

Лабораторная работа №1

Основное задание

Спроектировать и разработать систему авторизации пользователей на протоколе HTTP. Система должна обладать следующим функционалом:

Функциональность входа и выхода

Пароли должны храниться в хешированном виде

1. Описание пользовательских сценариев работы

На сайте пользователю доступны следующие возможности:

Регистрация

Авторизация

При вводе неверных данных, пользователю выводится сообщение об ошибке.

После авторизации пользователю открываются следующие возможности:

1. Выход из аккаунта

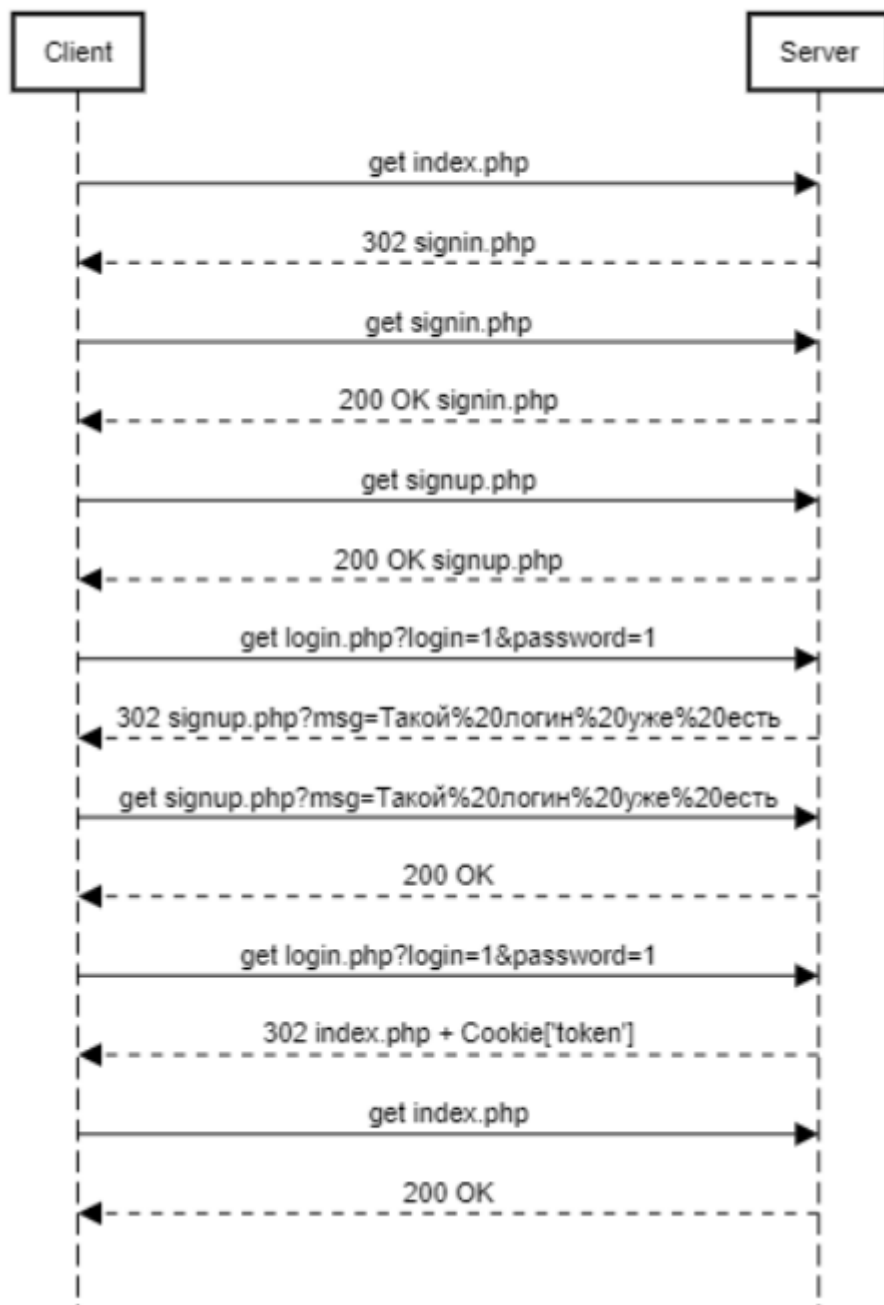
Если пользователь был ранее авторизован на сайте, у него есть cookie с токеном, и токен совпадает с токеном в БД, то при обновлении страницы, закрытия вкладки, закрытия браузера, он останется на главной странице.

После того как срок годности cookie закончится, при обновлении главной страницы, будет произведен автоматический выход из аккаунта. Пользователь будет перенаправлен на страницу входа.

