Оглавление

[Постановка задачи для разработки 3](#_Toc477978447)

[Макеты дизайна страниц сайта 3](#_Toc477978448)

[Структура файлов системы 5](#_Toc477978449)

[Исходный код реализации системы 5](#_Toc477978450)

[Руководство пользователя 17](#_Toc477978451)

[Стартовая страница системы 17](#_Toc477978452)

[Раздел «Регистрация» 17](#_Toc477978453)

[Раздел «Автор» 18](#_Toc477978454)

[Раздел «Модератор». 19](#_Toc477978455)

[Раздел «Новости». 21](#_Toc477978456)

[Руководство системного программиста 22](#_Toc477978457)

[Настройка СУБД 22](#_Toc477978458)

[Схема БД 26](#_Toc477978459)

[Структура полей таблиц БД 26](#_Toc477978460)

[UML (последовательность действий) 27](#_Toc477978461)

[UML (сценарий использования) 28](#_Toc477978462)

[Список литературы: 28](#_Toc477978463)

## Постановка задачи для разработки

В рамках контрольной работы разрабатывается web-приложение, позволяющее авторам добавлять новости на сайт, указывая заголовок, аннотацию, полный текст новости, изображение, а также период публикации (с какой даты по какую новость должна выводиться на сайте). Модераторы могут принимать новости или отклонять их.

Разрабатываемая система должна размещаться на хостинге со следующими требованиями: PHP 5.4, MySQL 5.6. К рабочим местам администратора и пользователей предъявляются следующие требования: любой из современных браузеров с включенной поддержкой JavaScript. При разработке необходимо применять языки **HTML**, **PHP**, **CSS**. Также использовать сессионные переменные, фреймворк **Yii2**, применять веб-приложение **PHPMyAdmin** для создания и первичного наполнения БД.

## Макеты дизайна страниц сайта

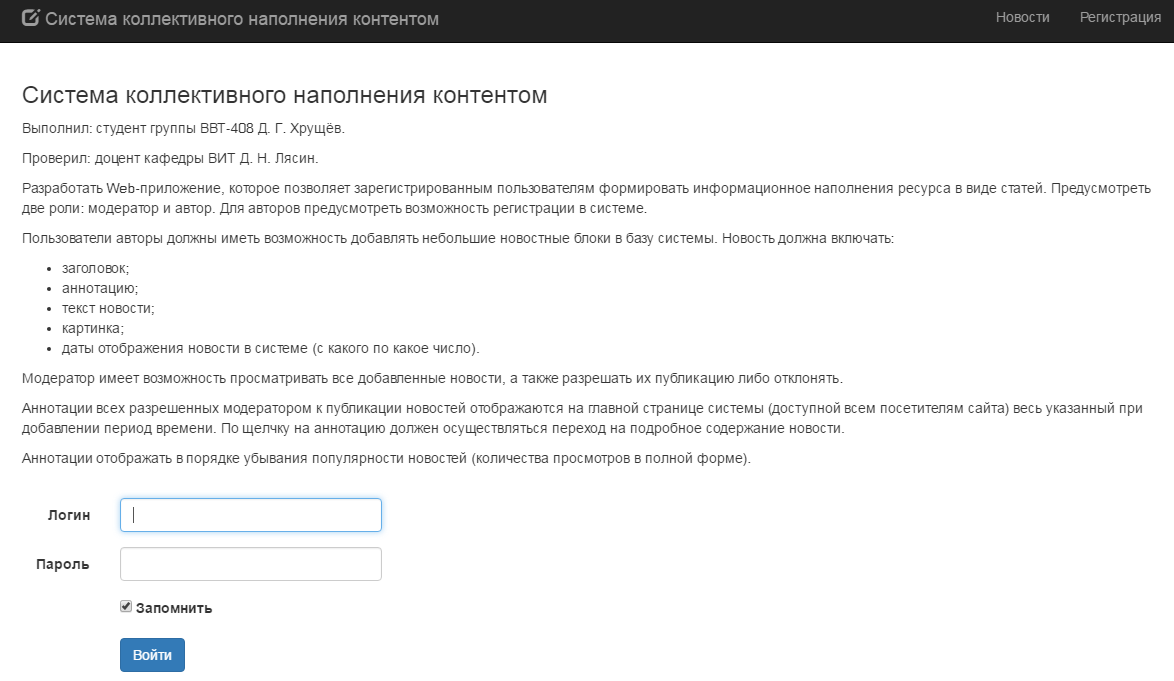


Рисунок 1 – Главная страница

Для реализации выбран фреймворк [Yii2](http://www.yiiframework.com/) с макетами по умолчанию, используются глиф-иконки из [Bootstrap](http://getbootstrap.com/components/). Для форм редактирования данных использовался [Gii](http://www.yiiframework.com/doc-2.0/guide-start-gii.html), входящий в состав Yii2.

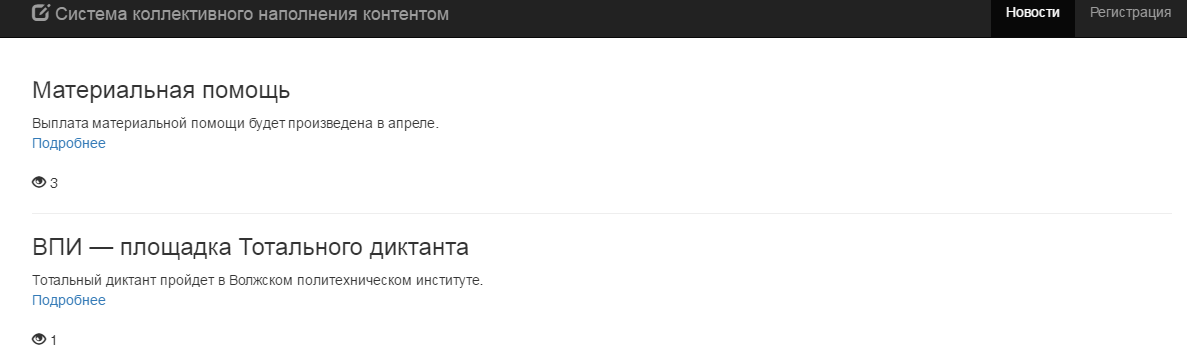


Рисунок 2 – Страница вывода новостей

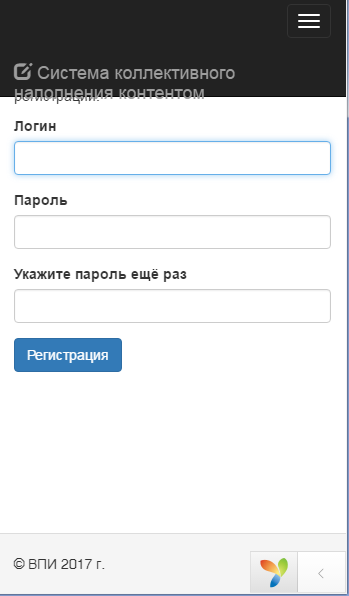
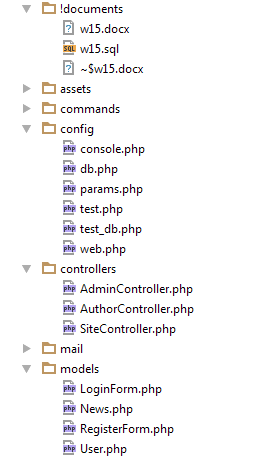
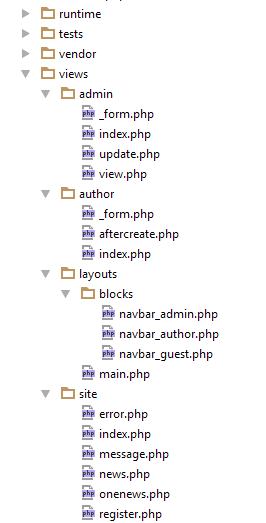
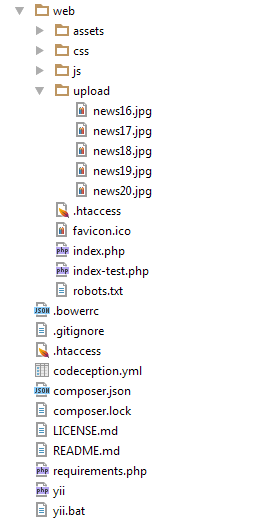


Рисунок 3 – Мобильная версия сайта

# Структура файлов системы







# Исходный код реализации системы

**/controllers/AdminController.php:**

**<?php  
  
namespace** app\controllers;  
  
**use** yii;  
**use** yii\filters\AccessControl;  
**use** yii\web\Controller;  
**use** yii\filters\VerbFilter;  
**use** yii\helpers\Url;  
**use** yii\web\UploadedFile;  
  
**use** app\models\News;  
**use** yii\data\ActiveDataProvider;  
  
  
**class** AdminController **extends** Controller  
{  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** behaviors()  
 {  
 **return** [  
 **'access'** => [  
 **'class'** => AccessControl::*className*(),  
 **'only'** => [**'index'**, **'view'**, **'update'**, **'delete'**],  
 **'rules'** => [  
 [  
 **'allow'** => **true**,  
 **'actions'** => [**'index'**, **'view'**, **'update'**, **'delete'**],  
 **'roles'** => [**'adminRole'**],  
 ],  
 ],  
 ],  
  
 **'verbs'** => [  
 **'class'** => VerbFilter::*className*(),  
 **'actions'** => [  
 **'logout'** => [**'post'**],  
 ],  
 ],  
 ];  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** actions()  
 {  
 **return** [  
 **'error'** => [  
 **'class'** => **'yii\web\ErrorAction'**,  
 ],  
 **'captcha'** => [  
 **'class'** => **'yii\captcha\CaptchaAction'**,  
 **'fixedVerifyCode'** => ***YII\_ENV\_TEST*** ? **'testme'** : **null**,  
 ],  
 ];  
 }  
  
 */\*\*  
 \* Displays homepage.  
 \*  
 \** ***@return*** *string  
 \*/* **public function** actionIndex()  
 {  
 $dataProvider = **new** ActiveDataProvider([  
 **'query'** => News::*find*()->orderBy(**'state'**), *//->where(['state' => News::STATUS\_NEW])* **'pagination'** => [  
 **'pageSize'** => 10,  
 ],  
 ]);  
  
 Url::*remember*(); *//remember current page (see Url::previous() below)* **return** $this->render(**'index'**, [  
 **'dataProvider'** => $dataProvider,  
 ]);  
 }  
  
 */\*\*  
 \* Displays a single News model.  
 \** ***@param*** *integer $id  
 \** ***@return*** *mixed  
 \*/* **public function** actionView($id)  
 {  
 **return** $this->render(**'view'**, [  
 **'model'** => $this->findModel($id),  
 ]);  
 }  
  
 */\*\*  
 \* Updates an existing News model.  
 \* If update is successful, the browser will be redirected to the 'view' page.  
 \** ***@param*** *integer $id  
 \** ***@return*** *mixed  
 \*/* **public function** actionUpdate($id)  
 {  
 $model = $this->findModel($id);  
  
