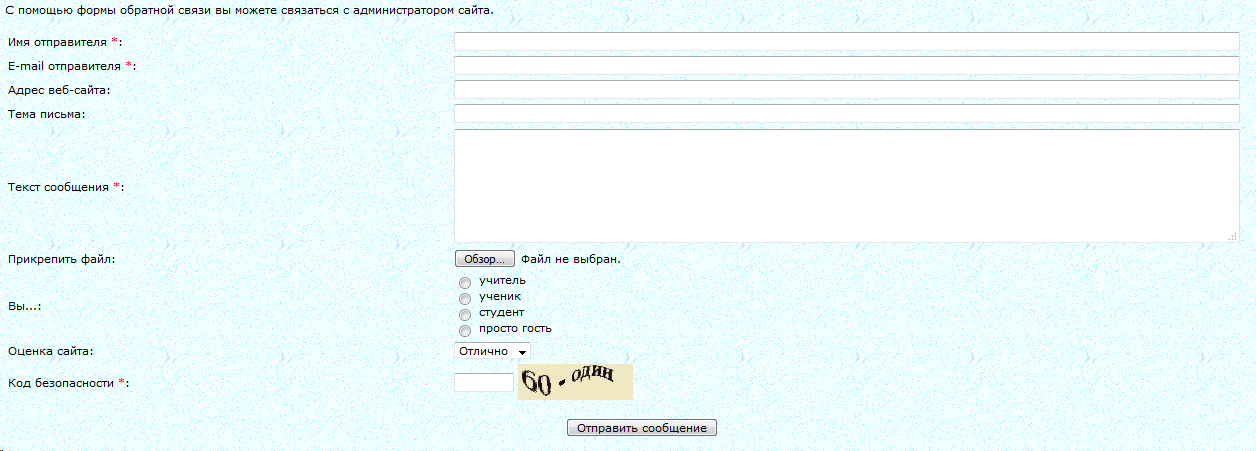
The Nazarbayev Intellectual Schools (next NIS) are a network of schools for gifted students from Kazakhstan. NIS was established in 2009. My client Zelenov Boris Alexandrovich works as Computer science teacher in NIS of Physics and Mathematics in Uralsk. There studies around 600 students. He has about 200 students for teaching them. In one week he has 24 lessons.

**The methods that are used now**

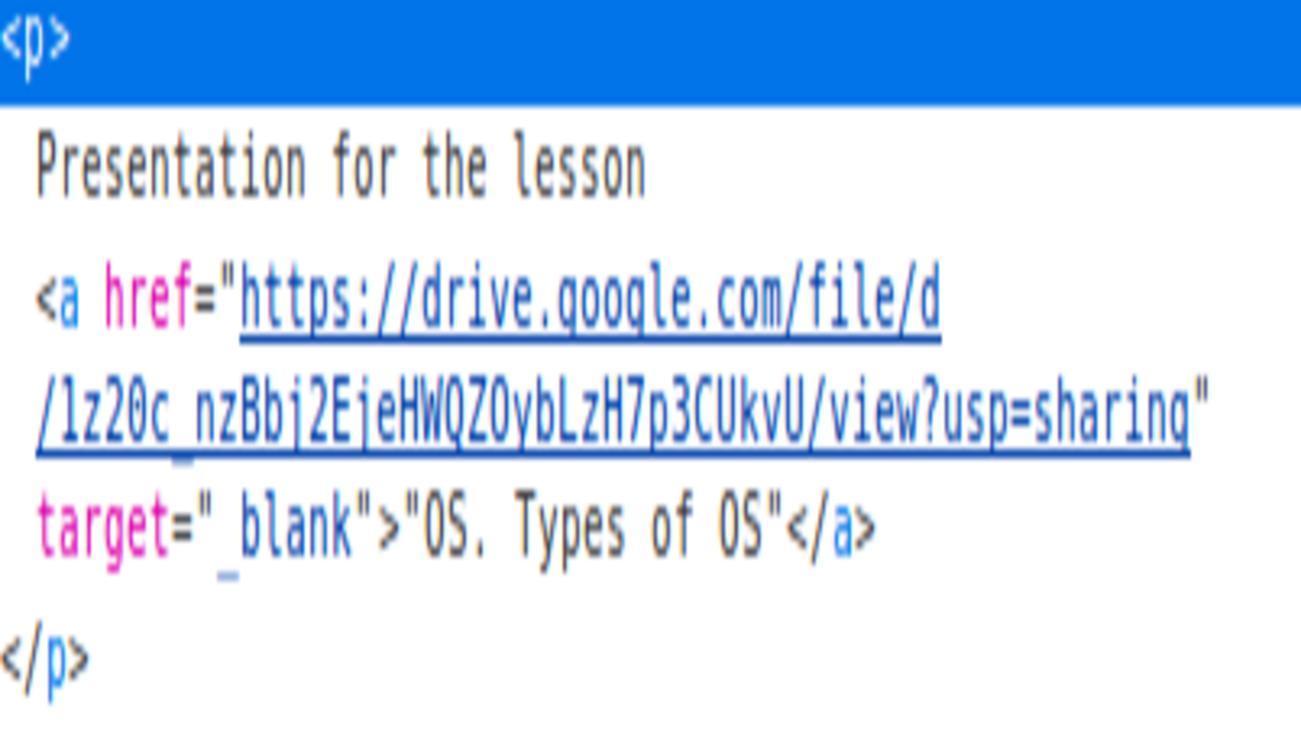
On his lessons, students don’t use books and there is no single educational resource, that students can use to learn more about subject. In addition, computer science is hard subject for students from 11-12 grades. So, my client uses Google Drive to store his data and some useful material. Nowadays, my client uses website for communicating with students. He uploads materials of lessons and put links to useful sources and uses feedback e-mail system to collect homework. In addition, on the system stores a lot of external information related to Computer Science, that could be helpful for students, who are interested in this subject to get more interesting information. However, some data stores on cloud services, such as Google Drive and students access them by links, (интерактивные материалы в сети) moreover on the site my client can’t give exclusive feedback or external homework for some students. However, there is no use of interactive conversation by using internal communication on website.

**Origin of the data**

Students send information (for example, homework) by form on the site, but this form re-sends this information to my client’s e-mail box and it could be hard to control all this flow of information.



In addition, my client uploads materials of lessons to cloud drive (for example, Google Drive) and then add link to the site to material on cloud drive.



**Description of the problem**

Firstly, it’s hard for my client to take messages from site to e-mail box. Secondly, materials from the lessons stores on Google’s servers and if there will be some problems in Google, students won’t access to the materials. Thirdly, on this system my client can’t give specific feedback or give extra homework to one student

## Investigation and Analysis

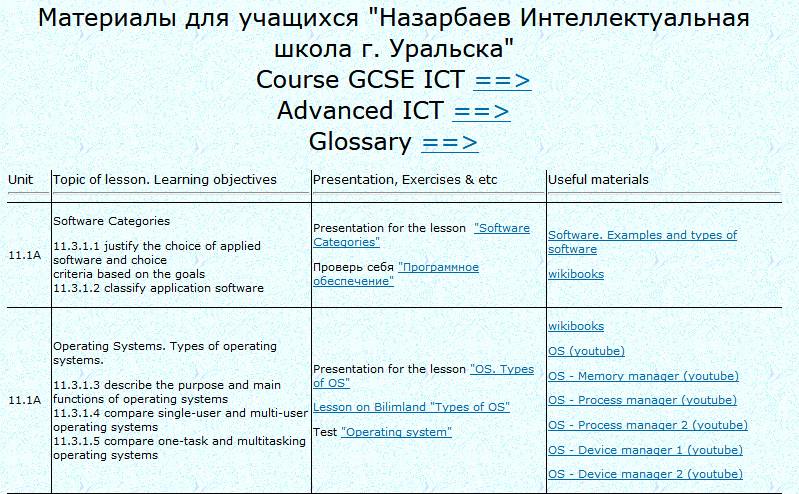
### Investigation

In my research, I have chosen methods of researching like interview and exam documentation, because by using them, I can collect information, what teacher needs on his own site and how current system works. Firstly, I made a exam documentation, because I can know, what problems does site have and ask about it my client in interview, so I will try to find ways how to solve them.

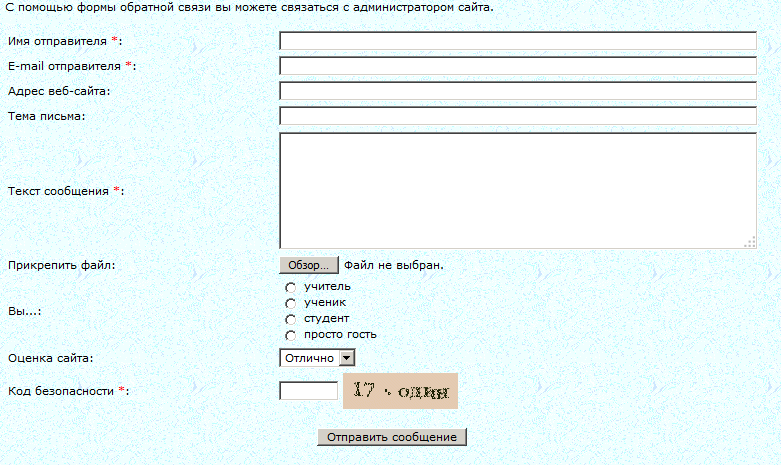
Results of exam documentation:



In this screenshot, you can notice that there used blog, authorization system and some useful links.



There you can see how lesson materials are presented and how students access them.



In this screenshot, you can see how students send messages to teacher. They use fill this form and message sends to teacher’s e-mail.

Interview with Zelenov B.A.:

- What’s your profession?

- I work as computer science teacher in NIS PhM in Uralsk

- How you interact with your students?

- I use my site to interact with them

- Can they send you some messages and can you send them messages by using your system?

