Спецификация требований   
проекта «Пробуй!»

[Введение 3](#_Toc75461517)

[Назначение 3](#_Toc75461518)

[Определения 4](#_Toc75461519)

[Бизнес-требования и метрики 4](#_Toc75461520)

[Границы системы 5](#_Toc75461521)

[Краткое описание контекста использования 5](#_Toc75461522)

[Типы пользователей 6](#_Toc75461523)

[Основные пользовательские требования 7](#_Toc75461524)

[Сценарии использования 9](#_Toc75461525)

[Общий бизнес-процесс использования сервиса 9](#_Toc75461526)

[Подпроцесс модерации 10](#_Toc75461527)

[Диаграмма сценариев 10](#_Toc75461528)

[Регистрация пользователя 10](#_Toc75461529)

[Авторизация пользователя 11](#_Toc75461530)

[Авторизация администратора в административном разделе сайта 13](#_Toc75461531)

[Восстановление забытого пароля пользователя 14](#_Toc75461532)

[Работа с личным кабинетом 16](#_Toc75461533)

[Поиск рецепта блюда 16](#_Toc75461534)

[Просмотр рецепта и переход на внешний сайт выбранной точки питания 17](#_Toc75461535)

[Комментирование рецепта 19](#_Toc75461536)

[Голосование по рецепту 20](#_Toc75461537)

[Занесение рецепта в избранное 22](#_Toc75461538)

[Редактирование сборника рецептов администратором 23](#_Toc75461539)

[Модерация комментариев администратором 26](#_Toc75461540)

[Функциональные требования 29](#_Toc75461541)

[Требования, налагаемые регулятором 30](#_Toc75461542)

[Производительность 30](#_Toc75461543)

[Удобство использования 30](#_Toc75461544)

[Логическая модель данных 32](#_Toc75461545)

[Статус пользователя 33](#_Toc75461546)

[Схема баз данных, реализующая логическую модель данных 34](#_Toc75461547)

[Пользовательский интерфейс 35](#_Toc75461548)

[Карта страниц пользовательского интерфейса приложения 35](#_Toc75461549)

[Стартовая страница 36](#_Toc75461550)

[Кулинарная география: страны 37](#_Toc75461551)

[Кулинария страны: общая характеристика 38](#_Toc75461552)

[Рецепт: карта точек питания, неавторизованный пользователь 39](#_Toc75461553)

[Рецепт: список точек питания, неавторизованный пользователь 40](#_Toc75461554)

[Поиск рецепта блюда 41](#_Toc75461555)

[Рецепт: карта точек питания, авторизованный пользователь 42](#_Toc75461556)

[Рецепт: создание комментария авторизованным пользователем 43](#_Toc75461557)

[Пользователь: регистрация/вход 44](#_Toc75461558)

[Пользователь: восстановления пароля 45](#_Toc75461559)

[Пользователь: регистрация 46](#_Toc75461560)

[Пользователь: личный кабинет 47](#_Toc75461561)

[Администратор: логон 48](#_Toc75461562)

[Администратор: редактирование рецепта 49](#_Toc75461563)

[Администратор: редактирование учетных записей 50](#_Toc75461564)

[Администратор: модерация 51](#_Toc75461565)

[Программные интерфейсы 51](#_Toc75461566)

[Ресурс аккаунта пользователя 51](#_Toc75461567)

[Создание нового аккаунта 53](#_Toc75461568)

[Изменение существующего аккаунта 54](#_Toc75461569)

[Получение данных аккаунта 56](#_Toc75461570)

[Получение списка аккаунтов с указанными параметрами 58](#_Toc75461571)

[Удаление аккаунта 60](#_Toc75461572)

[Ресурс комментария 61](#_Toc75461573)

[Создание нового комментария 61](#_Toc75461574)

[Изменение данных комментария 63](#_Toc75461575)

[Получение данных комментария 64](#_Toc75461576)

[Получение списка комментариев к рецепту с указанными параметрами 65](#_Toc75461577)

[Удаление комментария 67](#_Toc75461578)

[Ресурс кухни страны 67](#_Toc75461579)

[Создание новой кухни 68](#_Toc75461580)

[Изменение данных кухни страны 69](#_Toc75461581)

[Получение данных кухни страны 70](#_Toc75461582)

[Получение списка кухонь стран с указанными параметрами 71](#_Toc75461583)

[Удаление кухни страны 73](#_Toc75461584)

[Ресурс точки питания 73](#_Toc75461585)

[Создание новой точки питания 74](#_Toc75461586)

[Изменение данных точки питания 75](#_Toc75461587)

[Получение данных точки питания 76](#_Toc75461588)

[Получение списка точек питания с указанными параметрами 77](#_Toc75461589)

[Удаление точки питания 80](#_Toc75461590)

[Ресурс рецепта блюда 80](#_Toc75461591)

[Создание нового рецепта блюда 81](#_Toc75461592)

[Изменение данных рецепта блюда 86](#_Toc75461593)

[Получение данных рецепта блюда 89](#_Toc75461594)

[Получение списка рецептов с указанными параметрами 91](#_Toc75461595)

[Удаление рецепта блюда 94](#_Toc75461596)

[В качестве идеи для будущего развития: 95](#_Toc75461597)

# Введение

Документ содержит спецификацию требований для проекта «Пробуй!». «Пробуй!» — это онлайн-сервис, представляющий собой сборник кулинарных рецептов народов мира и дающий информацию о том, где в Москве блюда, представленные на сайте, можно попробовать.

Документ в целом соответствует рекомендациям стандарта IEEE-29148 Requirements Engineering с точки зрения как структуры, так и содержания.

В документе используются диаграммы, поясняющие его текст, созданные в соответствии с нотациями UML и BPMN.

## Назначение

Сервис «Пробуй!» предназначен для ознакомления пользователей с кухнями народов мира, обмена отзывами о тех или иных рецептах для зарегистрированных пользователей, а также для продвижения заведений общественного питания, что наряду с контекстной рекламой будет служить способом монетизации.

## Определения

Аккаунт — данные о пользователе, хранящиеся в Системе.

Администратор – выделенный аккаунт для редактирования БД рецептов и модерации поведения пользователей на сайте.

Административный раздел сайта – страницы сайта, с помощью которых осуществляется управление сборником рецептов и аккаунтами пользователей.

Бэкенд — серверная часть Системы, состоящая из набора компонентов и баз данных.

Голосование - оценка рецепта («нравится/не нравится»), которую оставляет зарегистрированный пользователь.

Комментарий — текстовый отзыв о рецепте зарегистрированного пользователя или ответ на отзывы других пользователей.

Модерация – процедура удаления комментария, нарушающего правила сайта, сопровождающаяся предупреждением или блокировкой автора такого комментария.

Рецепт — информация о каком-либо рецепте, хранящемся в Системе. Рецепты добавляются администратором сайта из кулинарных книг.

Сайт, Веб-сайт, Фронтенд — веб-приложение, предназначенное для работы в браузере, предоставляющее пользовательский интерфейс Системы.

Система — сайт и бэкенд, реализующие работу сервиса.

Точка питания – предприятие общественного питания любого типа: ресторан, кафе, закусочная, бар и т.д.

## Бизнес-требования и метрики

Основная задача проекта — создание сервиса, привлекательного для пользователей, и получение прибыли от переходов на сайты рекламируемых точек питания.

Основные метрики, по которым будет отслеживаться успешность сервиса:

* пользовательские:
  + месячная аудитория пользователей,
  + процент возврата пользователей (через день/неделю/месяц),
  + количество добавленных отзывов,
  + количество занесенных в избранное рецептов по отношению к общему числу рецептов
* бизнес-метрики:
  + количество переходов на продвигаемые сайты,
  + месячная выручка от рекламы.

## Границы системы

Сервис «Пробуй!» реализуется системой, состоящей из бэкенда и веб-сайта, предоставляющего пользовательский интерфейс.

Основные сценарии, реализуемые системой, показаны на диаграмме ниже. Информация о рецептах хранится в базе данных. Плата за переходы осуществляется на основании информации о траффике с сайтов продвигаемых точек питания. Для отрисовки карты с нанесенными на нее точками питания, отображения информации о точках питания, включая рейтинги, используется API Яндекс.Карт.

*Размышление вслух*: в этом месте нужно решать с финансистами, каким образом будут строиться отношения с точками питания:

1. мы партнеримся с каждым по типовому договору об отчислениях за переход на их сайт и отображаем информацию только о занесенных в нашу БД точках-партнерах (аналог Яндекс.Коллекции), использование API Яндекс.Карт вполне укладывается в бесплатный план. Правда, хранение информации о рейтингах – это платное удовольствие, а если их не хранить, то сортировка по ним может занимать длительное время. Договора с каждым – это довольно муторно и долго, но зато стабильно можно будет наращивать объемы.
2. мы берем все, что возвращают Яндекс.Карты, самостоятельно формируем базу данных о точках питания, изучая их меню (а ля inout.ru, молодые годы). Вопрос с рейтингами вполне можно забрать под свою ответственность. Т.е. API Яндекс.Карт будет использоваться исключительно по бесплатному тарифу. Раскручиваемся за счет рекламы, нарастив аудиторию, приходим к рестораторам и предлагаем партнерство. Тут можно довольно быстро договориться и всех обобрать, но этап первоначального накопления данных и раскрутки может затянуться.

## Краткое описание контекста использования

Система взаимодействует с пользователями:

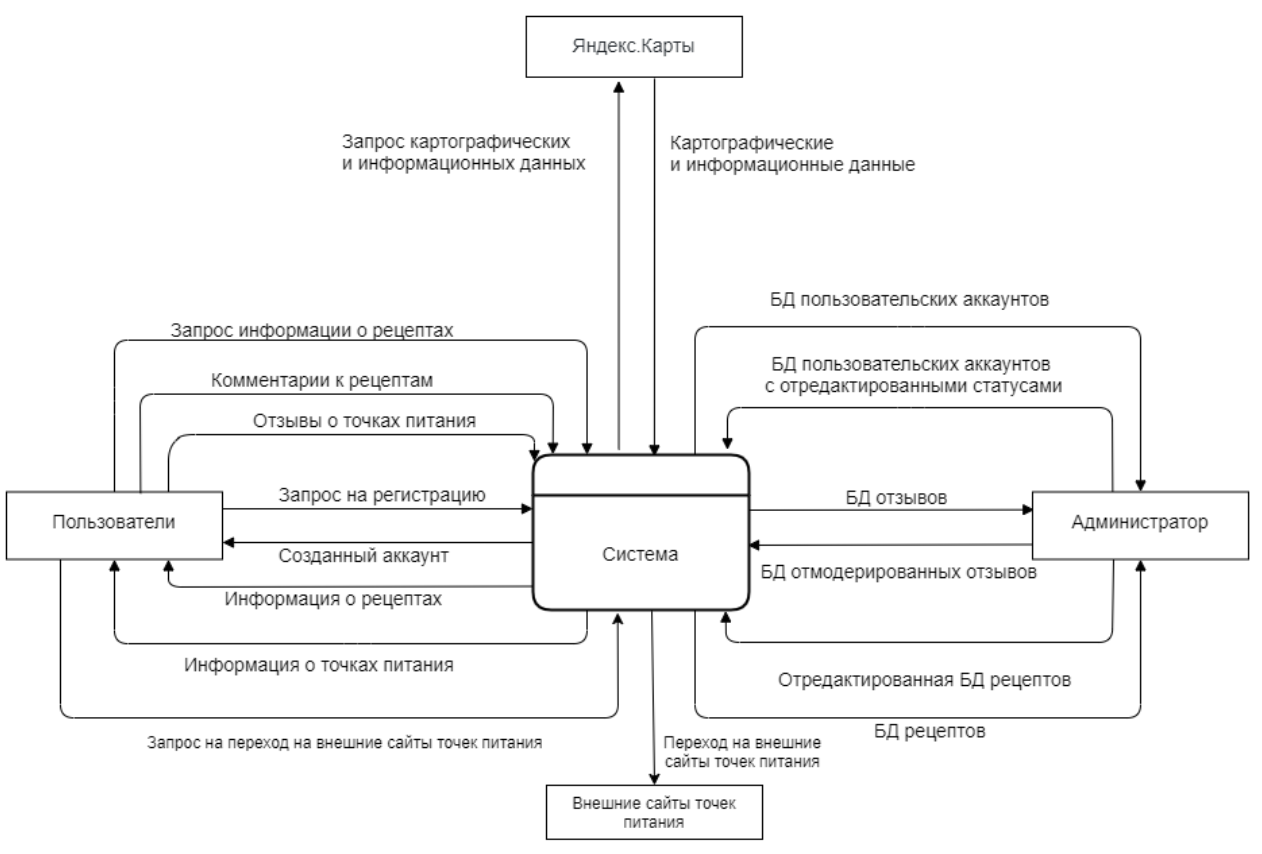
* пользователи регистрируются в системе;
* пользователи запрашивают информацию о рецептах;
* зарегистрированные пользователи добавляют комментарии о рецептах;
* пользователи получают информацию о точках питания, в которых можно попробовать блюда, рецепты которых представлены на сайте.
* зарегистрированные пользователи добавляют комментарии о точках питания.

Система взаимодействует с администратором:

* администратор редактирует БД рецептов;
* администратор может менять статус пользователя в случае его некорректного поведения на сайте вплоть до блокирования аккаунта;
* администратор модерирует отзывы пользователей в соответствии с требованиями Роскомнадзора и правилами сайта.

Система взаимодействует с внешними системами:

* система получает картографические и информационные данные о точках питания через API Яндекс.Карт;
* система совершает переход на внешние сайты точек питания по соответствующему действию пользователя.

**Контекстная диаграмма**

## Типы пользователей

*Здесь можно также добавить RACI-матрицу и описание заинтересованных лиц.*

**Т.к. подобных сервисов до сих пор нету, значит в них, кроме меня, особо никто не заинтересован, а поэтому я позволю себе этот пункт опустить :).**

* Незарегистрированный пользователь — пользователь, который может просматривать информацию о рецептах и точках питания, а также читать комментарии.
* Зарегистрированный пользователь, авторизованный пользователь — пользователь, имеющий аккаунт в системе; может просматривать информацию о рецептах и точках питания, писать комментарии и голосовать по рецептам, добавлять понравившиеся рецепты в избранное.
* Администратор – системный пользователь, обладающий правом редактирования БД рецептов, а также правом модерации комментариев зарегистрированных пользователей.

## Основные пользовательские требования

Пользователь может искать рецепт по названию блюда.

Пользователь может искать рецепты, содержащие указанные им ингредиенты.

Пользователь может искать рецепты, исключающие указанные им ингредиенты.

Пользователь может смотреть информацию о выбранном рецепте.

Пользователь может читать отзывы, чтобы сформировать мнение о рецепте перед тем, как готовить.

Пользователь может смотреть информацию о точках питания, в которых подают выбранное им блюдо.

Пользователь может переходить на сайты заинтересовавших его точек питания (если таковые имеются).

Пользователь может зарегистрироваться на сайте, чтобы расширить свои возможности до возможностей авторизованного пользователя (см. далее).

Пользователь может авторизоваться, чтобы получить доступ к своему аккаунту.

Авторизованный пользователь может отредактировать свой аккаунт, чтобы поддерживать актуальность своих данных.

Я как авторизованный пользователь хочу иметь возможность комментировать заинтересовавшие меня рецепты или высказывать свое мнение о точках питания.

Я как авторизованный пользователь хочу иметь возможность сохранить заинтересовавшие меня рецепты для быстрого доступа.

Я как авторизованный пользователь хочу иметь возможность удалить выбранные рецепты из сохраненных.

Я как авторизованный пользователь хочу иметь возможность писать заметки по сохраненным мною рецептам.

Я как авторизованный пользователь хочу иметь возможность редактировать заметки по сохраненным мною рецептам.

Я как администратор хочу иметь возможность редактировать сборник рецептов.

Я как администратор хочу получать уведомления о созданных авторизованными пользователями комментариях.

Я как администратор хочу иметь возможность модерировать (см. «Модерация") дискуссии авторизованных пользователей.