 **if** ($model->load(Yii::*$app*->**request**->post())) {  
 $image = UploadedFile::*getInstance*($model, **'image'**); *// картинка* **if** (!is\_null($image)) { *// если выбрана картинка  
 // удаляем старую картинку* @unlink(Yii::*$app*->**basePath** . Yii::*$app*->**params**[**'fullUploadPath'**] . **'news'** . $id . **'.'** . $model->**ext\_image**); *// @ – значит без exceptiona в случае ошибки  
  
 // сохраняем новую картинку* $model->**ext\_image** = end(explode(**"."**, $image->**name**));  
 $image->saveAs(Yii::*$app*->**basePath** . Yii::*$app*->**params**[**'fullUploadPath'**] . **'news'** . $id . **'.'** . $model->**ext\_image**);  
 }  
  
 **if** ($model->save()) {  
 *// Если запись обновилась в БД  
  
 // Направляем пользователя на предыдущую страницу* **return** $this->redirect(Url::*previous*());  
 } **else  
 return** $this->render(**'update'**, [  
 **'model'** => $model,  
 ]);  
 }  
 **else  
 return** $this->render(**'update'**, [  
 **'model'** => $model,  
 ]);  
 }  
  
 */\*\*  
 \* Deletes an existing News model.  
 \* If deletion is successful, the browser will be redirected to the 'index' page.  
 \** ***@param*** *integer $id  
 \** ***@return*** *mixed  
 \*/* **public function** actionDelete($id)  
 {  
 $model = $this->findModel($id);  
  
 *// При удалении новости необходимо удалить картинку* @unlink(Yii::*$app*->**basePath** . Yii::*$app*->**params**[**'fullUploadPath'**] . **'news'** . $id . **'.'** . $model->**ext\_image**);  
  
 $model->delete();  
  
 **return** $this->redirect([**'index'**]);  
 }  
  
 */\*\*  
 \* Finds the News model based on its primary key value.  
 \* If the model is not found, a 404 HTTP exception will be thrown.  
 \** ***@param*** *integer $id  
 \** ***@return*** *News the loaded model  
 \** ***@throws*** *NotFoundHttpException if the model cannot be found  
 \*/* **protected function** findModel($id)  
 {  
 **if** (($model = News::*findOne*($id)) !== **null**) {  
 **return** $model;  
 } **else** {  
 **throw new** NotFoundHttpException(**'Запрошенная страница не найдена.'**);  
 }  
 }  
  
  
}

**/controllers/AuthController.php:**

**<?php  
  
namespace** app\controllers;  
  
**use** yii;  
**use** yii\filters\AccessControl;  
**use** yii\web\Controller;  
**use** yii\filters\VerbFilter;  
**use** yii\web\UploadedFile;  
**use** app\models\News;  
  
  
  
**class** AuthorController **extends** Controller  
{  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** behaviors()  
 {  
 **return** [  
 **'access'** => [  
 **'class'** => AccessControl::*className*(),  
 **'only'** => [**'index'**],  
 **'rules'** => [  
 [  
 **'allow'** => **true**,  
 **'actions'** => [**'index'**],  
 **'roles'** => [**'authorRole'**],  
 ],  
 ],  
 ],  
  
 **'verbs'** => [  
 **'class'** => VerbFilter::*className*(),  
 **'actions'** => [  
 **'logout'** => [**'post'**],  
 ],  
 ],  
 ];  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** actions()  
 {  
 **return** [  
 **'error'** => [  
 **'class'** => **'yii\web\ErrorAction'**,  
 ],  
 **'captcha'** => [  
 **'class'** => **'yii\captcha\CaptchaAction'**,  
 **'fixedVerifyCode'** => ***YII\_ENV\_TEST*** ? **'testme'** : **null**,  
 ],  
 ];  
 }  
  
 */\*\*  
 \* Displays homepage.  
 \*  
 \** ***@return*** *string  
 \*/* **public function** actionIndex()  
 {  
 $model = **new** News();  
  
 **if** ($model->load(Yii::*$app*->**request**->post())) {  
 $image = UploadedFile::*getInstance*($model, **'image'**); *// картинка* **if** (!is\_null($image)) { *// если выбрана картинка  
 // сохраняем новую картинку* $model->**ext\_image** = end(explode(**"."**, $image->**name**));  
 }  
  
 $model->**id\_author** = Yii::*$app*->**user**->**identity**->getId();  
  
 **if** ($model->save()) {  
 *// Если запись обновилась в БД* **if** (!is\_null($image)) {  
 $image->saveAs(Yii::*$app*->**basePath** . Yii::*$app*->**params**[**'fullUploadPath'**] . **'news'** . $model->**id** . **'.'** . $model->**ext\_image**);  
 }  
  
 **return** $this->render(**'aftercreate'**);  
 } **else  
 return** $this->render(**'index'**, [  
 **'model'** => $model,  
 ]);  
 }  
 **else  
 return** $this->render(**'index'**, [  
 **'model'** => $model,  
 ]);  
  
 }  
  
 */\*\*  
 \* Finds the News model based on its primary key value.  
 \* If the model is not found, a 404 HTTP exception will be thrown.  
 \** ***@param*** *integer $id  
 \** ***@return*** *News the loaded model  
 \** ***@throws*** *NotFoundHttpException if the model cannot be found  
 \*/* **protected function** findModel($id)  
 {  
 **if** (($model = News::*findOne*($id)) !== **null**) {  
 **return** $model;  
 } **else** {  
 **throw new** NotFoundHttpException(**'Запрошенная страница не найдена.'**);  
 }  
 }  
   
}

**/controllers/SiteController.php:**

**<?php  
  
namespace** app\controllers;  
  
**use** yii;  
**use** yii\filters\AccessControl;  
**use** yii\web\Controller;  
**use** yii\filters\VerbFilter;  
**use** app\models\LoginForm;  
**use** app\models\RegisterForm;  
**use** app\models\News;  
**use** yii\data\Pagination;  
  
**class** SiteController **extends** Controller  
{  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** behaviors()  
 {  
 **return** [  
 **'access'** => [  
 **'class'** => AccessControl::*className*(),  
 **'only'** => [**'logout'**, **'register'**],  
 **'rules'** => [  
 [  
 **'actions'** => [**'logout'**],  
 **'allow'** => **true**,  
 **'roles'** => [**'@'**],  
 ],  
 [  
 **'actions'** => [**'register'**],  
 **'allow'** => **true**,  
 **'roles'** => [**'?'**],  
 ],  
 ],  
 ],  
 **'verbs'** => [  
 **'class'** => VerbFilter::*className*(),  
 **'actions'** => [  
 **'logout'** => [**'post'**],  
 ],  
 ],  
 ];  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** actions()  
 {  
 **return** [  
 **'error'** => [  
 **'class'** => **'yii\web\ErrorAction'**,  
 ],  
 **'captcha'** => [  
 **'class'** => **'yii\captcha\CaptchaAction'**,  
 **'fixedVerifyCode'** => ***YII\_ENV\_TEST*** ? **'testme'** : **null**,  
 ],  
 ];  
 }  
  
 *// Новости* **public function** actionNews() {  
 *// выполняем запрос (нужны только опубликованные новости с сортировкой от самой новой даты публикации к самой старой* $allNews = News::*find*()->where([**'state'** => News::***STATUS\_AGREE***])  
 ->andWhere(**'start\_date <= CURDATE()'**)  
 ->andWhere(**'finish\_date >= CURDATE()'**)  
 ->orderBy([**'count\_views'** => ***SORT\_DESC***, **'id'** => ***SORT\_DESC***]);  
 $countQuery = **clone** $allNews; *// делаем копию выборки  
  
 // подключаем класс Pagination* $pages = **new** Pagination([**'totalCount'** => $countQuery->count(), **'pageSize'** => 5]); *// 5 - количество новостей на странице* $pages->**pageSizeParam** = **false**; *// приводим параметры в ссылке к ЧПУ* $news = $allNews->offset($pages->**offset**)  
 ->limit($pages->**limit**)  
 ->all();  
  
 *// Передаем данные в представление* **return** $this->render(**'news'**, [  
 **'news'** => $news,  
 **'pages'** => $pages,  
 ]);  
 }  
  
 *// Выбранная новость* **public function** actionOneNews($id) {  
 $one\_news = News::*find*()->AsArray()->where([**'id'** => $id, **'state'** => News::***STATUS\_AGREE***])->limit(1)->one();  
 **if** (**empty**($one\_news))  
 $this->redirect(**'/'**); *// Если новость не найдена, то редирект на главную  
  
 // Добавляем +1 к количеству просмотров* $model = News::*findOne*($id);  
  
 $model->**count\_views** = $model->**count\_views** + 1;  
 $model->save();  
  
 **return** $this->render(**'onenews'**, compact(**'one\_news'**));  
 }  
   
 */\*\*  
 \* Displays homepage.  
 \*  
 \** ***@return*** *string  
 \*/* **public function** actionIndex()  
 {  
 *// Если не гость* **if** (!Yii::*$app*->**user**->**isGuest**) {  
 **if** (Yii::*$app*->**user**->can(**'adminRole'**)) { *// Если админ, то перенаправляем на его контроллер* **return** $this->redirect(**'/admin/'**);  
 }  
 **else** { *// Иначе - перенаправляем на контроллер автора* **return** $this->redirect(**'/author/'**);  
 }  
 }  
 **else** {  
 $model = **new** LoginForm();  
 **if** ($model->load(Yii::*$app*->**request**->post()) && $model->login()) {  
 **return** $this->goBack();  
 }  
 **return** $this->render(**'index'**, [  
 **'model'** => $model,  
 ]);  
 }  
 }  
  
 */\*\*  
 \* Login action.  
 \*  
 \** ***@return*** *string  
 \*/* **public function** actionLogin()  
 {  
 **if** (!Yii::*$app*->**user**->**isGuest**) {  
 **return** $this->goHome();  
 }  
  
 $model = **new** LoginForm();  
 **if** ($model->load(Yii::*$app*->**request**->post()) && $model->login()) {  
 **return** $this->goBack();  
 }  
 **return** $this->render(**'login'**, [  
 **'model'** => $model,  
 ]);  
 }  
  
 */\*\*  
 \* Logout action.  
 \*  
 \** ***@return*** *string  
 \*/* **public function** actionLogout()  
 {  
 Yii::*$app*->**user**->logout();  
  
 **return** $this->goHome();  
 }  
  
 **public function** actionRegister()  
 {  
 $model = **new** RegisterForm();  
 *// Если есть POST-данные* **if** ($model->load(Yii::*$app*->**request**->post())) {  
 *// Если успешно зарегистрировались* **if** ($user = $model->register()) {  
 *// Добавляем роль* $sql = **'INSERT INTO auth\_assignment(item\_name, user\_id) VALUES("author", '** . $user->**id** . **')'**;  
 \Yii::*$app*->**db**->createCommand($sql)->execute();  
  
