МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

*Факультет* *компьютерных наук*

*Кафедра информационных систем*

*Веб-приложение для просмотра рецептов блюд “YummYummY”*

*Курсовой проект*

09.03.02 *Информационные системы и технологии*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_*А.Ю. Бакулин*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_*Н.А. Бородин*

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_*А.Н. Пупыкин*

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *В.С. Тарасов.*

Воронеж 2020

# Содержание

[Содержание 2](#_Toc38893688)

[Введение 3](#_Toc38893689)

[1. Постановка задачи 4](#_Toc38893690)

[2. Анализ предметной области 6](#_Toc38893691)

[3. Диаграмма прецедентов 18](#_Toc38893694)

[4. Диаграмма классов 19](#_Toc38893695)

[5. Диаграмма объектов 20](#_Toc38893696)

[6. Диаграмма последовательностей 21](#_Toc38893697)

[7. Диаграмма взаимодействия 22](#_Toc38893698)

[8. Диаграмма состояний 23](#_Toc38893699)

[9. Диаграмма активностей 24](#_Toc38893700)

[10. Диаграмма развертывания 26](#_Toc38893701)

[11. Обоснование архитектуры проекта 27](#_Toc38893702)

[12. Обоснование выбора продуктовых воронок 28](#_Toc38893703)

[13. Разработка 29](#_Toc38893704)

# Введение

В повседневной жизни многие сталкиваются с приготовлением различных блюд. Люди экспериментирует, пытаются открыть для себя новые лакомства, узнавая информацию по приготовлению на просторах интернета, в кулинарных книгах или у знакомых. Многие хранят свои рецепты на бумаге или другом носителе информации, чтобы не забыть о недавно приготовленном деликатесе.

Сборник рецептов позволяет авторам делиться своими знаниями в области кулинарии, а читателю использовать их. Тем временем веб-приложение по отображению рецептов, даст возможность каждому поделиться своим опытом с остальными пользователями и создать свой сборник руководств по приготовлению. Также человек получит доступ к просмотру содержимого базы рецептов, сформированной другими пользователями или им самим. А главное, возможность всегда будет у него под рукой. Через веб-приложение он сможет записать в свой сборник какой-нибудь только что увиденный рецепт, где бы он ни был, в отличие от ситуации с бумажным носителем, когда присутствует возможность оставить его дома.

При разработке проекта используется каскадная модель, которая включает в себя следующие этапы:

* Определение требований
* Проектирование
* Конструирование (также «реализация» либо «кодирование»)
* Воплощение
* Тестирование и отладка (также «верификация»)
* Инсталляция
* Поддержка

Также в процессе разработки будет использоваться парадигма объектно-ориентированного программирования.

# Постановка задачи

Целью курсового проекта является формирование базы рецептов блюд, создание веб-сервиса для просмотра рецептов блюд. С точки зрения клиента, целью курсового проекта является доступ к базе рецептов через Глобальную Сеть Интернет, просмотр базы рецептов блюд, а также отказ от бумажных носителей для хранения и просмотра рецептов.

В системе предусмотрены четыре уровня доступа:

* Неавторизованный пользователь
* Авторизованный заблокированный пользователь
* Авторизованный открытый пользователь
* Администратор

Неавторизованный пользователь обладает следующими возможностями:

* Регистрация
* Авторизация
* Поиск по чужим рецептам
* Получить случайный рецепт
* Пошаговый просмотр выбранного рецепта

Авторизованный открытый пользователь обладает следующими возможностями:

* Поиск по чужим рецептам
* Получить случайный рецепт
* Просмотр своих созданных рецептов
* Добавление своих рецептов
* Удаление своих рецептов
* Изменение своих рецептов
* Пошаговый просмотр выбранного рецепта
* Изменение личных данных
* Выход из системы

Авторизованный заблокированный пользователь обладает следующими возможностями:

* Поиск по чужим рецептам
* Просмотр своих созданных рецептов
* Удаление своих рецептов
* Получение случайного рецепта
* Пошаговый просмотр выбранного рецепта
* Изменение личных данных
* Выход из системы

Администратор обладает всеми возможностями авторизованного открытого пользователя, а также следующими возможностями:

* Удаление рецепта любого пользователя
* Изменение рецепта любого пользователя
* Редактирование ингредиентов и национальностей
* Блокировка пользователя

# Анализ

# Анализ предметной области

## Анализ существующих решений.

Анализ существующих решений будем производить на основе следующих сайтов:

|  |  |
| --- | --- |
| <https://www.povarenok.ru/> | Поварёнок.ру |
| <https://eda.ru/> | Еда.ру |
| <https://www.gastronom.ru/> | Гастрономъ.ру |

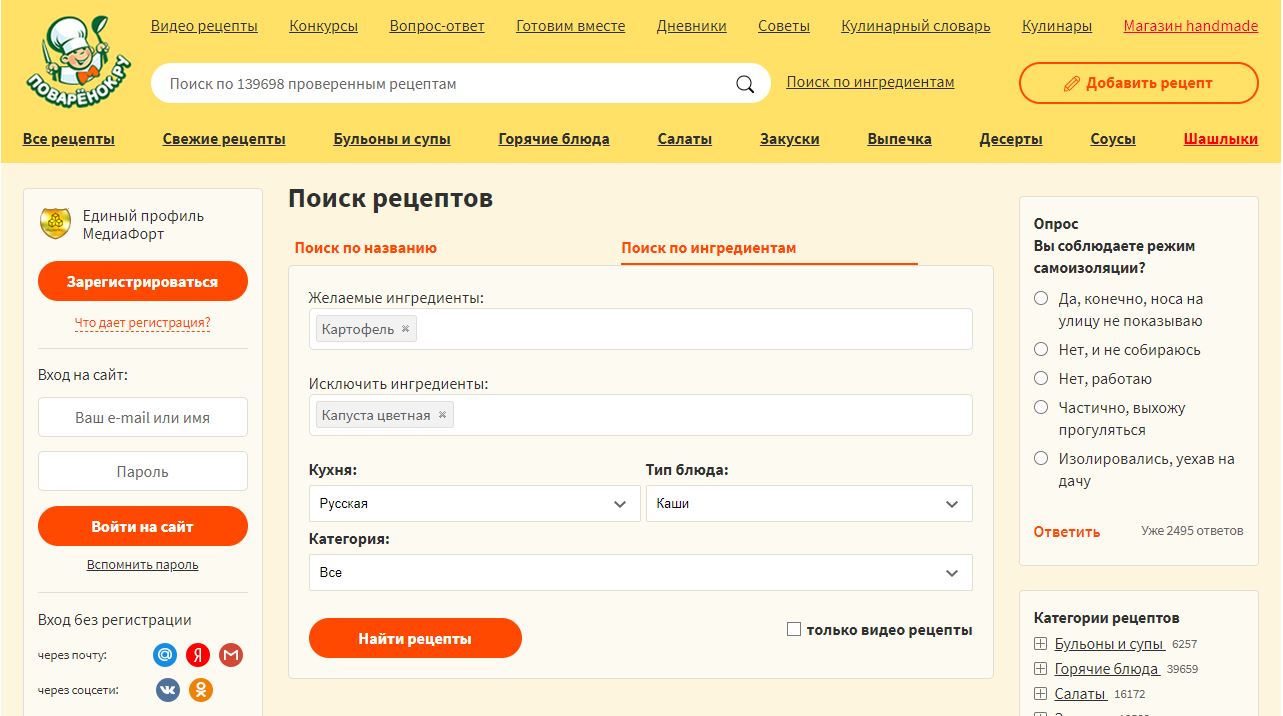
* + 1. Поварёнок.ру (<https://www.povarenok.ru/>)