- They can use feedback system to send messages to my e-mail box, but I can’t send them messages

- Is it comfortable to take messages from students to e-mail box?

- No, because, I have a lot of messages from a lot of people, companies and there is no sorting of them, so sometimes, it’s hard to find messages from students

- Do you need to send them messages?

- Yes, because, sometimes some students can’t understand topic, so by communicating with them, I can know what they didn’t understand actually and give them external homework for better understanding of topic

- How they access to the material of the lesson?

- I have sections for each grade, when they can find links to materials, usually to my presentations on Google Drive

- How do you think, using cloud drive as data store of lessons material is comfortable for you and students?

- It’s comfortable for me to upload materials, but there is 2 troubles. Firstly, it’s lack of space on Google Drive and secondly, if there would be some troubles on Google’s servers, students can’t access it.

- Do you have a lot of visitors on your own site and what is reason of this stats?

- Statistically, I have not so many visitors, in average 15-20 in a day. This numbers could be better, because of lack of functions, there is a little number of visitors

- How do you think, can changing your system to new one will increase this numbers?

- Yes, of course. As I said, adding new functions can radically change this numbers

- So, to remove some problems, I can provide you with some new alternatives to your system.

Firstly, it’s using local program on school PC. However, it has some advantages and disadvantages. For advantages, it’s easy to use for you and for students, you can connect database to use it as data store and it doesn’t require Internet. For disadvantages, students can’t access materials from everywhere and database should be local.

Secondly, to create a system, can be use mobile application. It has a lot of pros and cons. Examples of pros is, that mobile application is portable and it’s easy to access, it can be used offline, it has friendly interface and you can use database to store data. However, it can’t be accessed from computer, smartphone has small screen and hardware is more weaker that on PC, so there would be slow calculations.

Thirdly, there can be used web-site with personal cabinet of every student. Advantages of this system is everyone can access from everywhere and everytime, from every device, also it has friendly interface and you can directly connect database, that stores information on server. However, it has some drawbacks as paying for hostings, you should protect your site of attacks, as DDoS attack and it requires internet.

- I think the best way is website, because, comparing to another it has significant advantages and insignificant disadvantages.

- Do you need a function of messaging on your website?

- Yes, it can solve several problems

- Would it be useful, if we store materials of lessons on your own servers?

- Yes, it can help with elimination of issue, created by using Google Drive.

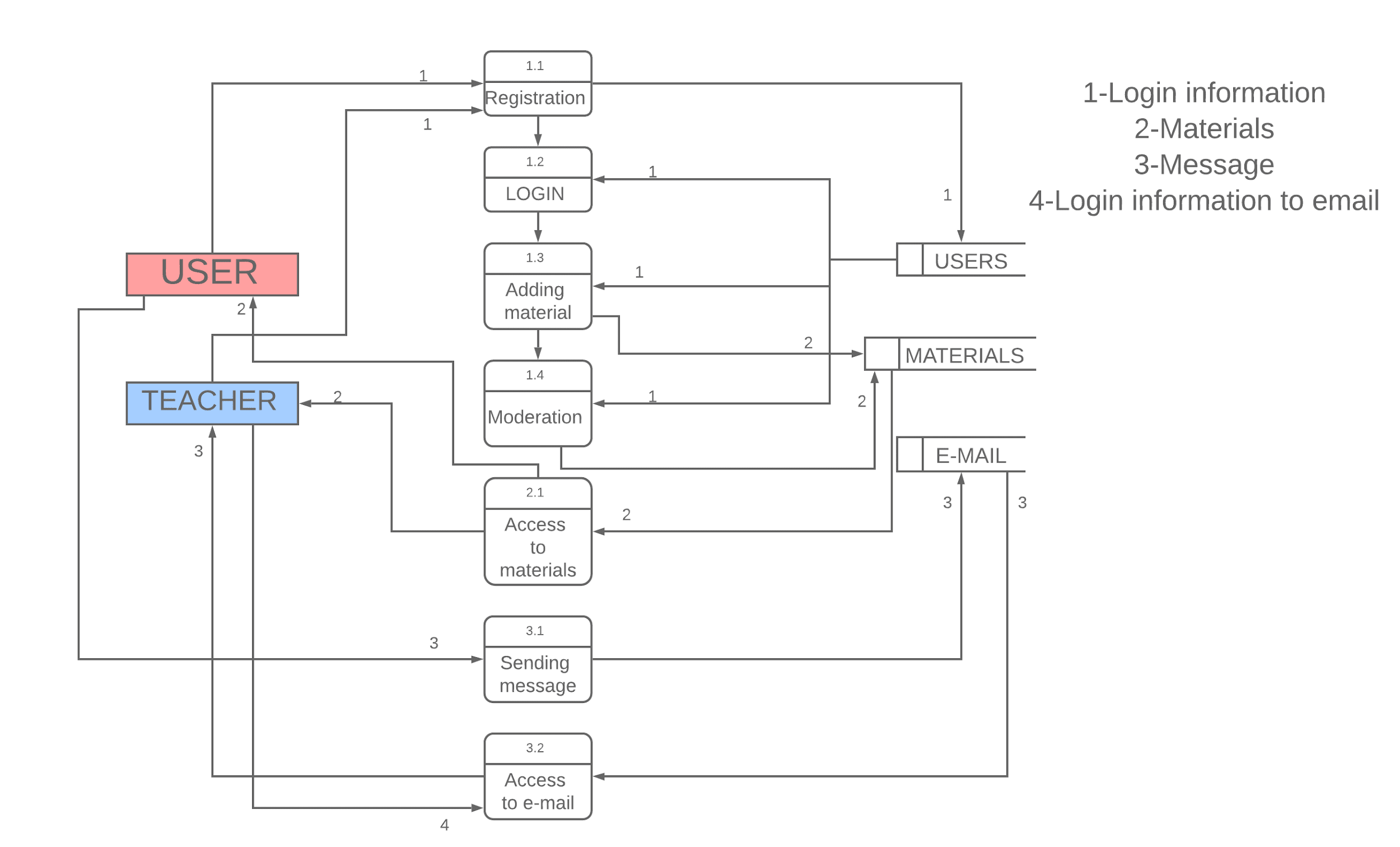
- Do you need system to be closed, so only registered students can access materials of the lessons?

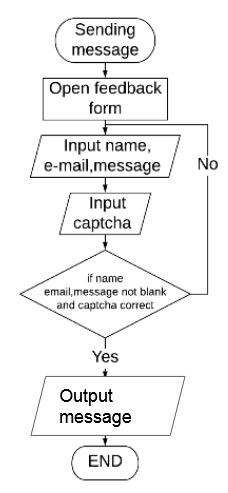
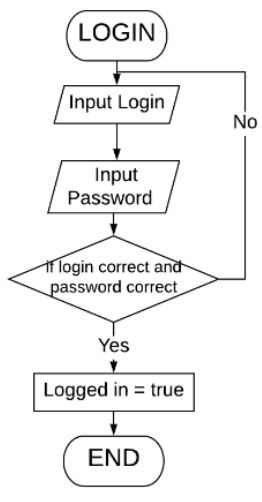
- Yes, because, by using it I can know who is using my site and if they would have own profiles, where parents can check the results of their child, it will be very useful

### Analysis

Спроектируйте работу текущей системы:

Current system has several problems. Firstly, it’s there is no feedback system with students, so my client can’t talk with them and give them extra homework. Secondly, messages from students sends to my client’s e-mail, that’s why, it’s hard to find some messages from students. Thirdly, materials of the lessons are stored on Google Drive, so using it has 2 problems, it’s that space on drive aren’t infinity and if there would be some troubles on Google’s servers and students can’t access to materials.





Current system has few advantages. Firstly, students can easily access to lesson’s material. Secondly, students can contact with teacher by using feedback form. Thirdly, students can add their own material to share it with another students.

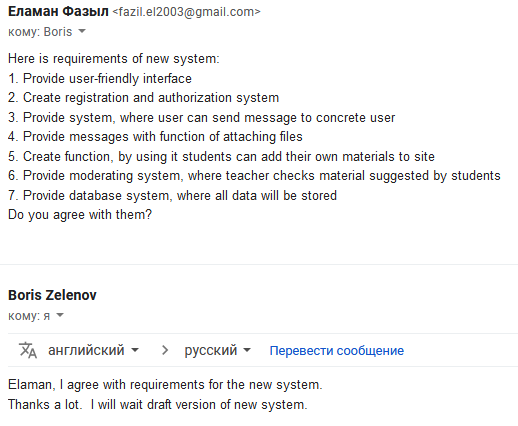
However, current system has several disadvantages. Firstly, teacher can’t send message to students and give feedback or external homework. Secondly, material stored on the site by using links, so if there will be some problems on server where material stores students couldn’t access it. Finally, it’s hard for teacher to take messages from site to email box.

From this information, I can conclude, that new system should have easy access to materials of lessons, messaging system where teacher can give feedback or give additional homework, adding information by students, store all materials on own server and system should be closed, so materials can be accessed only by using authorization system.