 *// flash-сообщением уведомляем пользователя, что регистрация прошла успешно* Yii::*$app*->**session**->setFlash(**'reg-success'**, **'Регистрация прошла успешно.'**);  
 **return** $this->redirect(**'/site/'**);  
 }  
 }  
  
 *// Отрисовываем страницу регистрации, если не было POST-данных или если регистрация прошла неуспешно* **return** $this->render(**'register'**, [  
 **'model'** => $model,  
 ]);  
 }  
  
}

**/models/LoginForm.php:**

**<?php  
  
namespace** app\models;  
  
**use** yii;  
**use** yii\base\Model;  
  
*/\*\*  
 \* LoginForm is the model behind the login form.  
 \*  
 \** ***@property*** *User|null $user This property is read-only.  
 \*  
 \*/***class** LoginForm **extends** Model  
{  
 **public $username**;  
 **public $password**;  
 **public $rememberMe** = **true**;  
  
 **private $\_user** = **false**;  
  
  
 */\*\*  
 \** ***@return*** *array the validation rules.  
 \*/* **public function** rules()  
 {  
 **return** [  
 *// username and password are both required* [[**'username'**, **'password'**], **'required'**],  
 *// rememberMe must be a boolean value* [**'rememberMe'**, **'boolean'**],  
 *// password is validated by validatePassword()* [**'password'**, **'validatePassword'**],  
 ];  
 }  
  
 **public function** attributeLabels()  
 {  
 **return** [  
 **'username'** => **'Логин'**,  
 **'password'** => **'Пароль'**,  
 **'rememberMe'** => **'Запомнить'** ];  
 }  
  
 */\*\*  
 \* Validates the password.  
 \* This method serves as the inline validation for password.  
 \*  
 \** ***@param*** *string $attribute the attribute currently being validated  
 \** ***@param*** *array $params the additional name-value pairs given in the rule  
 \*/* **public function** validatePassword($attribute, $params)  
 {  
 **if** (!$this->hasErrors()) {  
 $user = $this->getUser();  
  
 **if** (!$user || !$user->validatePassword($this->**password**)) {  
 $this->addError($attribute, **'Неверное имя пользователя или пароль.'**);  
 }  
 }  
 }  
  
 */\*\*  
 \* Logs in a user using the provided username and password.  
 \** ***@return*** *bool whether the user is logged in successfully  
 \*/* **public function** login()  
 {  
 **if** ($this->validate()) {  
 **return** Yii::*$app*->**user**->login($this->getUser(), $this->**rememberMe** ? 3600\*24\*30 : 0);  
 }  
 **return false**;  
 }  
  
 */\*\*  
 \* Finds user by [[username]]  
 \*  
 \** ***@return*** *User|null  
 \*/* **public function** getUser()  
 {  
 **if** ($this->**\_user** === **false**) {  
 $this->**\_user** = User::*findByUsername*($this->**username**);  
 }  
  
 **return** $this->**\_user**;  
 }  
}

**/models/News.php:**

**<?php  
  
namespace** app\models;  
  
**use** yii;  
**use** yii\helpers\ArrayHelper;  
  
*/\*\*  
 \* This is the model class for table "w15\_news".  
 \*  
 \** ***@property*** *integer $id  
 \** ***@property*** *string $header  
 \** ***@property*** *string $annotation  
 \** ***@property*** *string $full\_text  
 \** ***@property*** *string $ext\_image  
 \** ***@property*** *string $start\_date  
 \** ***@property*** *string $finish\_date  
 \** ***@property*** *integer $state  
 \** ***@property*** *integer $count\_views  
 \** ***@property*** *integer $id\_author  
 \*/***class** News **extends** \yii\db\ActiveRecord  
{  
 **public $image**;  
  
 **const *STATUS\_NEW*** = 0;  
 **const *STATUS\_AGREE*** = 1;  
 **const *STATUS\_DISCARD*** = 2;  
  
 *// Получить русское название статуса по константе (должна быть в поле status)* **public function** getStatusName()  
 {  
 **return** ArrayHelper::*getValue*(**self**::*getStatusesArray*(), $this->**state**);  
 }  
  
 *// Получить ассоциативный массив всех статусов с их русскими названиями* **public static function** getStatusesArray()  
 {  
 **return** [  
 **self**::***STATUS\_NEW*** => **'Ожидает подтверждения модератором'**,  
 **self**::***STATUS\_AGREE*** => **'Опубликована'**,  
 **self**::***STATUS\_DISCARD*** => **'Отклонена'** ];  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public static function** tableName()  
 {  
 **return 'w15\_news'**;  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** rules()  
 {  
 **return** [  
 [[**'header'**, **'annotation'**, **'full\_text'**, **'ext\_image'**, **'start\_date'**, **'finish\_date'**], **'required'**],  
 [[**'start\_date'**, **'finish\_date'**], **'date'**, **'format'** => **'php:Y-m-d'**],  
 [[**'state'**, **'count\_views'**], **'integer'**],  
 [[**'header'**], **'string'**, **'max'** => 100],  
 [[**'annotation'**], **'string'**, **'max'** => 200],  
 [[**'full\_text'**], **'string'**, **'max'** => 1024],  
 [[**'ext\_image'**], **'string'**, **'max'** => 4],  
 [[**'image'**], **'file'**, **'skipOnEmpty'** => **'false'**, **'extensions'**=>**'jpg, gif, png, jpeg'**, **'message'** => **'Не выбрано изображение'**]  
 ];  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** attributeLabels()  
 {  
 **return** [  
 **'id'** => **'ID'**,  
 **'header'** => **'Заголовок'**,  
 **'annotation'** => **'Аннотация'**,  
 **'full\_text'** => **'Текст новости'**,  
 **'image'** => **'Изображение'**,  
 **'start\_date'** => **'Дата начала'**,  
 **'finish\_date'** => **'Дата окончания'**,  
 **'state'** => **'Статус'**,  
 **'count\_views'** => **'Количество просмотров'**,  
 ];  
 }  
}

**/models/RegisterForm.php:**

**<?  
namespace** app\models;  
  
**use** yii;  
**use** yii\base\Model;  
  
*/\*\*  
 \* Register form  
 \*/***class** RegisterForm **extends** Model  
{  
 **public $username**;  
 **public $password**;  
 **public $password2**;  
  
 **public function** rules()  
 {  
 **return** [  
 [**'username'**, **'filter'**, **'filter'** => **'trim'**],  
 [**'username'**, **'required'**],  
 [**'username'**, **'unique'**, **'targetClass'** => User::*className*(), **'message'** => **'Логин уже занят.'**],  
 [**'username'**, **'string'**, **'min'** => 2, **'max'** => 255],  
  
 [**'password'**, **'required'**],  
 [**'password'**, **'string'**, **'min'** => 6],  
  
 [**'password2'**, **'required'**],  
 [**'password2'**, **'string'**, **'min'** => 6],  
 [**'password2'**, **'compare'**, **'compareAttribute'**=>**'password'**, **'message'** => **'Пароли не совпадают.'**], *// Проверка совпадения паролей* ];  
 }  
  
 **public function** attributeLabels()  
 {  
 **return** [  
 **'username'** => **'Логин'**,  
 **'password'** => **'Пароль'**,  
 **'password2'** => **'Укажите пароль ещё раз'**,  
 ];  
 }  
  
 **public function** register()  
 {  
 **if** ($this->validate()) {  
 $user = **new** User();  
 $user->**username** = $this->**username**;  
 $user->setPassword($this->**password**);  
 $user->generateAuthKey();  
  
 **if** ($user->save()) {  
 **return** $user;  
 }  
 }  
  
 **return null**;  
 }  
  
}

**/models/User.php:**

**<?php  
  
namespace** app\models;  
**use** yii;  
  
  
*/\*\*  
 \* This is the model class for table "w15\_users".  
 \*  
 \** ***@property*** *integer $id  
 \** ***@property*** *string $username  
 \** ***@property*** *string $password  
 \** ***@property*** *string $auth\_key  
 \** ***@property*** *string $access\_token  
 \*/  
  
//class User extends \yii\base\Object implements \yii\web\IdentityInterface***class** User **extends** \yii\db\ActiveRecord **implements** \yii\web\IdentityInterface  
{  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public static function** tableName()  
 {  
 **return 'w15\_users'**;  
 }  
  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public static function** findIdentity($id)  
 {  
 **return static**::*findOne*($id);  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public static function** findIdentityByAccessToken($token, $type = **null**)  
 {  
 **return static**::*findOne*([**'access\_token'** => $token]);  
 }  
  
 */\*\*  
 \* Finds user by username  
 \*/* **public static function** findByUsername($username)  
 {  
 **return static**::*findOne*([**'username'** => $username]);  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** getId()  
 {  
 **return** $this->**id**;  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** getAuthKey()  
 {  
 **return** $this->**authKey**;  
 }  
  
 */\*\*  
 \** ***@inheritdoc*** *\*/* **public function** validateAuthKey($authKey)  
 {  
 **return** $this->**authKey** === $authKey;  
 }  
  
 */\*\*  
 \* Validates password  
 \*  
 \** ***@param*** *string $password password to validate  
 \** ***@return*** *bool if password provided is valid for current user  
 \*/* **public function** validatePassword($password)  
 {  
 **return** $this->**password** === md5($password);  
 }  
  
 **public function** setPassword($password)  
 {  
 $this->**password** = md5($password);  
 }  
  
 **public function** generateAuthKey()  
 {  
 $this->**auth\_key** = Yii::*$app*->**security**->generateRandomString();  
 }  
}

**/views/admin/index.php:**

**<?php  
  
use** yii\helpers\Html;  
**use** yii\grid\GridView;  
  
*/\** ***@var*** *$this yii\web\View \*/  
/\** ***@var*** *$dataProvider yii\data\ActiveDataProvider \*/*$this->**title** = **'Новости'**;  
**?>**<**div**>  
 <**h1**>**<?=** Html::*encode*($this->**title**) **?>**</**h1**>  
  
 **<?=** GridView::*widget*([  
 **'dataProvider'** => $dataProvider,  
 **'columns'** => [  
 [**'class'** => **'yii\grid\SerialColumn'**],  
  
 **'id'**,  
 **'header'**,  
 **'annotation'**,  
 **'full\_text'**,  
 *//'ext\_image',* **'start\_date'**,  
 **'finish\_date'**,  
 **'state'**,  
 *// 'count\_views',  
 // 'id\_author',* [**'class'** => **'yii\grid\ActionColumn'**],  
 ],  
 ]); **?>**</**div**>

**/views/admin/\_form.php:**

**<?php  
  
use** yii\helpers\Html;  
**use** yii\widgets\ActiveForm;  
**use** mihaildev\ckeditor\CKEditor;  
**use** kartik\file\FileInput;  
  
*/\** ***@var*** *$this yii\web\View \*/  
/\** ***@var*** *$model app\models\News \*/  
/\** ***@var*** *$form yii\widgets\ActiveForm \*/***?>**<**div** class=**"news-form"**>  
  
 **<?php** $form = ActiveForm::*begin*([**'options'** => [**'enctype'** => **'multipart/form-data'**]]); **?>  
  
 <?=** $form->field($model, **'header'**)->textInput([**'maxlength'** => **true**]) **?>  
  
