# Министерство образования и науки Российской Федерации

# Федеральное государственное бюджетное образовательное учреждение

# высшего профессионального образования

# «Ижевский государственный технический университет имени М. Т. Калашникова»

# Кафедра «Программное обеспечение»

Отчет по лабораторной работе №3

по дисциплине: «Конструирование программного обеспечения»

на тему: «Система дистанционного обучения»

Выполнили

студенты гр. Б08-191-1 Д.О. Гамберов

Ю.А. Поздеева

Принял В.Г. Власов

Ижевск

2016

СОДЕРЖАНИЕ

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

5. КЛАССЫ КОДИРОВАНИЯ

5.1 UML схема.

5.2 Схема соответствий

1. ВВЕДЕНИЕ
   1. Назначение системы

Основное назначение системы – обеспечение автоматизированного

создания и проведения дистанционных учебных курсов с целью повышения вовлеченности студентов в учебный процесс.

* 1. Область применения системы

Система может применяться как в рамках государственного образования, так и репетиторами с целью обеспечения доставки образовательной информации учащимся посредством сети Интернет, что позволяет вовлечь в процесс обучения большее количество учащихся и сделать его более доступным. Система должна предоставлять необходимые инструменты для наглядного создания и редактирования курса, обеспечивать возможность прохождения курса без непосредственного участия учителя и осуществлять сбор статистики прохождения курсов и результатов тестирования учеников. В системе необходимо предусмотреть следующие группы пользователей: администратор, учитель, ученик.

* 1. Определения, акронимы, аббревиатуры

Администратор – пользователь, обладающий правами доступа ко всему функционалу сайта и правом добавления/удаления пользователей.

Учитель – пользователь, обладающий необходимыми правами для создания курса, его последующего редактирования и обладающий правами доступа к просмотру статистики по данному курсу и результатам тестирования Учеников.

Ученик – пользователь, занимающийся по курсу.

СДО – система дистанционного обучения.

* 1. Ссылки
  2. Обзор системы

Система делится на два главных модуля – сервер, обеспечивающий логику системы и пользовательский интерфейс, которые представляют собой отдельные независимые модули. Сервер в свою очередь делится на следующие модули: модуль аутентификации и авторизации, модуль сбора статистики, модуль взаимодействия с БД. Клиент делится на следующие модули: визуальный редактор курса, модуль прохождения курса, модуль просмотра статистики, модуль личного кабинета для просмотра созданных курсов и курсов, в которых пользователь принимал участие как ученик.

1. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ
   1. Системный контекст
   2. Режимы и состояния системы

Система может находиться в следующих состояниях:

1. Состояния сервера:

* Ожидание авторизации;
* Авторизация пользователя;
* Авторизация в панели администратора.

1. Состояния модуля работы с БД:

* ожидание приема данных;
* прием данных;
* сохранение данных в БД;
* загрузка данных из БД.

1. Состояния интерфейса

* ожидание команды пользователя;
* ввод команды пользователем.
  1. Основные функциональные возможности системы.



Рис.1. Классы анализа

* 1. Основные условия системы

Пользователи системы должны иметь постоянный доступ к интернету.

* 1. Основные ограничения системы

Так как в настоящий момент система размещается на бесплатном хостинге, существует ограничение по памяти – не более 2 Гб.

* 1. Характеристики пользователя

Ученик. Количество пользователей этого типа не ограничено. Использует систему для прохождения курсов.

Учитель. Количество пользователей этого типа не ограничено. Использует систему для создания курсов и проведения тестирования учащихся.

Администратор. Пользователь этого типа только один. Создается при создании системы.

2.7 Оперативные сценарии

Система предусматривает следующий сценарий. Администратор системы добавляет пользователя типа учитель. Учитель создает учебный курс посредством разработанного конструктора курсов, добавляет необходимые тесты и ответы к ним и отправляет данные учеников, которых он хочет добавить к курсу Администратору. Администратор добавляет пользователей типа Ученик по запросу Учителя и предоставляет им права, необходимые для прохождения курса. Ученик проходит курс и выполняет контрольные задания, результаты сохраняются в БД. Учитель может просмотреть результаты тестирования ученика и оценить творческие работы, не предполагающие возможность автоматической проверки. Ученик может просмотреть свою статистику прохождения.