New system requirements:

1. Provide user-friendly interface
2. Create registration and authorization system
3. Provide system, where user can send message to concrete user
4. Provide messages with function of attaching files
5. Create function, by using it students can add their own materials to site
6. Provide moderating system, where teacher checks material suggested by students
7. Provide database system, where all data will be stored

Prove of agreement:



# Design

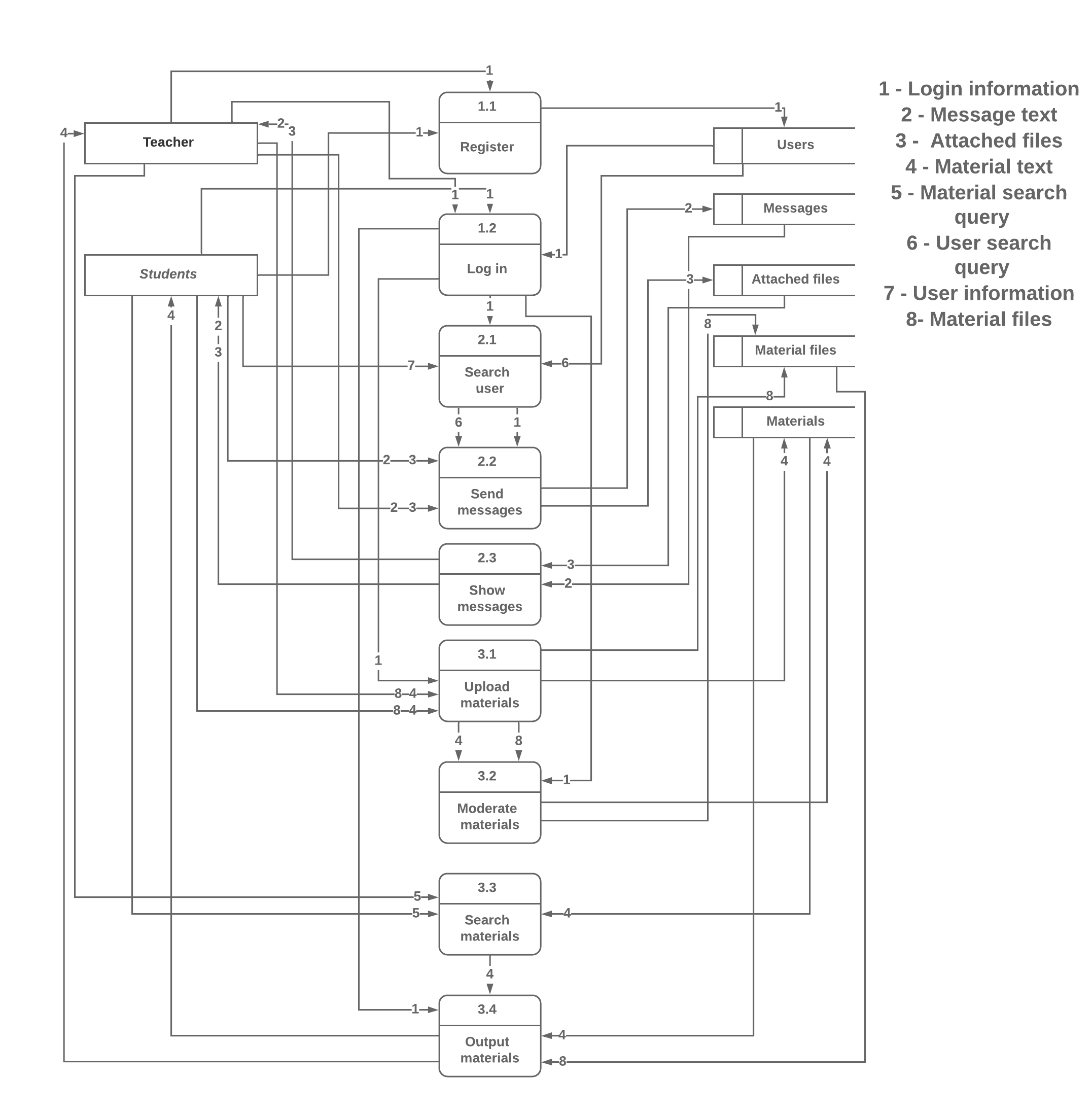
To realize new system I used this goals:

* To provide user-friendly interface
* To create registration system
* To handle the data entered by user in registration
* To create authorization system
* To create chatting system
* To provide to input materials
* To make moderation system to check new uploaded materials
* To provide search system for materials
* To provide the system of giving and removing permission of moderator
* To provide the checking of logined user
* To use database to contain materials, messages, uploaded files and user data

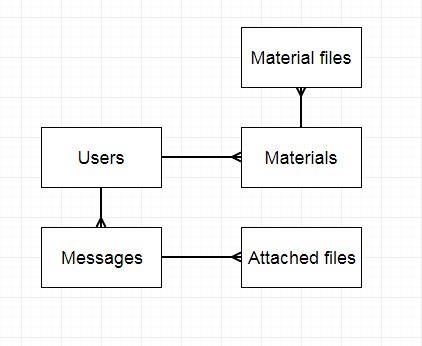
## Nature of the solution

(Сделайте небольшое введение перед проектированием. Что вы собираетесь сделать? На основе чего?)

DFD (Data Flow Diagram)



ERD (Entity Relationship Diagram)



Data dictionary

Table Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fields | Data type | Length | Description | Example |
| userID | Integer |  | user’s id | 1 |
| name | varchar | 50 | user’s first name | Elaman |
| surname | varchar | 50 | user’s surname | Fazyl |
| email | varchar | 50 | user’s email | fazil.el2003@gmail.com |
| password | varchar | 255 | user’s password | $2y$12$QjSH496pcT5CEbzjD/vtVeH03tfHKFy36d4J0Ltp3lRtee9HDxY3K |
| grade | integer | 5 | user’s grade | 11 |
| moderator | boolean |  | user’s status | false |

Table Messages

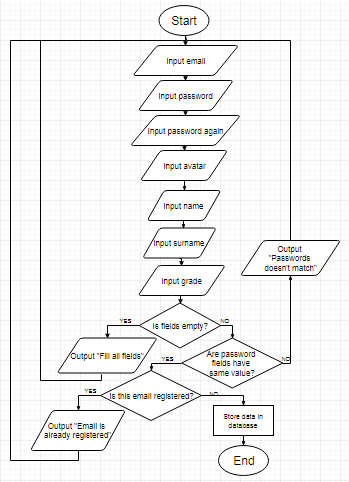
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fields | Data type | Length | Description | Example |
| messageID | Integer |  | message id | 1 |
| senderID | Integer |  | message’s sender | 1 |
| collectID | Integer |  | message’s collecter | 2 |
| text\_message | text | 65536 | message’s text | “Hello, how are you?” |

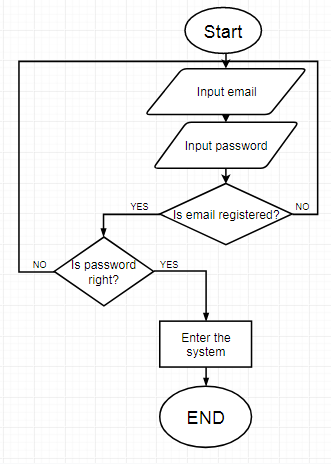
Table Materials

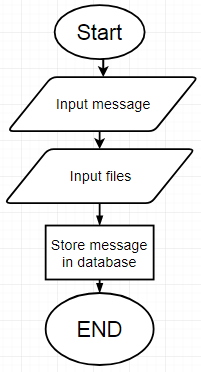
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fields | Data type | Length | Description | Example |
| materialID | Integer |  | material’s id | 1 |
| material\_name | varchar | 100 | material’s topic name | “How to use array in PHP?” |
| material\_text | text | 65536 | material’s text | PHP is object-oriented programming language |
| userID | Integer |  | material’s creator | 1 |
| moderated | boolean |  | material moderated? | 1 |

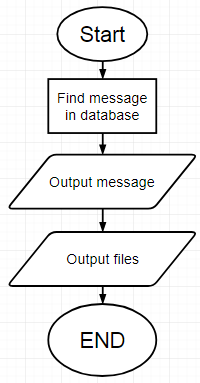
Table Material files

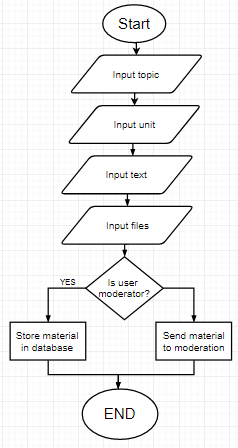
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fields | Data type | Length | Description | Example |
| fileID | Integer |  | file’s id | 1 |
| file\_path | varchar | 255 | file’s path | “php.pptx” |
| file\_material\_id | Integer |  | material’s id | 1 |

