**Министерство науки и высшего образования РФ**

**Государственное образовательное учреждение высшего профессионального образования**

**Белгородский Государственный Технологический Университет им В. Г. Шухова**

Кафедра программного обеспечения вычислительной техники и

автоматизированных систем.

Лабораторная работа №4 по дисциплине

«Технологии Web-программирования»

Тема: Разработка и проектирование базы данных

Выполнил:

студент группы ПВ-41

Колесников В.Д.

Проверил: Картамышев С. В.

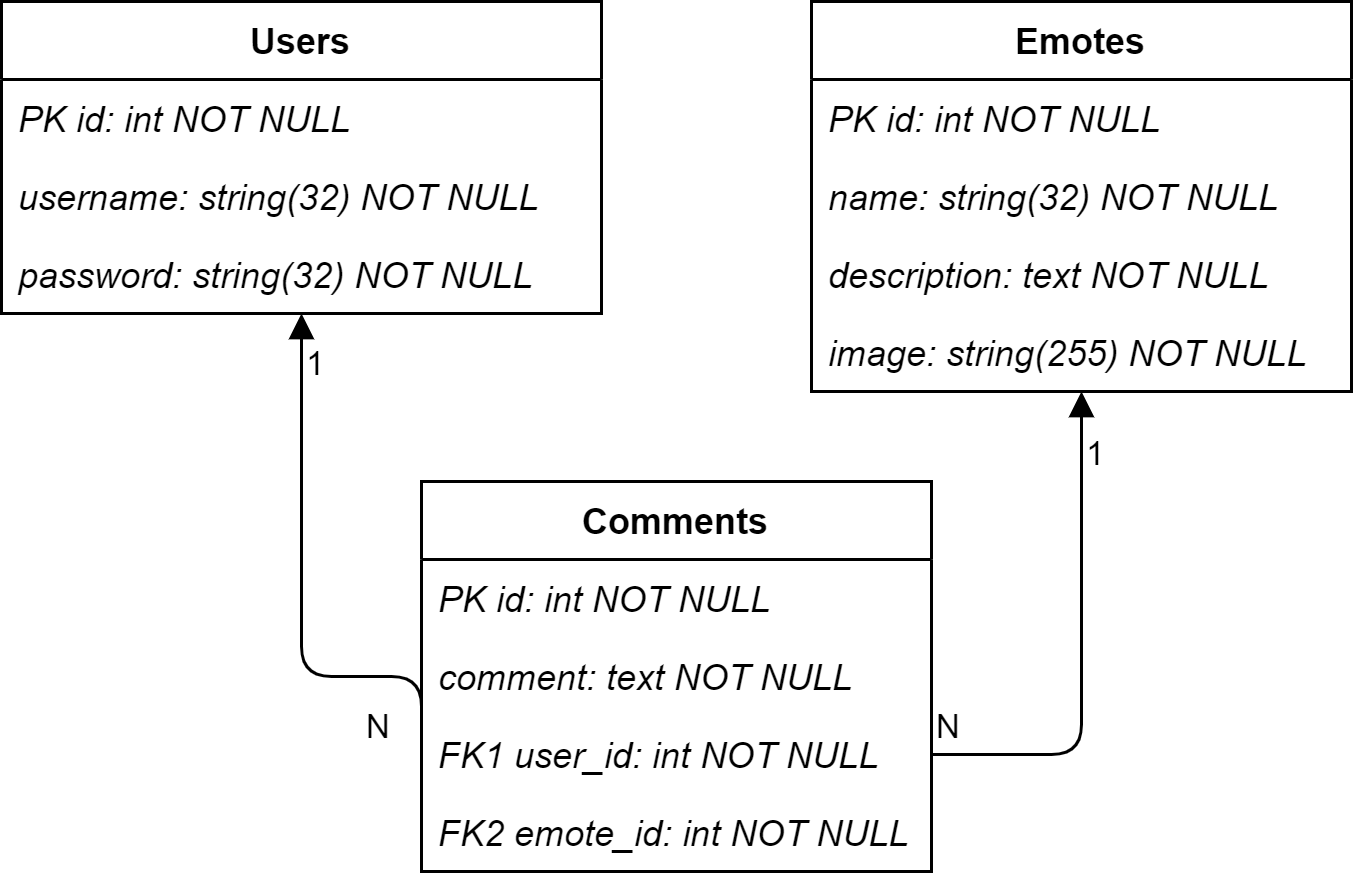
Белгород 2019 г.