Данный сайт обладает похожим функционалом, который мы собираемся реализовать в нашем проекте. Помимо просмотра, добавления рецептов на сайте можно участвовать в конкурсах, читать внутренний форум, вступать в тематические сообщества, создавать свою кулинарную команду (аналог закрытого форума) и ещё несколько не таких значимых функций.

Опишем тот функционал, который перекликается с нашим сайтом.

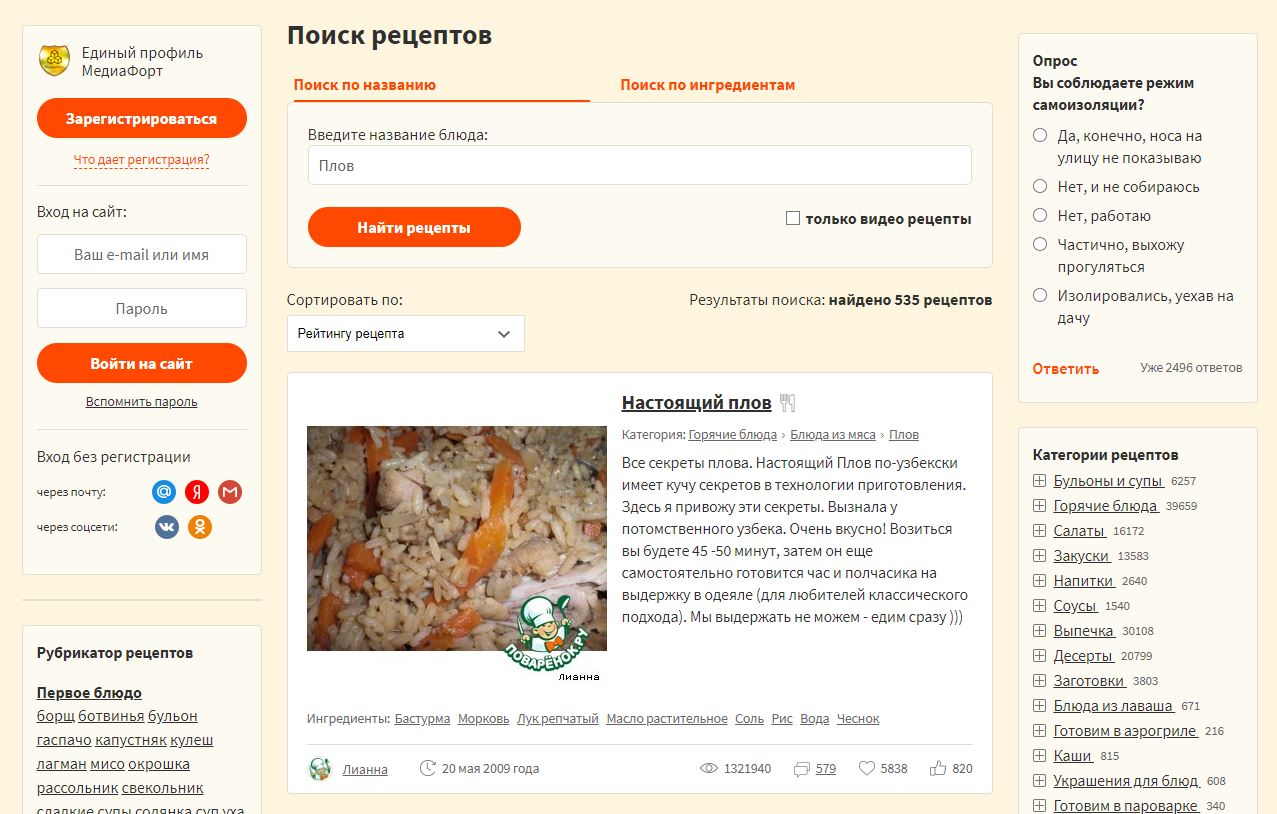
*Поиск рецепта по ингредиенту*

Есть возможность выбрать те ингредиенты, которые должны быть включены в рецепт, а также те, которые должны быть исключены из рецепта. Помимо этого, можно указать национальную принадлежность кухни, тип блюда, категорию и, если хотите, найти рецепт в видео формате.