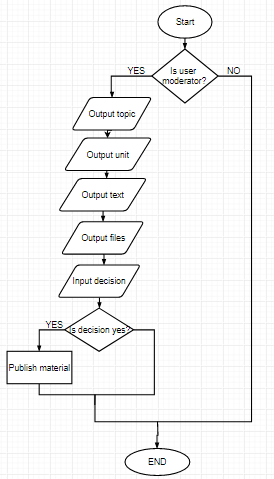
Register

Log in

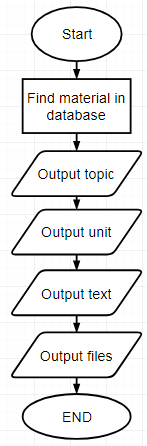
Send message

Show messages

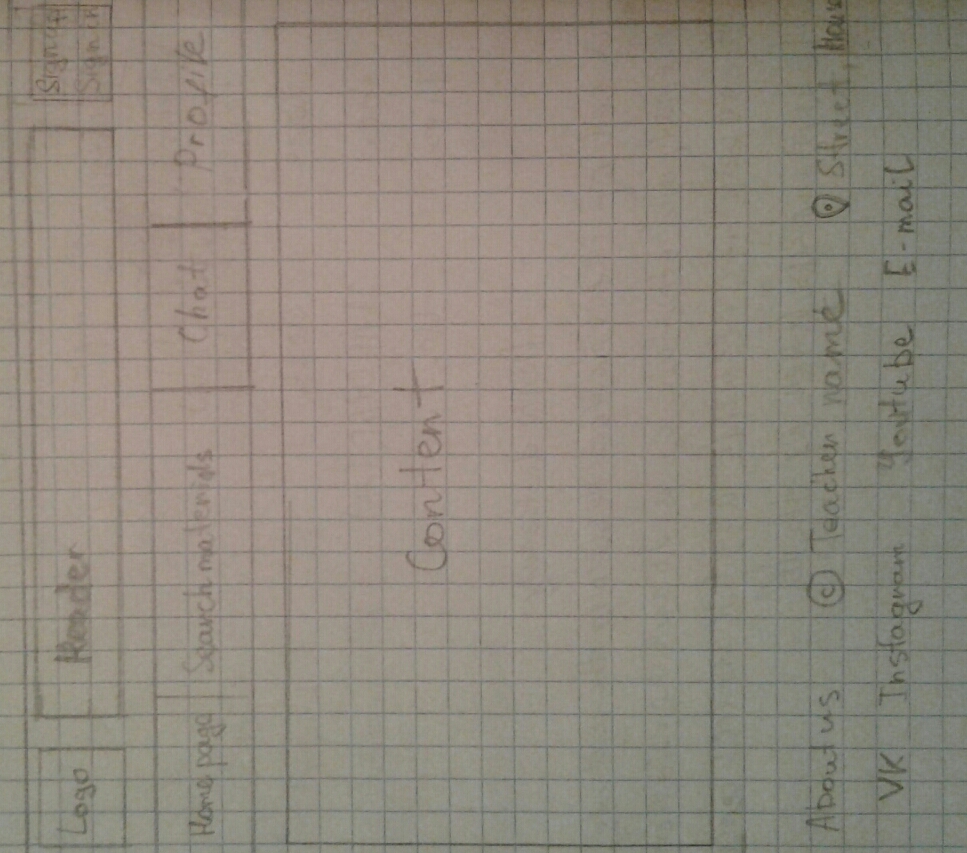
Upload materials

Moderate materials

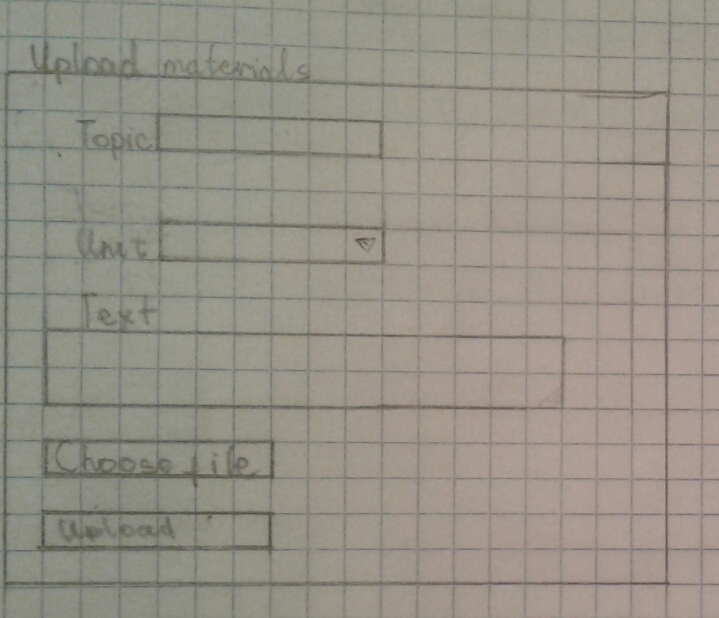
Search materials

Output materials

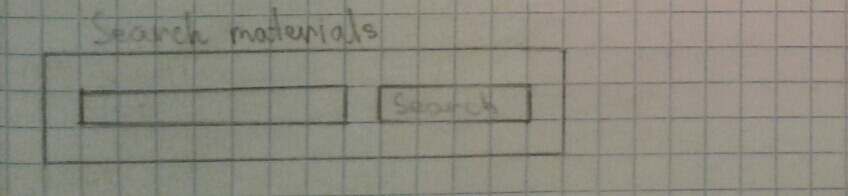
**Prototype:**

****

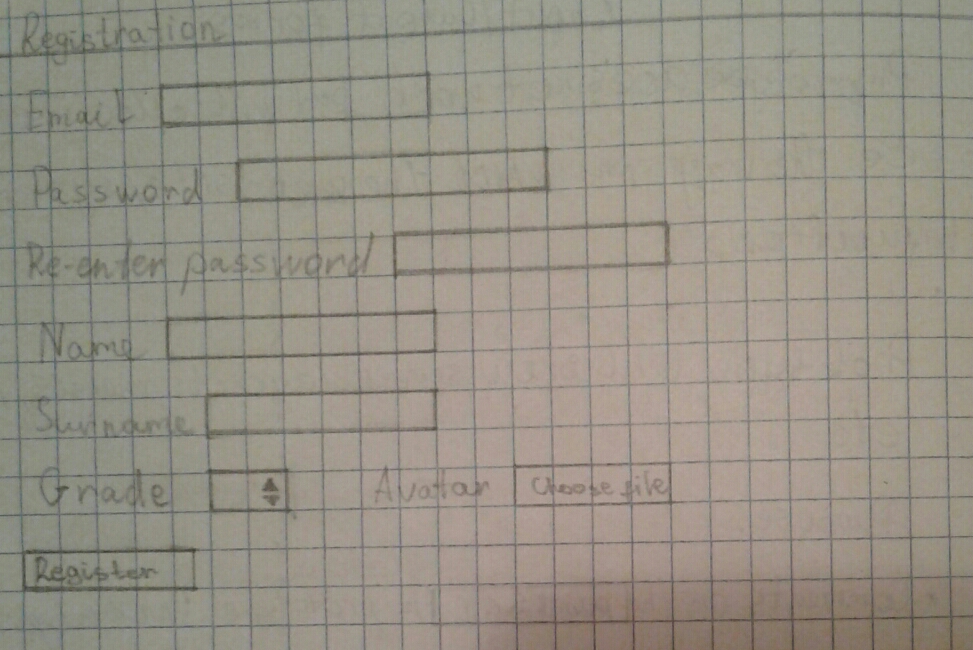
**Input/output forms**



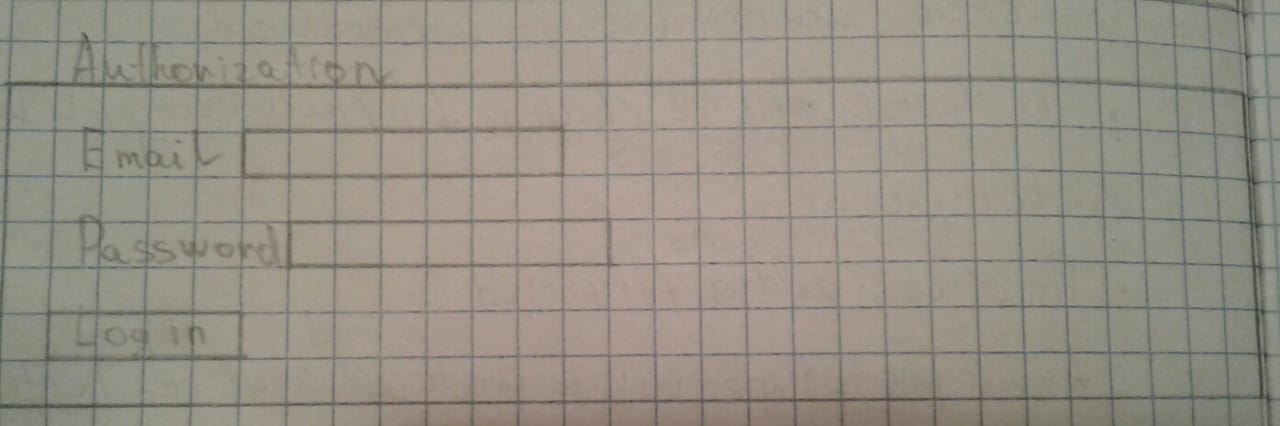
Upload materials



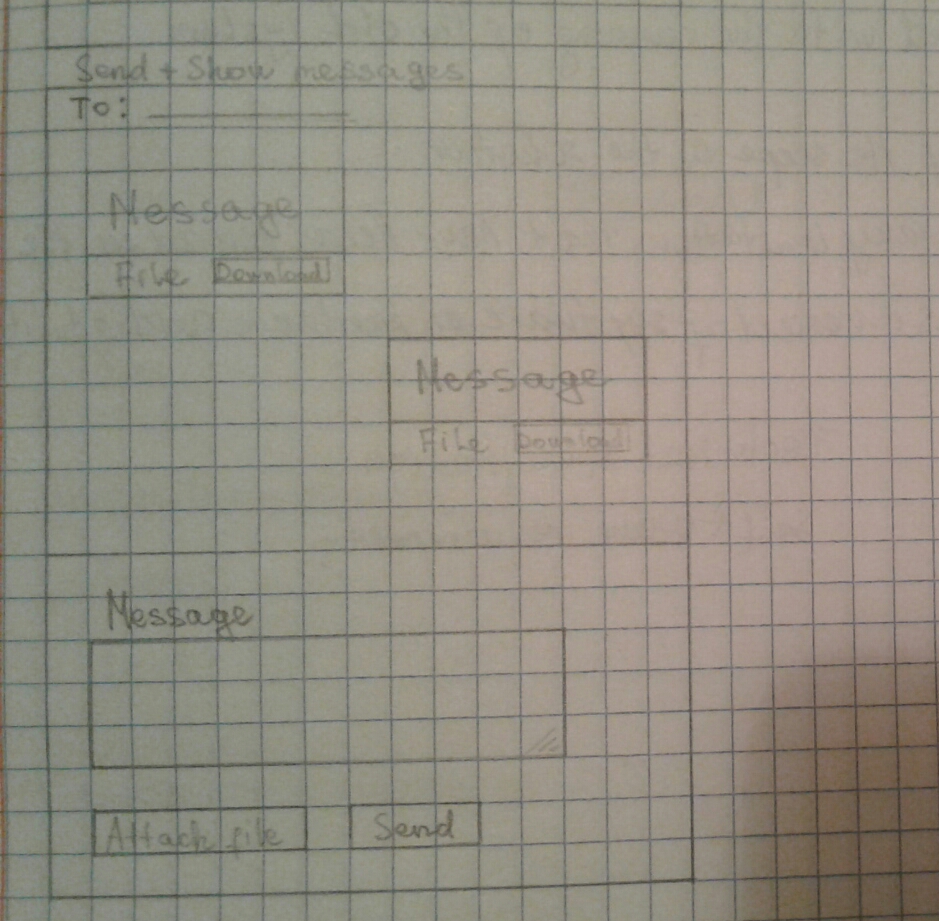
Search materials



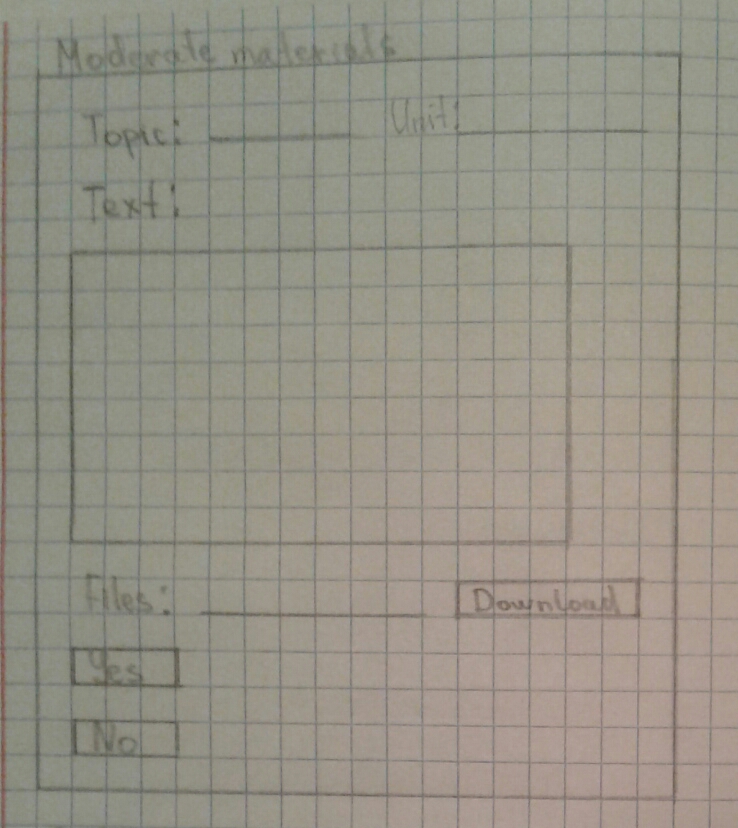
Register



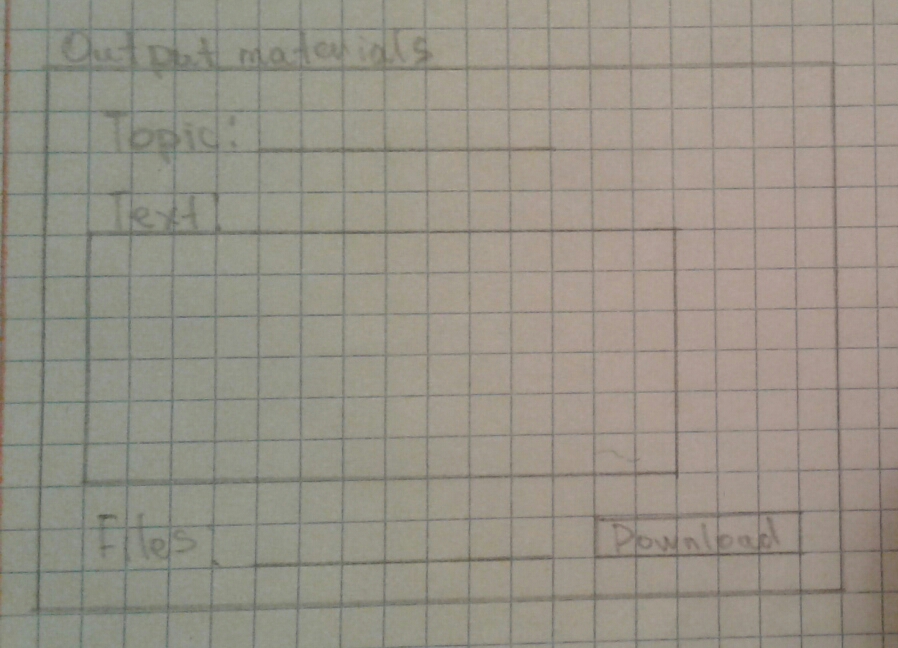
Log in



Send and Show messages



Moderate materials



Output materials

## Intended benefits

1. Using messaging system, where students can talk with each other without using external messengers and teacher can give any advice or feedback
2. Materials stored on own server, so students can easily access it
3. Password are contained in encrypted form, so it give protection

## Limit of the scope solution

1. Size of uploaded materials, attached files, avatars doesn’t have limit, so it could fastly fill servers
2. Size of web-site isn’t flexible, so it could be hard to use it at different devices
3. User can’t modify his profile information, so if you registered once with incorrect data, it will be stored without permission of changing it

Hardware requirements:

|  |  |  |
| --- | --- | --- |
| Device | Characteristics | Purpose of using |