 <?=** $form->field($model, **'annotation'**)->textInput([**'maxlength'** => **true**]) **?>  
  
 <?=** $form->field($model, **'full\_text'**)->widget(CKEditor::*className*(),[  
 **'editorOptions'** => [  
 **'preset'** => **'full'**, *//разработаны стандартные настройки basic, standard, full данную возможность не обязательно использовать* **'inline'** => **false**, *//по умолчанию false* ],  
 ])  
 **?>  
  
 <?php  
 if** (!**empty**($model->**id**)) {  
 $previewImg = **'/upload/news'** . $model->**id** . **'.'** . $model->**ext\_image**;  
 **echo "<label class='control-label'>Предыдущее изображение</label><br><img src='"** . $previewImg . **"' alt='Превью новости' width='200px'><br>"**;  
 }  
 **?>  
  
  
 <?=** $form->field($model, **'image'**)->widget(FileInput::*classname*(), [  
 **'options'** => [**'accept'** => **'image/\*'**],  
 **'pluginOptions'**=>[**'allowedFileExtensions'**=>[**'jpg'**,**'gif'**,**'png'**, **'jpeg'**],**'showUpload'** => **false**,],  
 ])  
 **?>  
  
 <?=** $form->field($model, **'start\_date'**)->widget(\yii\jui\DatePicker::*className*(), [**'dateFormat'** => **'yyyy-MM-dd'**]) **?>  
  
 <?=** $form->field($model, **'finish\_date'**)->widget(\yii\jui\DatePicker::*className*(), [**'dateFormat'** => **'yyyy-MM-dd'**]) **?>  
  
 <?=** $form->field($model, **'state'**)->dropDownList($model->getStatusesArray()) **?>  
  
 <?=** $form->field($model, **'count\_views'**)->textInput([**'readonly'** => **true**]) **?>** <**div** class=**"form-group"**>  
 **<?=** Html::*submitButton*($model->**isNewRecord** ? **'Добавить'** : **'Применить'**, [**'class'** => $model->**isNewRecord** ? **'btn btn-success'** : **'btn btn-primary'**]) **?>** </**div**>  
  
 **<?php** ActiveForm::*end*(); **?>**</**div**>

**/views/author/index.php:**

**<?php  
  
use** yii\helpers\Html;  
  
*/\** ***@var*** *$this yii\web\View \*/  
/\** ***@var*** *$model app\models\News \*/*$this->**title** = **'Добавление новости'**;  
**?>**<**div**>  
  
 <**h1**>**<?=** Html::*encode*($this->**title**) **?>**</**h1**>  
  
 **<?=** $this->render(**'\_form'**, [  
 **'model'** => $model,  
 ]) **?>**</**div**>

**/views/author/\_form.php:**

**<?php  
  
use** yii\helpers\Html;  
**use** yii\widgets\ActiveForm;  
**use** mihaildev\ckeditor\CKEditor;  
**use** kartik\file\FileInput;  
  
*/\** ***@var*** *$this yii\web\View \*/  
/\** ***@var*** *$model app\models\News \*/  
/\** ***@var*** *$form yii\widgets\ActiveForm \*/***?>**<**div** class=**"news-form"**>  
  
 **<?php** $form = ActiveForm::*begin*([**'options'** => [**'enctype'** => **'multipart/form-data'**]]); **?>  
  
 <?=** $form->field($model, **'header'**)->textInput([**'maxlength'** => **true**]) **?>  
  
 <?=** $form->field($model, **'annotation'**)->textInput([**'maxlength'** => **true**]) **?>  
  
 <?=** $form->field($model, **'image'**)->widget(FileInput::*classname*(), [  
 **'options'** => [**'accept'** => **'image/\*'**],  
 **'pluginOptions'**=>[**'allowedFileExtensions'**=>[**'jpg'**,**'gif'**,**'png'**, **'jpeg'**],**'showUpload'** => **false**,],  
 ])  
 **?>  
   
 <?=** $form->field($model, **'full\_text'**)->widget(CKEditor::*className*(),[  
 **'editorOptions'** => [  
 **'preset'** => **'full'**, *//разработаны стандартные настройки basic, standard, full данную возможность не обязательно использовать* **'inline'** => **false**, *//по умолчанию false* ],  
 ])  
 **?>  
  
 <?=** $form->field($model, **'start\_date'**)->widget(\yii\jui\DatePicker::*className*(), [**'dateFormat'** => **'yyyy-MM-dd'**]) **?>  
  
 <?=** $form->field($model, **'finish\_date'**)->widget(\yii\jui\DatePicker::*className*(), [**'dateFormat'** => **'yyyy-MM-dd'**]) **?>** <**div** class=**"form-group"**>  
 **<?=** Html::*submitButton*($model->**isNewRecord** ? **'Добавить'** : **'Применить'**, [**'class'** => $model->**isNewRecord** ? **'btn btn-success'** : **'btn btn-primary'**]) **?>** </**div**>  
  
 **<?php** ActiveForm::*end*(); **?>**</**div**>

**/views/layouts/blocks/navbar\_admin.php:**

**<?php  
 use** yii\bootstrap\NavBar;  
 **use** yii\bootstrap\Nav;  
 **use** yii\helpers\Html;  
  
 NavBar::*begin*([  
 **'brandLabel'** => **'<span class="glyphicon glyphicon-edit" aria-hidden="true"></span>&nbsp;Система коллективного наполнения контентом'**,  
 **'brandUrl'** => Yii::*$app*->**homeUrl**,  
 **'options'** => [  
 **'class'** => **'navbar-inverse navbar-fixed-top'**,  
 ],  
 ]);  
 **echo** Nav::*widget*([  
 **'options'** => [**'class'** => **'navbar-nav navbar-right'**],  
 **'items'** => [  
 (  
 **'<li>'** . Html::*beginForm*([**'site/logout'**], **'post'**)  
 . Html::*submitButton*(  
 **'Выйти ('** . Yii::*$app*->**user**->**identity**->**username** . **')'**,  
 [**'class'** => **'btn btn-link logout'**]  
 )  
 . Html::*endForm*()  
 . **'</li>'** )  
 ],  
 ]);  
  
  
 NavBar::*end*();  
**?>**

**/views/layouts/main.php:**

**<?php***/\** ***@var*** *$this \yii\web\View \*/  
/\** ***@var*** *$content string \*/***use** yii\helpers\Html;  
**use** app\assets\AppAsset;  
  
AppAsset::*register*($this);  
**?>  
<?php** $this->beginPage() **?>**<!DOCTYPE html>  
<**html** lang=**"<?=** Yii::*$app*->**language ?>"**>  
<**head**>  
 <**meta** charset=**"<?=** Yii::*$app*->**charset ?>"**>  
 <**meta** name=**"viewport"** content=**"width=device-width, initial-scale=1"**>  
 **<?=** Html::*csrfMetaTags*() **?>** <**title**>**<?=** Html::*encode*($this->**title**) **?>**</**title**>  
 **<?php** $this->head() **?>**</**head**>  
<**body**>  
**<?php** $this->beginBody() **?>**<**div** class=**"wrap"**>  
 **<?php  
 if** (Yii::*$app*->**user**->**isGuest**) { *// Если гость* $this->beginContent(**'@app/views/layouts/blocks/navbar\_guest.php'**);  
 $this->endContent();  
 }  
 **else** {  
 **if** (Yii::*$app*->**user**->can(**'adminRole'**)) { *// Если админ* $this->beginContent(**'@app/views/layouts/blocks/navbar\_admin.php'**);  
 $this->endContent();  
 }  
 **else** { *// Иначе клиент* $this->beginContent(**'@app/views/layouts/blocks/navbar\_author.php'**);  
 $this->endContent();  
 }  
 }  
 **?>** <**div** class=**"container"**>  
 **<?=** $content **?>** </**div**>  
</**div**>  
  
<**footer** class=**"footer"**>  
 <**div** class=**"container"**>  
 <**p** class=**"pull-left"**>**&copy;** ВПИ **<?=** date(**'Y'**) **?>** г.</**p**>  
 </**div**>  
</**footer**>  
  
**<?php** $this->endBody() **?>**</**body**>  
</**html**>  
**<?php** $this->endPage() **?>**

**/views/site/index.php:**

**<?php***/\** ***@var*** *$this yii\web\View \*/  
/\** ***@var*** *$form yii\bootstrap\ActiveForm \*/  
/\** ***@var*** *$model app\models\LoginForm \*/***use** yii\helpers\Html;  
**use** yii\bootstrap\ActiveForm;  
  
$this->**title** = **'Главная'**;  
**?>**<**div** class=**"site-index"**>  
 **<? if**(Yii::*$app*->**session**->hasFlash(**'reg-success'**)) : **?>** <**div** class=**"alert alert-success alert-dismissible"** role=**"alert"**>  
 <**button** type=**"button"** class=**"close"** data-dismiss=**"alert"** aria-label=**"Close"**><**span** aria-hidden=**"true"**>**&times;**</**span**></**button**>  
 **<?=** Yii::*$app*->**session**->getFlash(**'reg-success'**) **?>** </**div**>  
 **<? endif**; **?>** <**h3**>Система коллективного наполнения контентом</**h3**>  
 <**p**>Выполнил: студент группы ВВТ-408 Д. Г. Хрущёв.</**p**>  
 <**p**>Проверил: доцент кафедры ВИТ Д. Н. Лясин.</**p**>  
 <**p**>Разработать Web-приложение, которое позволяет зарегистрированным пользователям  
 формировать информационное наполнения ресурса в виде статей. Предусмотреть две роли:  
 модератор и автор. Для авторов предусмотреть возможность регистрации в системе. </**p**>  
 <**p**>Пользователи авторы должны иметь возможность добавлять небольшие новостные  
 блоки в базу системы. Новость должна включать:  
 </**p**>  
 <**ul**>  
 <**li**>заголовок;</**li**>  
 <**li**>аннотацию;</**li**>  
 <**li**>текст новости;</**li**>  
 <**li**>картинка;</**li**>  
 <**li**>даты отображения новости в системе (с какого по какое число).</**li**>  
 </**ul**>  
 <**p**>Модератор имеет возможность просматривать все добавленные новости, а также  
 разрешать их публикацию либо отклонять. </**p**>  
 <**p**>Аннотации всех разрешенных модератором к публикации новостей отображаются на  
 главной странице системы (доступной всем посетителям сайта) весь указанный при  
 добавлении период времени. По щелчку на аннотацию должен осуществляться переход на  
 подробное содержание новости. </**p**>  
 <**p**>  
 Аннотации отображать в порядке убывания популярности новостей (количества  
 просмотров в полной форме).  
 </**p**>  
 <**br**>  
  
 **<?php** $form = ActiveForm::*begin*([  
 **'id'** => **'login-form'**,  
 **'layout'** => **'horizontal'**,  
 **'fieldConfig'** => [  
 **'template'** => **"{label}\n<div class=\"col-lg-3\">{input}</div>\n<div class=\"col-lg-8\">{error}</div>"**,  
 **'labelOptions'** => [**'class'** => **'col-lg-1 control-label'**],  
 ],  
 ]); **?>  
  