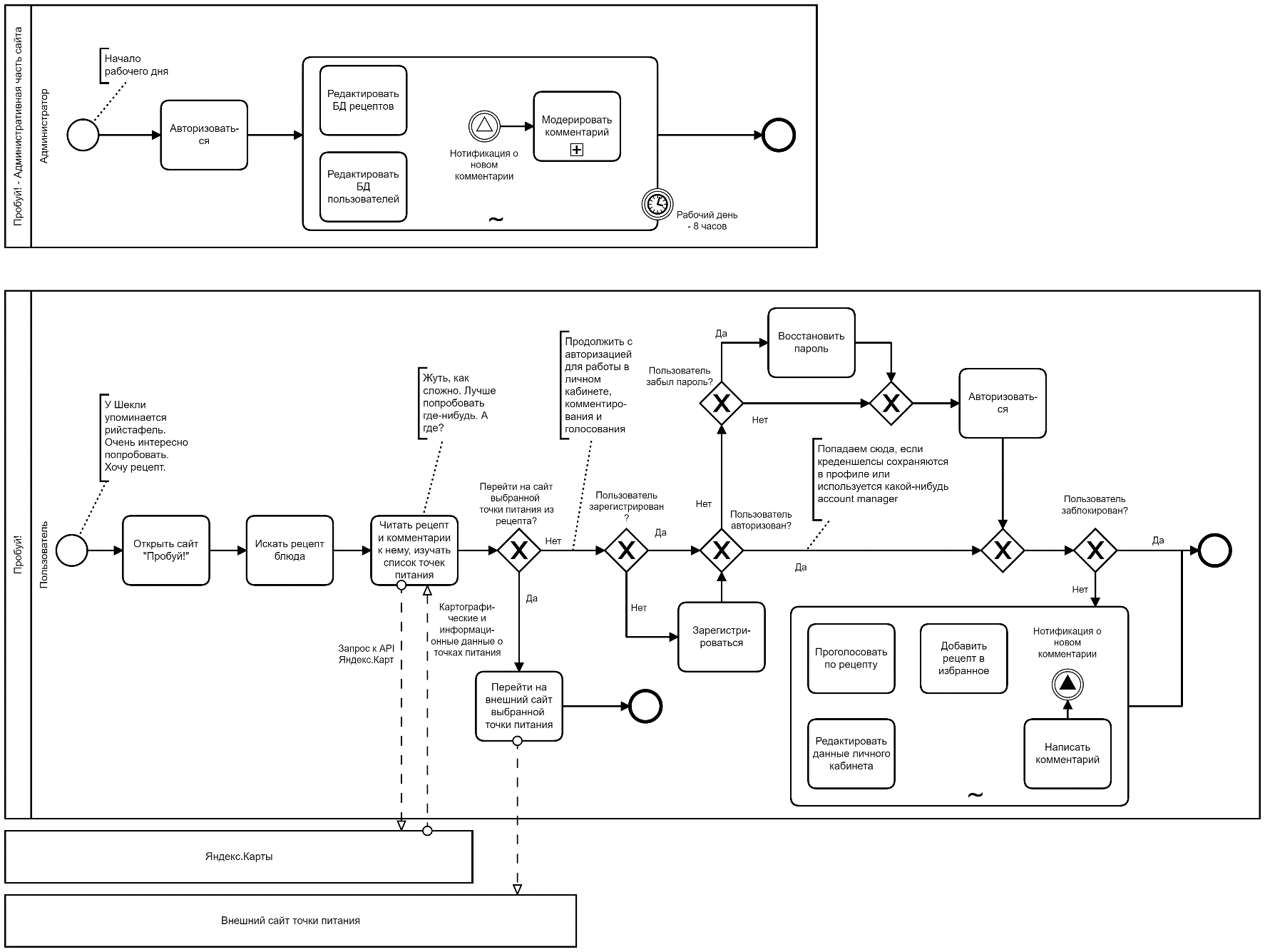
Я как администратор хочу иметь возможность предупреждать по e-mail’у впервые нарушивших правила поведения на сайте.

Я как администратор хочу иметь возможность временно блокировать аккаунты повторно нарушивших правила поведения на сайте.

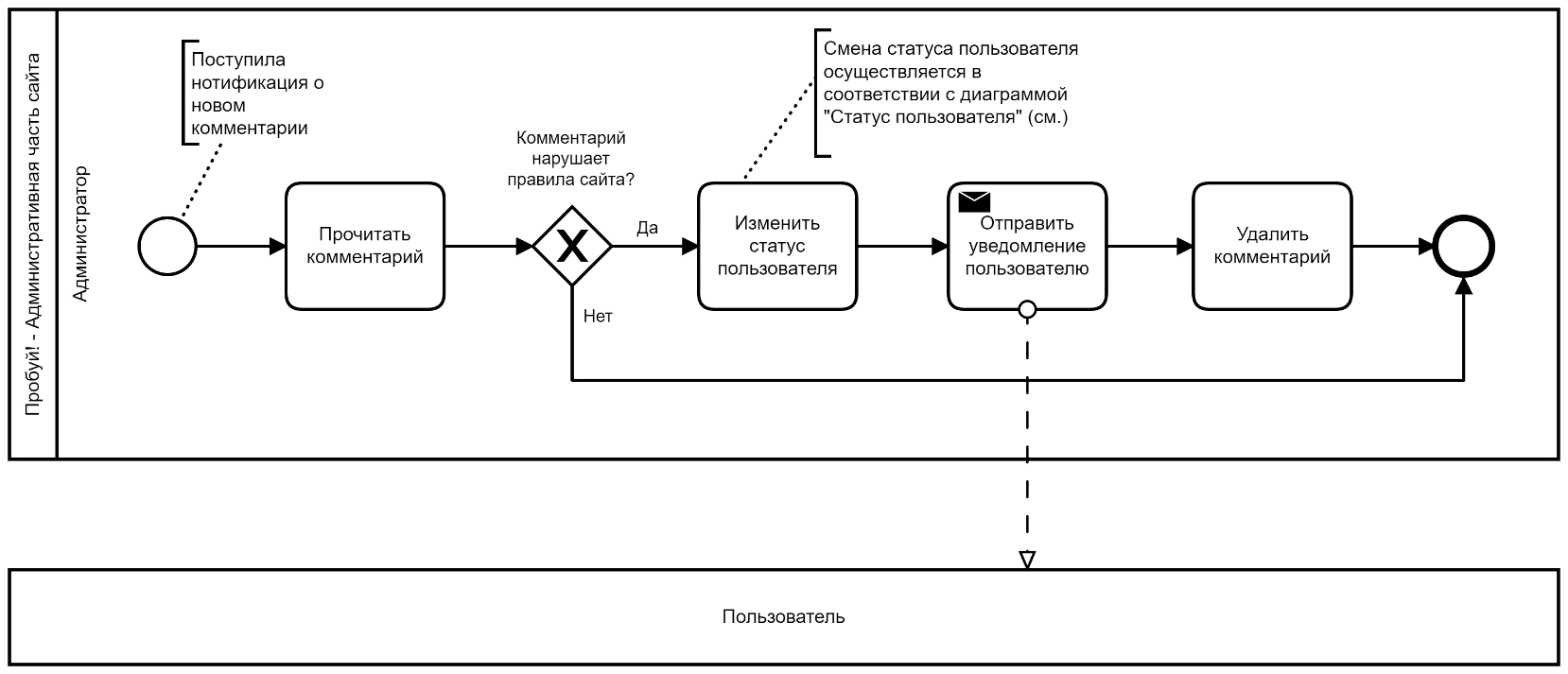
Я как администратор хочу иметь возможность навсегда блокировать аккаунты нарушивших более трех раз правила поведения на сайте.

## Сценарии использования

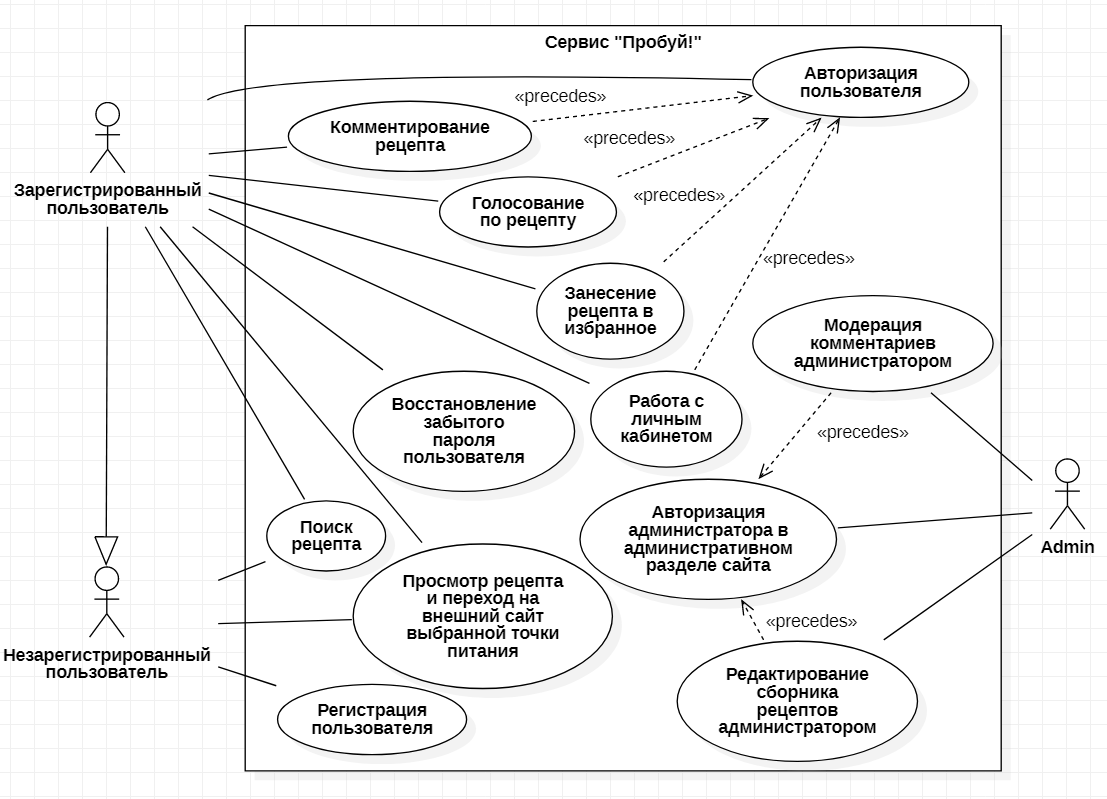
### Общий бизнес-процесс использования сервиса

****

### Подпроцесс модерации



### Диаграмма сценариев



### Регистрация пользователя

**Цель:** создать аккаунт в системе.

**Акторы:** пользователь, система (сайт, бэкенд)

**Контекст**: пользователь находится на сайте.

**Предусловие**: пользователь, не зарегистрированный в системе, зашел на сайт.

Основной сценарий:

1. Пользователь выбирает пункт меню “Регистрация/Вход”.
2. Система открывает страницу Пользователь: «Регистрация/Вход».
3. Пользователь выбирает пункт «Зарегистрироваться».
4. Система открывает страницу Пользователь: регистрация.
5. Пользователь заполняет информацию о себе и подтверждает создание аккаунта.
6. Система создаёт новый аккаунт, используя полученные данные.
7. Система открывает страницу Пользователь: «Регистрация/Вход» для ввода креденшелсов созданного аккаунта.



### Авторизация пользователя

**Цель**: авторизация пользователя.

**Акторы**: пользователь, система (сайт, бэкенд).

**Контекст**: пользователь хочет авторизоваться на сайте для того, чтобы комментировать рецепт, проголосовать по рецепту, занести его в избранные или редактировать данные личного кабинета.

**Предусловие**: неавторизованный пользователь зашел на сайт.

**Основной сценарий**:

1. Пользователь выбирает пункт меню “Регистрация/Вход”.
2. Система открывает страницу Пользователь: «Регистрация/Вход» для ввода креденшелсов уже существующего в системе аккаунта.
3. Пользователь вводит свои e-mail и пароль.
4. Система проверяет введенные данные и возвращает успешный результат проверки.
5. Система открывает страницу Пользователь: личный кабинет.

**Альтернативы и исключения:**

*Аккаунт не существует*

1. Результат проверки системой введенных данных: такой аккаунт не существует.
2. Система выдает соответствующее сообщение и предлагает перейти на страницу Пользователь: регистрация.

*Пароль неверен*

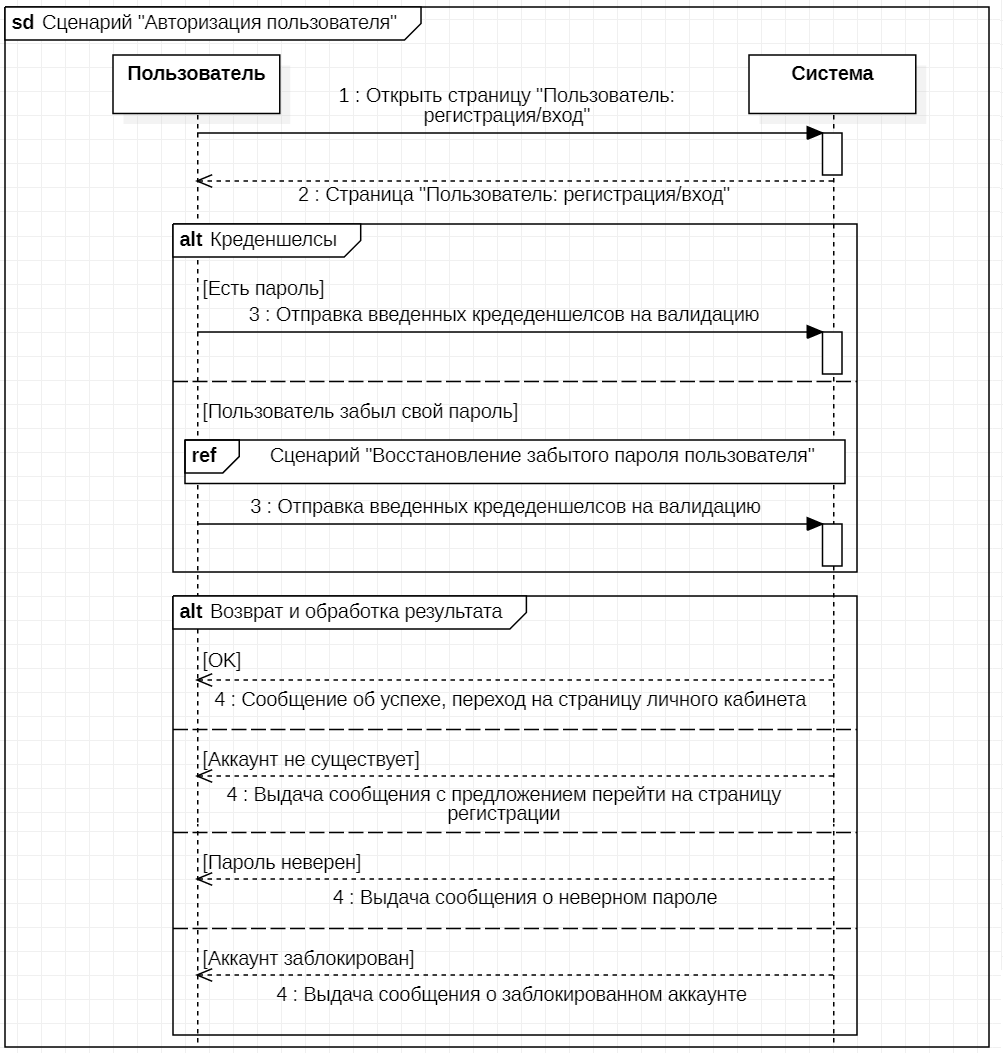
1. Результат проверки системой введенных данных: введенный пароль неверен.
2. Система выдает соответствующее сообщение и остается на странице ввода пароля, пока введенный пароль не будет принят системой как валидный.

*Пользователь забыл свой пароль*

1. Если пользователь забыл свой пароль, система предоставляет возможность восстановить его по указанному e-mail’у – см.Пользователь: восстановления пароля.
2. Получив пароль, пользователь возвращается к основному сценарию.

*Аккаунт* *заблокирован*

1. Результат проверки системой введенных данных: такой аккаунт уже существует и он заблокирован.
2. Система выдает соответствующее сообщение и авторизации не происходит.



### Авторизация администратора в административном разделе сайта

**Цель**: авторизация администратора.

**Акторы**: администратор, система (административный раздел сайта, бэкенд).

**Контекст**: администратор хочет авторизоваться на сайте для того, чтобы редактировать рецепты или аккаунты пользователей.