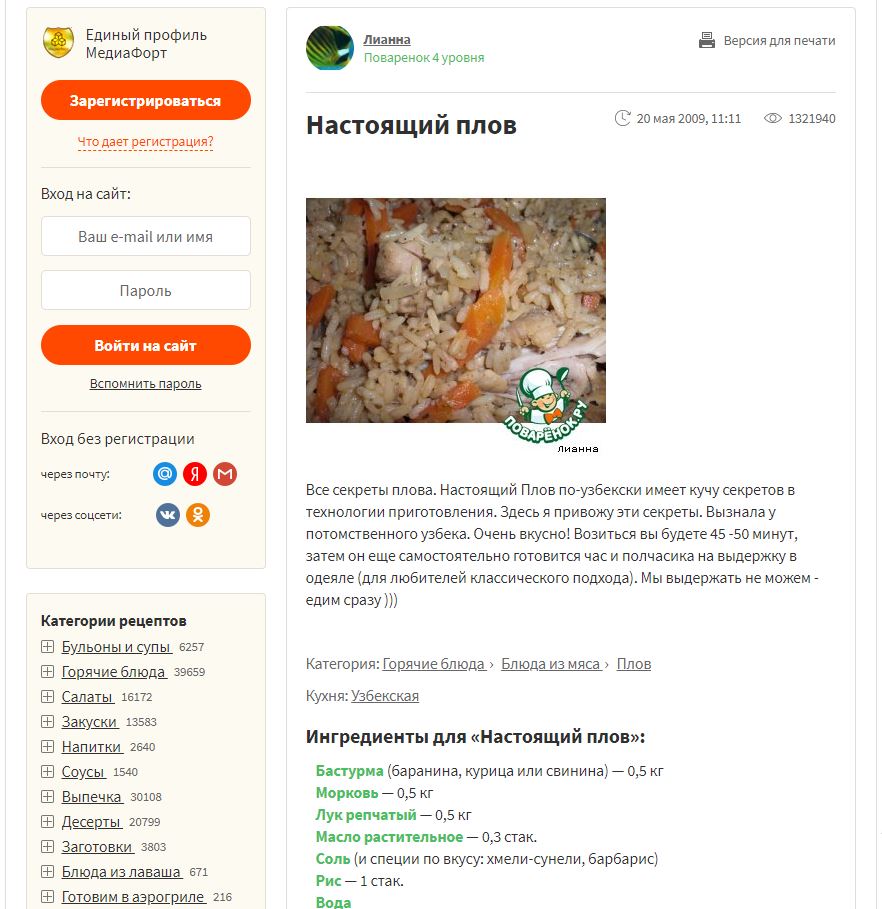
*Поиск рецепта по названию*

Пользователю предлагают ввести название рецепта, можно указать, что рецепты должны быть в видео формате.



*Просмотр выбранного рецепта*

При переходе к просмотру конкретного рецепта сайт резко меняет свой дизайн, возможно, ведётся работа по переводу сайта на новый дизайн. Вверху страницы указан автор рецепта, краткое описание и фотография готового блюда. Дальше помечены категории блюда и национальная принадлежность кухни. Потом список ингредиентов, примерное время приготовления, пищевая и энергетическая ценность. Дальше описана сама технология приготовления блюда, чёткого деления на шаги мы не заметили.



*Заключение*

Сайт наполнен достаточным функционалом, даже может быть избыточным. При первом заходе бросается в глаза большое количество текста и моргающей рекламы на экране.

*Достоинства:*

* Реализованная функция добавления собственного рецепта
* Подробный поиск рецепта по ингредиентам с возможностью не только указания тех, которые должны быть в рецепте, но и тех ингредиентов, которых быть не должно

*Недостатки:*

* Большое количество текста на станице
* Огромное количество рекламы
* Избыточный функционал, который нагружает страницы сайта
* Устаревший дизайн
* Нет возможности получить случайный рецепт
  + 1. Еда.ру (<https://eda.ru/>)

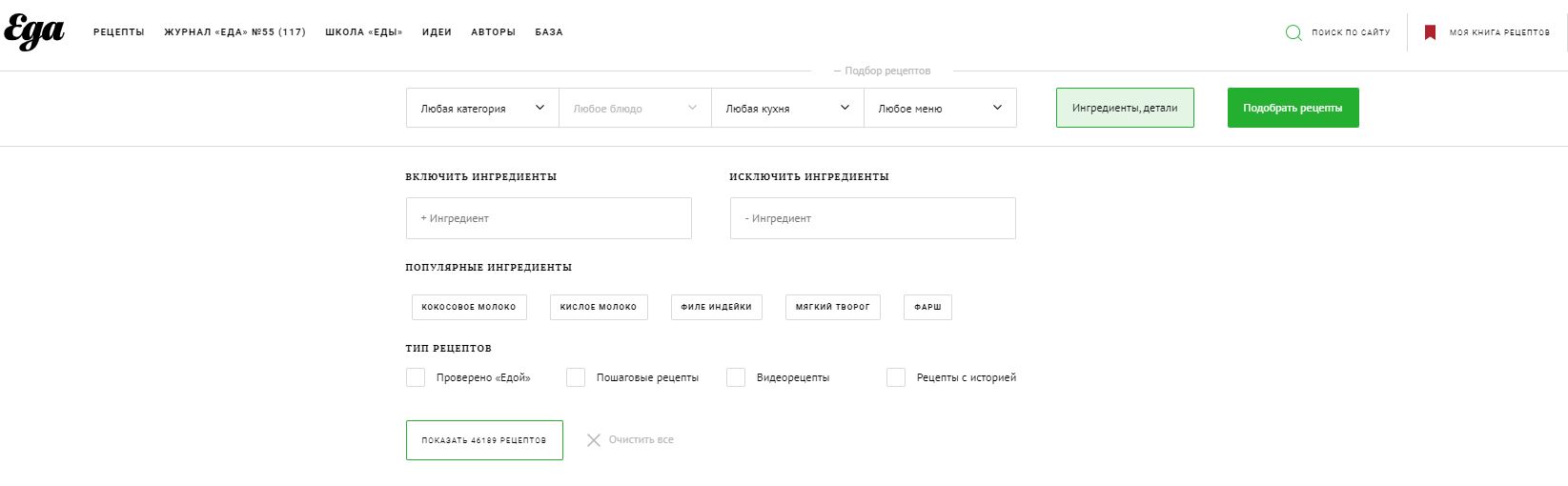
Сайт журнала «Еда».

Первое, что сразу бросается в глаза – утончённый дизайн. Страницы сайта приятно рассматривать, они не перегружены информацией, разбиты на блоки. Сайт позволяет просматривать рецепты, добавлять собственные, смотреть выпуски журнала, проходить обучение в интерактивной школе «Еды».

Рассмотрим интересующий нас функционал.