 <?=** $form->field($model, **'username'**)->textInput([**'autofocus'** => **true**]) **?>  
  
 <?=** $form->field($model, **'password'**)->passwordInput() **?>  
  
 <?=** $form->field($model, **'rememberMe'**)->checkbox([  
 **'template'** => **"<div class=\"col-lg-offset-1 col-lg-3\">{input} {label}</div>\n<div class=\"col-lg-8\">{error}</div>"**,  
 ]) **?>** <**div** class=**"form-group"**>  
 <**div** class=**"col-lg-offset-1 col-lg-11"**>  
 **<?=** Html::submitButton(**'Войти'**, [**'class'** => **'btn btn-primary'**, **'name'** => **'login-button'**]) **?>** </**div**>  
 </**div**>  
  
 **<?php** ActiveForm::*end*(); **?>**</**div**>

**/views/site/news.php:**

**<? use** yii\widgets\LinkPager; **?>  
<?php** $this->**title** = **"Лента новостей" ?>**<**div** id=**"news-block"**>  
 **<?php foreach** ($news **as** $one\_news) {**?>** <**h3**>  
 **<?=** $one\_news->**header ?>** </**h3**>  
 **<?=** $one\_news->**annotation ?>** <**div** style=**"**margin-bottom: 20**px**;**"**>  
 <**a** href=**"/site/one-news?id=<?=** $one\_news->**id ?>"** title=**"Прочитать новость"**>Подробнее</**a**>  
 </**div**>  
 <**span** class=**"glyphicon glyphicon-eye-openglyphicon glyphicon-eye-open"** aria-hidden=**"true"**></**span**> **<?=** $one\_news->**count\_views ?>** <**hr**>  
  
 **<?** } **?>  
 <?=** LinkPager::*widget*([**'pagination'** => $pages]) **?>**</**div**>

**/views/site/onenews.php:**

**<?php** $this->**title** = **"Новость" ?>  
<?  
 if** (!**empty**(Yii::*$app*->**request**->**referrer**))  
 **echo** \yii\helpers\Html::*a*(**'Назад'**, Yii::*$app*->**request**->**referrer**, [**'class'** => **'btn btn-primary'**]) .**"<br><br>"**;  
**?>**<**h3**>**<?=** $one\_news[**'header'**]; **?>**</**h3**>  
<**div** style=**"**margin-bottom: 20**px**;**"**>  
 **<?php  
 if** (!**empty**($one\_news[**'id'**])) {  
 $previewImg = **'/upload/news'** . $one\_news[**'id'**] . **'.'** . $one\_news[**'ext\_image'**];  
 **echo "<img src='"** . $previewImg . **"' alt='Новость' style='max-width:50%;'><br>"**;  
 }  
 **?>**</**div**>  
  
**<?=** $one\_news[**'full\_text'**]; **?>**<**br**><**br**>  
<**span** class=**"glyphicon glyphicon-eye-openglyphicon glyphicon-eye-open"** aria-hidden=**"true"**></**span**> **<?=** $one\_news[**'count\_views'**]+1 **?>**

**/views/site/register.php:**

**<?php***/\** ***@var*** *$this yii\web\View \*/  
/\** ***@var*** *$form yii\bootstrap\ActiveForm \*/  
/\** ***@var*** *$model app\models\RegisterForm \*/***use** yii\helpers\Html;  
**use** yii\bootstrap\ActiveForm;  
$this->**title** = **'Регистрация'**;  
  
**?>**<**div** class=**"site-register"**>  
 <**p**>Пожалуйста, заполните следующие поля для регистрации:</**p**>  
  
 <**div** class=**"row"**>  
 <**div** class=**"col-lg-5"**>  
 **<?php** $form = ActiveForm::*begin*([**'id'** => **'form-register'**]); **?>  
  
 <?=** $form->field($model, **'username'**)->textInput([**'autofocus'** => **true**]) **?>  
  
 <?=** $form->field($model, **'password'**)->passwordInput() **?>  
  
 <?=** $form->field($model, **'password2'**)->passwordInput() **?>** <**div** class=**"form-group"**>  
 **<?=** Html::*submitButton*(**'Регистрация'**, [**'class'** => **'btn btn-primary'**, **'name'** => **'register-button'**]) **?>** </**div**>  
  
 **<?php** ActiveForm::*end*(); **?>** </**div**>  
 </**div**>  
</**div**>

# Руководство пользователя

## Стартовая страница системы

На странице отображается главное меню, название задания, текст задания, кто выполнил и проверил работу, форма ввода логина-пароля (см. рисунок 1).

Для возврата к стартовой странице системы необходимо нажать левой кнопкой мыши на логотип или название системы в левом верхнем углу.

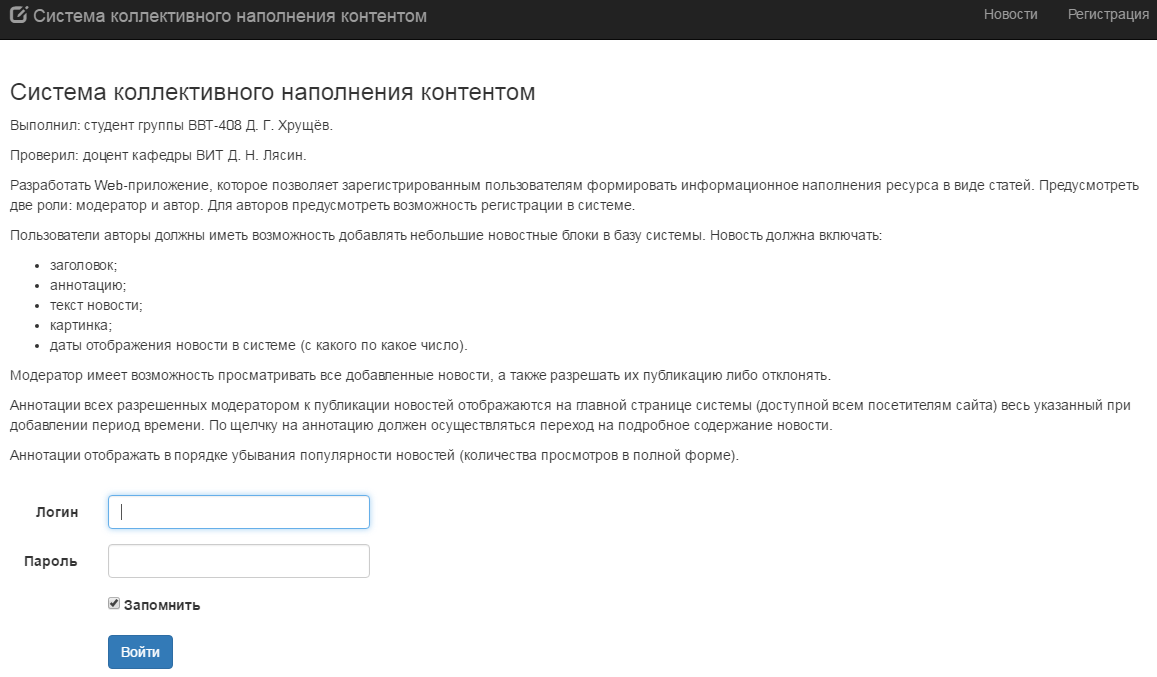


Рисунок 1 – Начальная страница системы

## Раздел «Регистрация»

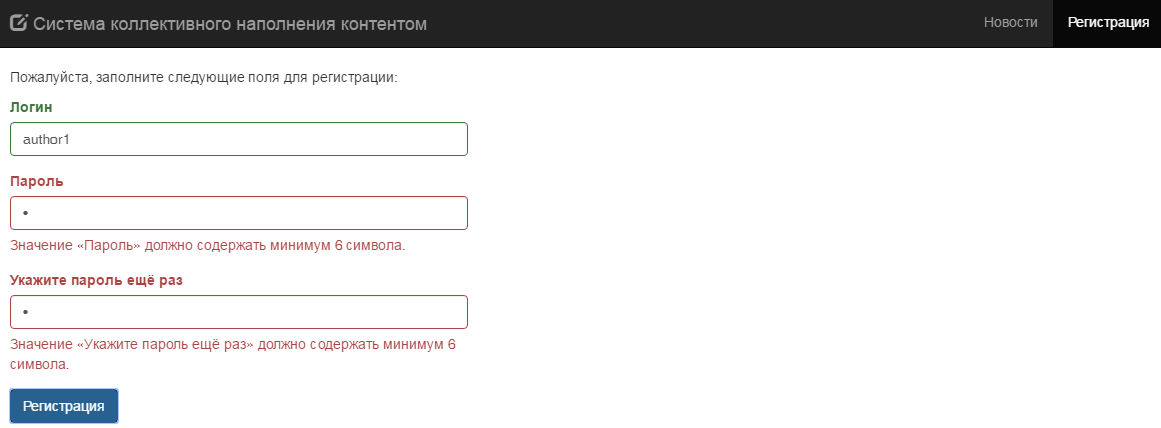


Рисунок 2 – Раздел «Регистрация»

В данном разделе выводится форма регистрации (см. рисунок 2). Пользователь должен указать логин (который не регистрировался ранее), а также два раза ввести пароль. Система выдаст сообщение, если введённые пароли не совпадут или не удовлетворяют условиям (не менее 6 символов).

После успешной регистрации система выдаёт соответствующее сообщение. Регистрируются *только авторы*, регистрация модераторов выполняется с помощью корректировки или добавления записей непосредственно в БД.

## Раздел «Автор»

В данном разделе автор добавляет новости.