| Hard drive | 50 MB free storage | To store all data |
| CPU | 1 GHz | To processing |
| Monitor | 1920\*1080 | To show content of site |
| Keyboard |  | To input some data |
| Mouse |  | To navigate on website |

Software requirements:

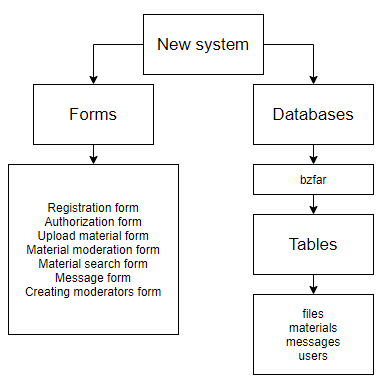
|  |  |
| --- | --- |
| Software | Purpose of using |
| Visual Studio Code | To write and edit programming code |
| Photoshop | To create and edit some graphical objects |
| Git | To control versions and access code easily |
| Opera | To see results of code |
| Apache | To create local web server |
| Mysql | To use database system |

# 

# Software development, programming, testing and installation

## Development

The new system uses the following forms, databases and tables:



**Tables used in the new system:**

The table with data of material files



Таблица “files”

The table containing information about materials:



Таблица “materials”

The table with data of messages:

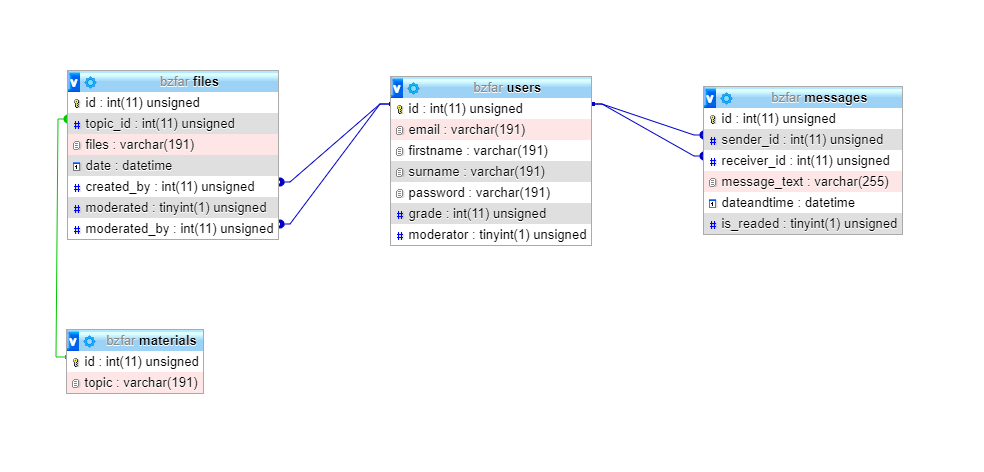


Таблица “messages”

The table with information of users



Таблица “users”



ERD (Entity Relationship Diagram)

## Programming

**Database connection**

**bd.php:**

<?php

// connecting redbeanphp ORM to with database

require 'assets/libs/rb.php';

// filling data about server to variables

$host = 'localhost';

$dbname = 'bzfar';

$username = 'root';

$password = 'root';

// setuping connection to database

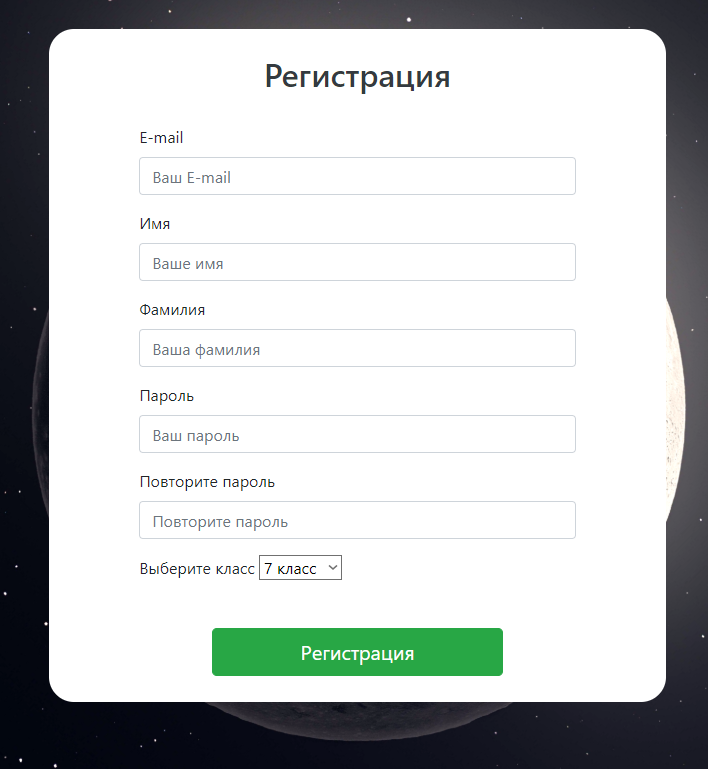
R::setup('mysql:host='.$host.';dbname='.$dbname,$username,$password);