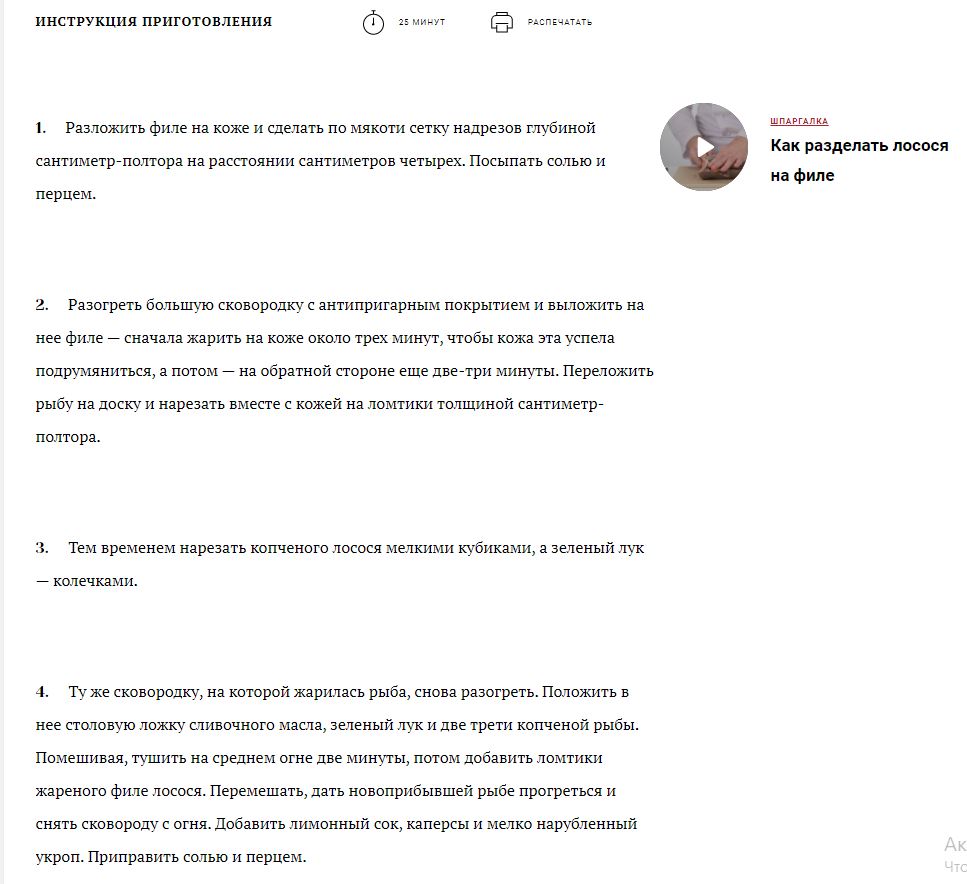
*Поиск рецепта*

Пользователя предлагается выбрать категорию, тип блюда, кухню, включить или исключить ингредиенты, а также выбрать будет ли этот рецепт проверен редакцией, видеорецепт, рецепт с историей. При выдаче результатов поиска приводятся краткое сведение об основных направлениях кухни выбранной национальности



*Просмотр рецепта*

Сначала указано название рецепта, автор, энергетическая ценность и список ингредиентов. Дальше описание шагов приготовление. Каждой шаг выделен в блоке – это позволяет отличить конец и начало очередного шага.



*Заключение*

Сайт выполнен довольно красиво визуально, позволяет производить поиск и просмотра рецепта.

*Достоинства:*

* Рецепты от известных поваров
* Красивый и понятный дизайн
* Большая база рецептов

*Недостатки:*

* Нет возможности поиска по названию рецепта
* Нет возможности получить случайный рецепт
  + 1. Гастрономъ.ру (<https://www.gastronom.ru/>)

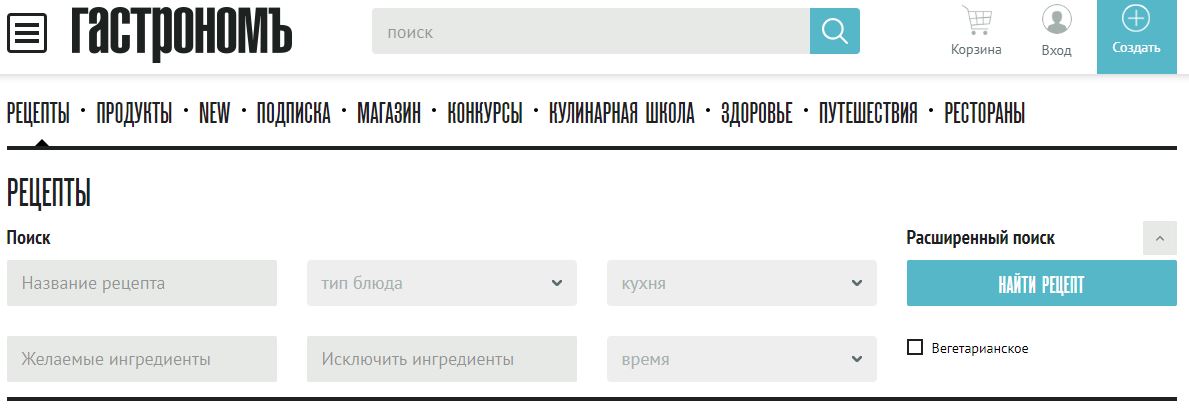
Такое описание приводят создатели сайта: «На нашем сайте собрано около 50 000 рецептов. И это не просто огромный массив кулинарных рецептур, а проверенные на домашних кухнях любимые блюда наших пользователей, а также оригинальные рецепты, которые придумывают или адаптируют наши редакторы. Затем штатные повара готовят эти блюда на редакционных кухнях, их, кстати, у нас 4. То, что нам не нравится, переделывается, поэтому практически все рецепты, которые вы найдете на сайте gastronom.ru, у вас получатся.»

Классное решение, что редакционные повара пробуют готовить представленные рецепты и вносят свои правки. Это первый сайт, где о таком сказано.

