



InstagramMephi

Подготовили студенты группы Б04-43
Калинин Александр и Игнатущенко Сергей

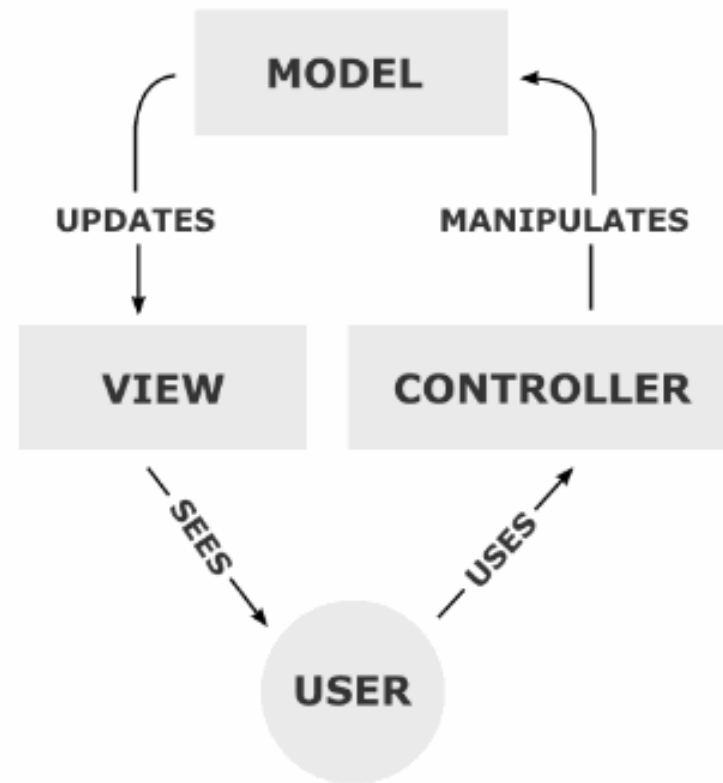
- Введение
- Паттерн проектирования MVC
- Используемые Framework'и
- Функциональность проекта
- Еще немного о проекте

Паттерн проектирования MVC

Паттерн проектирования - повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

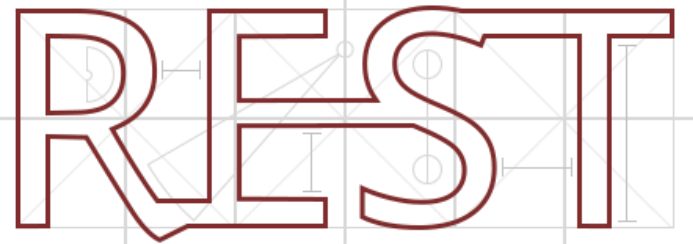
Паттерн проектирования MVC

Model-view-controller (MVC, «модель-представление-контроллер», «модель-вид-контроллер») – паттерн, при котором модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы изменение одного из компонентов оказывала минимальное воздействие на остальные.



Используемые Framework'и

django



framework



ANGULARJS
by Google™

Django

Django (Джанго) — свободный framework для веб-приложений на языке Python , использующий шаблон проектирования MVC.



Django

Отличительные особенности:

- Понятная документация
- Интуитивно понятный интерфейс администрирования с возможностью расширения
- Абстрагирование от низкого уровня баз данных
- Разделение контента и представления с помощью шаблонов
- Малое число внешних зависимостей удобно для пользования: все, что нам нужно – это питон
- Бесплатность
- Диспетчер URL

Django

Сайты и приложения на Python / Django:

Instagram



the ONION

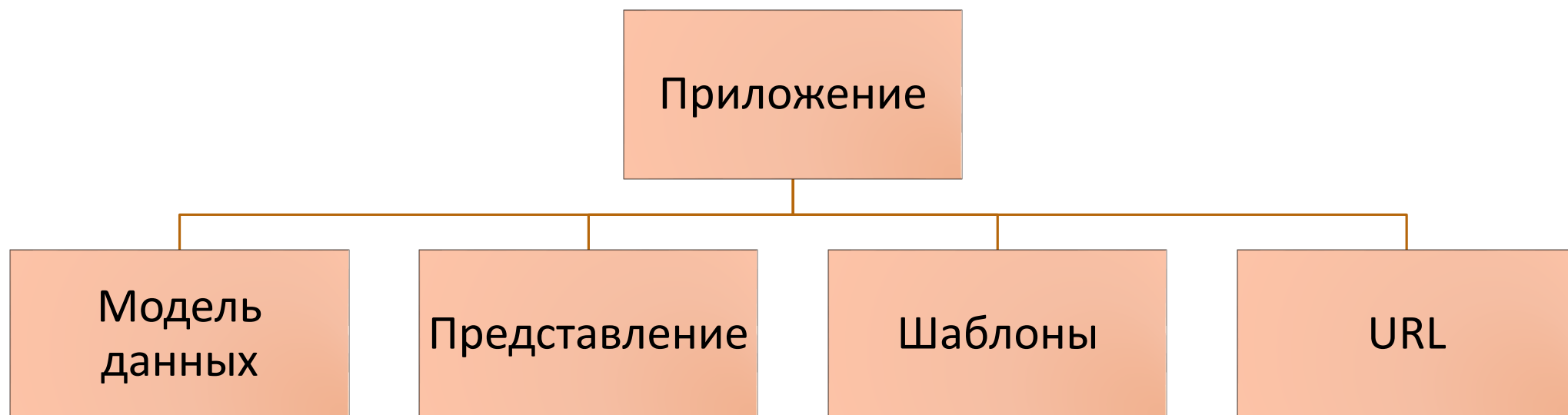
Pinterest



 **Discovery**
CHANNEL

Django

Django – приложение состоит из четырех основных компонентов:



Django

Пример модели:

```
from django.db import models
```

```
class Photo(models.Model):  
    class Meta:  
        db_table = 'photo'  
    user = models.ForeignKey(User, related_name='photos')  
    image = models.ImageField()  
    data = models.DateTimeField(auto_now_add=True)  
    comment_count = models.IntegerField(default=0)  
    likes = models.IntegerField(default=0)  
    dislikes = models.IntegerField(default=0)
```

Django

Пример URL:

```
urlpatterns = patterns("",  
  
    url(r'^api/', include('api.urls')),  
    url(r'^users/(?P<username>[0-9a-zA-Z_-]+)$', UserDetails.as_view()),  
    url(r'^register/res', Register.as_view())  
)
```

Django

Пример представления:

```
class UserPhotoList(generics.ListAPIView):
    serializer_class = PhotoSerializer
    permission_classes = [IsAuthenticated]
    queryset = Photo.objects.all()

    def get_queryset(self):
        return self.queryset.filter(user__username=self.kwargs.get('username')).order_by('-data')
```

Django Rest Framework

Django REST framework — удобный инструмент для работы с rest основанный на идеологии фреймворка Django.

Rest (сокр. англ. Representational State Transfer, «передача состояния представления») — стиль построения архитектуры распределенного приложения.



Django Rest Framework

Отличительные особенности:

- Четкое разделение на сериализаторы, которые описывают соответствие между моделью и ее форматом представления
- Набор базовых представлений, которые позволяют легко решать распространенные задачи связанные с CRUD
- Поставляется со встроенным браузером API для тестирования вашего API.

Django Rest Framework

GET /companies/

GET /api/1.0/companies/

HTTP 200 OK

Vary: Accept

Content-Type: text/html; charset=utf-8

Allow: GET, POST, PUT, PATCH, DELETE

```
{
  "company": [
    {
      "links": {
        "projects": "http://127.0.0.1:8000/api/1.0/companies/ngen-works/projects/",
        "profiles": "http://127.0.0.1:8000/api/1.0/companies/ngen-works/profiles/",
        "invitations": "http://127.0.0.1:8000/api/1.0/companies/ngen-works/invitations/"
      },
      "url": "www.ngenworks.com",
      "id": 1,
      "name": "nGen Works",
      "slug": "ngen-works",
      "tagline": "Building beautiful experiences for a multi-screen world.",
      "logo": "",
      "phone": "(904) 399-4411",
      "location": "Jacksonville, FL",
      "owner": 2,
      "main_contact": 2,
      "date_created": "2014-06-05 14:56:09",
      "description": ""
    }
  ],
}
```

