Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

«ИЖЕВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИМ. М.Т. КАЛАШНИКОВА»

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение»

Отчет по лабораторной работе №1, 2, 3

по дисциплине «Управление программными проектами»

на тему: РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ ДЛЯ ПРОГРАММНОЙ СИСТЕМЫ

|  |  |
| --- | --- |
| Выполнили:  студенты группы Б07-191-1        Принял: | Ю.А. Поздеева                                        Д.О. Гамберов                                                М.О. Еланцев |

Ижевск

2015

1.     ПОСТАНОВКА ЗАДАЧИ

Составить требования к интеллектуальной библиотеке.

2.     ТРЕБОВАНИЯ ДЛЯ БИБЛИОТЕКИ

1. Система регистрации (логин, пароль, подтверждение через электронную почту) с возможностью привязки аккаунта в социальных сетях (вконтакте, facebook)
2. Пользователь выбирает из списка интересные ему жанры, на основе которых формируется список рекомендаций
3. Поиск книг по названию, автору, жанру, поиск авторов
4. Раздел Favorite – список любимых книг, подписка на авторов
5. Формирование списка прочитанных пользователем книг (пользователь может сам отметить книгу, которую он прочел)
6. Раздел "Цитаты", куда пользователь может добавить любимые высказывания из книг.
7. Раздел "Задай вопрос любимому автору" (тем, кто зарегистрирован; отбираются интересные)
8. Система отзывов(оценка от 1 до 5), комментарии, на основе которых создаются различные рейтинги(топы по жанрам).
9. Диаграммы активности пользователя: общая(количество прочитанных книг), по жанрам, которые можно разместить в соцсетях
10. Wishlist, куда пользователь может сохранять книги, которые он хочет прочитать в будущем
11. Пользователи могут находить друзей по интересам

ЛАБОРАТОРНАЯ РАБОТА № 2

Модуль входа

1) Модуль регистрации

входные данные: логин, пароль, повторный пароль, почта

выходные данные: запись в базу данных или сообщение об ошибке

2) Модуль авторизации

входные данные: почта, пароль, запрос в соц.сеть

выходные данные: вход на аккаунт или сообщение об ошибке

Модуль библиотеки

1) Модуль поиска

а) Модуль поиска книги

входные данные: название, фамилия автора

выходные данные: список книг по совпадающему полю

б) Модуль поиска автора

входные данные: имя автора, фамилия автора

выходные данные: список авторов по совпадающему полю

2) Модуль книги

а) Модуль цитат

входные данные: текст, выбор соц.сети

выходные данные: запись в соц.сети

б) Модуль выбора книги

выходные данные: id\_книги

выходные данные: выбранная книга

в) Модуль добавления в любимое

входные данные: id\_книги, id\_аккаунта

выходные данные: добавление книги в личную библиотеку пользователя

г) Модуль добавления в прочитанное

входные данные: id\_книги, id\_аккаунта

выходные данные: добавление книги в личную библиотеку пользователя

д) Модуль оценки

входные данные: id\_аккаунта

выходные данные: оценка

е) Модуль комментариев

входные данные: id\_аккаунта

выходные данные: сообщение

3)Модуль автора:

а) Модуль добавления в любимое

входные данные: id\_автора, id\_аккаунта

выходные данные: добавление автора в личную библиотеку пользователя

б) Модуль выбора автора

выходные данные: id\_автора

выходные данные: список книг автора

в) Модуль вопроса автору

выходные данные: id\_автора

выходные данные: сообщение

Модуль аккаунта

1) Модуль выбора жанров

входные данные: id\_аккаунта

выходные данные: добавление жанра в личную библиотеку пользователя

2) Модуль активности

входные данные: id\_аккаунта, id\_книг

выходные данные: список книг

3) Модуль личной библиотеки

входные данные: id\_аккаунта

выходные данные: список любимых авторов, книг и жанров

4) Модуль рекомендаций

входные данные: id\_аккаунта

выходные данные: список рекомендаций

5) Модуль поиска друзей по интересам

входные данные: id\_аккаунта

выходные данные: список рекомендаций

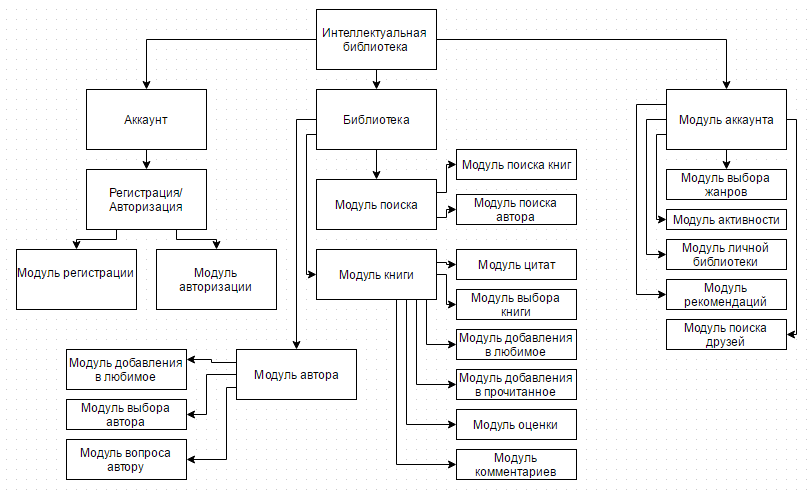
Структура книги:

id\_книги, id\_автора, ФИО автора, название, год публикации, краткое описание, отметка "Favorite"

Структура автора:

id\_автора, список книг с указанием id\_книг, ФИО автора, краткая биография, отметка "Favorite"

СХЕМА СИСТЕМЫ



Содержание

Перечень рисунков

Перечень таблиц

1.ВВЕДЕНИЕ

1.1 Назначение системы

1.2 Область применения системы

1.3 Определения, акронимы, аббревиатуры

1.4 Ссылки

1.5 Обзор системы

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

2.1 Системный контекст

2.2 Режимы и состояния системы

2.3 Основные функциональные возможности системы

2.4 Основные условия системы

2.5 Основные ограничения системы

2.6 Характеристики пользователя

2.7 Допущения и зависимости

2.8 Оперативные сценарии

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

3.1.2 Износостойкость

3.1.3 Адаптируемость

3.1.4 Условия окружающей среды

3.2 Рабочие характеристики системы

3.3 Безопасность системы

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

3.5.2 Ремонтопригодность системы

3.5.3 Надежность системы

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

4. ИНТЕРФЕЙСЫ СИСТЕМЫ

1. ВВЕДЕНИЕ

1.1 Назначение системы

Данное веб-приложение используется для ведения собственной библиотеки, для получения доступа к книгам из любого удобного для пользователя места, отслеживания публикаций определенных авторов. А так же возможна публикация собственных небольших произведений и обсуждение различных тем на форуме.

1.2 Область применения системы

Система «Интеллектуальная библиотека» должна создавать аккаунт пользователя, из которого будут доступны следующие функции:

* составление собственной библиотеки из имеющихся в базе книг
* подписка на публикации выбранных авторов
* поиск книг и авторов по различным критериям
* возможность задать вопрос автору
* нахождение друзей по интересам

1.3 Определения, акронимы, аббревиатуры

Пользователь. Конечный пользователь системы.

1.4 Обзор системы

Система имеет вид веб-приложения. Есть несколько режимов: режим аккаунта пользователя, режим обзора книги, режим обзора автора, режим обзора библиотеки.

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

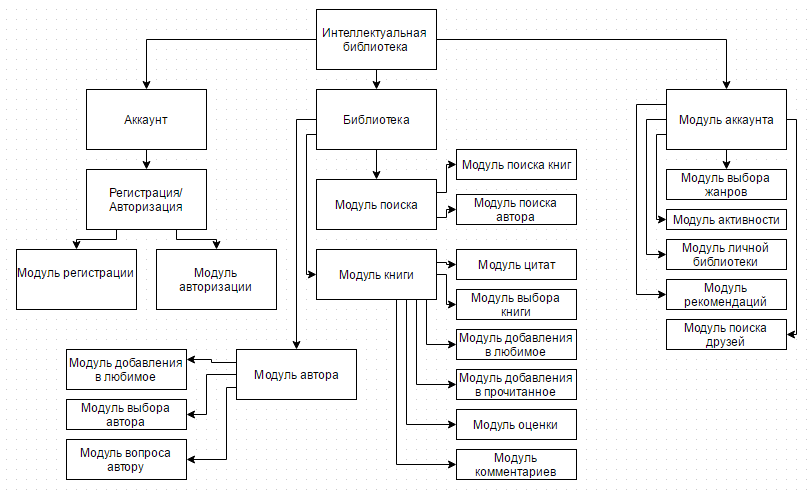
2.1 Системный контекст

2.2 Режимы и состояния системы

Система может находиться в следующих состояниях:

- состояние системы без аутентификации;

- состояние системы авторизованного пользователя;

2.3 Основные функциональные возможности системы

2.4 Основные условия системы

-Доступ в интернет

2.5 Основные ограничения системы

2.6 Характеристики пользователя

Пользователь. Конечный пользователь системы, количество может быть любое. Использует систему для поиска книг и составления личной библиотеки

2.7 Допущения и зависимости

2.8 Оперативные сценарии

Данная система подразумевает под собой следующий сценарий:

Пользователь-клиент получает доступ к сервису через браузер. Для доступа к сервису необходимо пройти первоначальную авторизацию.

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

Данная система располагается на хостинге, соответственно на предоставляемом им сервере.

3.1.2 Износостойкость

Данный параметр системы не зависит от нас, т.к. сервер арендован.

3.1.3 Адаптируемость

В случае увеличения нагрузки на сервер, количество арендованных серверов будет увеличено, а нагрузка равномерно распределится между ними.

3.1.4 Условия окружающей среды

Серверы не зависят от погодных условий.

3.2 Рабочие характеристики системы

Критическое количество запросов обрабатываемое системой равно 1000 запросов в час. Долговечность оборудования зависит от поставщика сервера. До превышения критического количества запросов в час пользователь не замечает нагрузки сервера.

3.3 Защита системы

Система должна быть защищена от sql - инъекций и xss - атак.

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

Требования по распределению функций для персонала описано в ТЗ.

3.5.2 Эксплуатационная технологичность

Техническая составляющая эксплуатационной технологичности зависит от арендодателя сервера. Программные изменения происходят в фоновом режиме и не вызывают задержек при данных.

3.5.3 Надежность системы

Техническая составляющая надежности системы зависит от арендодателя сервера.

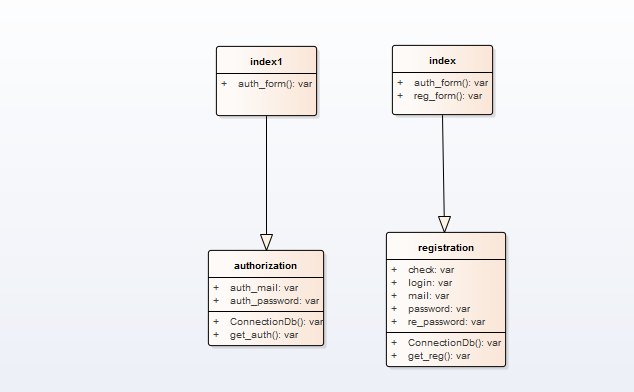
3.6 Стратегия и регулирование

Модерация и наполнение сайта производится разработчиками

3.7 Устойчивость жизненного цикла системы

Для устойчивости жизненного цикла системы используется обратная связь с пользователем.

UML-диаграмма



ИНТЕРФЕЙСЫ СИСТЕМЫ

index.php

<?

require\_once('reg.php');

class index extends registration

{

public function reg\_form(){

?>

<!DOCTYPE html>

<html lang="ru">

<head>

<link href="style.css" rel="stylesheet">

</head>

<body bgcolor="#FFEC8B">

<div id="second">

<p>&#1056;&#1077;&#1075;&#1080;&#1089;&#1090;&#1088;&#1072;&#1094;&#1080;&#1103;</p>

<form method="POST" >

<p>Login</p><input type="text" name="login"/><br />

<p>&#1055;&#1072;&#1088;&#1086;&#1083;&#1100;</p><input type="text" name="pass" /><br />

<p>&#1055;&#1086;&#1074;&#1090;&#1086;&#1088;&#1080;&#1090;&#1077; &#1087;&#1072;&#1088;&#1086;&#1083;&#1100;</p><input type="text" name="re\_pass" /><br />

<p>Email</p><input type="text" name="mail" /><br /><br />

<input type="submit" name="submit1" value="Send">

</form>

</div>

</body>

</html>

<?