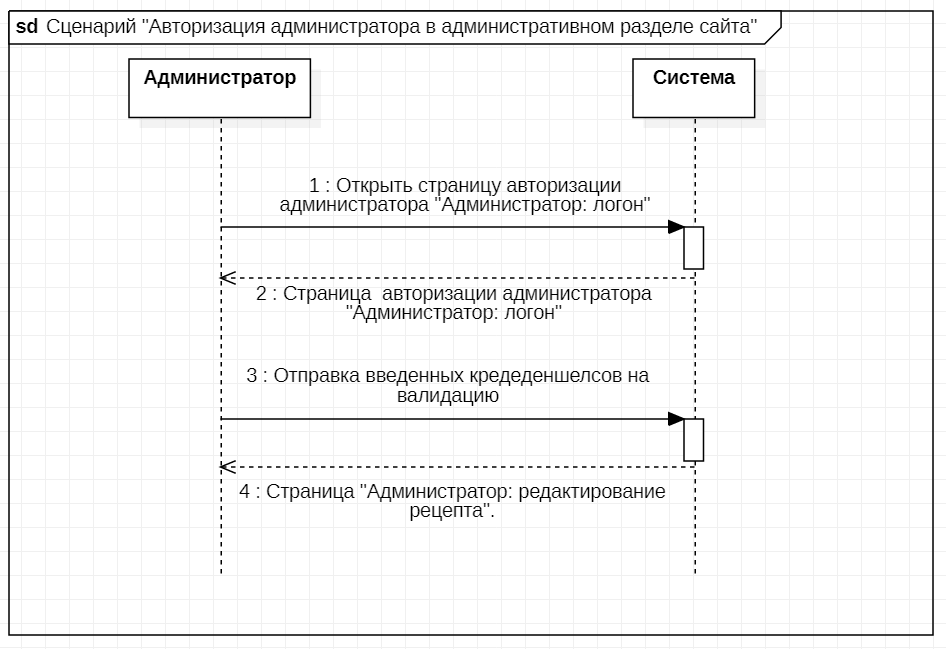
**Предусловие**: администратор открыл административный раздел сайта.

**Основной сценарий**:

1. Администратор открыл страницу авторизации в административном разделе сайта – см. Администратор: логон.
2. Администратор вводит свои e-mail и пароль.

***Примечание****: в качестве административного используется доменный аккаунт (AD), поэтому ни регистрации, ни восстановления пароля здесь не предполагается.*

1. Система проверяет введенные данные и возвращает успешный результат проверки.
2. Система открывает страницу редактирования рецептов Администратор: редактирование рецепта.



### Восстановление забытого пароля пользователя

**Цель**: восстановление пароля пользователя.

**Акторы**: пользователь, система (сайт, бэкенд).

**Контекст**: пользователь хочет авторизоваться на сайте для того, чтобы комментировать рецепт, проголосовать по рецепту, занести его в избранные или редактировать данные личного кабинета, но не может вспомнить свой пароль.

**Предусловие**: неавторизованный пользователь зашел на сайт.

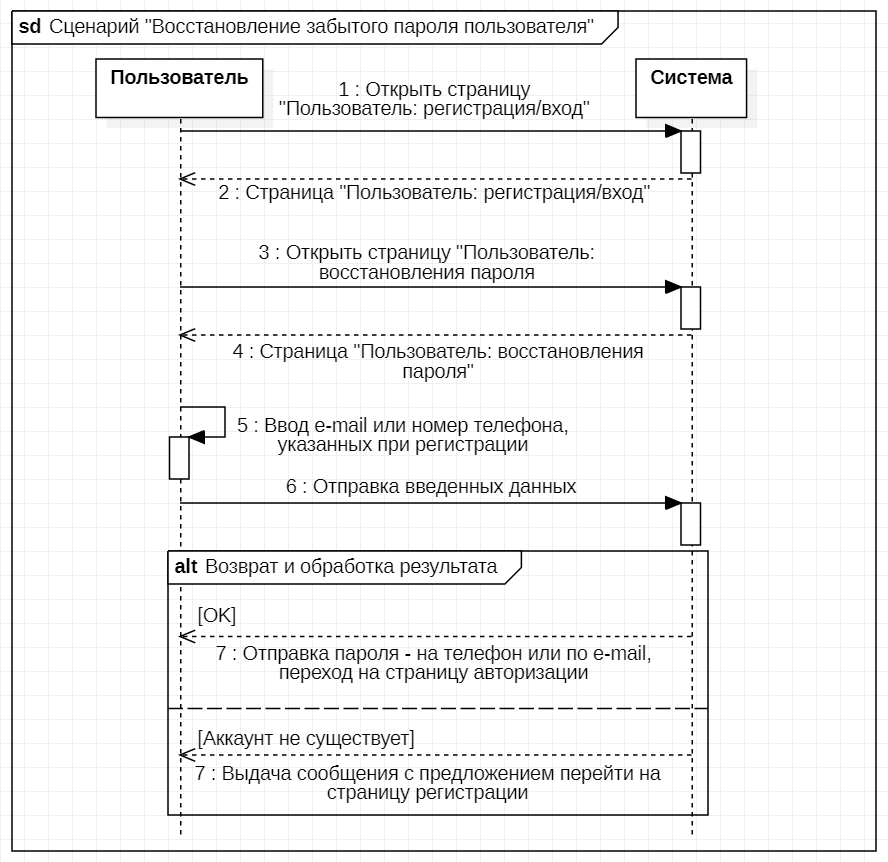
**Основной сценарий**:

1. Пользователь выбирает пункт меню “Регистрация/Вход”.
2. Система открывает страницу Пользователь: «Регистрация/Вход».
3. Пользователь выбирает пункт «Не помню пароль».
4. Система открывает страницу Пользователь: восстановления пароля.
5. Пользователь вводит свой e-mail или номер телефона, указанные при регистрации.
6. Система проверяет введенные данные и отправляет пароль выбранным пользователем способом – на телефон или по e-mail.
7. Система открывает страницу Пользователь: «Регистрация/Вход» для авторизации с полученным паролем.

**Альтернативы и исключения:**

*Аккаунт не существует*

1. Результат проверки системой введенных данных: такой аккаунт не существует.
2. Система выдает соответствующее сообщение и предлагает перейти на страницу Пользователь: регистрация.



### Работа с личным кабинетом

**Цель**: редактировать данные своего личного кабинета.

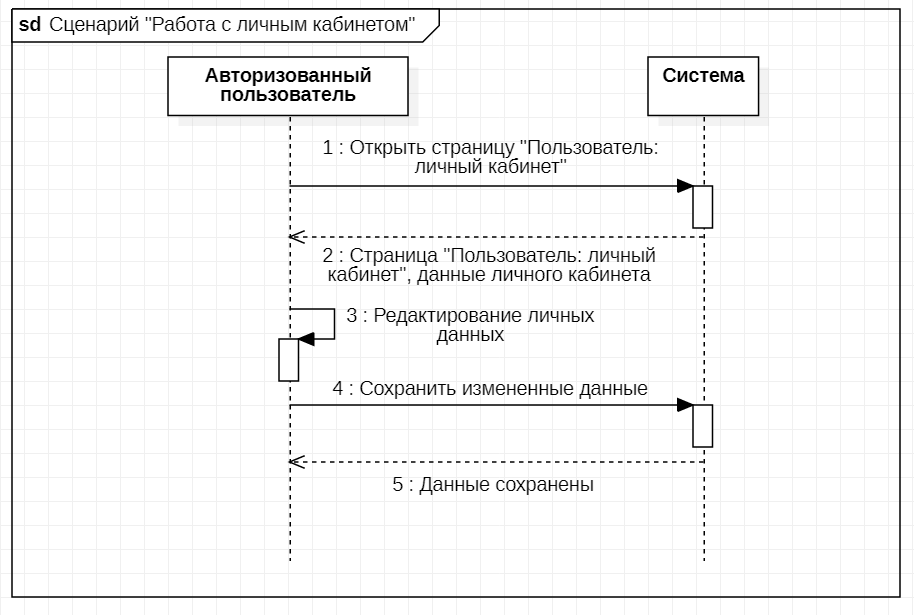
**Акторы**: авторизованный пользователь, система (сайт, бэкенд).

**Контекст**: авторизованный пользователь зашел на сайт и открыл свой кабинет.

**Предусловие**: пользователь зашел на сайт и авторизовался.

**Основной сценарий**:

1. Пользователь открывает страницу личного кабинета (*пункт меню, появляющийся для авторизованных пользователей*).
2. Система отображает данные личного кабинета: ФИО, номер телефона, e-mail, DOB, раздел с избранными рецептами, а также раздел для функционала по смене текущего пароля – см.Пользователь: личный кабинет.
3. Пользователь редактирует свои данные (ФИО, номер телефона, избранные рецепты и заметки к ним), меняет пароль, а также пишет заметки к рецептам.
4. Система получает обновленные данные личного кабинета сохраняет их.



### Поиск рецепта блюда

**Цель:** найти нужный рецепт блюда.

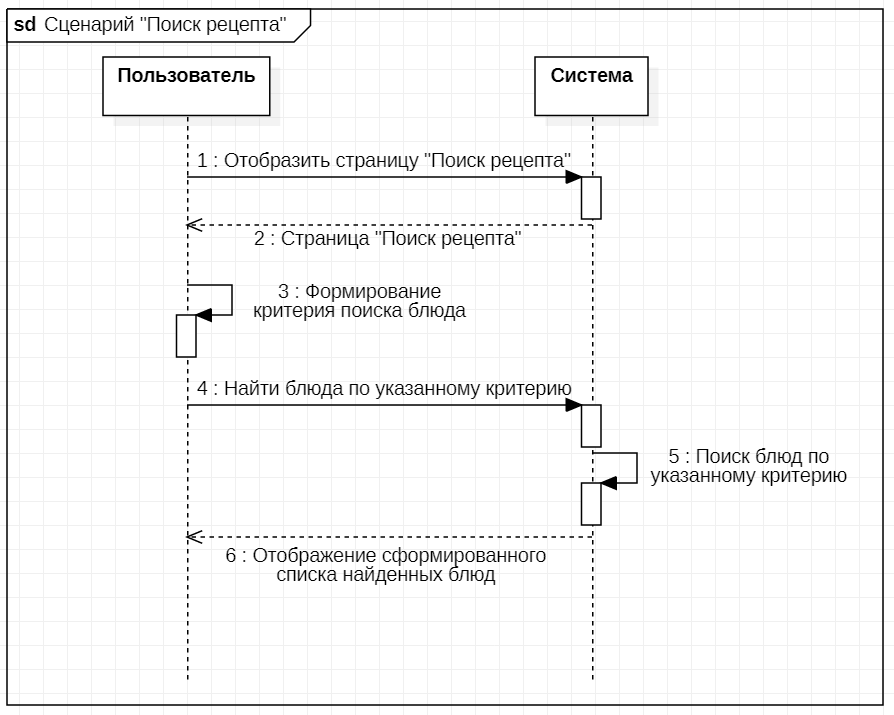
**Акторы:** пользователь, система (сайт, бэкенд)

**Контекст**: пользователь или авторизованный пользователь находится на сайте.

**Предусловие**: пользователь или авторизованный пользователь зашел на сайт.

**Основной сценарий:**

1. Пользователь выбрал пункт меню «Поиск рецепта блюда».
2. Система отображает страницу Поиск рецепта блюда.
3. Пользователь указал какие-либо параметры поиска: название блюда, ингредиенты, входящие, либо не входящие в состав блюда.
4. Система формирует список рецептов, удовлетворяющих заданным условиям.
5. Система отображает список найденных рецептов.



### Просмотр рецепта и переход на внешний сайт выбранной точки питания

**Цель:** посмотреть рецепт и комментарии к нему, выбрать точку питания и перейти на ее сайт.

**Акторы:** пользователь, система (сайт, бэкенд)

**Контекст**: пользователь или авторизованный пользователь находится на сайте.

**Предусловие**: пользователь или авторизованный пользователь зашел на сайт.

**Основной сценарий:**

1. Пользователь открыл страницу Кулинарная география: страны и выбрал интересующую его страну.
2. Сайт отображает страницу с описанием особенностей кухни выбранной страны – см. Кулинария страны: общая характеристика.
3. Пользователь выбирает из списка интересующий его раздел кухни.
4. Система отображает список рецептов этого раздела.
5. Пользователь выбирает интересующий его рецепт.
6. Система открывает страницу рецепта – см. Рецепт: карта точек питания, неавторизованный пользователь или Рецепт: карта точек питания, авторизованный пользователь.
7. Система обращается к API Яндекс.Карт для отрисовки карты с нанесенными точками питания, в которых готовят блюдо по выбранному рецепту, а также для получения рейтинга (*опосредовано из Яндекс.Справочника, входящего в Яндекс.Карты*) этих точек питания на Яндексе.
8. С помощью API Яндекс.Карт система отрисовывает карту с нанесенными на нее точками питания и сохраняет в своей БД их рейтинги (*для возможности сортировки по рейтингам*).
9. Изучив комментарии и данные точек питания, пользователь выбирает точку питания и переходит на ее сайт.

**Альтернативы и исключения:**

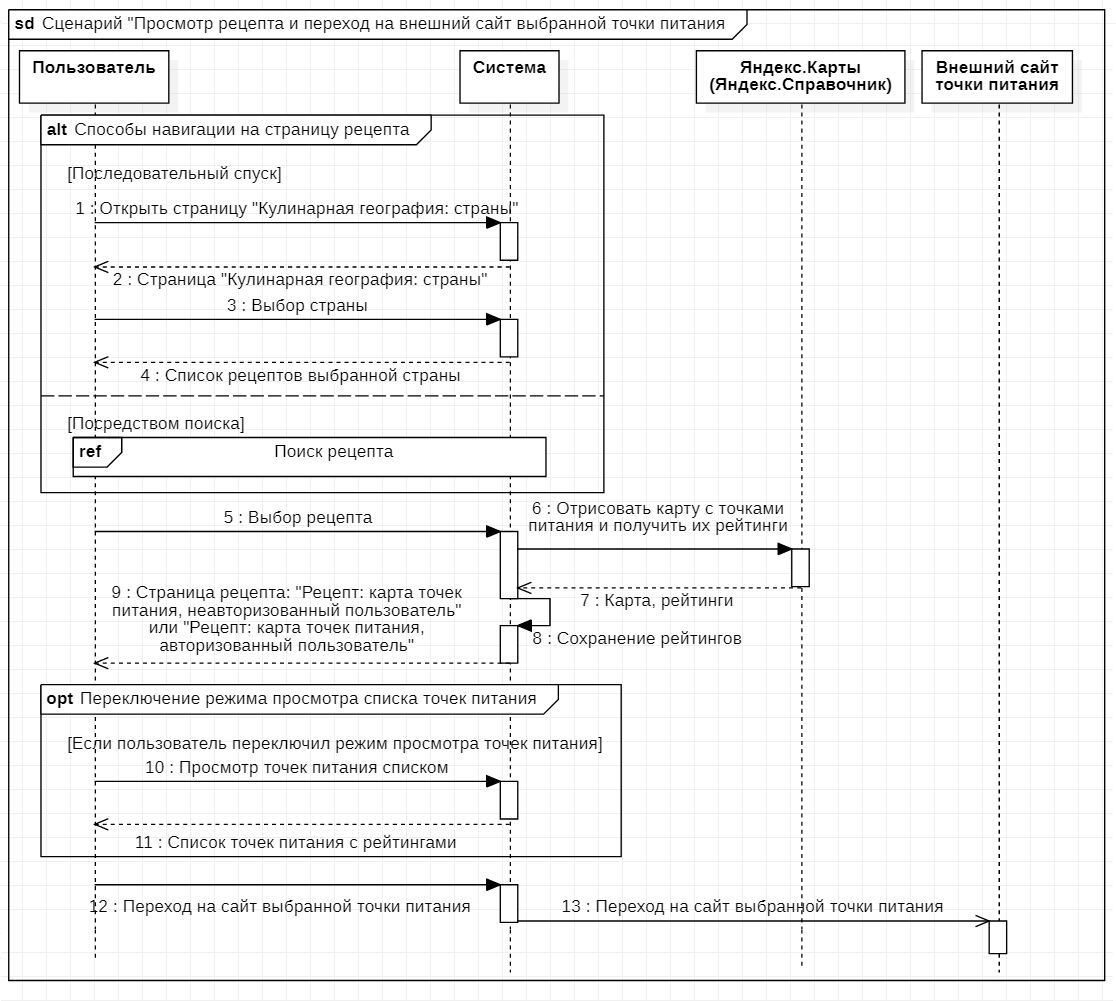
*Просмотр рецепта, найденного через пункт меню «Поиск* *рецепта блюда»:*

1. Выполнение сценария Поиск рецепта блюда.
2. Пользователь выбирает какой-либо рецепт из найденных.
3. Система открывает страницу рецепта выбранного рецепта и комментарии к нему – см. Рецепт: карта точек питания, неавторизованный пользователь или Рецепт: карта точек питания, авторизованный пользователь.

Далее – шаги 7-9 по основному сценарию.

*Переключение режима просмотра списка точек питания*

1. Пользователь выбирает способ отображения соответствующих рецепту точек питания с «Картой» на «Списком».
2. Система отображает данные точек питания текстом в виде списка – см. Рецепт: список точек питания, неавторизованный пользователь.
3. Изучив комментарии и данные точек питания, пользователь выбирает точку питания и переходит на ее сайт.