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

3.1 Физические

3.1.1 Конструкция

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 распределена между ними.

3.1.4 Условия окружающей среды

Шанс выхода сервера из строя из-за условий окружающей среды существует, но он невелик.

3.2 Рабочие характеристики системы

Критическое количество запросов, обрабатываемых системой, зависит от арендованного сервера. До превышения критического количества запросов в час пользователь не замечает нагрузку оборудования.

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 обработке команд.

3.5.3 Надежность системы

Система защищена от sql - инъекций и xss - атак.

3.6 Стратегия и регулирование

Базовые настройки установленной системы регулируются при установке системы администратором, в зависимости от пожеланий владельца.

3.7 Устойчивость жизненного цикла системы

Для устойчивости жизненного цикла системы используется обратная связь с пользователем.

4. СИСТЕМНЫЕ ИНТЕРФЕЙСЫ

# C:\Users\Юлия\Desktop\ДИПЛОМ\KП.png

1. Интерфейс модуля Администратора

interface iAdminOffice

{

function ShowAuth();

function AddNewUser();

function DeleteUser();

function ShowUsers();

}

2. Интерфейс модуля Авторизации

interface iAuthorization

{

function CheckAuth();

function MakeAuth();

}

3. Интерфейс модуля работы с БД

interface iDB

{

function Connect($param);

function Query($query);

function Die(\Exception $e);

}

5. Интерфейс модуля Учитель

interface iTeacherOffice

{

function CourseList ();

function NewCourse ();

function StudentsList ();

function TasksCheck ();

function ChangePassword ();

}

6. Интерфейс модуля Ученик

interface iStudentOffice

{

static function doHomeWork ();

function Statistic ();

function Course ();

function ChangePassword ();

}

7. Интерфейс модуля Пользователь

interface iUsersGui

{

function GetPermission (DB $db);

}

8. Интерфейс модуля добавления статьи

interface iArticle

{

function createArticle ();

function uploadArticle ();

function editArticle ();

}

8. Интерфейс модуля оценивания заданий

interface iGiveMark

{

function giveMark ();

}

8. Интерфейс модуля добавления тестов

interface iTestConstruct

{

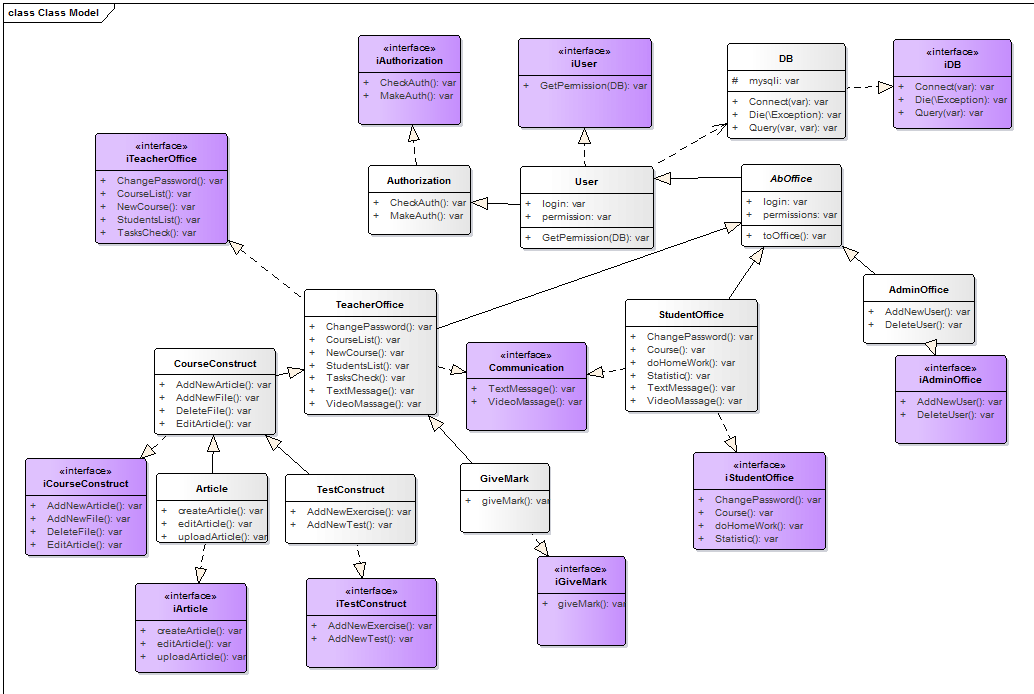
function AddNewTest ();