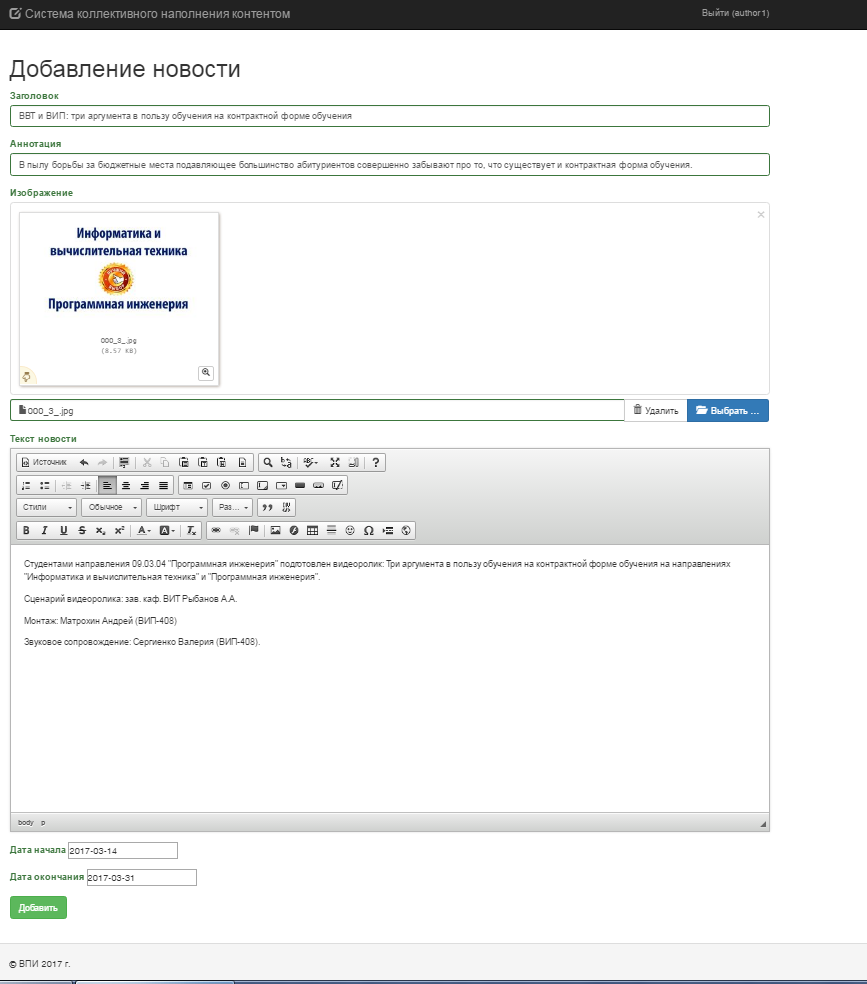


Рисунок 3 – Раздел «Автор»

Пользователь заполняет все поля новости. Полный текст новости может содержать HTML-разметку. В случае превышения длины содержимого, сайт выдаёт соответствующее предупреждение. После заполнения всех полей пользователь должен нажать кнопку «*Добавить*», сайт должен выдать сообщение, что новость отправлена модератору на рассмотрение.



Рисунок 4 – Сообщение после добавления новости

## Раздел «Модератор».

Данный раздел доступен только для пользователей, которым выставлена роль «Модератор» (выставляется в базе данных).

В личном кабинете модератора в табличной форме выдаются все новости с сортировкой по статусу новости (0 – новость на рассмотрении модератора, 1 – новость опубликована, 2 – новость отклонена).

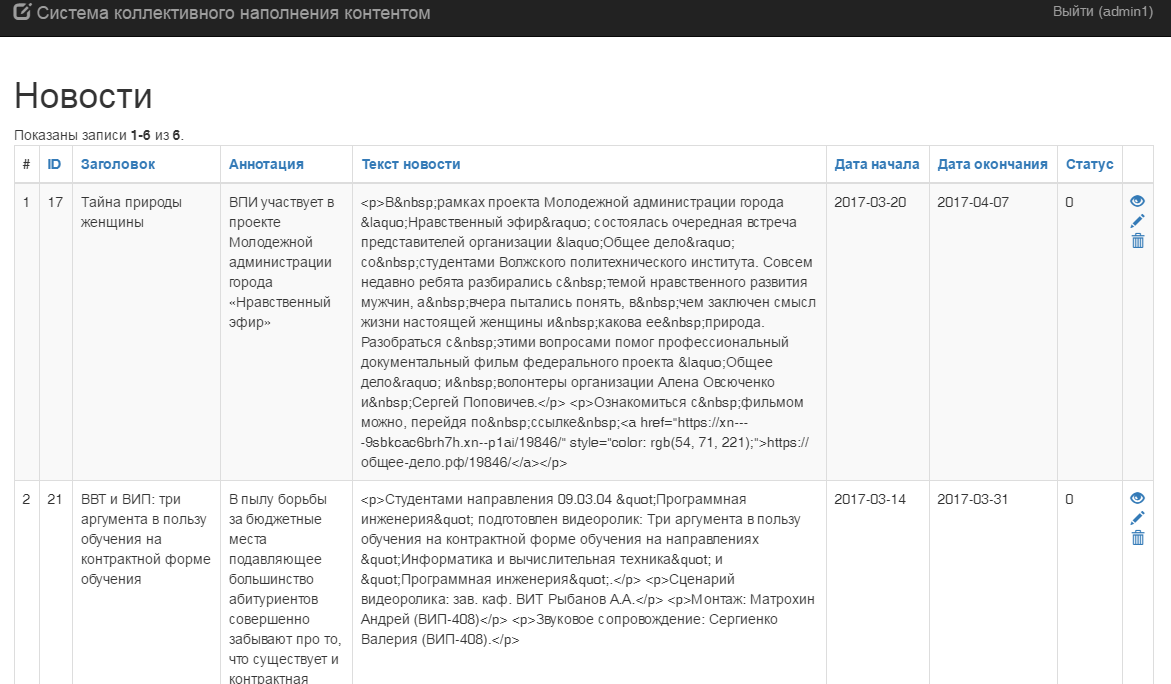


Рисунок 5 – Стартовая страница ЛК модератора

Модератор может просмотреть, отредактировать или удалить новость.

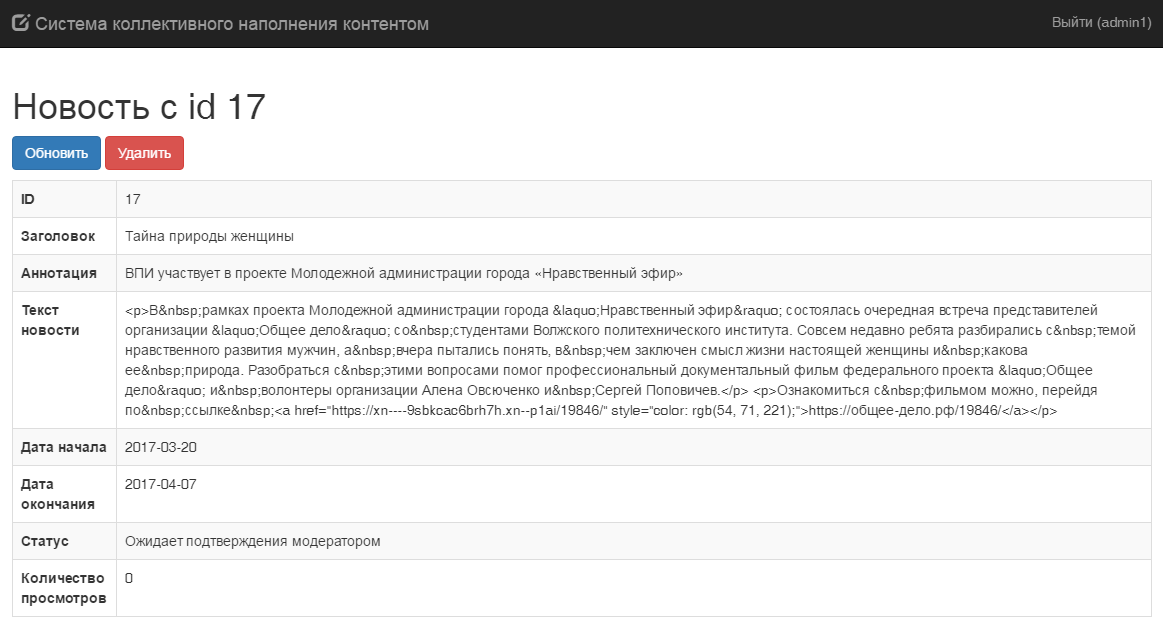


Рисунок 6 – Просмотр новости

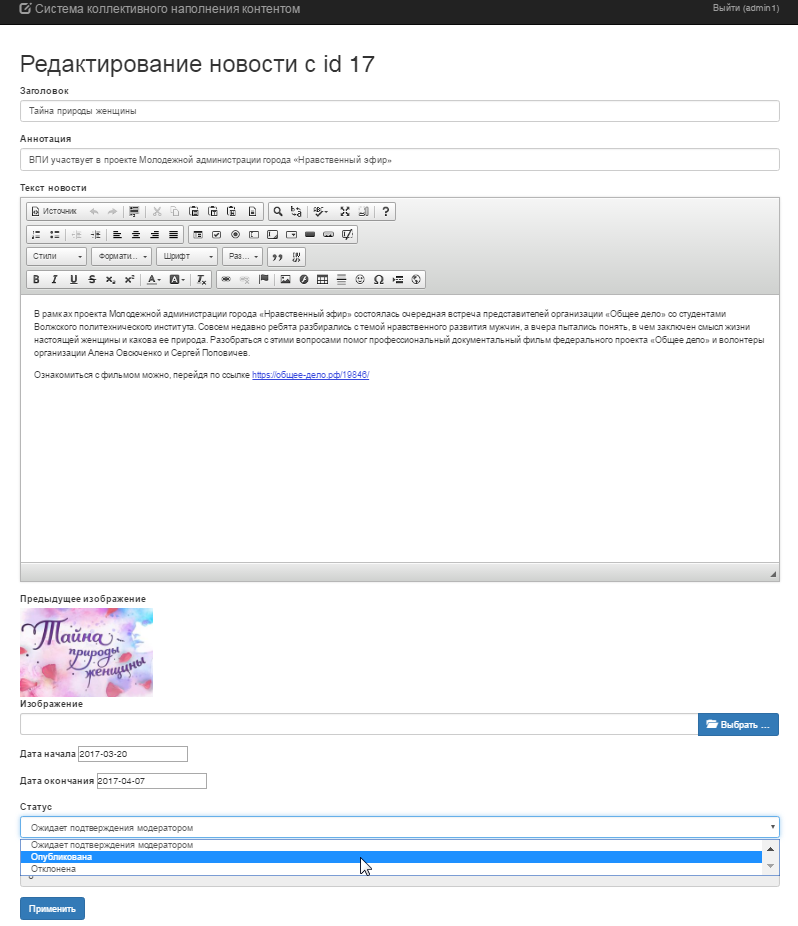


Рисунок 7 – Редактирование новости

## Раздел «Новости».

Любой посетитель сайта может просмотреть опубликованные новости, воспользовавшись соответствующим пунктом меню в правом верхнем углу сайта.

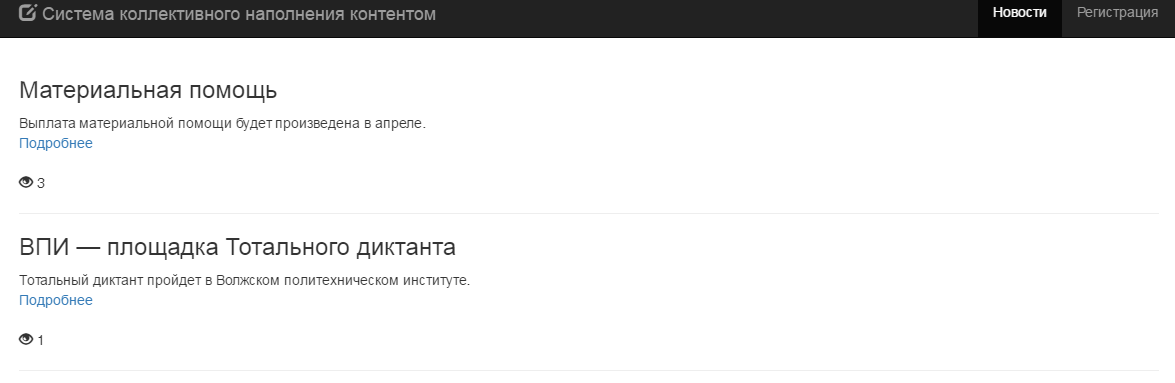


Рисунок 8 – Опубликованные новости

Новости выводятся списком, с сортировкой от самых просматриваемых к наименее просматриваемым. Также осуществляется проверка, что текущая дата принадлежит периоду публикации новости.