**Цель работы:** изучить основы взаимодействия web-приложения с базой данных. Спроектировать базу данных для хранения информации приложения (страницы, пользователи и т.п.) и создать модели данных из базы данных.

**Выполение**

Предполагается следующая структура приложения: основной альбом содержит набор эмоций, имеющих имя, описание и изображение. Зарегистрированные пользователи (имеющие логин и пароль) могут просматривать, добавлять, редактировать и изменять вышеописанные эмоции. Также под каждой эмоцией любой пользователь может оставить свой собственный комментарий.

Итого получается следующая структура бд:



Для реализации данной структуры необходимо создать миграции:

**create\_comments\_table**

<?php

use yii\db\Migration;

/\*\*

\* Handles the creation of table `{{%comments}}`.

\*/

class m201204\_150646\_create\_comments\_table extends Migration

{

/\*\*

\* {@inheritdoc}

\*/

public function safeUp()

{

$this->createTable('{{%comments}}', [

'id' => $this->primaryKey(),

'comment' => $this->text()->notNull()->comment('Comment text'),

'createdAt' => $this->dateTime()->notNull()->comment('Creation time'),

'updatedAt' => $this->dateTime()->notNull()->comment('Update time')

]);

}

/\*\*

\* {@inheritdoc}

\*/

public function safeDown()

{

$this->dropForeignKey('fk\_comments\_user\_id', '{{%comments}}');

$this->dropColumn('{{%comments}}', 'user\_id');

$this->dropTable('{{%comments}}');

}

}

**create\_users\_table**

<?php

use yii\db\Migration;

/\*\*

\* Handles the creation of table `{{%users}}`.

\*/

class m201204\_151307\_create\_users\_table extends Migration

{

/\*\*

\* {@inheritdoc}

\*/

public function safeUp()

{

$this->createTable('{{%users}}', [

'id' => $this->primaryKey(),

'username' => $this->string(32)->notNull()->comment('Username'),

'password' => $this->string(32)->notNull()->comment('Password'),

'createdAt' => $this->dateTime()->notNull()->comment('Creation time'),

'updatedAt' => $this->dateTime()->comment('Update time')

]);

$this->addColumn('{{%comments}}', 'user\_id', $this->integer()->comment('User'));

$this->addForeignKey('fk\_comments\_user\_id', '{{%comments}}', 'user\_id', '{{%users}}', 'id', 'SET NULL');

}

/\*\*

\* {@inheritdoc}

\*/

public function safeDown()

{

$this->dropTable('{{%users}}');

}

}

**create\_emotes\_table**

<?php

use yii\db\Migration;

/\*\*

\* Handles the creation of table `{{%emotes}}`.

\*/

class m201204\_151413\_create\_emotes\_table extends Migration

{

/\*\*

\* {@inheritdoc}

\*/

public function safeUp()

{

$this->createTable('{{%emotes}}', [

'id' => $this->primaryKey(),

'name' => $this->string(32)->notNull()->comment('Name'),

'description' => $this->text()->notNull()->comment('Description'),

'image' => $this->string()->notNull()->comment('Image'),

'createdAt' => $this->dateTime()->notNull()->comment('Creation time'),

'updatedAt' => $this->dateTime()->comment('Update time')

]);

$this->addColumn('{{%comments}}', 'emote\_id', $this->integer()->comment('Emote'));

$this->addForeignKey('fk\_comments\_emote\_id', '{{%comments}}', 'emote\_id', '{{%emotes}}', 'id', 'SET NULL');

}

/\*\*

\* {@inheritdoc}

\*/

public function safeDown()

{

$this->dropForeignKey('fk\_comments\_emote\_id', '{{%comments}}');

$this->dropColumn('{{%comments}}', 'emote\_id');

$this->dropTable('{{%emotes}}');

}

}

Далее по созданным миграциям генерируются модели:

**BaseModel**

<?php

namespace app\modules\v1\models;

use yii\behaviors\TimestampBehavior;

use yii\db\ActiveRecord;

use yii\db\Expression;

class BaseModel extends ActiveRecord {

/\*\*

\* {@inheritdoc}

\*/

public function behaviors() {

return [

[

'class' => TimestampBehavior::class,

'createdAtAttribute' => 'createdAt',

'updatedAtAttribute' => 'updatedAt',

'value' => new Expression('now()')

]

];

}

}

**Comment**

<?php

namespace app\modules\v1\models;

use Yii;

/\*\*

\* This is the model class for table "comments".