// starting session to contain user data

session\_start();

?>

**Registration**

****

**Registration form**

**signup.php:**

<?php

// Connecting database

require "utils/bd.php";

// variable for user input

$data = $\_POST;

// array with errors

$errors = array();

// check if user trying to signup

if(isset($data['do\_signup'])){

// checking emptiness of email input

if(empty($data['email'])){

// fill errors variable with error

$errors[] = "Введите e-mail";

}

// checking emptiness of firstname input

if(empty($data['firstname'])){

// fill errors variable with error

$errors[] = "Введите имя";

}

// checking emptiness of surname input

if(empty($data['surname'])){

// fill errors variable with error

$errors[] = "Введите фамилию";

}

// checking emptiness of password input

if(empty($data['password1'])){

// fill errors variable with error

$errors[] = "Введите пароль";

}

// checking emptiness of password input

if(empty($data['password2'])){

// fill errors variable with error

$errors[] = "Введите повтор пароля";

}

// checking passwords aren't equal

if($data['password1'] != $data['password2']){

// fill errors variable with error

$errors[] = "Пароли не совпадают";

}

// checking emptiness of grade input

if(empty($data['grade'])){

// fill errors variable with error

$errors[] = "Выберите класс";

}

// checking another user with this email

$another = R::findOne("users","email = ?",array($data['email']));

if($another){

// fill errors variable with error

$errors[] = "Пользователь с таким e-mail зарегистрирован";

}

// checking no errors

if(empty($errors)){

// registration process

// creating new record

$user = R::dispense("users");

// filling the attributes

$user->email = $data['email'];

$user->firstname = $data['firstname'];

$user->surname = $data['surname'];

$user->password = password\_hash($data['password1'],PASSWORD\_DEFAULT);

$user->grade = $data['grade'];

$user->moderator = false;

// storing the record

R::store($user);

$\_SESSION['user\_info'] = $user;

// redirect to main page

header('Location: /index.php');

die();

}

}

?>

**Code of form:**

<form class="col-10 col-md-6 col-lg-4 bg-white py-2 formsignup" method="post">

<?php

// cheking if there's errors

if(!empty($errors)) { ?>

<div style="color: red; font-size:24px;" class="col-12 text-center">

<?php

// showing errors one by one

echo array\_shift($errors);

?>

</div>

<?php

}

?>

<div class="form-group">

<h2 class="text-center text-dark py-3">Регистрация</h2>

</div>

<div class="form-group w-75 mx-auto">

<label for="formGroupExampleInput">E-mail</label>

<input name="email" type="email" class="form-control" placeholder="Ваш E-mail">

</div>

<div class="form-group w-75 mx-auto">

<label for="formGroupExampleInput">Имя</label>

<input name="firstname" type="text" class="form-control" placeholder="Ваше имя">

</div>

<div class="form-group w-75 mx-auto">

<label for="formGroupExampleInput">Фамилия</label>

<input name="surname" type="text" class="form-control" placeholder="Ваша фамилия">

</div>

<div class="form-group w-75 mx-auto">

<label for="formGroupExampleInput">Пароль</label>

<input name="password1" type="password" class="form-control" placeholder="Ваш пароль" minlength="8">

</div>

<div class="form-group w-75 mx-auto">

<label for="formGroupExampleInput">Повторите пароль</label>

<input name="password2" type="password" class="form-control" placeholder="Повторите пароль">

</div>

<div class="form-group w-75 mx-auto">

<label for="formGroupExampleInput">Выберите класс</label>

<select name="grade">

<option value="7">7 класс</option>

<option value="8">8 класс</option>

<option value="9">9 класс</option>

<option value="10">10 класс</option>

<option value="11">11 класс</option>

<option value="12">12 класс</option>

</select>

</div>

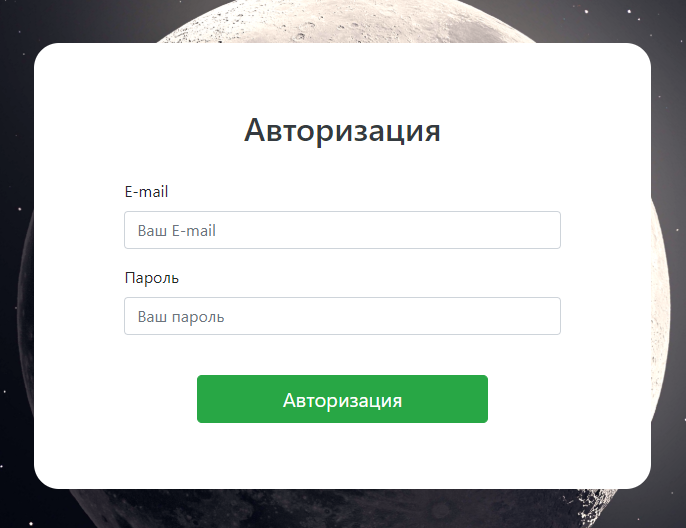
<div class="form-group pt-4">

<button class="btn btn-lg btn-block w-50 btn-success mx-auto" type="submit" name="do\_signup">Регистрация</button>

</div>

</form>

**Authorization**

****

**Form of authorization**

**signin.php:**

<?php

// connecting database

require "utils/bd.php";

// variable for user input

$data = $\_POST;

// array with errors

$errors = array();

// checking if user is trying to log in

if(isset($data['do\_login'])){

// checking emptiness of email input

if(empty($data['email'])){

// fill errors variable with error

$errors[] = "Введите e-mail";

}

// checking emptiness of password input

if(empty($data['password'])){

// fill errors variable with error

$errors[] = "Введите пароль";

}

// finding any user with inputted email

$user = R::findOne("users","email = ?",array($data['email']));

// checking if there is no user with this email

if(empty($user)){

// fill errors variable with error

$errors[] = "Пользователь с таким email не найден";

}

// checking if there is any user with this email

if($user){

// cheking password correction

if(!(password\_verify($data['password'],$user->password))){

// fill errors variable with error

$errors[] = "Пароль введен неправильно";

}

}

// checking if there is no errors

if(empty($errors)){

// finding info about user

$user = R::findOne("users","email = ?",array($data['email']));

// filling session variable with user data

$\_SESSION['user\_info'] = $user;

// redirecting to main page

header('Location: /index.php');

die();

}

}

?>

**Code of form:**

<form class="col-10 col-md-6 col-lg-4 bg-white py-5 formsignup" method="post">

<?php

// checking if there's errors

if(!empty($errors)) { ?>

<div style="color: red; font-size:24px;" class="col-12 text-center">

<?php

// showing errors one by one

echo array\_shift($errors);

?>

</div>

<?php

}

?>

<div class="form-group">

<h2 class="text-center text-dark py-3">Авторизация</h2>

</div>

<div class="form-group w-75 mx-auto">

<label for="formGroupExampleInput">E-mail</label>

<input name="email" type="email" class="form-control" placeholder="Ваш E-mail">

</div>

<div class="form-group w-75 mx-auto">

<label for="formGroupExampleInput">Пароль</label>

<input name="password" type="password" class="form-control" placeholder="Ваш пароль">

</div>

<div class="form-group pt-4">

<button class="btn btn-lg btn-block w-50 btn-success mx-auto" type="submit" name="do\_login">Авторизация</button>

</div>

</form>

**Logging out**

**logout.php:**

<?php

// connecting database

require "utils/bd.php";

// destroying the session with user data

session\_destroy();

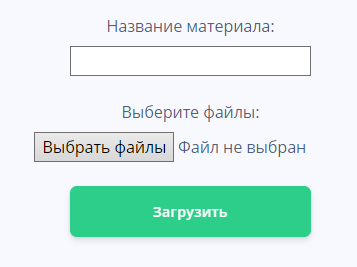
// redirecting to main page

header("Location: /index.php");

die();

?>

**Upload of materials**

****

**Form to upload materials**

**upload.php:**

<?php

// connecting database

require "utils/bd.php";

// declaring variables

$data = $\_POST;

$errors = array();

$success = array();

// checking if user pressed the button

if(isset($data['do\_upload'])){

// checking emptiness of topic name

if(empty(trim($data['material\_topic']))){

$errors[] = "Не введено название материала";

}

// checking if there's no errors

if(empty($errors)){

// variables to upload materials

$total = count($\_FILES["materials"]["name"]);

$target\_dir = "uploads/";

$topic = R::findOne('materials', 'topic = ?', array($data['material\_topic']));

// checking if topic with same name exists

if($topic){

$materials = $topic;

} else{

$materials = R::dispense("materials");

$materials->topic = $data["material\_topic"];

R::store($materials);

}

// loop to upload several materials

for ($j=0; $j < $total; $j++){

// variable to contain the path of file

$target\_file = $target\_dir . basename($\_FILES["materials"]["name"][$j]);

// variable to check is everything ok

$status = 1;

// check if file already exists

if (file\_exists($target\_file)) {

$errors[] = "Файл с именем " . $\_FILES["materials"]["name"][$j] . " уже существует";

$status = 0;

}

// check is there no errors

if ($status == 1) {

// checking is everything with the upload of file

if (move\_uploaded\_file($\_FILES["materials"]["tmp\_name"][$j], $target\_file)) {

// creating new record in files table

$files = R::dispense("files");

$files->topicId = $materials->id;

$files->files = $target\_file;

$files->date = date("Y-m-d H:i:s");

$files->created\_by = $\_SESSION['user\_info']->id;

// checking if user is moderator

if($\_SESSION['user\_info']->moderator){

$files->moderated = true;

$files->moderated\_by = $\_SESSION['user\_info']->id;

} else {

$files->moderated = false;

}

// storing record

R::store($files);

// filling array with success message

$success[] = "Файл ". basename( $\_FILES["materials"]["name"][$j]). " загружен";

} else {

// filling array with error message

$errors[] = "Извините, произошла ошибка при загрузке вашего файла";

}

}

}

}

}

?>

-Название операции

-Скриншот формы, которая обрабатывается (если это возможно)

-Имя файла обработчика (например, testreg.php)

-Листинг файла обработчика с отступами структуры и комментариями.