Для новости выводится заголовок, аннотация, ссылка «Подробнее» и количество просмотров. После нажатия ссылки «Подробнее» посетителю выдаётся страница с изображением и полным текстом новости.

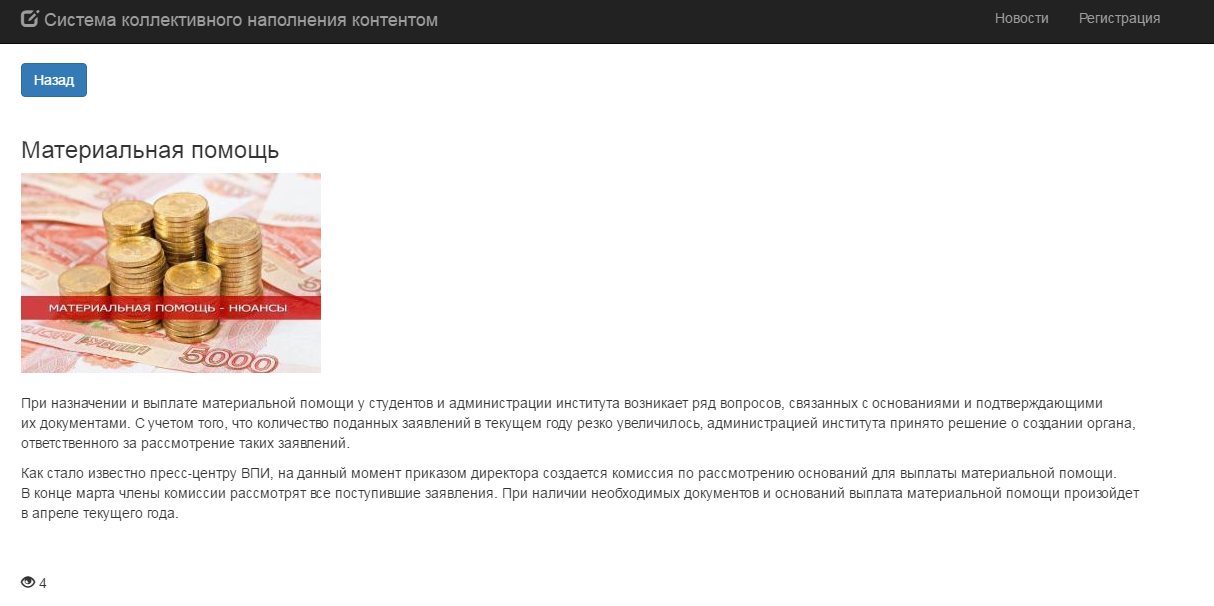


Рисунок 9 – Просмотр выбранной новости

С помощью кнопки «*Назад*» посетитель может вернуться к списку новостей.

# Руководство системного программиста

## Настройка СУБД

Запустить phpMyAdmin, создать БД w15, оставить учётную запись root с пустым паролем для доступа к ней.

Экспортировать в БД следующий SQL-файл:

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET time\_zone = "+00:00";

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- База данных: `w15`

--

-- --------------------------------------------------------

--

-- Структура таблицы `auth\_assignment`

--

CREATE TABLE IF NOT EXISTS `auth\_assignment` (

`item\_name` varchar(64) COLLATE utf8\_unicode\_ci NOT NULL,

`user\_id` varchar(64) COLLATE utf8\_unicode\_ci NOT NULL,

`created\_at` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Дамп данных таблицы `auth\_assignment`

--

INSERT INTO `auth\_assignment` (`item\_name`, `user\_id`, `created\_at`) VALUES

('admin', '1', 1486809442),

('admin', '2', 1486809442),

('author', '11', NULL),

('author', '12', NULL);

-- --------------------------------------------------------

--

-- Структура таблицы `auth\_item`

--

CREATE TABLE IF NOT EXISTS `auth\_item` (

`name` varchar(64) COLLATE utf8\_unicode\_ci NOT NULL,

`type` smallint(6) NOT NULL,

`description` text COLLATE utf8\_unicode\_ci,

`rule\_name` varchar(64) COLLATE utf8\_unicode\_ci DEFAULT NULL,

`data` blob,

`created\_at` int(11) DEFAULT NULL,

`updated\_at` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Дамп данных таблицы `auth\_item`

--

INSERT INTO `auth\_item` (`name`, `type`, `description`, `rule\_name`, `data`, `created\_at`, `updated\_at`) VALUES

('admin', 1, NULL, NULL, NULL, 1486809442, 1486809442),

('adminRole', 2, 'Администратор', NULL, NULL, 1486809442, 1486809442),

('author', 1, NULL, NULL, NULL, 1486809442, 1486809442),

('authorRole', 2, 'Автор', NULL, NULL, 1486809442, 1486809442);

-- --------------------------------------------------------

--

-- Структура таблицы `auth\_item\_child`

--

CREATE TABLE IF NOT EXISTS `auth\_item\_child` (

`parent` varchar(64) COLLATE utf8\_unicode\_ci NOT NULL,

`child` varchar(64) COLLATE utf8\_unicode\_ci NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

--

-- Дамп данных таблицы `auth\_item\_child`

--

INSERT INTO `auth\_item\_child` (`parent`, `child`) VALUES

('admin', 'adminRole'),

('author', 'authorRole');

-- --------------------------------------------------------

--

-- Структура таблицы `auth\_rule`

--

CREATE TABLE IF NOT EXISTS `auth\_rule` (

`name` varchar(64) COLLATE utf8\_unicode\_ci NOT NULL,

`data` blob,

`created\_at` int(11) DEFAULT NULL,

`updated\_at` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

-- --------------------------------------------------------

--

-- Структура таблицы `w15\_news`

--

CREATE TABLE IF NOT EXISTS `w15\_news` (

`id` int(11) NOT NULL,

`header` varchar(100) NOT NULL,

`annotation` varchar(200) NOT NULL,

`full\_text` varchar(1024) NOT NULL,

`ext\_image` varchar(4) NOT NULL,

`start\_date` date NOT NULL,

`finish\_date` date NOT NULL,

`state` tinyint(4) NOT NULL,

`count\_views` int(11) NOT NULL DEFAULT '0',

`id\_author` int(11) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=22 DEFAULT CHARSET=utf8;

--

-- Дамп данных таблицы `w15\_news`

--

INSERT INTO `w15\_news` (`id`, `header`, `annotation`, `full\_text`, `ext\_image`, `start\_date`, `finish\_date`, `state`, `count\_views`, `id\_author`) VALUES

(16, 'ВПИ — площадка Тотального диктанта', 'Тотальный диктант пройдет в Волжском политехническом институте.', '<p>Приглашаем принять участие в&nbsp;Тотальном диктанте&nbsp;&mdash; ежегодной образовательной акции, которая состоится 8&nbsp;апреля 2017 года в&nbsp;14:00 и&nbsp;впервые пройдет на&nbsp;площадке Волжского политехнического института. Всего на&nbsp;территории города Волжского работают две такие площадки&nbsp;&mdash; ВПИ и&nbsp;МЭИ.</p>\r\n\r\n<p>Для участия необходимо зарегистрироваться на&nbsp;официальном сайте&nbsp;<a href="http://www.totaldict.ru/" style="color: rgb(54, 71, 221);">totaldict.ru</a>. Регистрация будет открыта с&nbsp;29&nbsp;марта.</p>\r\n\r\n<p>Материал подготовлен с&nbsp;использованием информации сайта&nbsp;<a href="http://www.totaldict.ru/" style="color: rgb(54, 71, 221);">totaldict.ru</a></p>\r\n', 'jpg', '2017-03-22', '2017-04-09', 1, 1, 11),

(17, 'Тайна природы женщины', 'ВПИ участвует в проекте Молодежной администрации города «Нравственный эфир»', '<p>В&nbsp;рамках проекта Молодежной администрации города &laquo;Нравственный эфир&raquo; состоялась очередная встреча представителей организации &laquo;Общее дело&raquo; со&nbsp;студентами Волжского политехнического института. Совсем недавно ребята разбирались с&nbsp;темой нравственного развития мужчин, а&nbsp;вчера пытались понять, в&nbsp;чем заключен смысл жизни настоящей женщины и&nbsp;какова ее&nbsp;природа. Разобраться с&nbsp;этими вопросами помог профессиональный документальный фильм федерального проекта &laquo;Общее дело&raquo; и&nbsp;волонтеры организации Алена Овсюченко и&nbsp;Сергей Поповичев.</p>\r\n\r\n<p>Ознакомиться с&nbsp;фильмом можно, перейдя по&nbsp;ссылке&nbsp;<a href="https://xn----9sbkcac6brh7h.xn--p1ai/19846/" style="color: rgb(54, 71, 221);">https://общее-дело.рф/19846/</a></p>\r\n', 'jpg', '2017-03-20', '2017-04-07', 0, 0, 11),

(18, 'Материальная помощь', 'Выплата материальной помощи будет произведена в апреле.', '<p>При назначении и&nbsp;выплате материальной помощи у&nbsp;студентов и&nbsp;администрации института возникает ряд вопросов, связанных с&nbsp;основаниями и&nbsp;подтверждающими их&nbsp;документами. С&nbsp;учетом того, что количество поданных заявлений в&nbsp;текущем году резко увеличилось, администрацией института принято решение о&nbsp;создании органа, ответственного за&nbsp;рассмотрение таких заявлений.</p>\r\n\r\n<p>Как стало известно пресс-центру ВПИ, на&nbsp;данный момент приказом директора создается комиссия по&nbsp;рассмотрению оснований для выплаты материальной помощи. В&nbsp;конце марта члены комиссии рассмотрят все поступившие заявления. При наличии необходимых документов и&nbsp;оснований выплата материальной помощи произойдет в&nbsp;апреле текущего года.</p>\r\n', 'jpg', '2017-03-20', '2017-04-18', 1, 4, 11),