2. Описание API сервера и хореографии

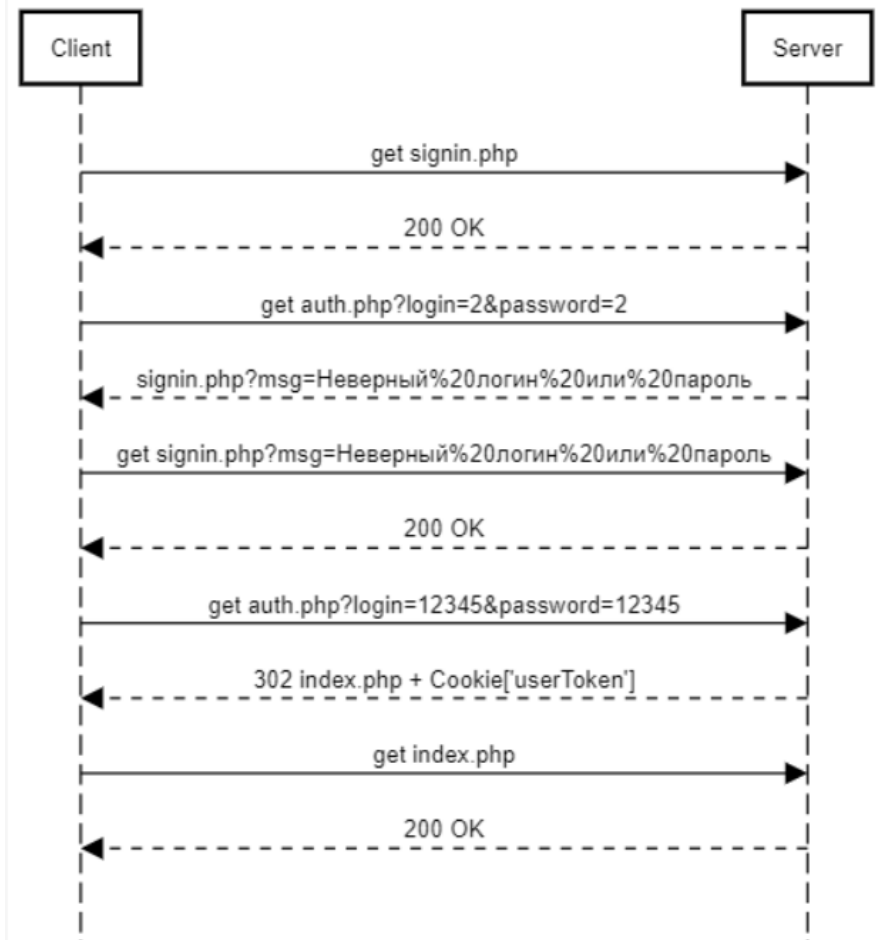
- Пример запросов, когда пользователь впервые заходит на страницу index.php, а после регистрируется:

Регистрация



Пример запросов, когда пользователь заходит на страницу авторизации, вводит неверное имя пользователя, а после вводит верные данные и входит в аккаунт:

Вход



3. Описание структуры базы данных

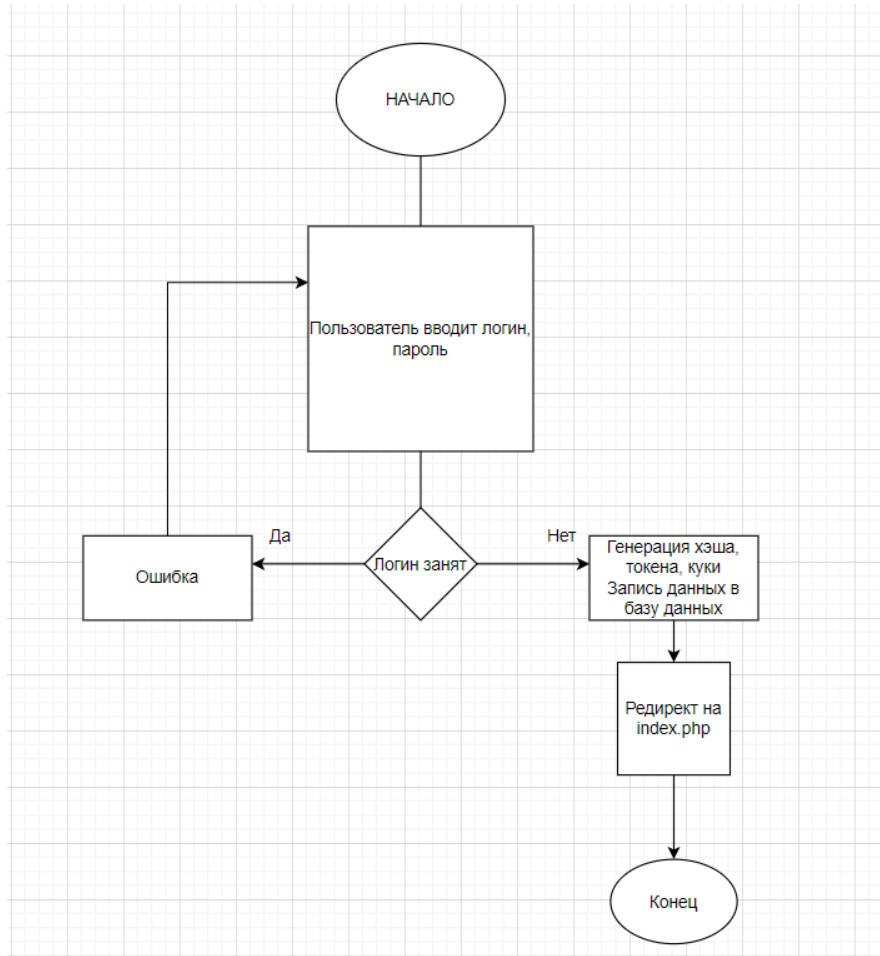
Для хранения данных пользователей используется JSON файл. Каждый объект содержит в себе `login` пользователя, хэш пароля и токен.