function AddNewExercise ();

}

5. КЛАССЫ КОДИРОВАНИЯ

5.1 UML схема программы



Классы кодирования

5.2 Классы кодирования программы соответствуют по функциональному назначению интерфейсам, по которым построены классы кодирования (см. п.4 Системные интерфейсы). Для системы так же разработаны следующие классы представления данных:

1. Класс AdminOffice

class AdminOffice extends AbOffice implements iAdminOffice

{

function DeleteUser()

{

}

function AddNewUser()

{

}

}

2. Класс DB

class DB implements iDB

{

protected $mysqli;

function Connect($param)

{

$param = func\_get\_args();

$this->mysqli = new mysqli($param['host'], $param['user'], $param['pass'], $param['world']);

if ($this->mysqli->connect\_errno) {

printf("Не удалось подключиться: %s\n", $this->mysqli->connect\_error);

exit();

}

}

function Query($query, $row = null)

{

$res = $this->mysqli->query($query);

return $row === null ? $res : mysqli\_fetch\_array($res);

}

function Die(\Exception $e)

{

die($e->getMessage() . ' at line: ' . $e->getLine());

}}

3. Класс AbOffice

abstract class AbOffice extends User

{

public $login,$permissions;

public function toOffice()

{

if($this->permissions=="student")

{

$usr=new StudentOffice();

}

if($this->permissions=="teacher")

{

$usr=new TeacherOffice();

}

if($this->permissions=="admin")

{

$usr=new AdminOffice();

}

}

}

4. Класс User

class User extends Authorization implements iUser

{

public $login,$permission;

function GetPermission(DB $db)

{

$db->Connect(1,2,3,4);

}

}

5. Класс Article

class Article extends CourseConstruct implements iArticle

{

function createArticle(){}

function uploadArticle(){}

function editArticle(){}

}

6. Класс Authorization

class Authorization implements iAuthorization

{

function CheckAuth() {

$db = new DB();

$db->Connect('param');

$pass = $db->Query("select pass from $this->type where name=$this->name", 1);

if(md5($this->pass) === $pass) :

$this->MakeAuth();

else :

return false;

endif;

}

function MakeAuth() {

header('Location: ');

}

}

7. Класс CourseConstruct

class CourseConstruct extends TeacherOffice implements iCourseConstruct

{

function AddNewFile()

{}

function AddNewArticle()

{}

function EditArticle()

{}

function DeleteFile()

{}

}

8. Класс GiveMark

class GiveMark extends TeacherOffice implements iGiveMark

{

function giveMark( )

{

}

}

9. Класс StudentOffice

class StudentOffice extends AbOffice implements Communication,iStudentOffice

{

function doHomeWork()

{

}

function Statistic()

{}

function Course()

{}

function VideoMassage()

{}

function TextMessage()

{}

function ChangePassword()

{}

}

10. Класс TeacherOffice

class TeacherOffice extends AbOffice implements Communication, iTeacherOffice

{

function TextMessage()

{}

function VideoMassage()

{}

function CourseList()

{}

function NewCourse()

{

$constract1 = new CourseConstruct();

$constract2 = new TestConstruct();

}

function StudentsList()

{}

function TasksCheck()

{

function ShowResults()

{}

$marktasks = new GiveMark();

}

function ChangePassword()

{}

}

11. Класс TestConstruct

class TestConstruct extends CourseConstruct implements iTestConstruct

{

function AddNewTest()

{}

function AddNewExercise()

{}

}