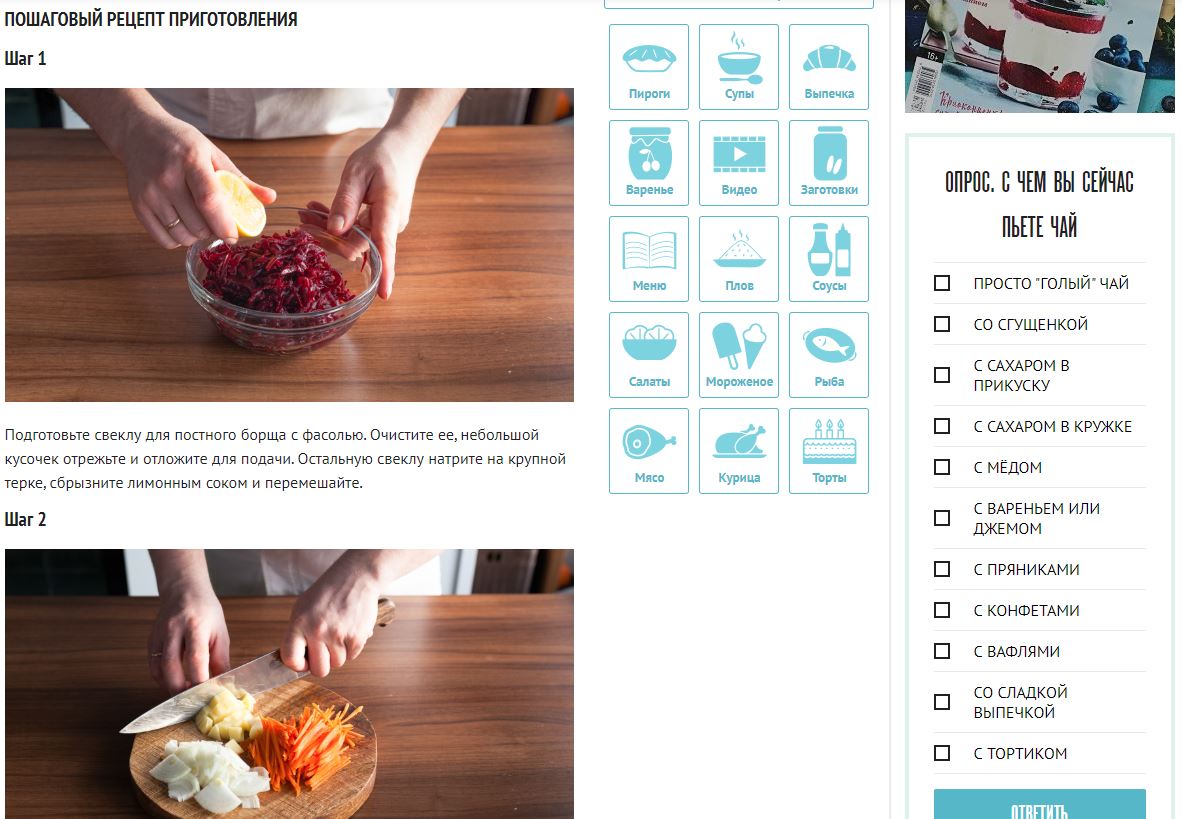
Рассмотрим функции поиска и просмотра рецептов.

*Поиск рецепта*

Пользователю предлагается ввести название блюда, тип и кухню, а также ввести желаемые ингредиенты и те, которые стоит исключить. Можно выбрать примерное время приготовления. Есть отдельный флажок «вегетарианское»



*Просмотр рецепта*

Сначала представлено описание рецепта и необходимые ингредиенты. Сам рецепт разбит на шаги, к каждому шагу прикреплена иллюстрация.

*Заключение*

Сайт выполнен очень хорошо и выполняет все функции которые требует сайт такого рода. Сохранено единообразие и стиль на всех страницах.

*Достоинства:*

* Красивый и понятный дизайн
* Большая база рецептов

*Недостатки:*

* Нет возможности поиска по автору
* Нет возможности получить случайный рецепт

## Итог анализа

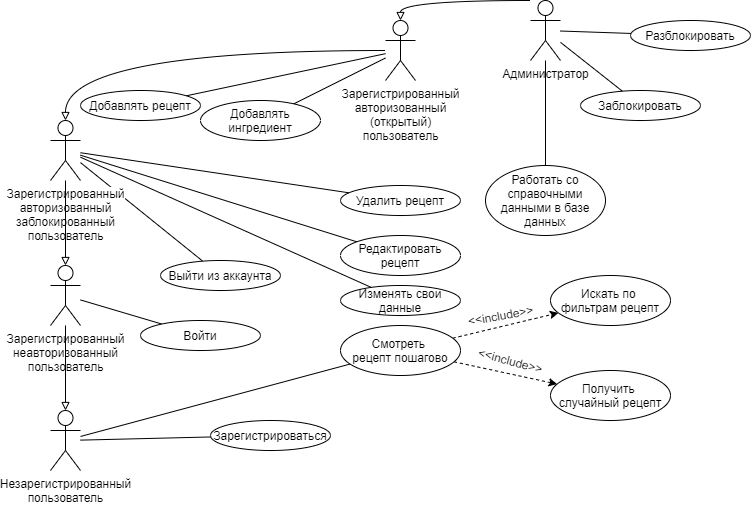
После анализа этих трех сервисов мы можем выявить для себя некоторые аспекты, которых желательно придерживаться при разработке приложения.

1. Простой и интуитивный интерфейс, который не должен быть перегружен ненужным функционалом
2. Красивый и запоминающийся дизайн, при этом он не должен быть слишком контрастным и броским
3. Иметь такие критерии для поиска рецепта, как национальность, автор, главный ингредиент и название рецепта
4. Возможность смотреть рецепт пошагово
5. Разделение пользователей по ролям, для реализации добавления рецептов
6. Возможность изменять зарегистрированным пользователям свои текущие данные профиля
7. Возможность добавлять рецепт и после чего изменять/удалять его
8. Функция администрирования для редактирования существующих рецептов, ингредиентов и национальностей, с возможностью блокировки пользователя за неподобающее поведение (определяется администратором)

Так же у нас появится новый функционал, который отличает нас от сервисов, описанных выше, – возможность получения случайного рецепта.

# Анализ задачи

## Диаграмма прецедентов



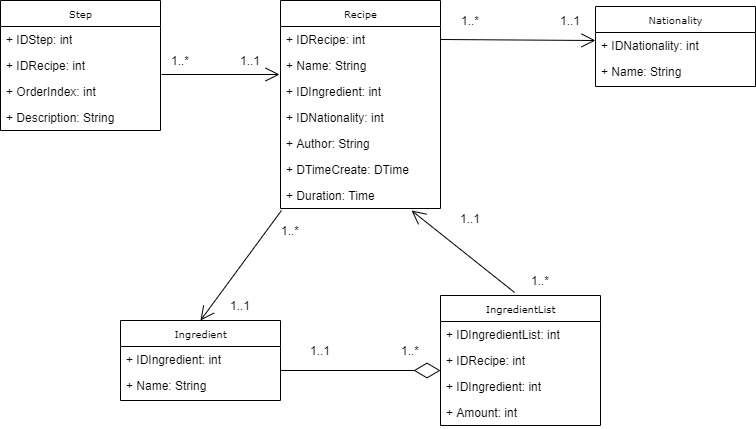
Комментарий:

Пользователи, имеющие возможность удалять и редактировать рецепты, могут это делать только со своими рецептами (рецепты, которые они сами создали).