### Комментирование рецепта

**Цель**: написать комментарий к рецепту.

**Акторы**: авторизованный пользователь, система (сайт, бэкенд), администратор.

**Контекст**: авторизованный пользователь находится на странице заинтересовавшего его рецепта.

**Предусловие**: пользователь зашел на сайт, авторизовался и открыл страницу заинтересовавшего его рецепта.

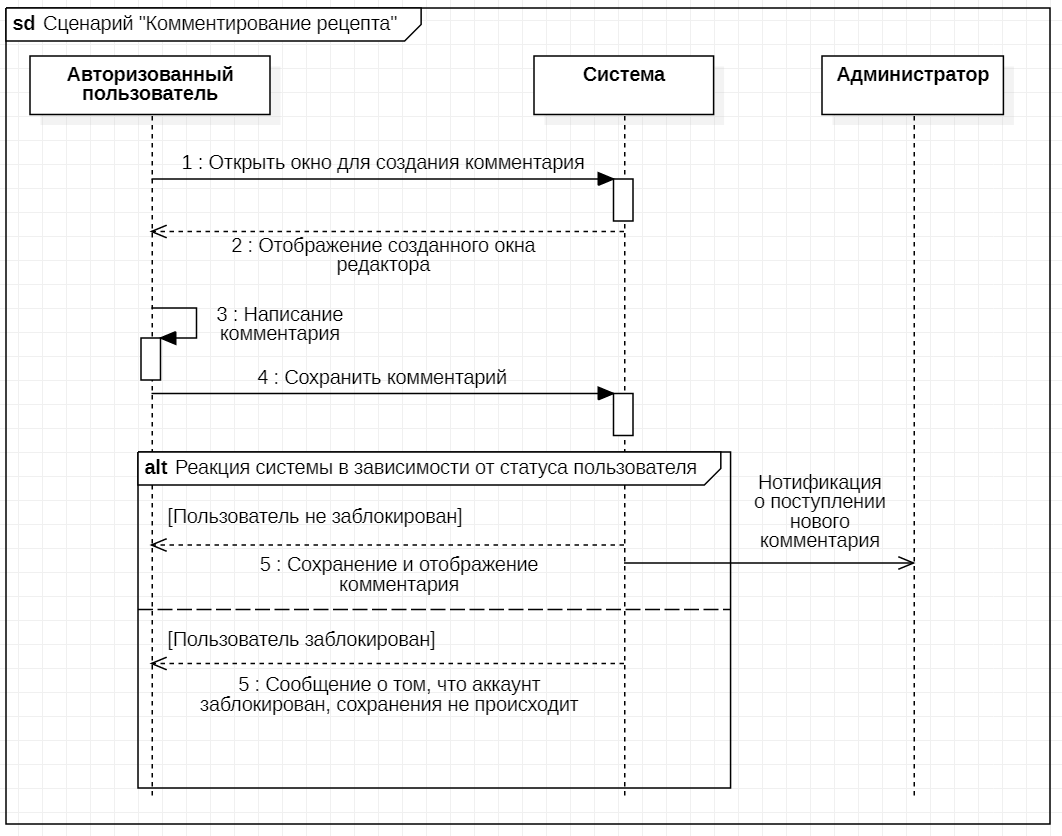
**Основной сценарий**:

1. Пользователь нажимает на кнопку добавления комментария – к рецепту или в ответ на чей-то комментарий.
2. Система открывает на странице окно для создания комментария.
3. Пользователь пишет комментарий и отправляет его, нажав на соответствующую кнопку.
4. Система сохраняет комментарий.
5. Система отправляет нотификацию администратору о новом комментарии.
6. Система отображает добавленный комментарий.

**Альтернативы и исключения:**

*Аккаунт заблокирован*

1. Система выдает сообщение о том, что аккаунт заблокирован и записи комментария не происходит.



### Голосование по рецепту

**Цель**: проголосовать по рецепту.

**Акторы**: авторизованный пользователь, система (сайт, бэкенд).

**Контекст**: авторизованный пользователь находится на странице заинтересовавшего его рецепта.

**Предусловие**: пользователь зашел на сайт, авторизовался и открыл страницу заинтересовавшего его рецепта.

**Основной сценарий**:

1. Пользователь нажимает на кнопку голосования – за или против.
2. Система сохраняет измененный рейтинг рецепта в соответствии со сделанным выбором.
3. Система отображает факт голосования и обновленный рейтинг рецепта.

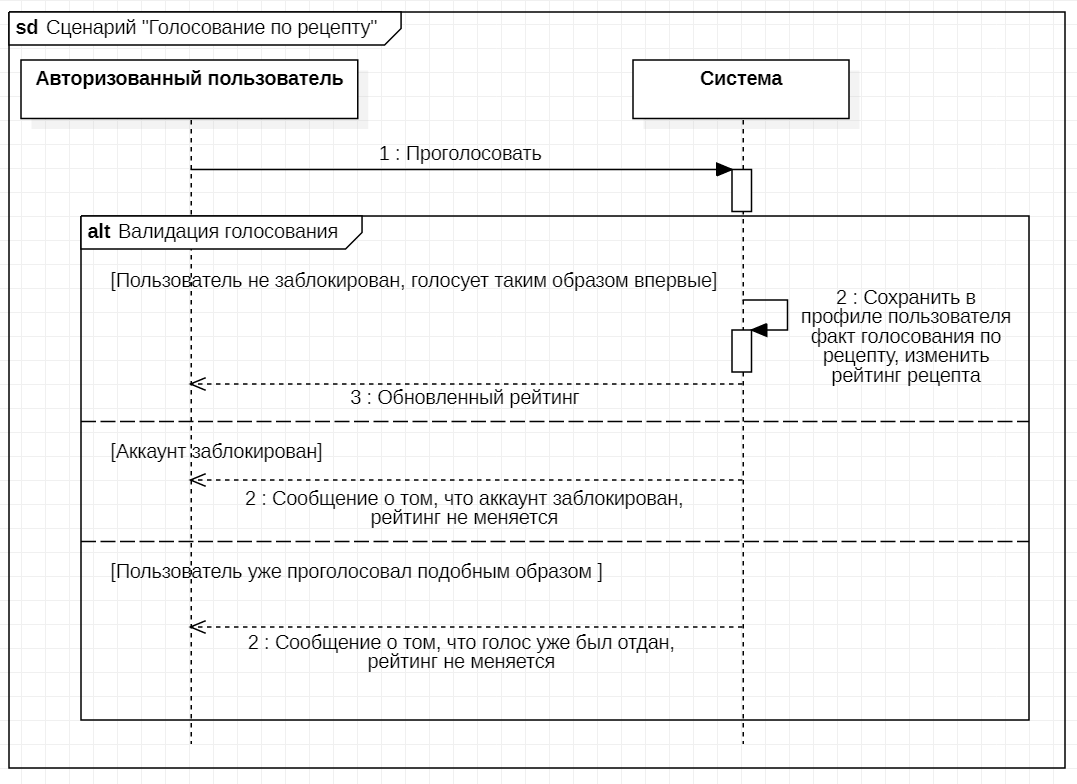
**Альтернативы и исключения:**

*Аккаунт заблокирован*

1. Система выдает сообщение о том, что аккаунт заблокирован, изменения рейтинга рецепта не происходит.

*Пользователь уже проголосовал подобным образом за выбранный рецепт (борьба с накрутками)*

1. Система выдает сообщение о том, что голос уже был отдан, изменения рейтинга рецепта не происходит.



### Занесение рецепта в избранное

**Цель**: сохранить рецепт в избранном, чтобы всегда был под рукой.

**Акторы**: авторизованный пользователь, система (сайт, бэкенд).

**Контекст**: авторизованный пользователь находится на странице заинтересовавшего его рецепта.

**Предусловие**: пользователь зашел на сайт, авторизовался и открыл страницу заинтересовавшего его рецепта.

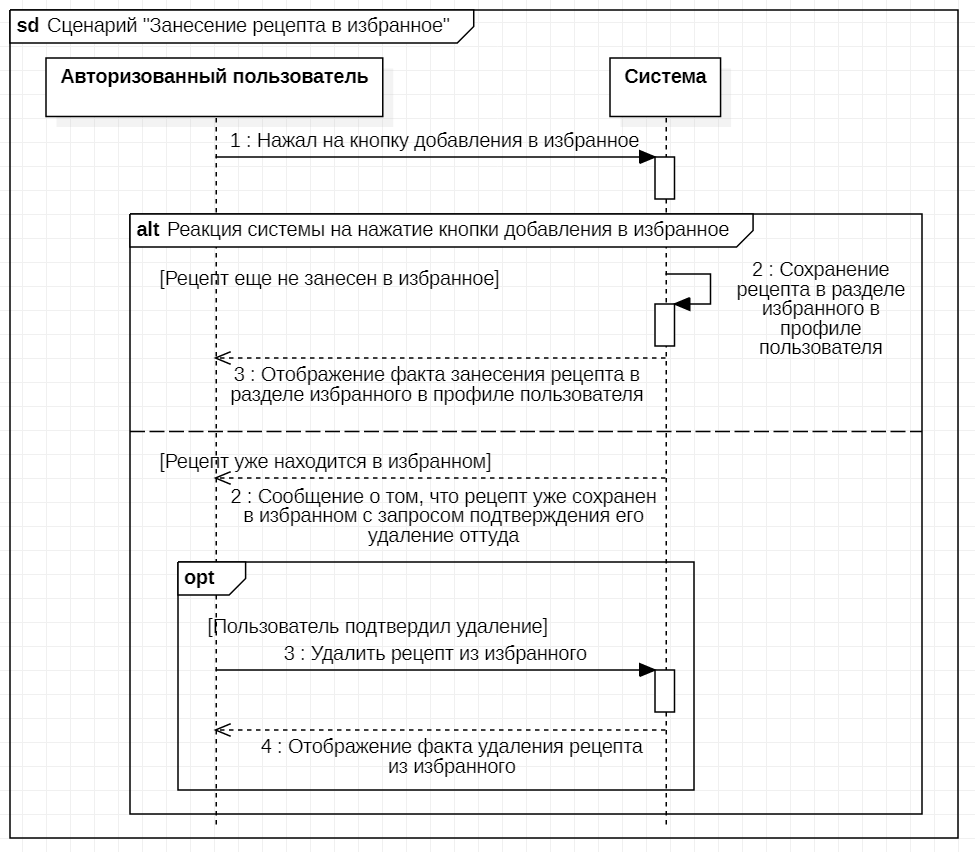
**Основной сценарий**:

1. Пользователь нажимает на кнопку сохранения в избранном.
2. Система сохраняет рецепт разделе избранного в профиле пользователя.
3. Система отображает факт сохранения рецепта в избранном.

**Альтернативы и исключения:**

*Пользователь нажимает на кнопку сохранения в избранном для уже находящегося там рецепта (т.е. команда «отжать кнопку»)*

1. Система выдает сообщение о том, что рецепт уже сохранен в избранном, и просит подтвердить его удаление оттуда.
2. При подтверждении рецепт удаляется из избранного вместе с заметками по нему, иначе нажатие игнорируется.



### Редактирование сборника рецептов администратором

**Цель**: редактирование сборника рецептов.

**Акторы**: администратор, система (административный раздел сайта, бэкенд).

**Контекст**: администратор авторизовался на сайте для того, чтобы редактировать рецепты.

**Предусловие**: администратор авторизовался в административном разделе сайта.

**Основной сценарий**:

1. Администратор открыл страницу Администратор: редактирование рецепта.
2. Система отобразила список кухонь мира, зарегистрированных в системе.
3. Администратор выбрал страну, рецепты которой он хочет редактировать.
4. Система отобразила список имеющихся рецептов выбранной кухни.
5. Администратор выбрал рецепт для редактирования.
6. Система отобразила данные выбранного рецепта.
7. Администратор внес изменения в рецепт и нажал на кнопку сохранения.
8. Система сохранила изменения.

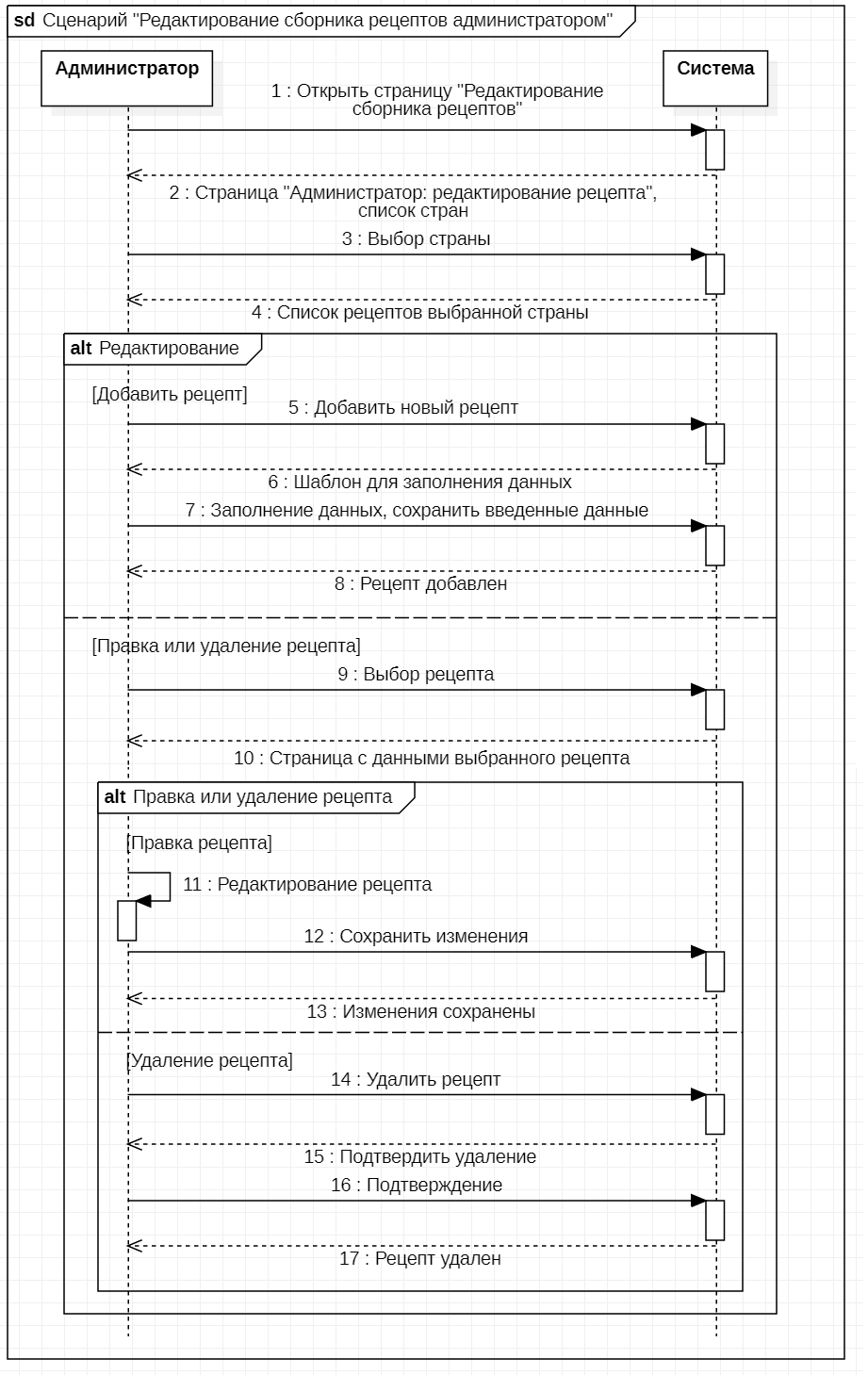
**Альтернативы и исключения:**

*Добавление рецепта*

1. Администратор нажал на кнопку добавления рецепта.
2. Система предоставила шаблон для заполнения информации о новом рецепте.
3. Администратор заполнил всю необходимую информацию о рецепте и нажал на кнопку сохранения.
4. Система сохранила изменения.

*Удаление рецепта*

1. Администратор нажал на кнопку удаления рецепта.
2. Система запросила подтверждение удаления рецепта.
3. Администратор подтвердил удаление рецепта.
4. Система удалила рецепт из списка и сохранила изменения.



### Модерация комментариев администратором

**Цель**: отслеживание соблюдения пользователями правил сайта.

**Акторы**: администратор, система (сайт, бэкенд).

**Контекст**: администратор авторизовался на сайте для того, чтобы модерировать комментарии пользователей.

**Предусловие**: администратор получил уведомление о новом комментарии, прочитав комментарий, обнаружил, что этот комментарий нарушает правила сайта, администратор авторизовался на сайте.

**Основной сценарий**:

1. Администратор открыл страницу рецепта, где содержится комментарий, нарушающий правила сайта.
2. Администратор открыл визитку пользователя, написавшего комментарий, нарушающий правила сайта.
3. Администратор изменил статус пользователя с учетом его текущего статуса (предупреждение, если это первое нарушение, или блокировка, если это уже повторный случай).
4. Система записала это изменение и выслала соответствующее уведомление на e-mail нарушителя.
5. Администратор нажал на кнопку удаления комментария, нарушающего правила сайта.