}

public function auth\_form()

{

?>

<br />

<form method="POST">

<input type="text" name="re\_mail" />

<input type="text" name="re\_pass" /><br />

<input type="submit" name="submit" value="Log in">

</form>

<?

}

}

$r= new index();

$r->reg\_form();

$r->get\_reg();

?>

Index1.php

<?

require\_once('auth.php');

class index1 extends authorization

{

public function auth\_form(){

?>

<!DOCTYPE html>

<html lang="ru">

<head> <link href="style.css" rel="stylesheet">

</head>

<body bgcolor="#FFEC8B">

<div id="first">

<p>&#1040;&#1074;&#1090;&#1086;&#1088;&#1080;&#1079;&#1072;&#1094;&#1080;&#1103;</p>

<form method="POST">

<p>Email</p><input type="text" name="auth\_mail" />

<p>&#1055;&#1072;&#1088;&#1086;&#1083;&#1100;</p><input type="text" name="auth\_pass" /><br /><br />

<input type="submit" name="submit" value="Log in">

</form>

</div>

</body>

</html>

<?

}

}

$t = new index1();

$t->auth\_form();

$t->get\_auth();

?>

Auth.php

<?

/\*\*

\*

\*/

class authorization

{

public $auth\_mail;

public $auth\_password;

public function get\_auth()

{

$this->auth\_mail = $\_POST['auth\_mail'];

$this->auth\_password = $\_POST['auth\_pass'];

$this->ConnectionDb();

if(isset($\_POST['submit']))

{

$query = mysql\_query("SELECT \* FROM registration");

do

{

if($row['mail'] == $this->auth\_mail && $row['pass'] == $this->auth\_password && $this->auth\_mail != '' && $this->auth\_password != '')

{

echo "Success";

}

}while($row = mysql\_fetch\_array($query));

}

}

public function ConnectionDb() //ÐŸÐ¾Ð´ÐºÐ»ÑŽÑ‡ÐµÐ½Ð¸Ðµ Ðº Ð±Ð´

{

$host = 'localhost';

$user = 'u924420666\_upp1';

$pass = '4815162342';

$db = 'u924420666\_upp';

$connection = mysql\_connect($host,$user,$pass);

mysql\_select\_db($db,$connection);

mysql\_set\_charset('UTF-8',$connection);

if(!$connection || !mysql\_select\_db($db,$connection))

{

exit(mysql\_error());

}

}

}

Reg.php

<?

class registration

{

public $login;

public $password;

public $re\_password;

public $mail;

public $check;

public function ConnectionDb() //ÐŸÐ¾Ð´ÐºÐ»ÑŽÑ‡ÐµÐ½Ð¸Ðµ Ðº Ð±Ð´

{

$host = 'localhost';

$user = 'u924420666\_upp1';

$pass = "4815162342";

$db = "u924420666\_upp";

$connection = mysql\_connect($host,$user,$pass);

mysql\_select\_db($db,$connection);

mysql\_set\_charset('UTF-8',$connection);

if(!$connection || !mysql\_select\_db($db,$connection))

{

exit(mysql\_error());

}

}

public function get\_reg() //ÐŸÐ¾Ð»ÑƒÑ‡ÐµÐ½Ð¸Ðµ Ð´Ð°Ð½Ð½Ñ‹Ñ… Ñ Ñ„Ð¾Ñ€Ð¼Ñ‹ Ð¸ Ð·Ð°Ð¿Ð¸ÑÑŒ Ð² Ð±Ð´

{

if(isset($\_POST['submit1']))

{

$this->login=$\_POST['login'];

$this->password=$\_POST['pass'];

$this->mail=$\_POST['mail'];

$this->re\_password=$\_POST['re\_pass'];

$this->ConnectionDb(); //ÐšÐ¾Ð½Ð½ÐµÐºÑ‚Ð¸Ð¼ÑÑ Ðº Ð±Ð´

if($this->password == $this->re\_password && $this->login !='' && $this->password!='' && $this->re\_password!='' && $this->mail!='')

{

$query = mysql\_query("INSERT INTO registration(usr,pass,mail) VALUES ('$this->login', '$this->password','$this->mail')");

echo "Success";

}

else{

echo "Try again";

}

}

}

}

?>