Администратор же имеет возможность удаления и редактирования всех существующих рецептов.

Под фразой «Работать со справочными данными в базе данных» имеется в виду добавление, изменение и удаление ингредиентов и национальностей.

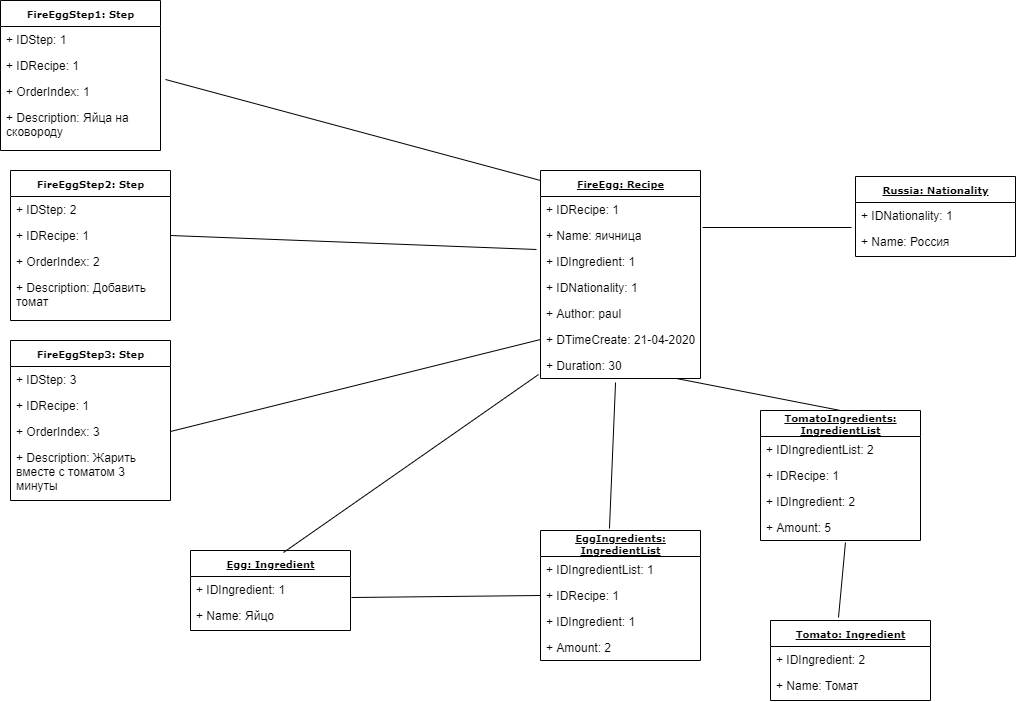
## Диаграмма классов



Комментарий:

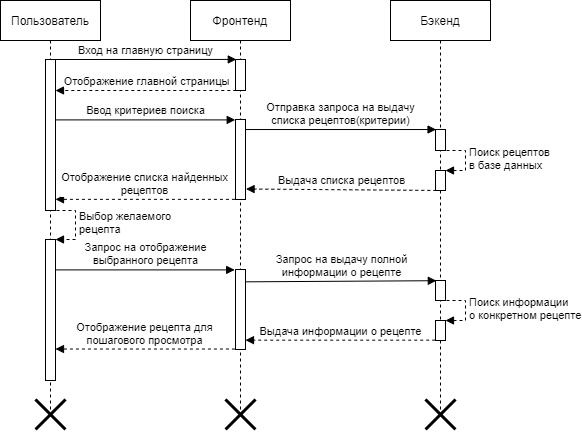
На диаграмме отражены основные классы, которые нужны для понимания работы приложения.

## Диаграмма объектов

Комментарий:

Под диаграммой объектов мы понимаем, состояние объектов в какой-то случайный промежуток времени, для отображения возможных объектов и взаимосвязей между ними.

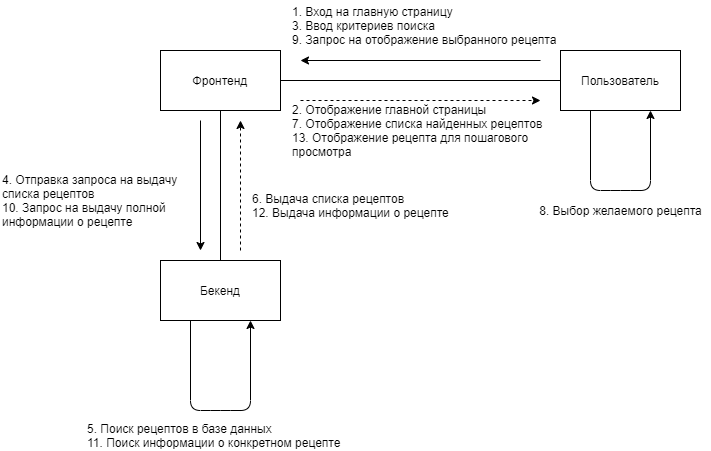
## Диаграмма последовательностей



Комментарий:

Под фронтендом мы понимаем браузер, с помощью которого пользователь взаимодействует с Системой.

## Диаграмма взаимодействия

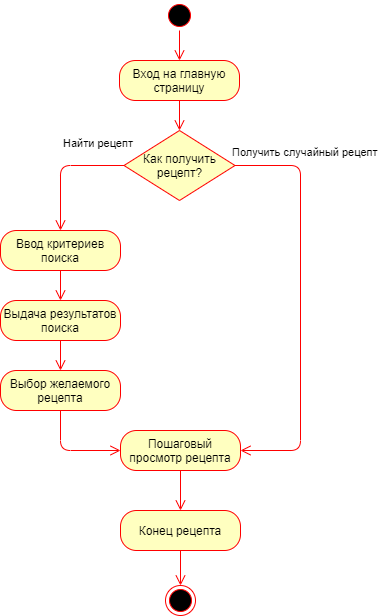


Комментарий:

Под фронтендом мы понимаем браузер, с помощью которого пользователь взаимодействует с Системой.