Django Rest Framework

Пример сериализатора:

```
from rest_framework import serializers
```

```
class PhotoSerializer(serializers.ModelSerializer):
```

```
    class Meta:
```

```
        model = Photo
```

```
        user = UserSerializer(required=False)
```

```
        image = serializers.ImageField()
```

```
        data = serializers.DateTimeField(format='%d %B %Y')
```

```
        comment_count = serializers.IntegerField(default=0)
```

```
        likes = serializers.IntegerField(default=0)
```

```
        dislikes = serializers.IntegerField(default=0)
```


AngularJS

AngularJS — JavaScript-фреймворк с открытым исходным кодом. Его цель — расширение браузерных приложений на основе MVC шаблона, а также упрощение тестирования и разработки.



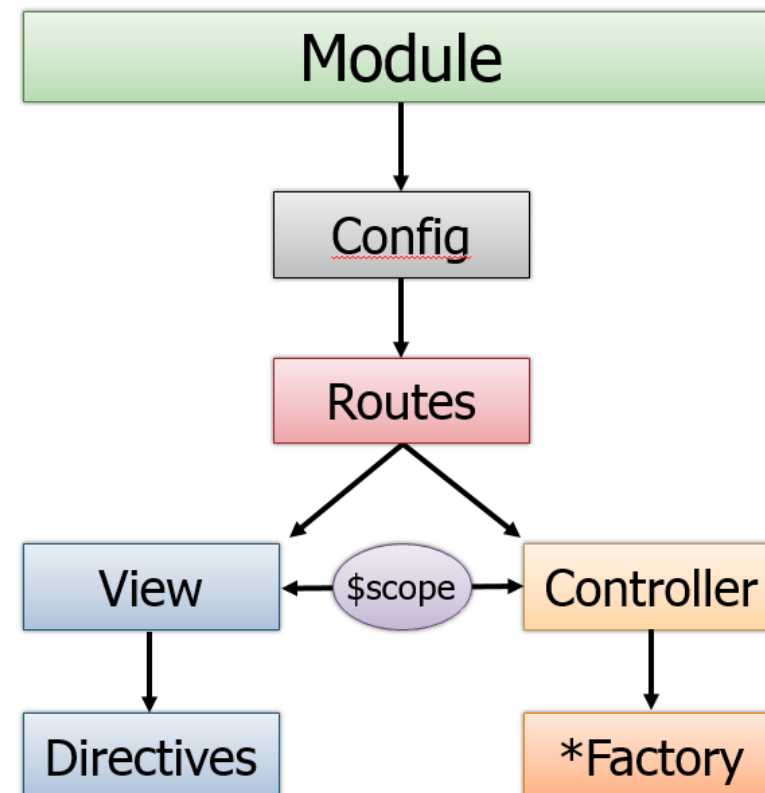
AngularJS

Отличительные особенности:

- Двустороннее связывание данных
- Позволяет использовать данные прямо в разметке
- Позволяет писать меньше кода
- Замечательная документация
- Постоянная поддержка инструмента

AngularJS

- **Module** – глобальное хранилище для вашего приложения(контроллеры, роуты, вьюхи)
- **Config** – настройки модуля
- **Routes** – “менеджер” контроллеров для url
- **Controller** – логика отдельных частей модуля
- **View** – представление html
- **\$scope** – глобальный объект со всеми данными контроллера
- **Directives** – кастомное поведение html
- **Factory** – фабрика (сервис)



AngularJS

Пример:

```
app.factory('AllPhotos',[
  '$resource', function($resource){
    return $resource('/api/photos/getallphotos',{},
      {get:{method:'GET',isArray:false}
    })
  }
]);
```

```
$scope.refresh = function() {
  $scope.all_photos = AllPhotos.get();
};
```