***Примечание***: *комментарий удаляется вместе со всеми комментариями-потомками.*

1. Система запросила подтверждение удаления комментария.
2. Администратор подтвердил удаление комментария.
3. Система удалила комментарий, нарушающий правила сайта.

**Альтернативы и исключения:**

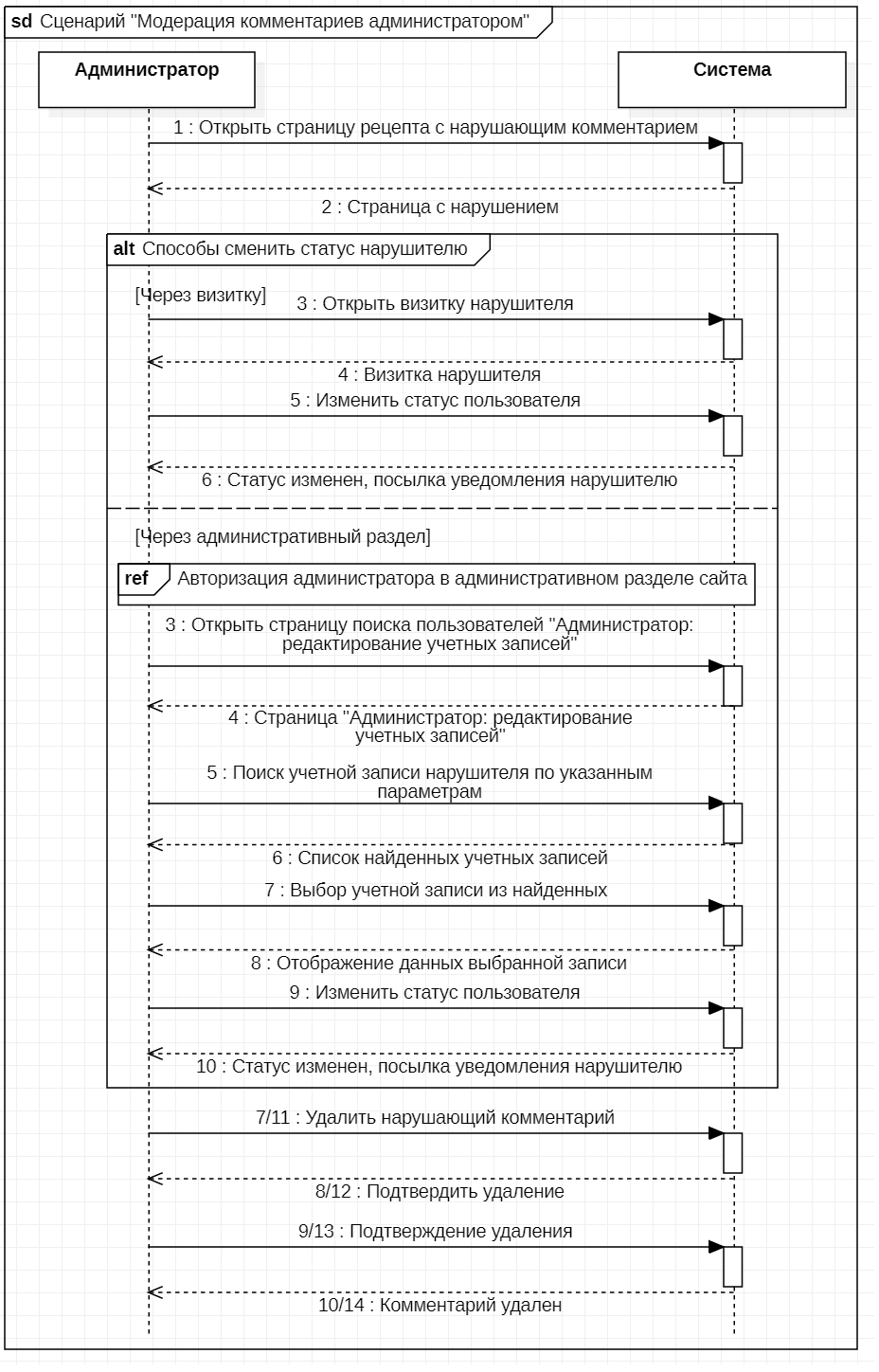
*Администратор использует административный раздел сайта*

1. Администратор авторизовался в административном разделе сайта.
2. Администратор открыл страницу поиска пользователей Администратор: редактирование учетных записей.
3. Администратор выполнил поиск нарушителя.
4. Администратор изменил статус пользователя с учетом его текущего статуса (предупреждение, если это первое нарушение, или блокировка, если это уже повторный случай).
5. Система записала это изменение и выслала соответствующее уведомление на e-mail нарушителя.
6. Администратор на сайте нажал на кнопку удаления комментария, нарушающего правила сайта.

***Примечание***: *комментарий удаляется вместе со всеми комментариями-потомками.*

1. Система запросила подтверждение удаления комментария.
2. Администратор подтвердил удаление комментария.
3. Система удалила комментарий, нарушающий правила сайта.

*Примечание: не знаю, как правильно нумеровать пункты после alt, если они при переходе по одной ветке будут 7-10, а по второй – 11-14, поэтому поставила нумерацию через слэш.*



## Функциональные требования

По завершению новым пользователем процедуры регистрации система должна создать ему аккаунт.

После ввода пользователем логина (e-mail) и пароля, система должна авторизовать пользователя для осуществления доступа к личному кабинету.

При вводе неверно указанных креденшелсов система должна выдать пользователю сообщение об ошибке.

В том случае, когда зарегистрированный пользователь не может вспомнить свой пароль, но хочет авторизоваться, система должна предоставить ему функционал по восстановлению пароля.

После ввода пользователем своих личных данных (ФИО, контактный номер телефона, e-mail) система должна обеспечить их сохранение.

По завершению зарегистрированным пользователем процедуры смены пароля система должна изменить его пароль на новый.

При изменении зарегистрированным пользователем личных данных (ФИО, контактный номер телефона, e-mail) система должна обеспечить их обновление.

После входа авторизованного пользователя в личный кабинет система должна предоставлять пользователю список сохраненных в избранном рецептов для просмотра.

После входа авторизованного пользователя в личный кабинет система должна предоставлять пользователю список сохраненных в избранном рецептов для его редактирования.

После входа авторизованного пользователя в личный кабинет система должна предоставлять пользователю список сохраненных в избранном рецептов для редактирования заметок к ним.

При добавлении нового комментария авторизованного пользователя о рецепте или связанных с ним точках питания, система должна сохранять его наряду с остальными комментариями к этому рецепту.

При редактировании заметки к рецепту из избранного, система должна сохранять сделанное изменение.

Система должна предоставлять пользователю функционал для поиска рецептов по заданным критериям (название блюда, ингредиенты, входящие в состав блюда, ингредиенты, не входящие в состав блюда).

После того, как пользователь указал критерий для поиска рецепта, система должна отобразить все найденные по этому критерию рецепты.

После того, как пользователь выбрал для просмотра рецепт, система должна отобразить всю доступную информацию по этому рецепту.

Для выбранного рецепта блюда в зависимости от выбора пользователя система должна отобразить информацию о точках питания, в которых готовятся блюда по этому рецепту: в виде карты или списком.

Система должна позволять проголосовать не более одного раза «за» или «против» рецепта одному и тому же авторизованному пользователю.

После авторизации в административном разделе сайта система должна предоставить администратору доступ к редактированию сборника рецептов.

После авторизации в административном разделе сайта система должна предоставить администратору доступ к поиску и редактированию данных пользовательских аккаунтов.

После авторизации на пользовательской части сайта система должна предоставить администратору возможность удалять комментарии к рецептам, нарушающие правила сайта.

В случае обнаружения администратором нарушения правил сайта система должна предоставлять администратору доступ к аккаунту нарушителя для посылки предупреждения, если пользователь провинился впервые, или блокировки аккаунта (день/неделя/навечно), если это повторный случай.

В случае блокировки аккаунта система должна не разрешать пользователю авторизовываться, оставлять комментарии и голосовать.

## Требования, налагаемые регулятором

Система должна обеспечивать передачу, обработку, защиту и хранение персональных данных в соответствии с 152-ФЗ.

Система должна хранить данные пользовательских действий в течение 30 дней по требованиям федерального закона.

## Производительность

Система должна обеспечивать полную работоспособность при нагрузке до 10 000 пользователей в сутки.

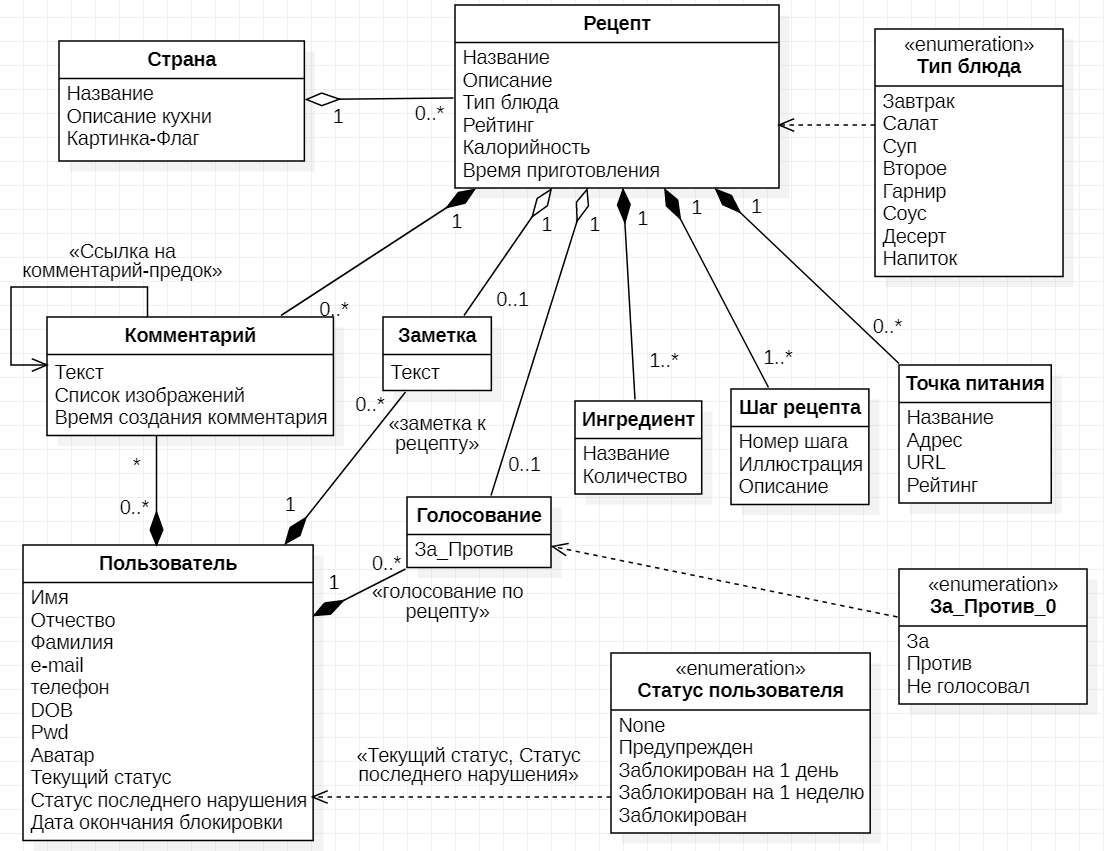
## Удобство использования

Пользовательский интерфейс системы должен соответствовать требованиям стандарта Web Accessibility Initiative (WAI) доступности веб-приложений для людей с ограниченными возможностями.

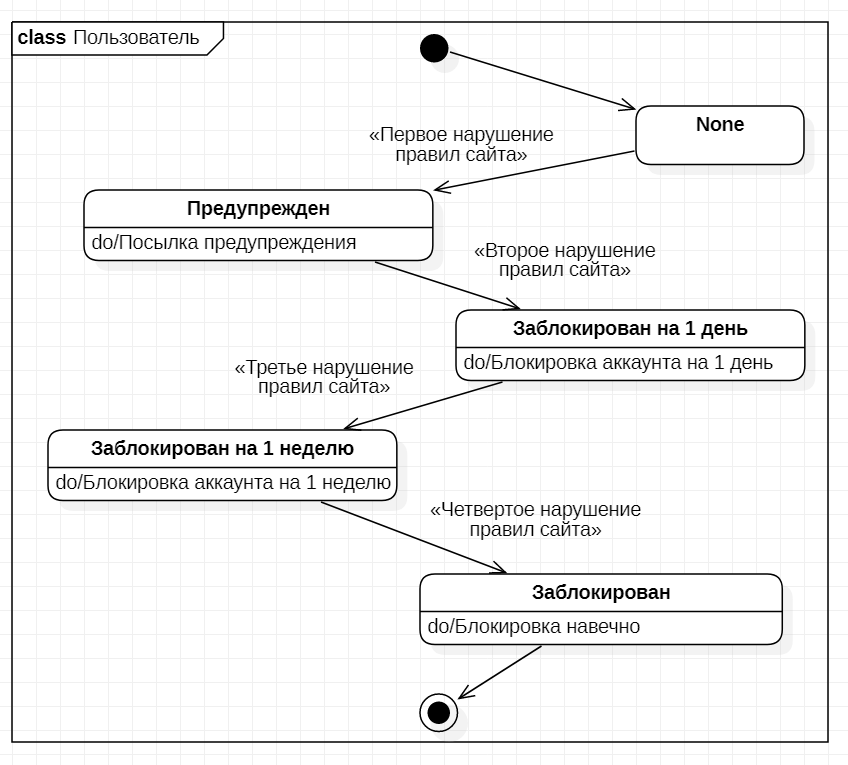
Цвета и шрифты пользовательского интерфейса веб-сайта должны соответствовать требованиям стандарта по инклюзивному дизайну WCAG 2.0.

Пользовательский интерфейс должен быть адаптивным — поддерживать работу веб-браузерах на устройствах: десктопный компьютер, планшетный компьютер, смартфон.