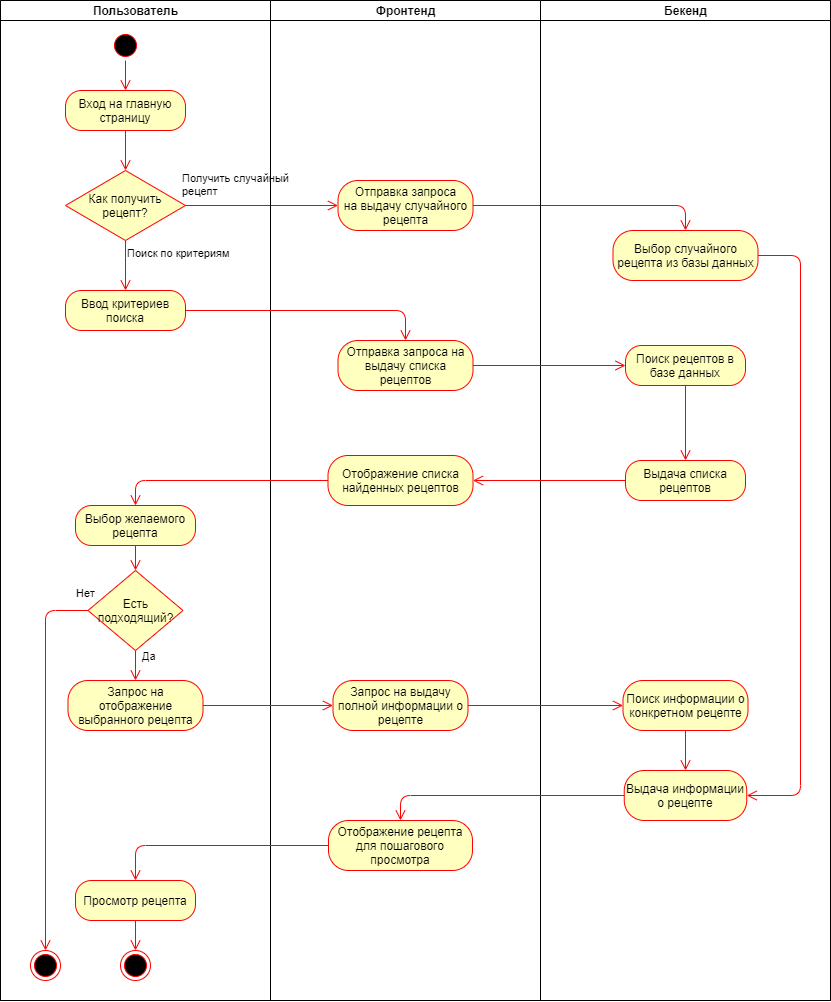
В действии №9 мы понимаем нажатие на определённый рецепт

## Диаграмма состояний



Комментарий:

## Диаграмма активностей

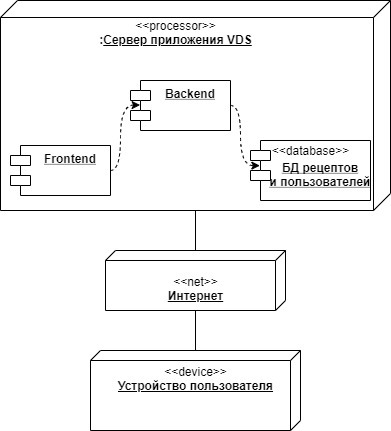


Комментарий:

Под фронтендом мы понимаем браузер, с помощью которого пользователь взаимодействует с Системой.

Под выражением «запрос на отображение выбранного рецепта» мы понимаем нажатие на определённый рецепт

## Диаграмма развертывания



Комментарий:

База данных развернута с помощью программы Microsoft SQL Server.

Передача данных между сервером и устройством пользователя осуществляется через Глобальную Сеть Интернет посредством протокола HTTPS.

# Обоснование архитектуры проекта

При проектировании приложения был выбран паттерн MVC.

Основные элементы паттерна:

Модель (Model) – предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.

Представление (View) – отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Контроллер (Controller) - интерпретирует действия пользователя, оповещая модель о необходимости изменений.

В качестве ORM в проекте выступает технология Entity Framework Core, являющаяся ORM решением для .NET Core от Microsoft. Использование подхода Database First в EF Core позволило автоматически по готовой базе данных сгенерировать модель, а также легко обновлять модель в случае изменений структуры базы данных.

Причины выбора данного паттерна:

* Чётко разделена логика приложения
* Возможность автоматической генерации контроллера в ASP.NET Core Web API
* Автоматическая генерация модели в EF Core

# Обоснование выбора продуктовых воронок

Нашей командой было принято составить три продуктовых воронки, основанных на целевых действиях пользователя.

Основываясь на проведённом анализе предметной области и существующих решений, были выделены три сценария действия пользователя, по которым мы сможем оценить на сколько достигнута цель создания Системы:

1. Просмотр случайно выданного рецепта до конца

Благодаря отслеживанию этой воронки мы сможем понять на сколько пользователю подходят те случайные рецепты, которые выдаёт наша Система, тем самым увидеть потребность в добавлении какого-то специального алгоритма выдачи случайного рецепта, основанного на пользовательских предпочтениях.

1. Добавление рецепта после регистрации на сайте

Увидев динамику прохождения пользователем этого сценария, мы сможем сделать вывод: способна ли наша Система увеличивать базу рецептов исключительно за счёт добавленных рецептов пользователей или требуется пополнять базу за счёт сил создателей Системы

1. Просмотр рецепта после поиска

Количество просмотров рецепта после поиска позволит оценить полноту базы рецептов, а также функционал самого поиска: сделать ли его более детальным или наоборот убрать некоторые поля.

# Выбор методов и средств разработки

После составления Технического Задания и проектирования нашей Системы началась стадия разработки.

Основным стэком технологий для бэкенда были заявлены .NET Framework 4.6.1 и Entity Framework 6.2.0.

В ходе разработки выяснилось, что в качестве основной платформы для разработки веб-приложений Microsoft на данный момент предлагает использовать кросс-платформенный фреймворк с открытым исходным кодом ASP.NET Core 3.1. Более того, сейчас политика Microsoft направлена на продвижение использования ASP.NET Core, как современной альтернативы .NET Framework. Особенно это касается новых разработок. Также на сайте https://www.thoughtworks.com/radar/platforms платформа .NET Core указана как наиболее перспективная.

С целью разработки Системы на современных технологиях бэкенд было принято решение писать на ASP.NET Core 3.1, а также переписать уже написанную часть приложения на использование современной платформы.