\*

\* @property int $id

\* @property string $comment Comment text

\* @property string $createdAt Creation time

\* @property string $updatedAt Update time

\* @property int|null $user\_id User

\* @property int|null $emote\_id Emote

\*

\* @property Users $user

\* @property Emotes $emote

\*/

class Comment extends BaseModel

{

/\*\*

\* {@inheritdoc}

\*/

public static function tableName()

{

return 'comments';

}

/\*\*

\* {@inheritdoc}

\*/

public function rules()

{

return [

[['comment', 'createdAt', 'updatedAt'], 'required'],

[['comment'], 'string'],

[['createdAt', 'updatedAt'], 'safe'],

[['user\_id', 'emote\_id'], 'integer'],

[['user\_id'], 'exist', 'skipOnError' => true, 'targetClass' => Users::className(), 'targetAttribute' => ['user\_id' => 'id']],

[['emote\_id'], 'exist', 'skipOnError' => true, 'targetClass' => Emotes::className(), 'targetAttribute' => ['emote\_id' => 'id']],

];

}

/\*\*

\* {@inheritdoc}

\*/

public function attributeLabels()

{

return [

'id' => 'ID',

'comment' => 'Comment text',

'createdAt' => 'Creation time',

'updatedAt' => 'Update time',

'user\_id' => 'User',

'emote\_id' => 'Emote',

];

}

/\*\*

\* Gets query for [[User]].

\*

\* @return \yii\db\ActiveQuery

\*/

public function getUser()

{

return $this->hasOne(Users::className(), ['id' => 'user\_id']);

}

/\*\*

\* Gets query for [[Emote]].

\*

\* @return \yii\db\ActiveQuery

\*/

public function getEmote()

{

return $this->hasOne(Emotes::className(), ['id' => 'emote\_id']);

}

}

**User**

<?php

namespace app\modules\v1\models;

use Yii;

/\*\*

\* This is the model class for table "users".

\*

\* @property int $id

\* @property string $username Username

\* @property string $password Password

\* @property string $createdAt Creation time

\* @property string|null $updatedAt Update time

\*

\* @property Comments[] $comments

\*/

class User extends BaseModel

{

/\*\*

\* {@inheritdoc}

\*/

public static function tableName()

{

return 'users';

}

/\*\*

\* {@inheritdoc}

\*/

public function rules()

{

return [

[['username', 'password', 'createdAt'], 'required'],

[['createdAt', 'updatedAt'], 'safe'],

[['username', 'password'], 'string', 'max' => 32],

];

}

/\*\*

\* {@inheritdoc}

\*/

public function attributeLabels()

{

return [

'id' => 'ID',

'username' => 'Username',

'password' => 'Password',

'createdAt' => 'Creation time',

'updatedAt' => 'Update time',

];

}

/\*\*

\* Gets query for [[Comments]].

\*

\* @return \yii\db\ActiveQuery

\*/

public function getComments()

{

return $this->hasMany(Comments::className(), ['user\_id' => 'id']);

}

}

**Emote**

<?php

namespace app\modules\v1\models;

use Yii;

/\*\*

\* This is the model class for table "emotes".

\*

\* @property int $id

\* @property string $name Name

\* @property string $description Description

\* @property string $image Image

\* @property string $createdAt Creation time

\* @property string|null $updatedAt Update time

\*

\* @property Comments[] $comments

\*/

class Emote extends BaseModel

{

/\*\*

\* {@inheritdoc}

\*/

public static function tableName()

{

return 'emotes';

}

/\*\*

\* {@inheritdoc}

\*/

public function rules()

{

return [

[['name', 'description', 'image', 'createdAt'], 'required'],

[['description'], 'string'],

[['createdAt', 'updatedAt'], 'safe'],

[['name'], 'string', 'max' => 32],

[['image'], 'string', 'max' => 255],

];

}

/\*\*

\* {@inheritdoc}

\*/

public function attributeLabels()

{

return [

'id' => 'ID',

'name' => 'Name',

'description' => 'Description',

'image' => 'Image',

'createdAt' => 'Creation time',

'updatedAt' => 'Update time',

];

}

/\*\*

\* Gets query for [[Comments]].

\*

\* @return \yii\db\ActiveQuery

\*/

public function getComments()

{

return $this->hasMany(Comments::className(), ['emote\_id' => 'id']);

}

}

**Вывод:** в результате выполнения работы мы изучили основы взаимодействия web-приложения с базой данных, спроектировали базу данных для хранения информации приложения (новости, пользователи, комментарии) и создали модели таблиц данных из самой базы данных.