Функциональность проекта

- Регистрация
- Добавление и удаление фотографий
- Возможность отметить фотографию как «понравилось» или «не понравилось»
- Просмотр пользователей, которым «понравилась» или «не понравилась» фотография
- Подписка на других пользователей
- Просмотр пользователей на которых вы подписаны и тех, кто подписан на вас
- Возможность оставлять комментарии под фотографиями
- Поиск пользователей
- Лента фотографий пользователей, на которых вы подписаны
- Лента фотографий всех пользователей(на каждом обновлении показывает 25 случайных фотографий)
- Возможность загрузки фотографии просто перетаскив ее в окно браузера

Структура

- **urls.py** : Конфигурация URL-ов
- **settings.py** : Настройки/конфигурация проекта
- **models.py** : Модели
- **serializers.py** : Сериализаторы моделей
- **api.py** : Представления
- каталог **templates** : Содержатся html шаблоны
- каталог **static** : Содержатся js, css файлы

[Войти](#)[Регистрация](#)

Instagram Mephi

Запечатляйте и поделитесь моментами из жизни

Instagram-mephi это **быстрый, красивый и интересный** способ поделиться событиями из жизни с семьей и друзьями.

Сделайте фото или видео и опубликуйте на Instagram-mephi — что может быть проще? Это новое видение мира.

Ах, да, мы мы говорили, что это бесплатно?

Авторизация и регистрация



InstagramMephi

Вход в личный кабинет

admin

Войти

InstagramMephi

Регистрация

Имя пользователя:

Пароль:

Пароль еще раз:

E-mail:

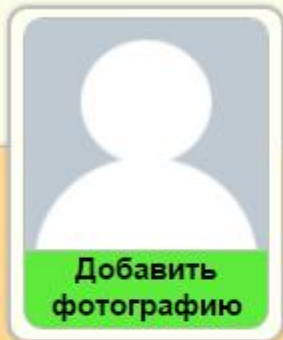
Зарегистрироваться



InstagramMephi



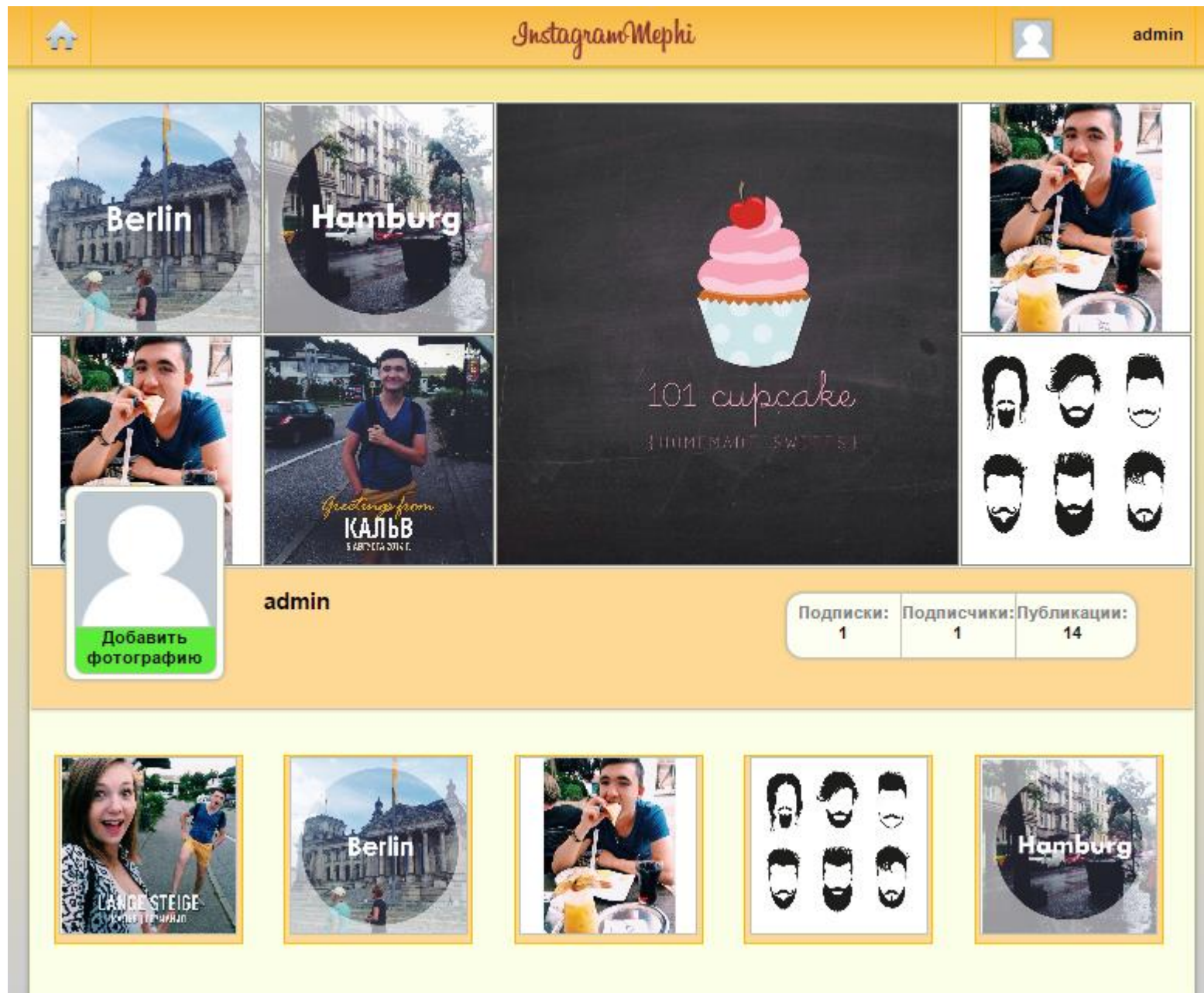
admin

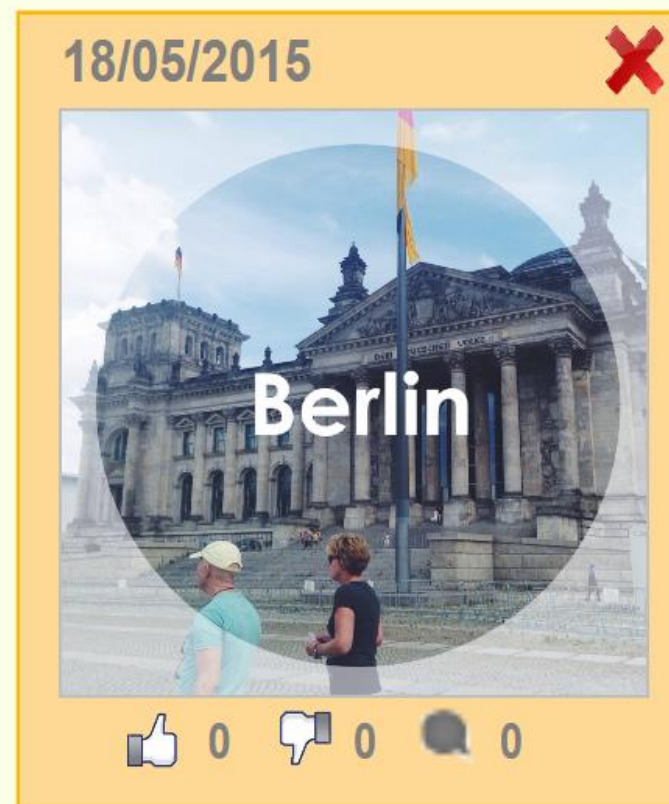
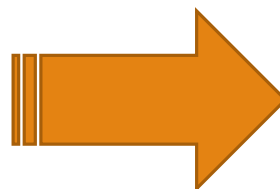