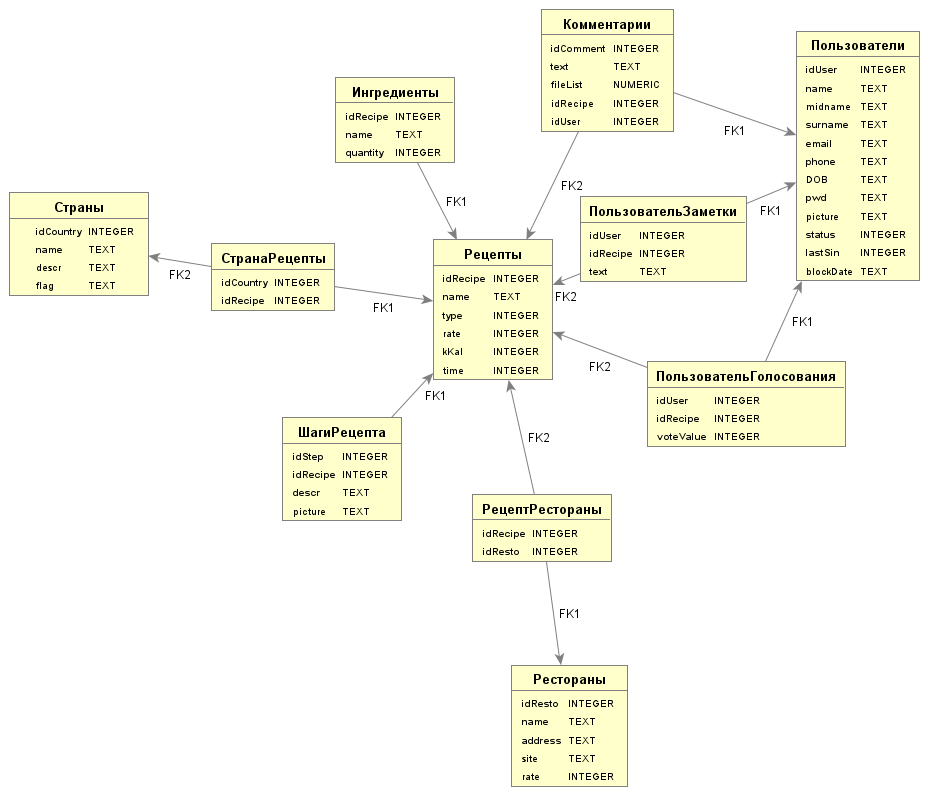
## Логическая модель данных



### Статус пользователя

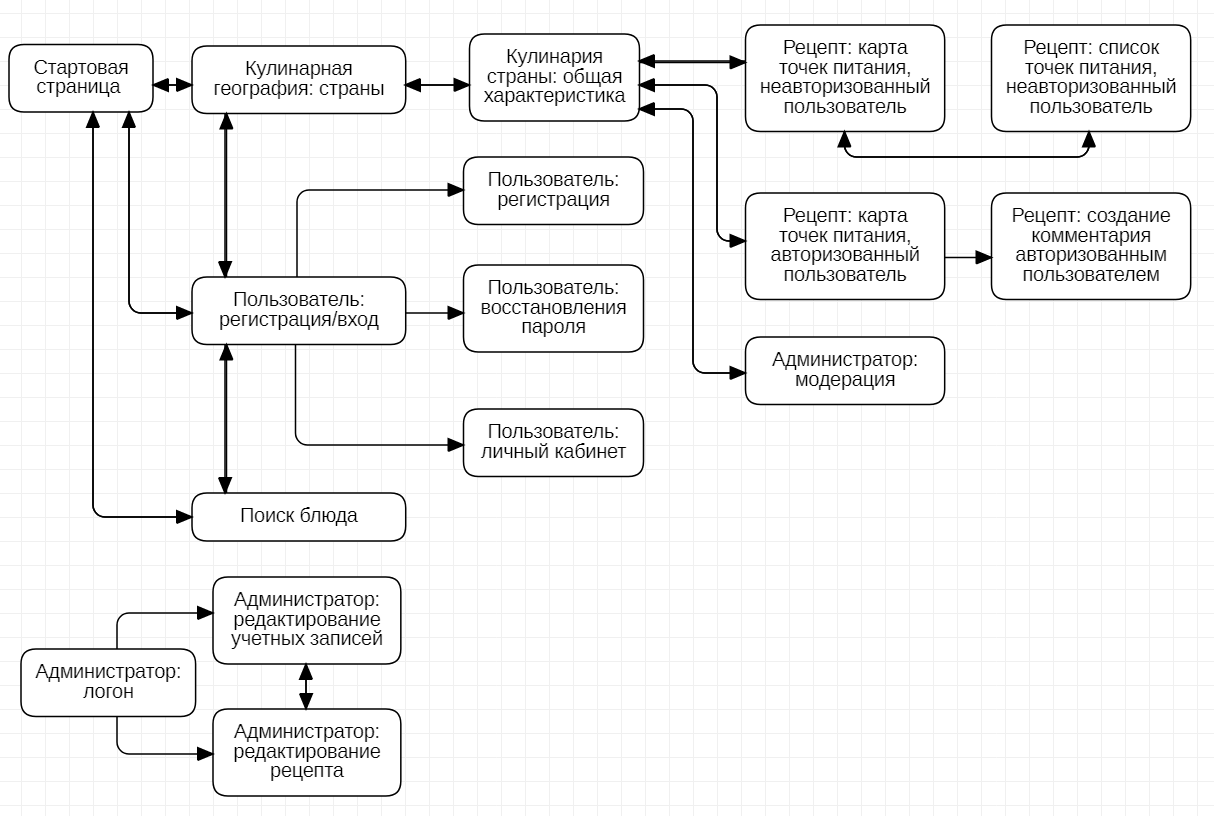


### Схема баз данных, реализующая логическую модель данных



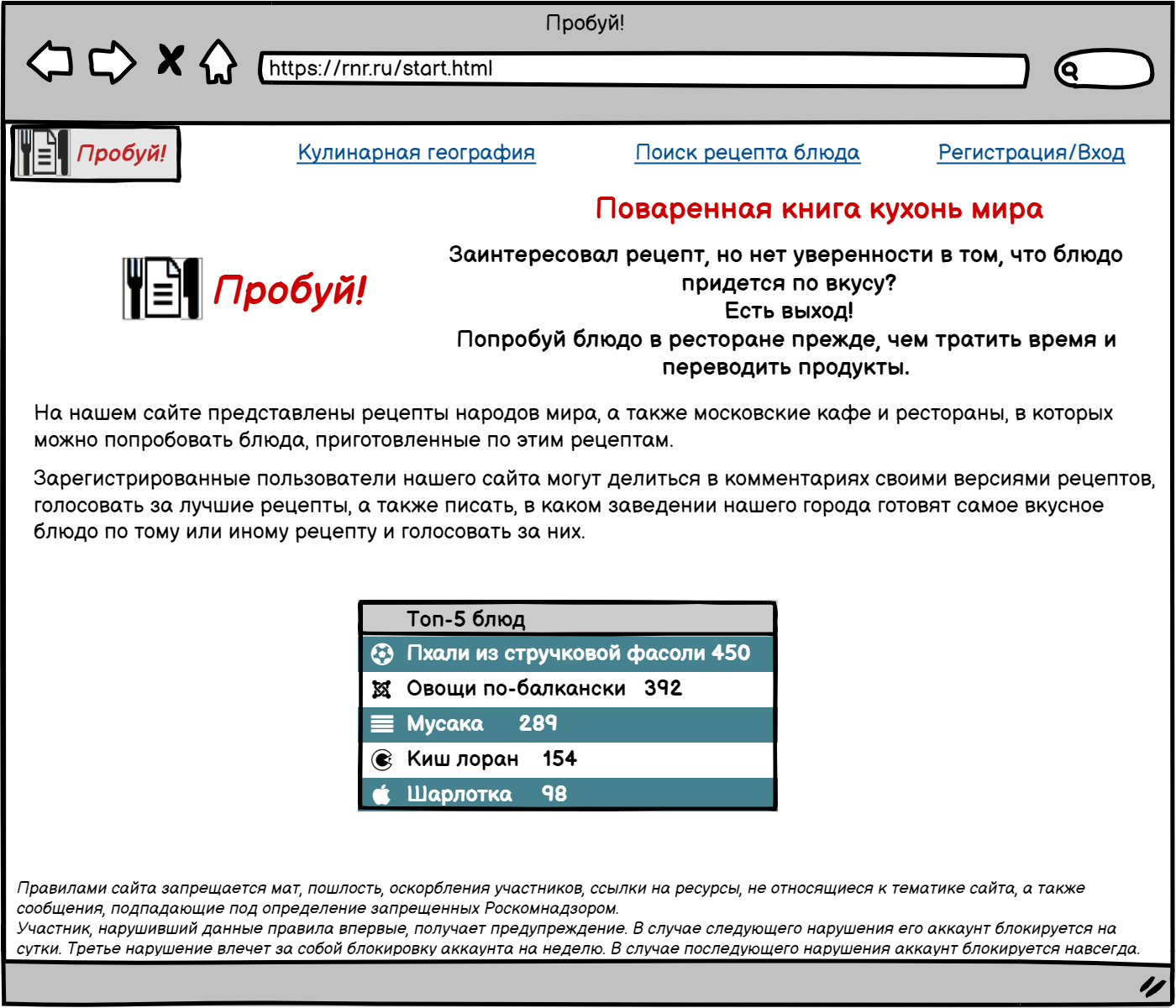
## Пользовательский интерфейс

### Карта страниц пользовательского интерфейса приложения

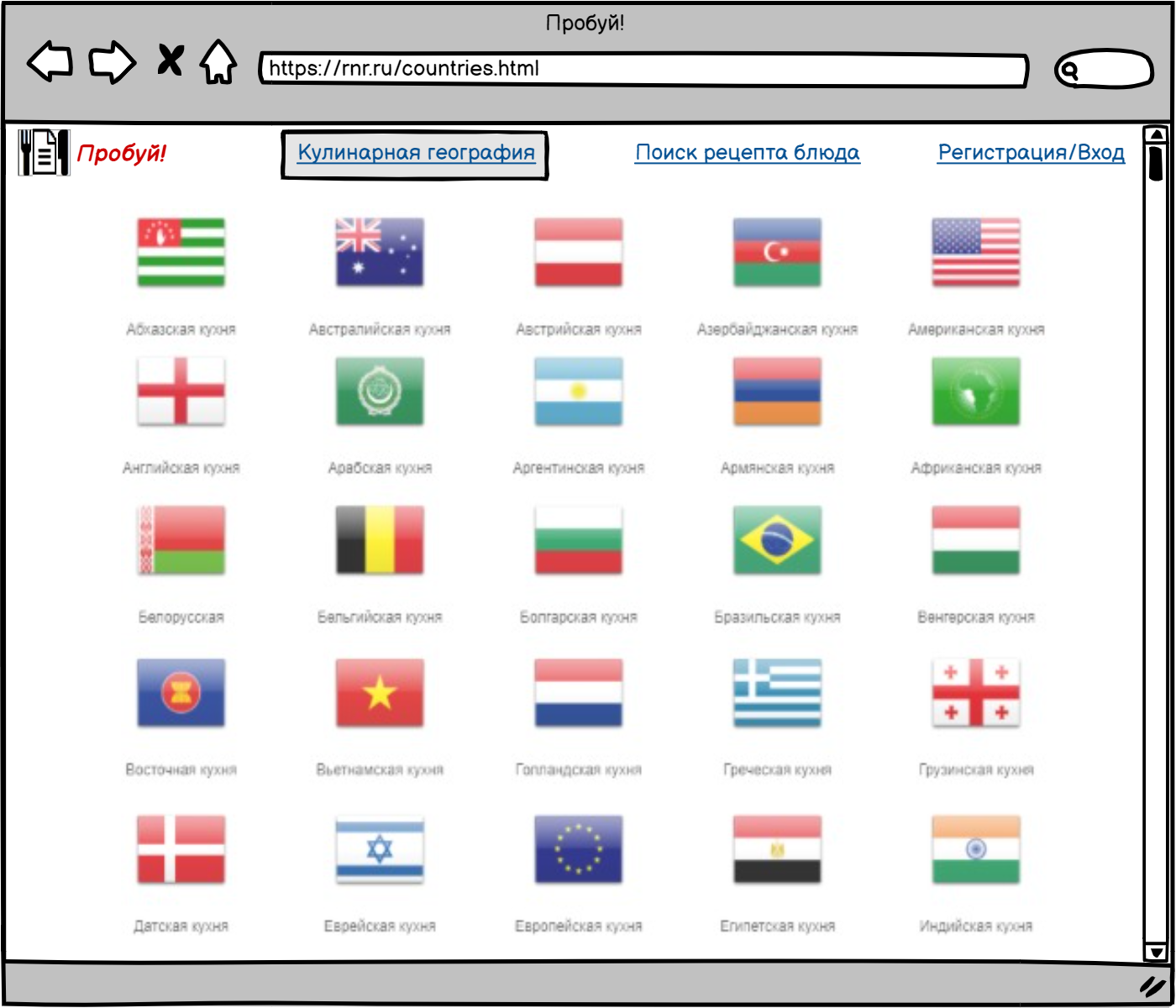


*Примечание*: на страницы «Стартовая страница», «Кулинарная география: страны», «Пользователь: регистрация/вход»/«Пользователь: личный кабинет» и «Поиск блюда» можно попасть с любой страницы через главное меню.

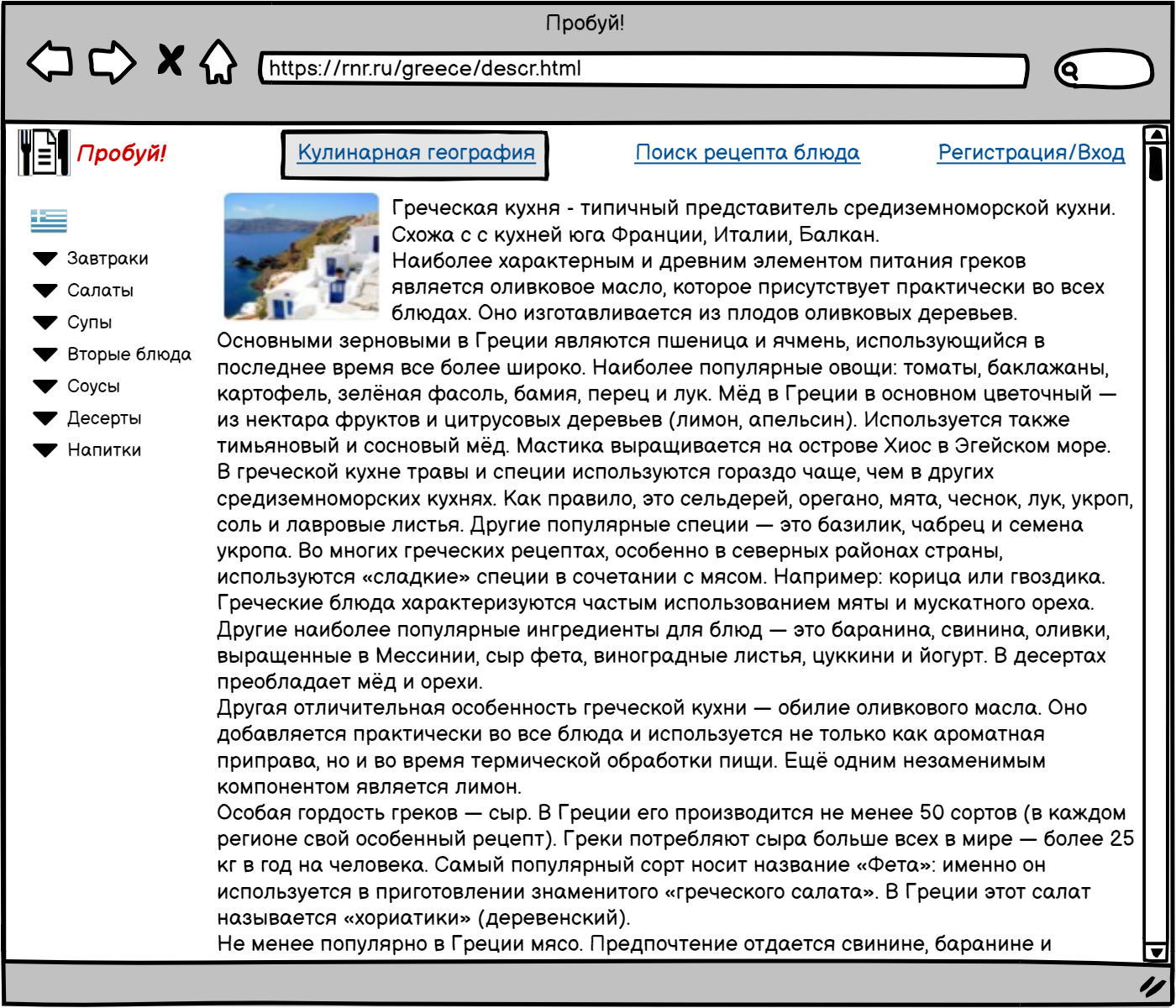
### Стартовая страница



### Кулинарная география: страны



### Кулинария страны: общая характеристика



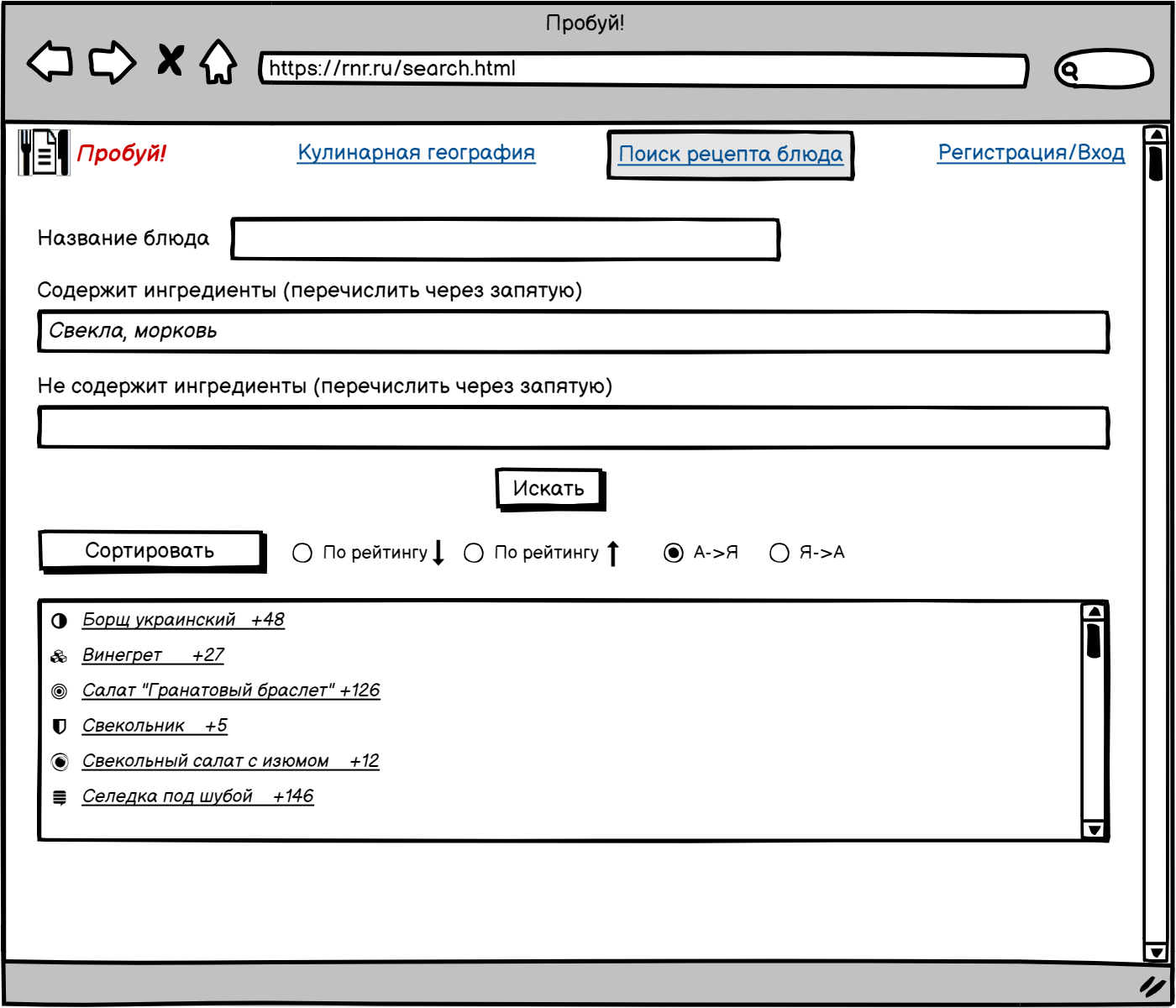
### Рецепт: карта точек питания, неавторизованный пользователь