Пример данных пользователя в базе данных:

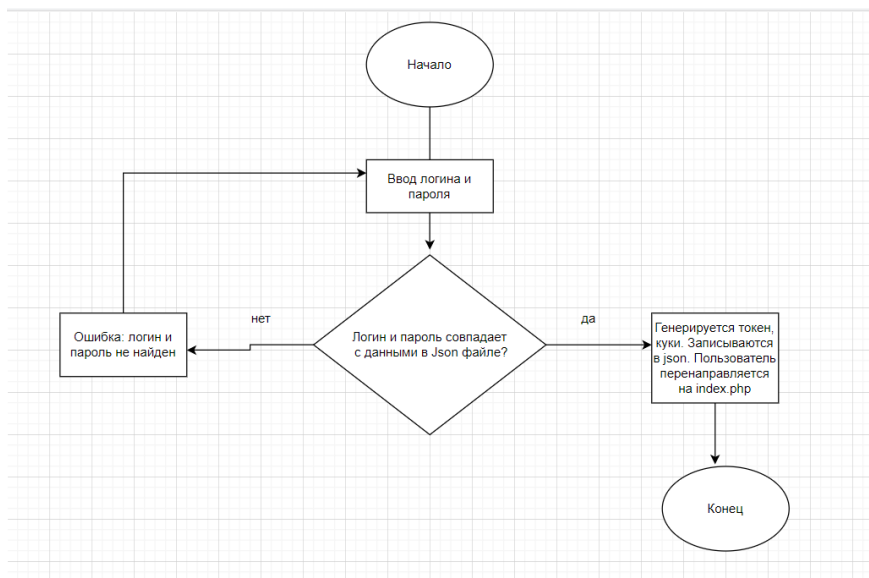
```
[{"login": "1", "password": "$2y$10$TcqLKccC1P5.pkfyZXSpme2fc/MCSItxc1sv.unwvOhW68NPH3XyW", "token": "2fce80e291a45781bd5f"}]
```

4. Описание алгоритмов

- Алгоритм страницы регистрации



- Алгоритм страницы авторизации:



Значимые фрагменты кода

Фрагмент кода проверки данных, полученных из формы регистрации, и запись их в БД:

```

if (empty($taskList)) {
header('Location: /index.php');
}
else{

```

```

foreach ($taskList as $value){
if(($login == $value->login)){
header('Location: /signup.php?msg=Такой логин уже есть');
exit();
}
}
}
$token = bin2hex(random_bytes(10));
$file = file_get_contents('db.json');
$taskList = json_decode($file,TRUE);
unset($file);
$taskList[] = array('login'=>$login, 'password'=>$hash, 'token'=>$token);
file_put_contents('db.json',json_encode($taskList));
unset($taskList);
setcookie("token", $token, time() + 30);
header('Location: /index.php');
exit();

```

Фрагмент кода проверки введенных данных, обновления токена при авторизации:

```

foreach ($taskList as $value){
if(($login == $value->login && password_verify($password, $value->password))){
if (!isset($_COOKIE['token'])){
foreach ($taskList as $value){
if(($login == $value->login)){
$token = bin2hex(random_bytes(10));
setcookie("token", $token, time() + 30);
$value->token = $token;
file_put_contents('db.json',json_encode($taskList));
}
}
}
header('Location: /index.php');
exit();
}
}
}

```