**Продуктовые требования**

**“GeekTest – закрытая социальная сеть для тестировщиков”**

**\*запрашивать доступ к файлу не надо\***

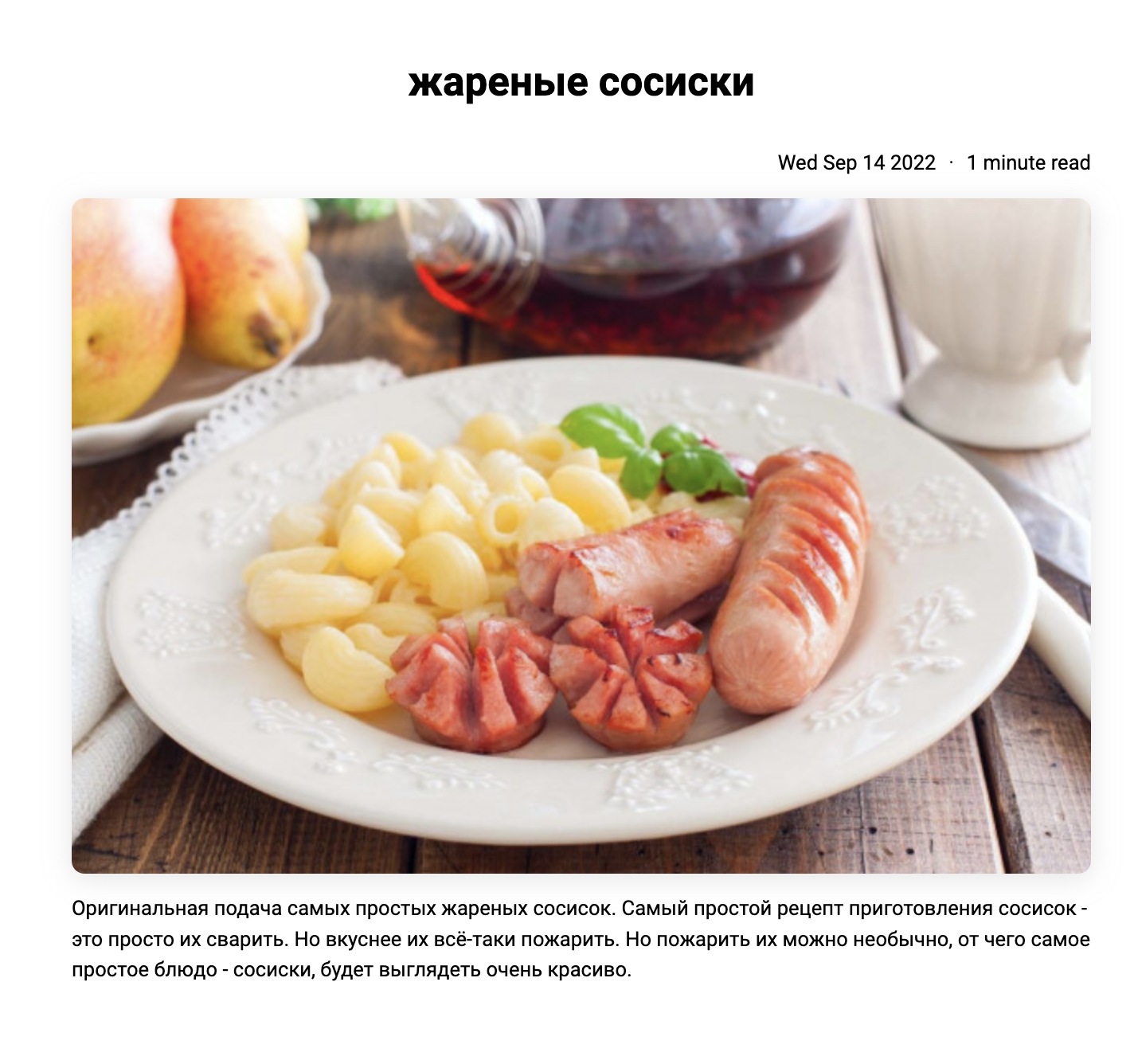
Информационная система "GeekTest – закрытая cоциальная сеть для тестировщиков" предназначена для общения начинающих и опытных QA-специалистов.

**2.4. Страница поста**

Детальное отображение поста. Показывает заголовок, фотографию и контент.

**URL:** https://test-stand.gb.ru/posts/{number}

**Дизайн:** <https://miro.com/app/board/uXjVPXrEyJU=/>

****

**Пользовательский сценарий**

**Пользователь:** нажимает на пост

**Система:** открывает экран просмотра поста

**Пользователь:** может просматривать, редактировать или удалить свой пост; может только просматривать чужой пост

**Требования, условия и ограничения**

| **Требование** | **Отработка ошибки** |
| --- | --- |
| Содержит дату публикации, время на чтение, тайтл, фото и контент поста | - |
| Для своего поста также содержит значки управления (редактирование, удаление) | - |
| Время на чтение рассчитывается по след алгоритму: – менее 100 символов = 1 минута  – от 100 до 400 = 3 минуты  – 400-700 = 5 минут  – больше 700 = больше 5 минут | - |
| Если нет картинки с сервера, вешается заглушка | - |

**Получение поста**

GET https://test-stand.gb.ru/api/posts/{post\_id}

В ручке передается {post\_id} – Integer значение, уникальный id поста.

В query параметрах передается timestamp – значение которого - время отправки запроса.

Если передать в query параметрах admin\_mode – значение которого - true, то можно посмотреть посты, которые опубликованы админом (это некий пользователь в системе, например, игуана - админ) уже ранее в системе, их id – 999-xxx, где xxx - любое значение от 0 до 999 (не забыть это же значение передать в {post\_id}).

В случае успешного запроса, в response мы получаем:

| {  "id": Integer,  "title": "String",  "description": "String",  "content": "String",  "authorId": Integer,  "mainImage": {  "id": Integer,  "cdnUrl": "String"  },  "updatedAt": "String",  "createdAt": "String",  "labels": [],  "delayPublishTo": null,  "draft": Boolean  } |
| --- |

где:

id – идентификатор поста,

title – название поста,

description – описание,

content – текст поста,

authorId – идентификатор пользователя,

mainImage – информация про изображение, тип значения – объект,

mainImage.id – id изображения,

mainImage.cdnUrl – ссылка на картинку,

updatedAt – дата обновления поста,

createdAt – дата создания поста,

labels – лейблы,

delayPublishTo – отложенная публикация,  
 draft – черновик [false | true].

В случае, если выполняется запрос без передачи квери параметра, то в ответе приходит пост пользователя.

Если запрашивается несуществующий пост, то в ответе получаем:

| {  "error": "critical",  "code": 404,  "message": "Такого поста не существует"  } |
| --- |

где:

error – описание типа ошибки. Всего есть два типа: critical – если пользователь запрашивает впервые чей-то пост, major – если запрашивает чей-то пост второй раз,

code – 404 – данный статус код возвращается всегда,

message – параметр, содержащий текст ошибки “такого поста не существует”.

Если отправить запрос без токена, то будет в ответе:

| {  "error": "Необходимо авторизоваться",  "code": 403,  "message": "No API token provided or is not valid"  } |
| --- |