Отступы используйте, чтобы показать структуры ветвления и цикла.

Комментарии вставляются через // **до** фрагмента, который они комментируют.

## Testing

(Сделайте предисловие, почему вы собираетесь проводить тестирование.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Purpose** | **Description** | **Test data** | **Expected result** | **Actual result** |
| **1.1** | **To check the registration process** | **All fields are empty** | **email=””**  **name=””**  **surname=””**  **password=””**  **password1=””**  **grade=”7”** | **Output:  “Введите e-mail”** | **Output:  “Введите e-mail”** |
| **Evidences:**    **1.1** | | | | | |
| **1.2** | **To check the registration process** | **Input valid data, but user with this email is already registered** | **email=”**[**fazil.el2003@gmail.com**](mailto:fazil.el2003@gmail.com)**”**  **name=”Yelaman”**  **surname=”Fazyl”**  **password=”elaman10082003”**  **password1=”elaman10082003”**  **grade=”12”** | **Output:**  **”Пользователь с таким e-mail зарегистрирован”** | **Output:**  **”Пользователь с таким e-mail зарегистрирован”** |
| **Evidences:**    **1.2** | | | | | |
| **1.3** | **To check the registration process** | **Input valid data** | **email=”**[**bzfar07@gmail.com**](mailto:bzfar07@gmail.com)**”**  **name=”Boris”**  **surname=”Zelenov”**  **password=”bzfar078”**  **password1=”bzfar078”**  **grade=”12”** | **Redirected to main page and user appeared on database and hyperlink to dashboard is available** | **Redirected to main page and user appeared on database and hyperlink to dashboard is available** |
| **Evidences:**      **1.3** | | | | | |
| **2.1** | **To check the validation in registration process** | **Input valid data, but email is written without “@” symbol** | **email=”**[**bzfar07gmail.com**](mailto:bzfar07@gmail.com)**”**  **name=”Boris”**  **surname=”Zelenov”**  **password=”bzfar078”**  **password1=”bzfar078”**  **grade=”12”** | **Output: “Адрес электронный почты должен содержать символ “@” . В адресе “**[**bzfar07gmail.com**](mailto:bzfar07@gmail.com)**” отсутствует символ “@”.”** | **Output: “Адрес электронный почты должен содержать символ “@” . В адресе “**[**bzfar07gmail.com**](mailto:bzfar07@gmail.com)**” отсутствует символ “@”.”** |
| **Evidences:**    **2.1** | | | | | |
| **2.2** | **To check the validation in registration process** | **Input valid data, but password data is invalid** | **email=”**[**bzfar07@gmail.com**](mailto:bzfar07@gmail.com)**”**  **name=”Boris”**  **surname=”Zelenov”**  **password=”bzfar”**  **password1=”bzfar”**  **grade=”12”** | **Output:“Минимально доступное количество символов: 8. Длина текста сейчас: 5.”** | **Output:“Минимально доступное количество символов: 8. Длина текста сейчас: 5.”** |
| **Evidences:**      **2.2** | | | | | |
| **2.3** | **To check the validation in registration process** | **Input valid data, but password data is extreme** | **email=”**[**bzfar07@gmail.com**](mailto:bzfar07@gmail.com)**”**  **name=”Boris”**  **surname=”Zelenov”**  **password=”bzfar078”**  **password1=”bzfar078”**  **grade=”12”** | **User is registered and user appeared in database** | **User is registered and user appeared in database** |
| **Evidences:**      **2.3** | | | | | |
| **2.4** | **To check the validation in registration process** | **Input valid data, but password data is valid** | **email=”**[**sultanidcze@gmail.com**](mailto:sultanidcze@gmail.com)**”**  **name=”Sultan”**  **surname=”Ishangaliyev”**  **password=”sultan120303”**  **password1=”sultan120303”**  **grade=”12”** | **User is registered and user appeared in database** | **User is registered and user appeared in database** |
| **Evidences:**      **2.4** | | | | | |
| **3.1** | **To check the authorization process** | **All fields are empty** | **email=””**  **password=””** | **Output: “Введите e-mail”** | **Output: “Введите e-mail”** |
| **Evidences:**      **3.1** | | | | | |
| **3.2** | **To check the authorization process** | **Input valid data, but user with entered email is not registered** | **email= “**[**sultanidcze123@gmail.com**](mailto:sultanidcze@gmail.com)**”**  **password= “sultan120303”** | **Output: “Пользователь с таким email не найден”** | **Output: “Пользователь с таким email не найден”** |
| **Evidences:**      **3.2** | | | | | |
| **3.3** | **To check the authorization process** | **Input valid data, but password of user is not correct** | **email= “**[**fazil.el2003@gmail.com**](mailto:fazil.el2003@gmail.com)**”**  **password= “sultan120303”** | **Output: “Пароль введен неправильно”** | **Output: “Пароль введен неправильно”** |
| **Evidences:**      **3.3** | | | | | |
| **3.4** | **To check the authorization process** | **Input valid data** | **email= “**[**fazil.el2003@gmail.com**](mailto:fazil.el2003@gmail.com)**”**  **password= “elaman10082003”** | **Redirected to main page and hyperlink to dashboard is available** | **Redirected to main page and hyperlink to dashboard is available** |
| **Evidences:**    **3.4** | | | | | |
| **4.1** | **To check the chatting system** | **User opened the chat with another user** | **contacter\_id = “3”** | **Opened the chat with user and shows messages written before** | **Opened the chat with user and shows messages written before** |
| **Evidences:**      **4.1** | | | | | |
| **4.2** | **To check the chatting system** | **Write new message** | **message\_text = “Привет”** | **New record in database created and message is shown in dialogue** | **New record in database created and message is shown in dialogue** |
| **Evidences:**        **4.2** | | | | | |
| **4.3** | **To check the chatting system** | **Unread message** | **is\_readed = 0** | **written “unread” before name of contacter** | **written “unread” before name of contacter** |
| **Evidences:**      **4.3** | | | | | |
| **5.1** | **To check upload of materials** | **All fields are empty** | **material\_topic = “”**  **materials[] = “”** | **Output: “Не введено название материала”** | **Output: “Не введено название материала”** |
| **Evidences:**      **5.1** | | | | | |
| **5.2** | **To check upload of materials** | **Input valid data, but the material with same name is already** | **material\_topic = “AI”**  **materials[] = “Deep Learning.pptx”** | **The file is uploaded and appeared on server and table “files”, but it refers to topic\_id of AI that was created before**  **Output: “Файл Deep Learning.pptx загружен”** | **The file is uploaded and appeared on server and table “files”, but it refers to topic\_id of AI that was created before**  **Output: “Файл Deep Learning.pptx загружен”** |
| **Evidences:**          **5.2** | | | | | |
| **5.3** | **To check upload of materials** | **Input valid data, but file with this name is already uploaded** | **material\_topic = “Deep Learning”**  **materials[] = “Deep Learning.pptx”** | **Output: “Файл с именем Deep learning.pptx уже существует”** | **Output: “Файл с именем Deep learning.pptx уже существует”** |
| **Evidences:**      **5.3** | | | | | |
| **5.4** | **To check upload of materials** | **Input valid data** | **material\_topic = “Machine learning”**  **materials[] = “Machine learning.pptx”** | **Output: “Файл Machine learning.pptx загружен” and file appeared on “files” table and uploaded to server and material appeared in “materials” table** | **Output: “Файл Machine learning.pptx загружен” and file appeared on “files” table and uploaded to server and material appeared in “materials” table** |
| **Evidences:**          **5.4** | | | | | |
| **6.1** | **To check the moderation system of materials uploaded by non-moderaotrs** | **If user is admin, show all unmoderated materials** | **email=“**[**fazil.el2003@gmail.com**](mailto:fazil.el2003@gmail.com)**”**  **password = “elaman10082003”** | **All unmoderated materials are shown** | **All unmoderated materials are shown** |
| **Evidences:**      **6.1** | | | | | |
| **6.2** | **To check the moderation system of materials uploaded by non-moderaotrs** | **Moderator accepted suggested material** | **clicked button “Одобрить”** | **In the field moderated appeared true and in the field moderated\_by appeared the id of moderator** | **In the field moderated appeared true and in the field moderated\_by appeared the id of moderator** |
| **Evidences:**      **6.2** | | | | | |
| **6.3** | **To check the moderation system of materials uploaded by non-moderaotrs** | **Moderator declined suggested material** | **clicked button “Отказать”** | **File on server is deleted and file disappeared from database** | **File on server is deleted and file disappeared from database** |
| **Evidences:**            **6.3** | | | | | |
| **7.1** | **To check the search system of materials** | **Search field is empty** | **search\_text = “”** | **Output: “Введите запрос для поиска”** | **Output: “Введите запрос для поиска”** |
| **Evidences:**      **7.1** | | | | | |
| **7.2** | **To check the search system of materials** | **Search materials that doesn’t exist** | **search\_text = “Gravity”** | **Output: “Тема не была найдена”** | **Output: “Тема не была найдена”** |
| **Evidences:**      **7.2** | | | | | |
| **7.3** | **To check the search system of materials** | **Search materials that exists** | **search\_text = “AI”** | **Output all materials found** | **Output all materials found** |
| **Evidences:**      **7.3** | | | | | |
| **7.4** | **To check the search system of materials** | **Delete found material if user is moderator** | **search\_text = “AI”**  **clicked button “Удалить”** | **The file is deleted from the server and deleted from database** | **The file is deleted from the server and deleted from database** |
| **Evidences:**          **7.4** | | | | | |
| **8.1** | **To check the system of giving and removing permission of moderator** | **Giving moderator rights** | **clicked button “Сделать модератором”** | **User becomes moderator and field “moderator” of the user becomes true**  **Output: “Successful”** | **User becomes moderator and field “moderator” of the user becomes true**  **Output: “Successful”** |
| **Evidences:**          **8.1** | | | | | |
| **8.2** | **To check the system of giving and removing permission of moderator** | **Removing moderator rights** | **clicked button “Убрать модераторство”** | **User is not moderator anymore and field “moderator” of the user becomes false**  **Output: “Successful”** | **User is not moderator anymore and field “moderator” of the user becomes false**  **Output: “Successful”** |
| **Evidences:**          **8.2** | | | | | |
| **9.1** | **To check if the user is logged to use the site** | **User is not logged in** | **email = “”**  **password = “”** | **Output: “Пожалуйста, авторизуйтесь или зарегистрируйтесь”** | **Output: “Пожалуйста, авторизуйтесь или зарегистрируйтесь”** |
| **Evidences:**    **9.1** | | | | | |
| **9.2** | **To check if the user is logged to use the site** | **User is logged but trying to access moderator page without moderator rights** | **email = “**[**sultanidcze@gmail.com**](mailto:sultanidcze@gmail.com)**”**  **password = “sultan120303”** | **Output: “Пожалуйста, авторизуйтесь или зарегистрируйтесь”** | **Output: “Пожалуйста, авторизуйтесь или зарегистрируйтесь”** |
| **Evidences:**          **9.2** | | | | | |
| **9.3** | **To check if the user is logged to use the site** | **User is logged and can access page** | **email= “**[**fazil.el2003@gmail.com**](mailto:fazil.el2003@gmail.com)**”**  **password = “elaman10082003”** | **User can access the page and use it** | **User can access the page and use it** |
| **Evidences:**      **9.3** | | | | | |