(19, 'Практика по программированию встраиваемых систем автобусов', '17 марта студенты направлений «Информатика и вычислительная техника» и «Программная инженерия» посетили с экскурсией ООО «Волгабас Волжский».', '<p>Экскурсию проводил инженер-программист встраиваемых систем ООО&nbsp;&laquo;Волгабас Волжский&raquo; Билялов Мухамед Харисович.</p>\r\n\r\n<p>Информационные технологии&nbsp;&mdash; сама эта сфера и&nbsp;её&nbsp;специалисты на&nbsp;современном рынке труда пользуются огромнейшей популярностью. Инженер-программист встраиваемых систем (Embedded Software Engineer)&nbsp;&mdash; одна из&nbsp;самых востребованных профессий в&nbsp;данной сфере.</p>\r\n\r\n<p>По&nbsp;результатам экскурсии зав. каф. Рыбанов А. А. и&nbsp;инженер-программист встраиваемых систем ООО&nbsp;&laquo;Волгабас Волжский&raquo; Билялов М. Х. согласовали ряд тем выпускных квалификационных работ по&nbsp;направлениям &laquo;Информатика и&nbsp;вычислительная техника&raquo; и&nbsp;&laquo;Программная инженерия&raquo; на&nbsp;2017/2018 учебный год.</p>\r\n', 'jpg', '2017-03-14', '2017-03-28', 2, 0, 11),

(20, 'ВПИ посетил президент', 'Президент Волжской торгово-промышленной палаты провел лекцию для студентов института.', '<p>Открытая лекция президента Волжской торгово-промышленной палаты и&nbsp;депутата городской Думы Владимира Глухова стала центральным событием первого весеннего дня для преподавателей, бакалавров и&nbsp;магистрантов кафедры экономики и&nbsp;менеджмента ВПИ. Анализируя проблемы развития предпринимательской среды Волжского в&nbsp;условиях глобальной нестабильности, Владимир Николаевич заметил: &laquo;Я&nbsp;очень рад, что сегодня Россия, развиваясь, не&nbsp;забывает про молодёжь. На&nbsp;сегодняшний день в&nbsp;нашем городе порядка десяти тысяч предпринимателей. Каждый год появляется несколько сот новых предпринимателей&raquo;.</p>\r\n\r\n<p>Владимир Николаевич пояснил, как в&nbsp;Волжской торгово-промышленной палате оказывают поддержку самой активной и&nbsp;деловой части студенчества, самостоятельно делающей первые шаги в&nbsp;малом бизнесе, той молодежи, которая хочет заниматься предпринимательством или развивать уже своё имеющееся дело.</p>\r\n', 'jpg', '2017-03-03', '2017-03-15', 1, 0, 11),

(21, 'ВВТ и ВИП: три аргумента в пользу обучения на контрактной форме обучения', 'В пылу борьбы за бюджетные места подавляющее большинство абитуриентов совершенно забывают про то, что существует и контрактная форма обучения.', '<p>Студентами направления 09.03.04 &quot;Программная инженерия&quot; подготовлен видеоролик: Три аргумента в пользу обучения на контрактной форме обучения на направлениях &quot;Информатика и вычислительная техника&quot; и &quot;Программная инженерия&quot;.</p>\r\n\r\n<p>Сценарий видеоролика: зав. каф. ВИТ Рыбанов А.А.</p>\r\n\r\n<p>Монтаж: Матрохин Андрей (ВИП-408)</p>\r\n\r\n<p>Звуковое сопровождение: Сергиенко Валерия (ВИП-408).</p>\r\n', 'jpg', '2017-03-14', '2017-03-31', 0, 0, 11);

-- --------------------------------------------------------

--

-- Структура таблицы `w15\_users`

--

CREATE TABLE IF NOT EXISTS `w15\_users` (

`id` int(11) NOT NULL,

`username` varchar(32) NOT NULL,

`password` varchar(32) NOT NULL,

`is\_admin` tinyint(4) NOT NULL,

`auth\_key` varchar(32) NOT NULL,

`access\_token` varchar(32) NOT NULL

) ENGINE=InnoDB AUTO\_INCREMENT=13 DEFAULT CHARSET=utf8 COMMENT='Пользователи';

--

-- Дамп данных таблицы `w15\_users`

--

INSERT INTO `w15\_users` (`id`, `username`, `password`, `is\_admin`, `auth\_key`, `access\_token`) VALUES

(1, 'admin1', 'e00cf25ad42683b3df678c61f42c6bda', 1, '', ''),

(2, 'admin2', 'c84258e9c39059a89ab77d846ddab909', 1, '', ''),

(11, 'author1', 'b312ba4ffd5245fa2a1ab819ec0d0347', 0, 'Ao0gYO1tnsT3NSLP5QHZdFP9Z5BsdC7K', ''),

(12, '123', '4297f44b13955235245b2497399d7a93', 0, 'KlKlQcMJnvVzs6h7pjqMw5CvlEmPIPwZ', '');

--

-- Индексы сохранённых таблиц

--

--

-- Индексы таблицы `auth\_assignment`

--

ALTER TABLE `auth\_assignment`

ADD PRIMARY KEY (`item\_name`,`user\_id`);

--

-- Индексы таблицы `auth\_item`

--

ALTER TABLE `auth\_item`

ADD PRIMARY KEY (`name`),

ADD KEY `rule\_name` (`rule\_name`),

ADD KEY `idx-auth\_item-type` (`type`);

--

-- Индексы таблицы `auth\_item\_child`

--

ALTER TABLE `auth\_item\_child`

ADD PRIMARY KEY (`parent`,`child`),

ADD KEY `child` (`child`);

--

-- Индексы таблицы `auth\_rule`

--

ALTER TABLE `auth\_rule`

ADD PRIMARY KEY (`name`);

--

-- Индексы таблицы `w15\_news`

--

ALTER TABLE `w15\_news`

ADD PRIMARY KEY (`id`);

--

-- Индексы таблицы `w15\_users`

--

ALTER TABLE `w15\_users`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT для сохранённых таблиц

--

--

-- AUTO\_INCREMENT для таблицы `w15\_news`

--

ALTER TABLE `w15\_news`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=22;

--

-- AUTO\_INCREMENT для таблицы `w15\_users`

--

ALTER TABLE `w15\_users`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=13;

--

-- Ограничения внешнего ключа сохраненных таблиц

--

--

-- Ограничения внешнего ключа таблицы `auth\_assignment`

--

ALTER TABLE `auth\_assignment`

ADD CONSTRAINT `auth\_assignment\_ibfk\_1` FOREIGN KEY (`item\_name`) REFERENCES `auth\_item` (`name`) ON DELETE CASCADE ON UPDATE CASCADE;

--

-- Ограничения внешнего ключа таблицы `auth\_item`

--

ALTER TABLE `auth\_item`

ADD CONSTRAINT `auth\_item\_ibfk\_1` FOREIGN KEY (`rule\_name`) REFERENCES `auth\_rule` (`name`) ON DELETE SET NULL ON UPDATE CASCADE;

--

-- Ограничения внешнего ключа таблицы `auth\_item\_child`

--

ALTER TABLE `auth\_item\_child`

ADD CONSTRAINT `auth\_item\_child\_ibfk\_1` FOREIGN KEY (`parent`) REFERENCES `auth\_item` (`name`) ON DELETE CASCADE ON UPDATE CASCADE,

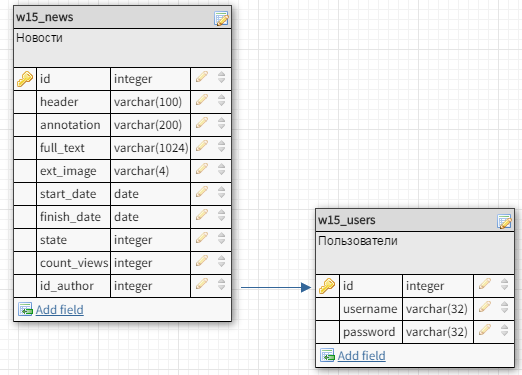
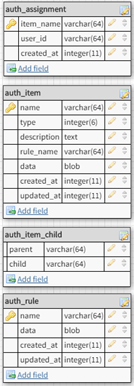
ADD CONSTRAINT `auth\_item\_child\_ibfk\_2` FOREIGN KEY (`child`) REFERENCES `auth\_item` (`name`) ON DELETE CASCADE ON UPDATE CASCADE;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

# Схема БД

# Структура полей таблиц БД

|  |  |  |
| --- | --- | --- |
| **w5\_authors** | **Пользователи системы** | |
| id (PK) | integer | ID пользователя |
| username | varchar(32) | Имя пользователя |
| password | varchar(32) | Пароль |
| **w15\_news** | **Новости** | |
| id (PK) | integer | ID новости |
| header | varchar(100) | Заголовок |
| annotation | varchar(200) | Аннотация |
| full\_text | varchar(1024) | Полный текст новости |
| ext\_image | varchar(4) | Расширение файла с изображением |
| start\_date | date | Дата начала публикации |
| finish\_date | date | Дата окончания публикации |
| state | smallint | Статус новости |
| count\_views | integer | Количество просмотров новости |
| id\_author | integer | ID пользователя |

# UML (последовательность действий)

Автор

Модератор

Система

1.авторизация

2.проверка данных

3. просмотр новостей

4. модерация и удаление новостей

5.выход из системы

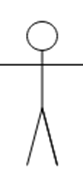
3. добавить новость

1.авторизация

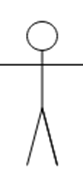
2.проверка данных

5.выход из системы

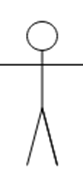
# UML (сценарий использования)



автор



модератор



посетитель

# Список литературы:

1. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя.-М.:ДМК ПРЕСС; СПб.:Питер,2004.-429 с.
2. Зандстра М., PHP: объекты, шаблоны и методики программирования, 3-е издание = PHP Objects, Patterns and Practice, Third Edition — М.: [«Вильямс»](http://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2010. — С. 560. — [ISBN 978-5-8459-1689-1](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:BookSources/9785845916891).
3. Суэринг С., Конверс Т., Джойс П. PHP и MySQL. Библия программиста, 2-е издание = PHP 6 and MySQL 6 Bible — М.: [«Диалектика»](http://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2010. — 912 с. — [ISBN 978-5-8459-1640-2](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:BookSources/9785845916402).
4. Кормен Т., Лейзерсон И. Ч., Ривест Р. Л., Штайн К. Алгоритмы: построение и анализ = INTRODUCTION TO ALGORITHMS — 2-е изд. — М.: [«Вильямс»](http://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2006. — С. 1296. — [ISBN 0-07-013151-1](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:BookSources/0070131511).
5. [Кнут](http://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%B4_%D0%9A%D0%BD%D1%83%D1%82) Д. Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. Fundamental Algorithms — 3-е изд. — М.: [«Вильямс»](http://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2006. — С. 720. — [ISBN 0-201-89683-4](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:BookSources/0201896834)
6. Титтел Э., Ноубл Дж. HTML, XHTML и CSS для чайников, 7-е издание = HTML, XHTML & CSS For Dummies, 7th Edition — М.: [«Диалектика»](http://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2011. — 400 с. — [ISBN 978-5-8459-1752-2](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:BookSources/9785845917522).
7. Zakas N., McPeak J., Fawcett J. Professional Ajax — 2nd ed. — Wrox, 2007. — 624 p. — (Programmer to Programmer). — [ISBN 0470109491](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:BookSources/0470109491).
8. Lindley C. jQuery Cookbook. Solutions & Examples for jQuery Developers. O'Reilly Media, 2009. – 478 c.