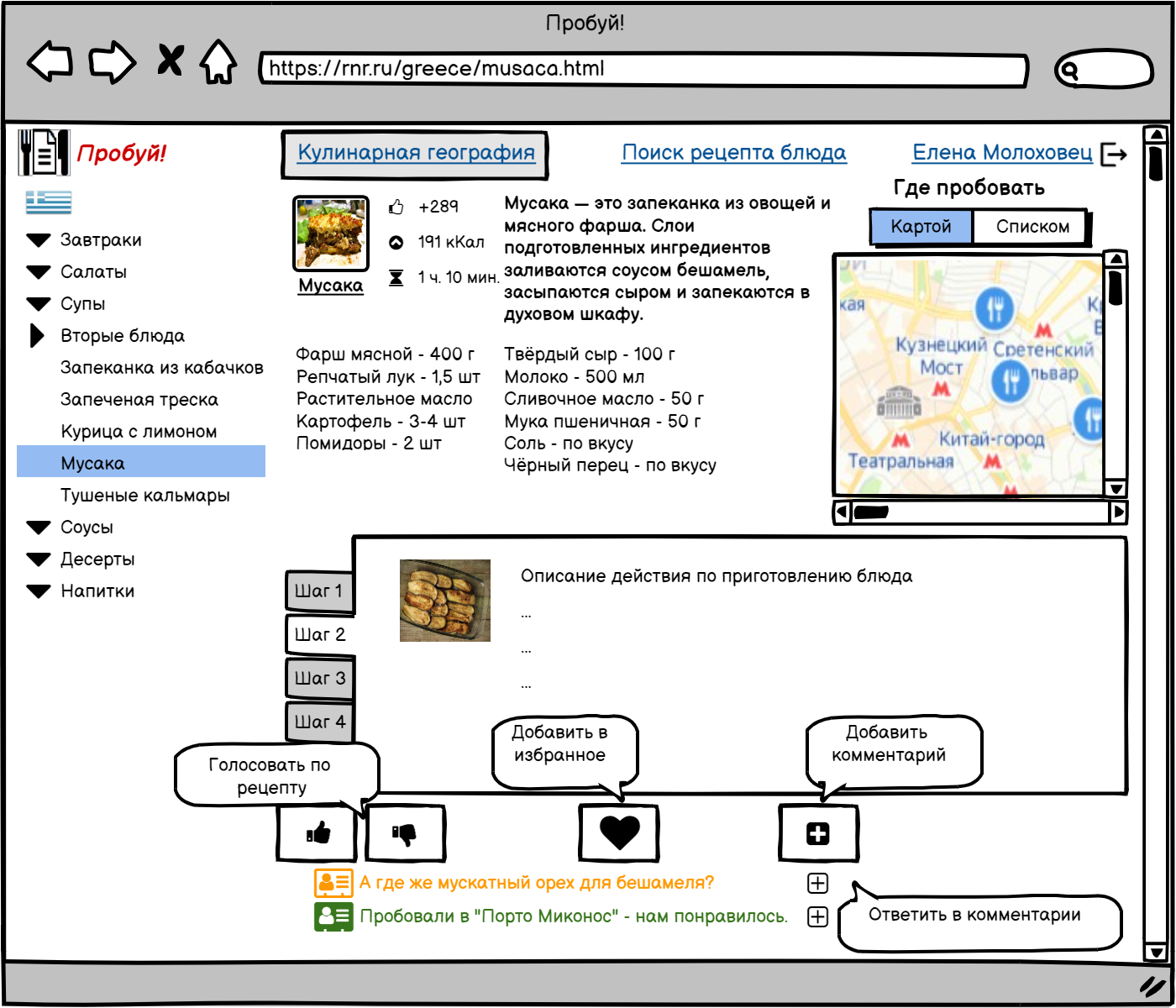
### Рецепт: список точек питания, неавторизованный пользователь



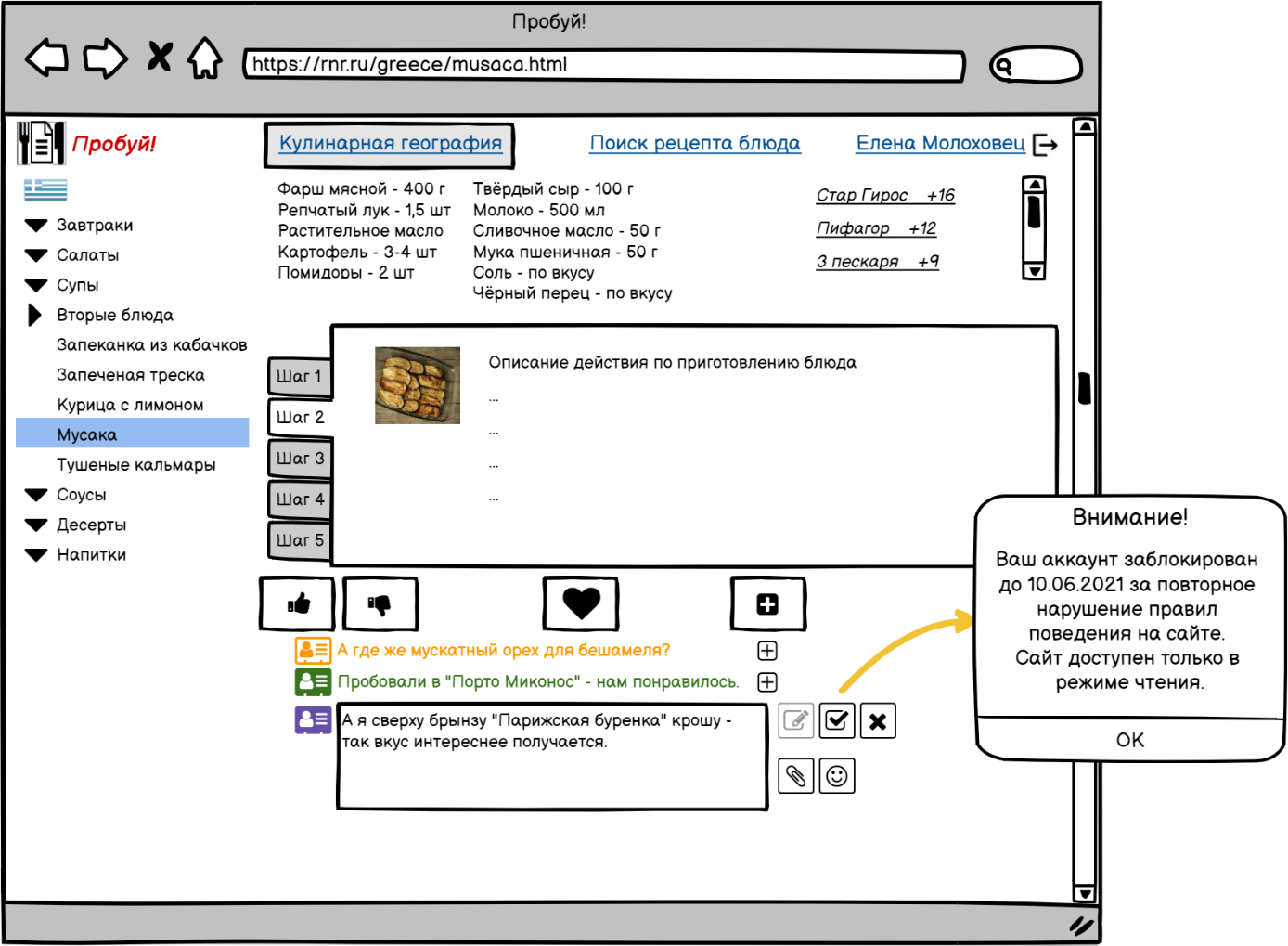
### Поиск рецепта блюда



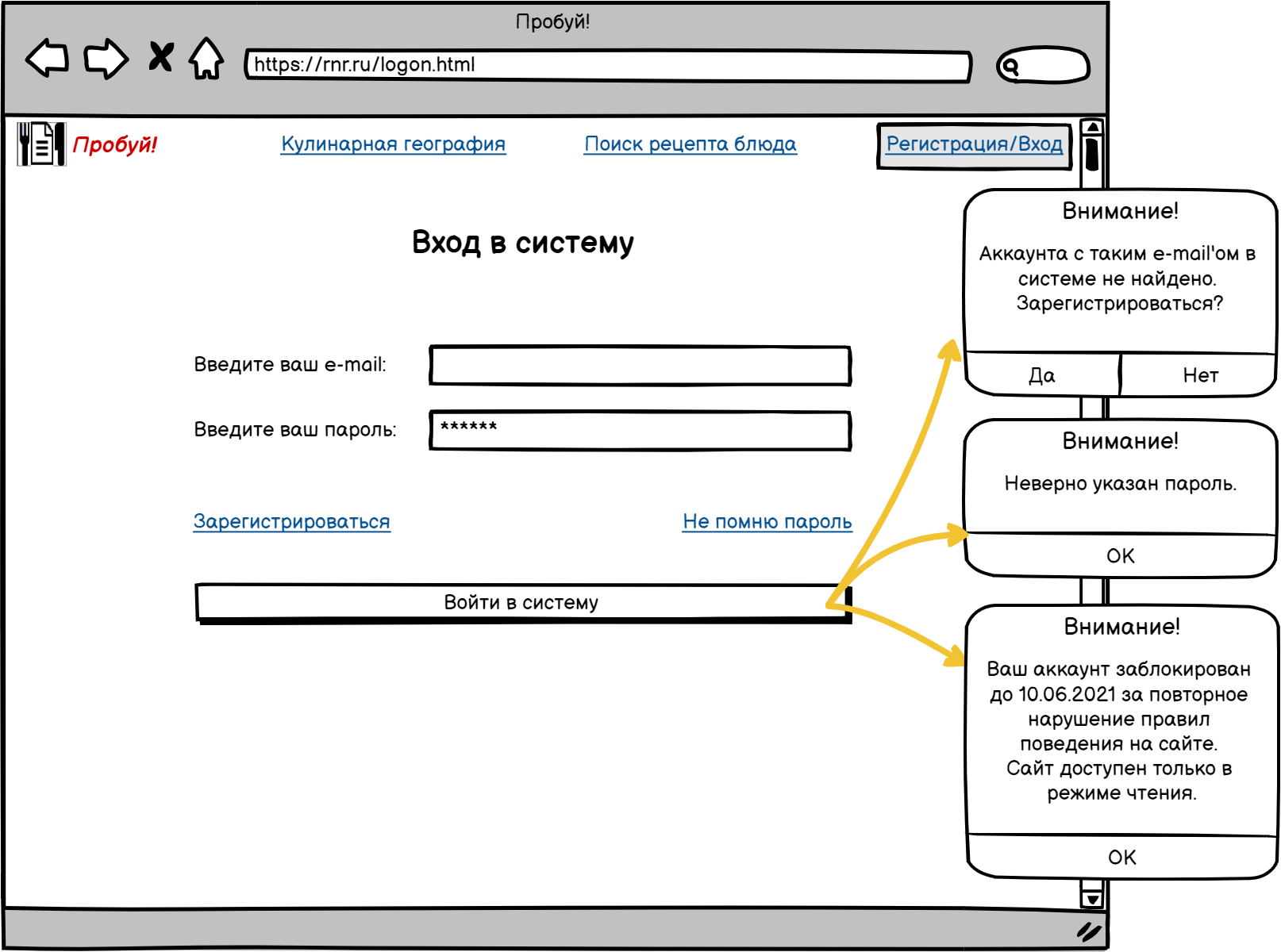
### Рецепт: карта точек питания, авторизованный пользователь



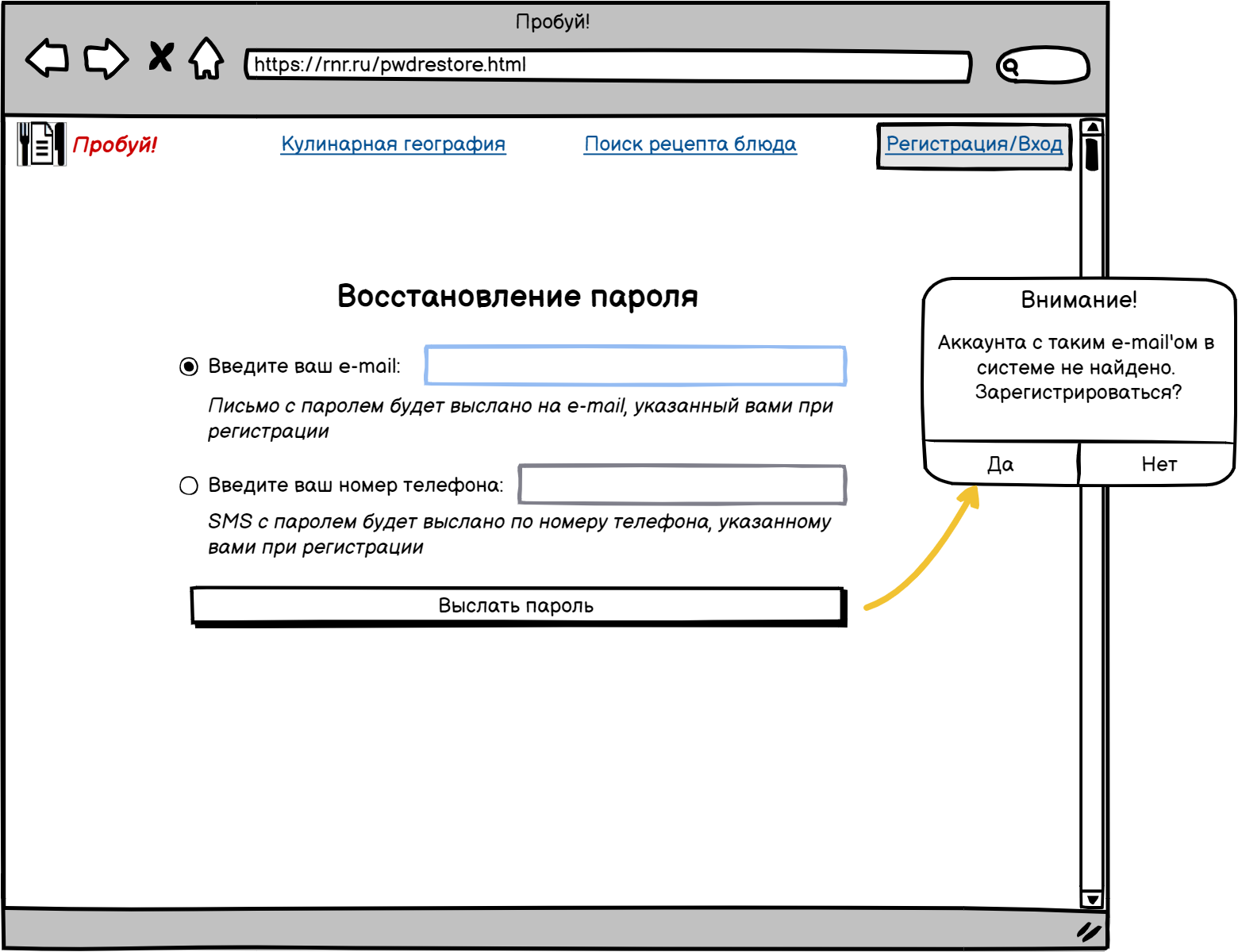
### Рецепт: создание комментария авторизованным пользователем