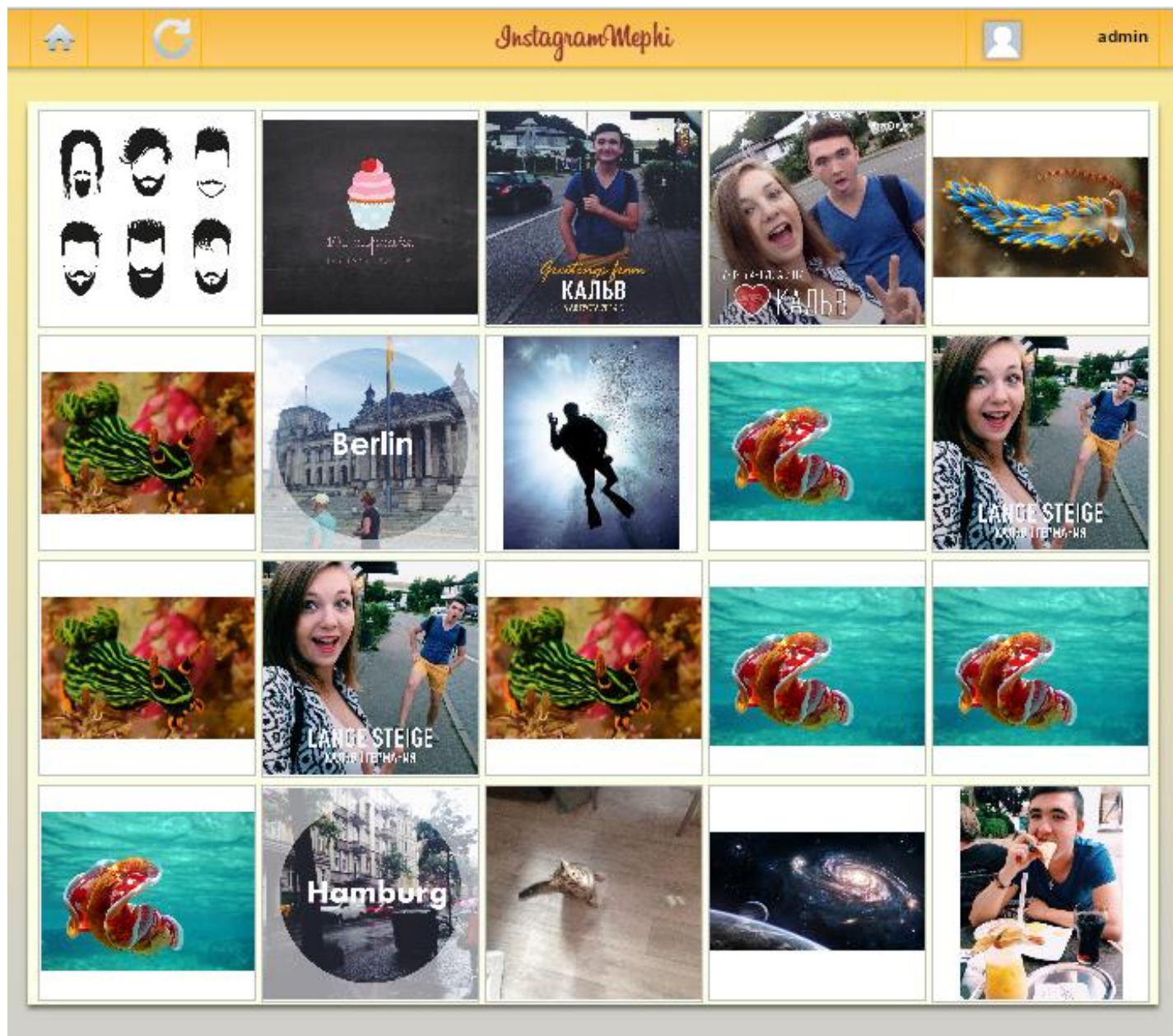
admin

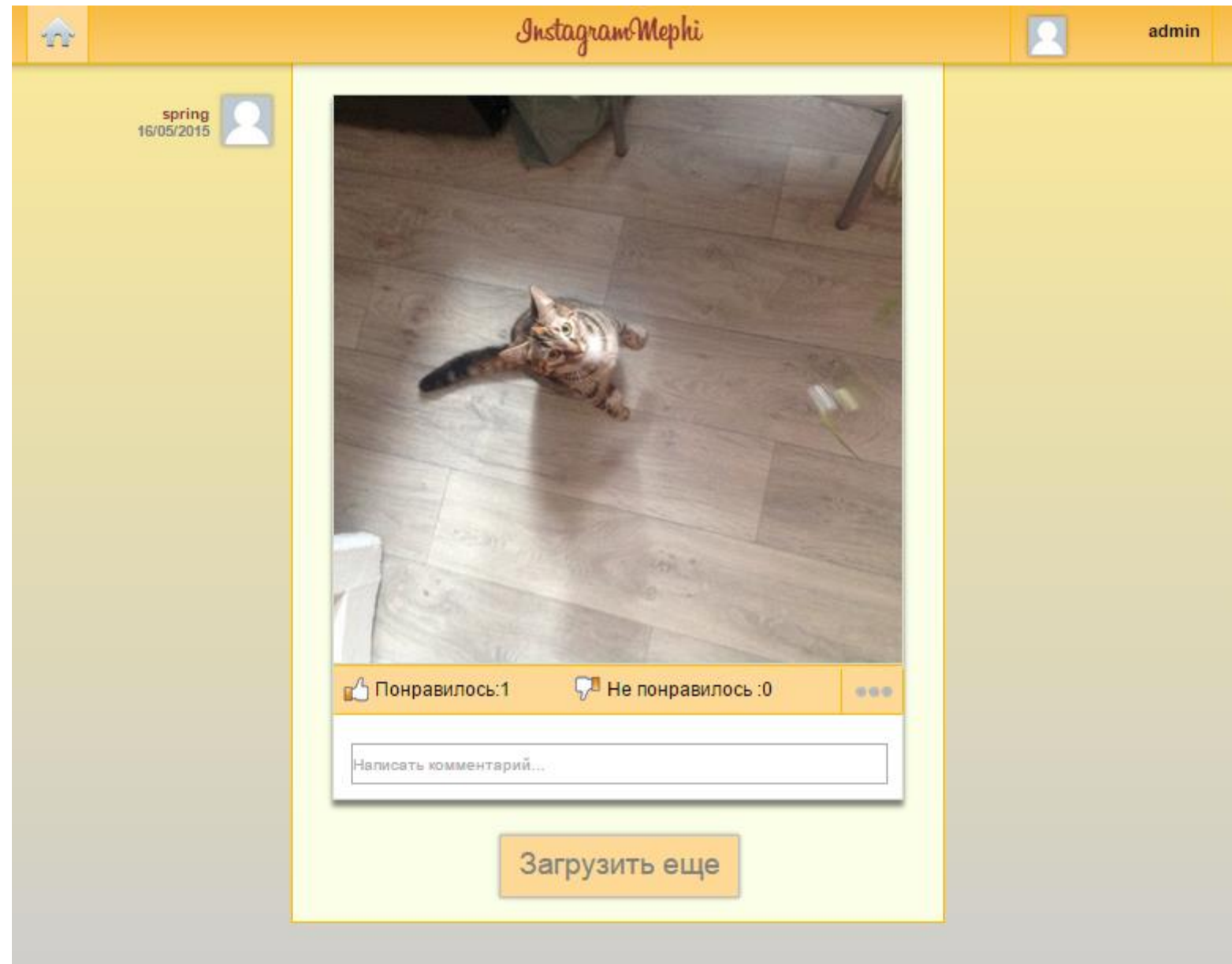
Подписки:	Подписчики:	Публикации:
1	1	5















 Понравилось:1


 Не понравилось :0




Написать комментарий...



 Понравилось:1

 Не понравилось :0



Понравилось:

Не понравилось:

spring

Написать комментарий...



Пользователи



admin



testuser1



testuser



hodor



mytest



spring



Alex



asdf



Your_mother





InstagramMephi



admin

Подписки admin



spring



asdf



testuser1



testuser



mytest

Посмотреть профиль

Найти пользователя

Лента фотографий

Выйти





Спасибо за внимание !

Ссылка на GitHub: <https://github.com/alsedi28/InstagramMEPhI>