## Installation

Plan:

1. Choose installation date
2. Choose method of installation
3. Choose time of studying
4. Choose method of studying
5. Approve everything with customer
6. Send customer software, so he can test it
7. Take feedback from customer about the software

**Installation date:** November, 29

Reason: One week after teaching, new system will be installed, because it’s the day when client is available for installing the system

**Method:** Parallel conversion

Reason: Old data could be used by users, and it won’t be lost. In addition, users will be adapting to use the new system and workflow won’t become slow. Furthermore, some problems could be found while using the new system without risk of losing data.

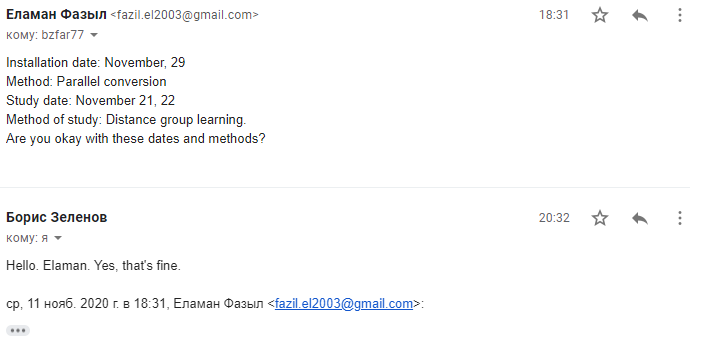
**Training plan:**

Study date: November 21, 22

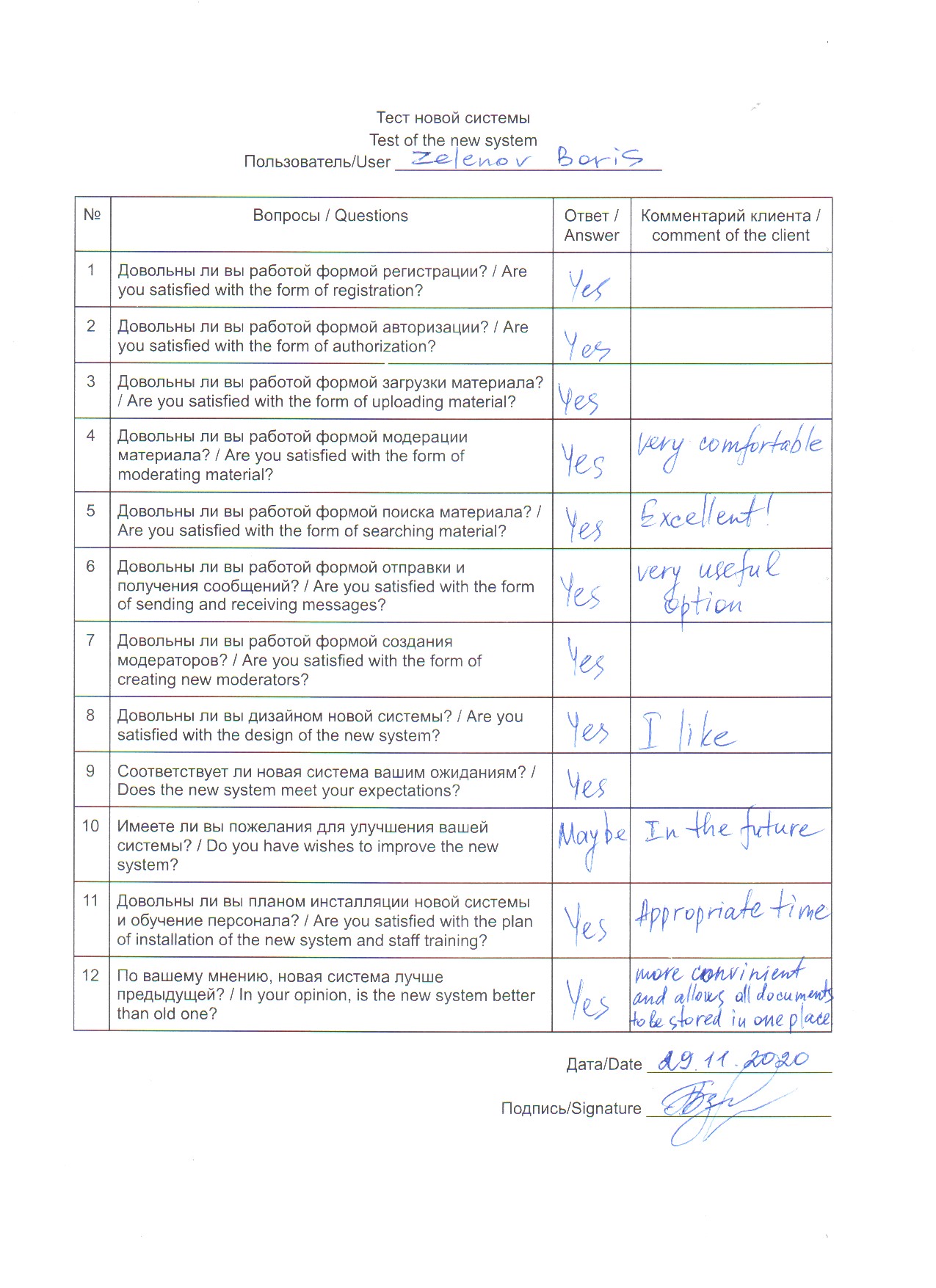
Reason: Most of the users are students and most of them have free time only on weekends, so they could spend some time on learning how to use the new system. Besides, customer and I have free time on these dates

Method of study: Distance group learning. It could be done by using Microsoft Teams platform, because most of software users have accounts on this service and due to pandemic situation in the country, it will be more safe to teach without social contact

**The evidence of user agreement:**

****

**The evidence of user tested the software:**

****

# Documentation

## User guide

2) Content (Заголовки основных действий, которые может выполнить пользователь в системе)

3) A guide on how to use the main function of the system (Пошаговая инструкция как выполнять каждое действие, сопровождающаяся скриншотами)

4) A troubleshooting section for possible errors and how to fix them (Руководство по ошибкам пользователя)

Common error guide

|  |  |  |
| --- | --- | --- |
| Error | Cause | Solution |
| Not all the fields have been filled! Try again! | During the entering the system or registration process not all required fields were filled in. | Fill in all the fields and repeat the operation. |
| Sorry, the login doesn't exist! | The login is incorrect (doesn’t exist in the data base). | Enter the existent login (which was entered on registration) or if you don’t have a login please, follow the registration form. |

5) **Back up routines** (Создайте пошаговую инструкцию резервного копирования данных для пользователя)

**6) Glossary** (Создайте список терминов и определений.

!!!Термины берем только из руководства пользователя)

**Back up** -  the process ….

 7) **Index** (Создайте предметный указатель средствами Word.

!!!Термины берем только из руководства пользователя)

Admin, 79

Log in, 78

….

Evaluation

## Discussion of the degree of success in meeting the original objectives

Еще раз перечислите цели, которые вы поставили перед разработкой проекта. Используйте подводку (например, Для реализации своего проекта перед разработкой я поставил следующие цели…)

|  |  |  |
| --- | --- | --- |
| № | Evaluation | Evidence |
| 1 | *Для каждой цели оцените степень ее выполнения. Что удалось выполнить? Какие сложности возникли при достижении этой цели? Как удалось их преодолеть?*  *Если цель не достигнута, то опишите почему не достигнута? что требуется для того, чтобы ее достичь в будущем?* | *Укажите раздел, где есть доказательства того, что цель достигнута и страницу*  Например, Testing  Page 56 |
| 2 |  |  |

## Evaluate the client's and user's response to the system

Получите достоверную обратную связь от клиента по итогам разработки проекта. Обратная связь должна быть подробная. Можно предложить клиенту набор вопросов, на которые он должен ответить.

Далее проведите анализ обратной связи клиента, подведите summary проекта.