# Реализация

# Безопасность

При реализации приложения были поставлены требования к обеспечению безопасности:

1. Приложение должно предоставлять доступ к некоторым методам неавторизованным пользователям. Например, к методу добавления рецепта или блокировки пользователей.
2. Приложение должно иметь несколько уровней доступа в зависимости от роли, которая закреплена за пользователем. Например, только администратор может получить информацию о пользователях, а заблокированный пользователь не должен иметь доступ к добавлению рецепта.

Механизм авторизации **веб сервисов** преимущественно полагается на  
**JWT**-токены. JWT представляет собой веб-стандарт, который определяет способ передачи информации о пользователе в формате **JSON** в зашифрованном виде. Пример токена: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJodHRwOi8vc2NoZW1hcy54bWxzb2FwLm9yZy93cy8yMDA1LzA1L2lpZW50eXR5L2NsYWltcy9uYW1lIjoic3RyaW5nIiwiaHR0cDovL3NjaGVtYXMubWljcm9zb2Z0LmNvbS88cy8yMDA4LzA2L2lkZW50aXR5L2NsYWltcy9yb2xlIjoib3BlbiIsIm5iZiI6MTU5MTcyOTcyXywiZXhwIjoxNjA5NzI5NzIzLCJpc3MiOiJZdW1tWXVtbVktQmFja2VuZCIsImF1ZCI6Ill1bW1ZdW1tWS1Vc2VyIn0.UM4X4\_xBcvGBQGRNVYosEITD96ZCjMnqNWYUEBre2Jo, который используется у нас в приложении. Токен имеет время жизни. По истечении отведённого времени, токен не проходит проверку, и пользователю будет предложено снова авторизоваться на сайте.

# Тестирование

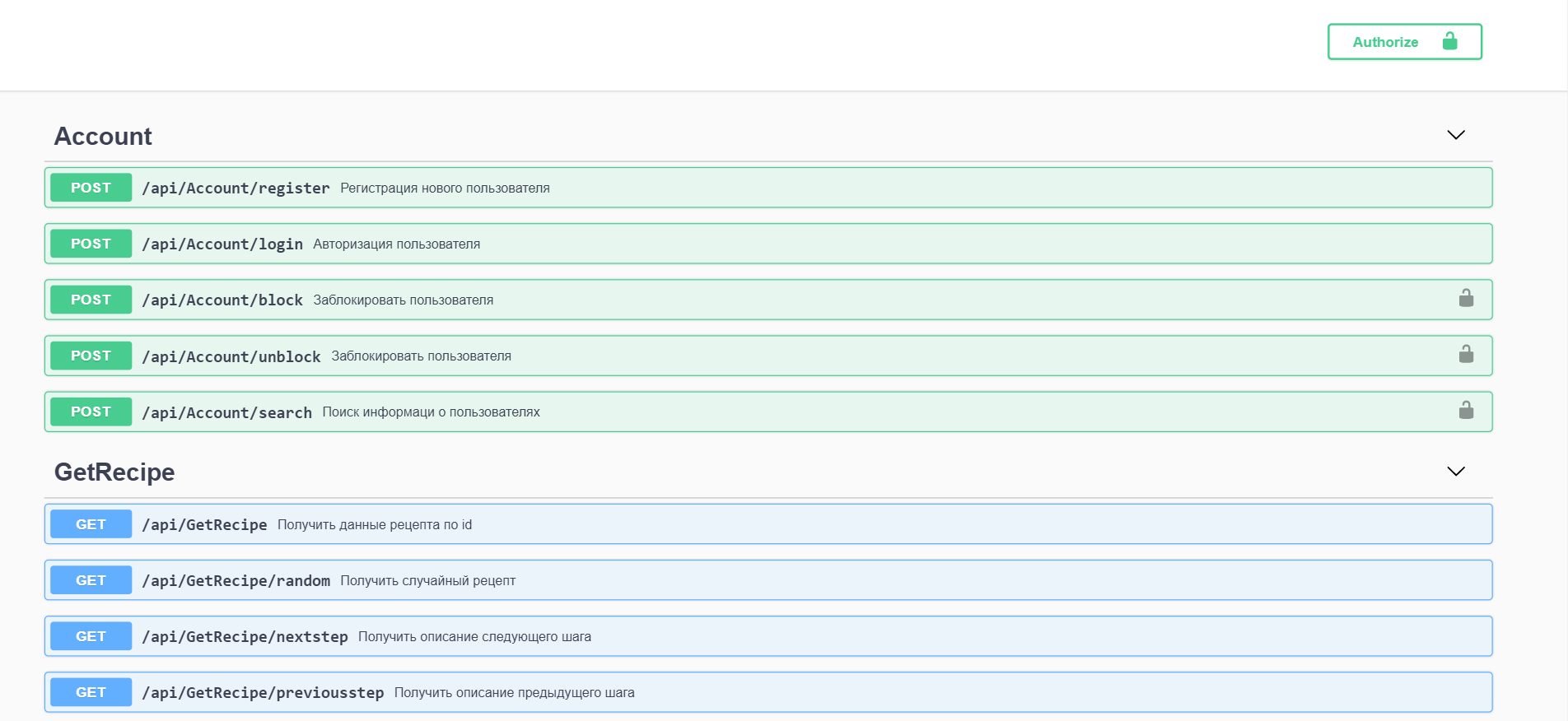
# Документирование

Документирование **бэкенда** приложения ведётся с помощью ПО с открытым исходным кодом Swagger. Этот инструмент позволяет разработчикам вест и предоставлять документацию **RESTful** веб-сервисов с возможностью тестирования запросов к методам **API**.

Структуру API Swagger понимает благодаря специальным аннотациям перед методом, например, [SwaggerResponse(HttpStatusCode.BadRequest)] или с помощью разбора xml комментариев, вот комментарий-аналог аннотации выше: /// <response code="400">Некорректные значения</response>

Также ПО Swagger позволяет вызывать методы, требующие авторизации пользователя. Для этого предусмотрена специальная настройка. Достаточно в специальное поле внести значение действительного токена.

На рисунке \*номер\* представлен скриншот Swagger-документации нашего приложения.



Для каждого контроллера написаны методы, которые есть возможность вызвать. Замочком помечены методы, требующие авторизации перед вызовом.

# API

# Заключение

# Планы

# Список использованных источников