### Пользователь: регистрация/вход



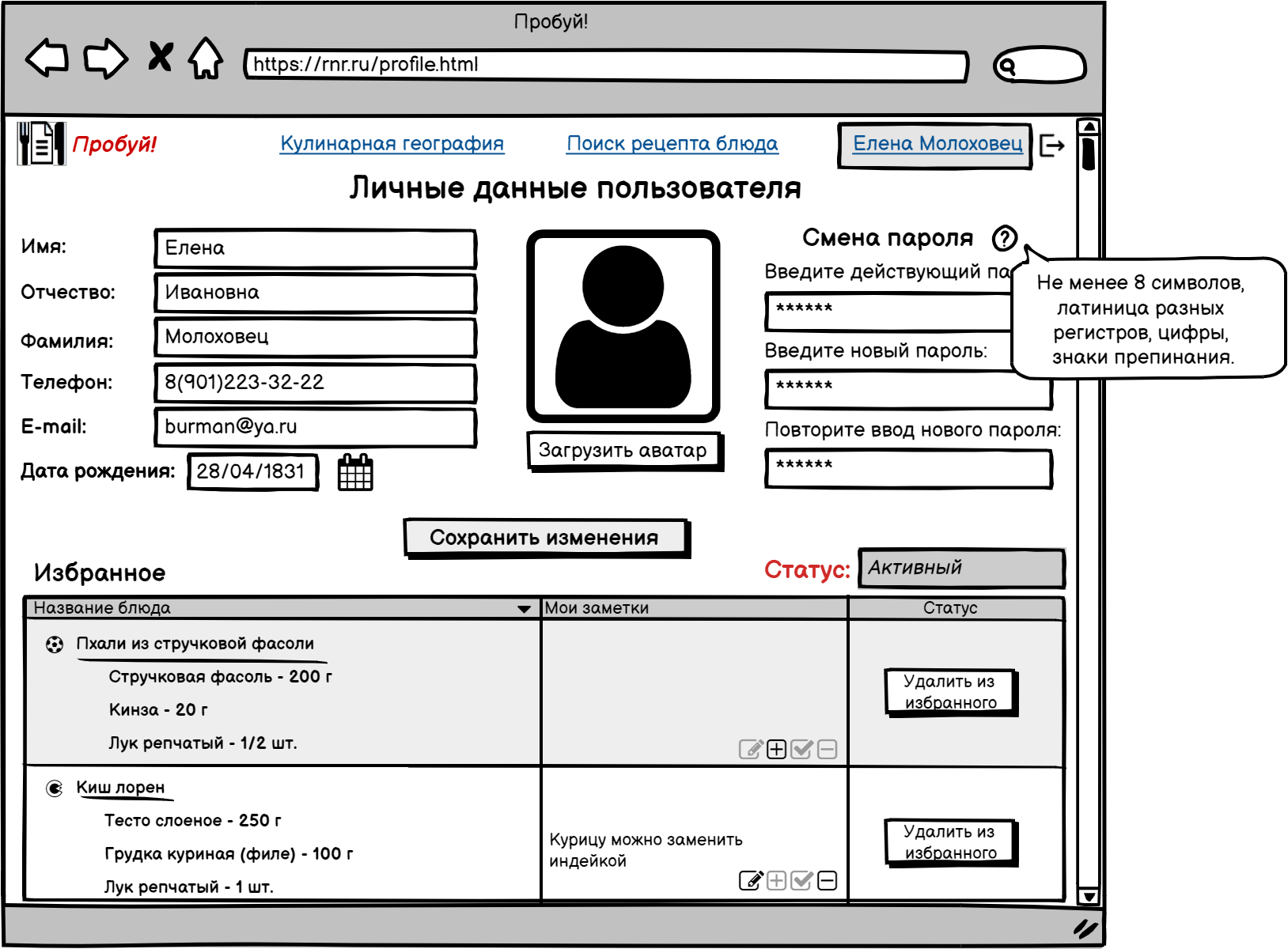
### Пользователь: восстановления пароля



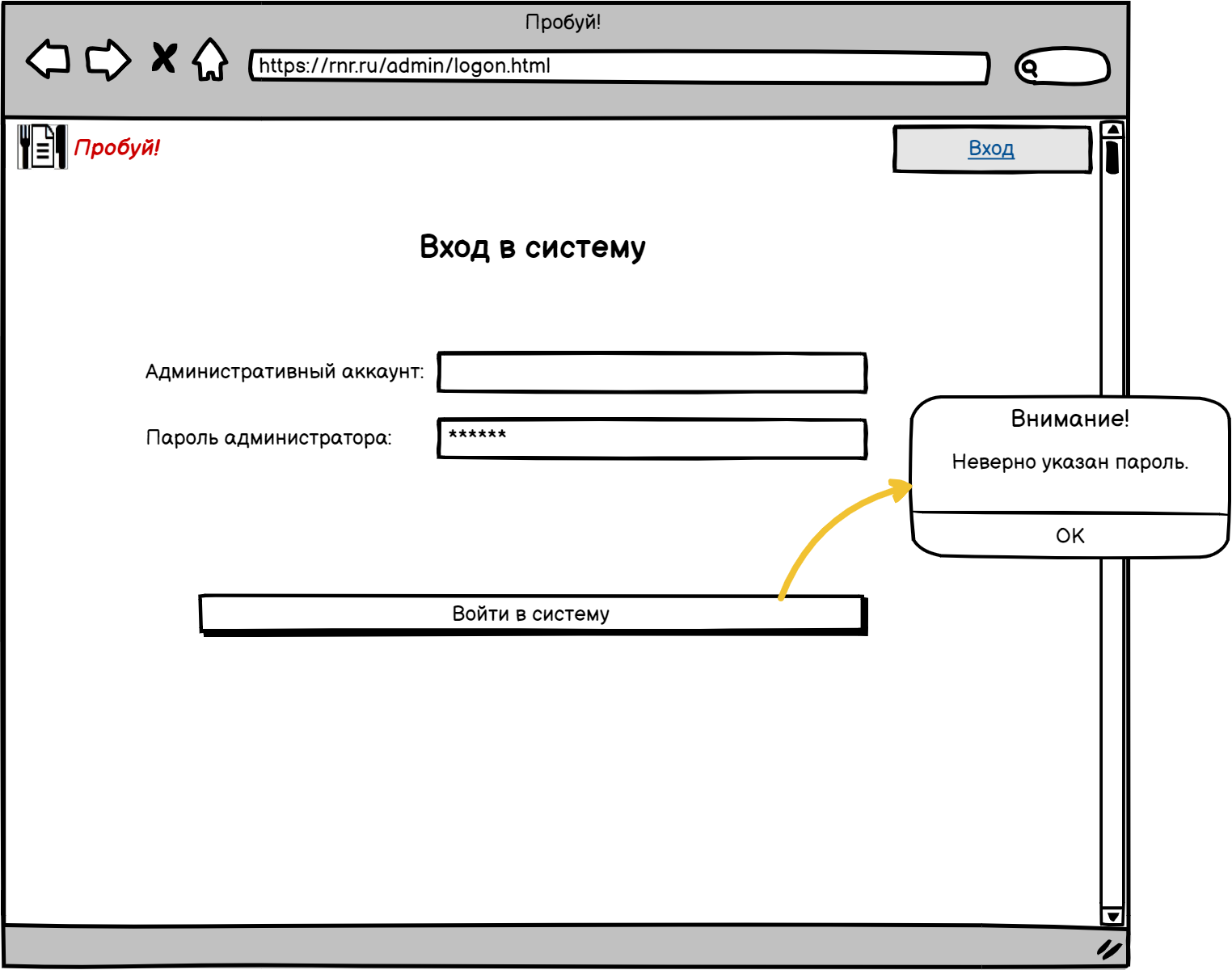
### Пользователь: регистрация



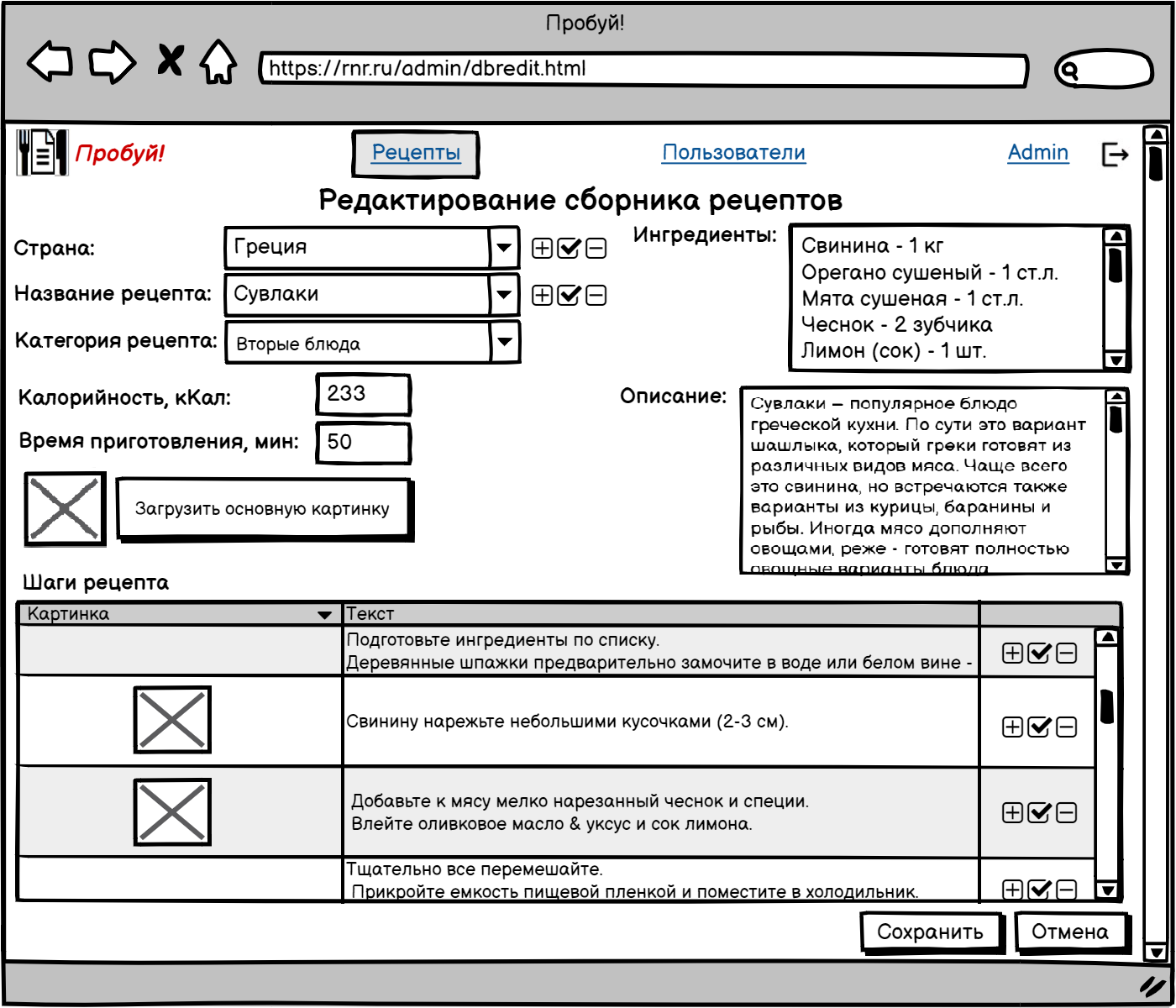
### Пользователь: личный кабинет



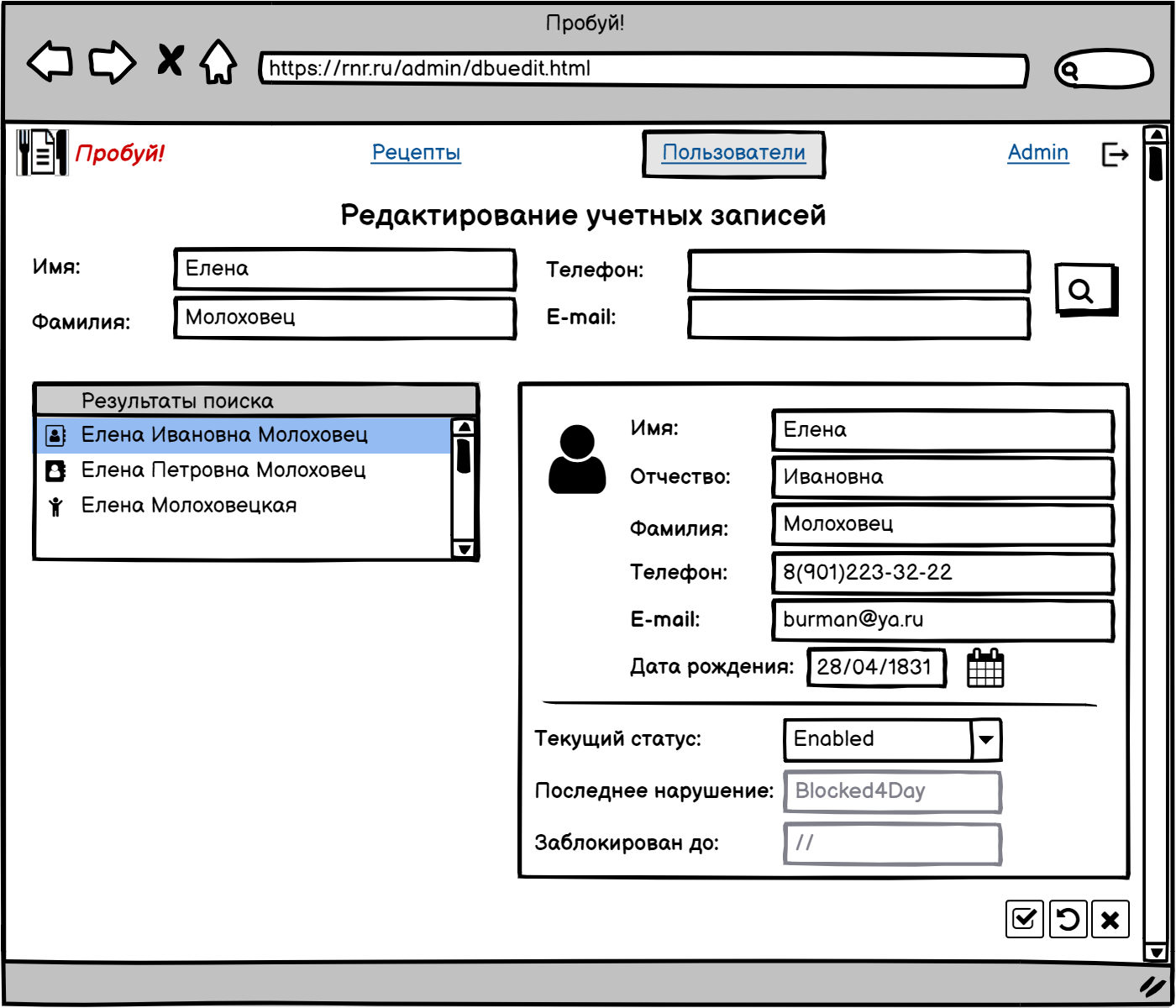
### Администратор: логон



### Администратор: редактирование рецепта

s

### Администратор: редактирование учетных записей



### Администратор: модерация



## Программные интерфейсы

Бэкенд системы должен предоставлять REST API по протоколу HTTP для доступа к данным.

**URL сайта**: <https://rnr.ru>

### Ресурс аккаунта пользователя

URL: /accounts

Предоставляет управление информацией об аккаунте пользователя: создание, изменение, чтение.

***Примечание****: удаление пока не используется, но может быть востребовано в перспективе, например, при самоудалении аккаунта. Заблокированных навечно можно по прошествии года списывать в архив в виде e-mail’а и проверять аккаунт при регистрации на заблоченность (т.е. не содержится ли он в архиве).*

**Объекты, используемые в API аккаунта пользователя**

**Объект** **account**

***Примечание****: поля объекта account, выделенные курсивом, в тело запроса на создание не включаются.*

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| *idUser* | *int* | *идентификатор пользователя* |
| name | string | имя пользователя |
| midname | string | отчество пользователя |
| surname | string | фамилия пользователя |
| photo | string | ссылка на фото пользователя |
| email | string | адрес электронной почты |
| phone | string | номер мобильного телефона |
| DOB | datetime | дата рождения |
| pwd | string | пароль (*шифруется base64 на клиенте при передаче*) |
| *status* | *int* | *текущий статус пользователя* |
| *lastSin* | *int* | *последнее нарушение* |
| *blockTillDate* | *datetime* | *дата окончания блокировки* |
| *favorits* | *array[int recId, strnig note]* | *массив объектов, каждый из которых представляет собой пару: идентификатор рецепта recId (см. поле idRecipe объекта* **recipe***) и текстовая заметка к нему note* |
| *votes* | *array[int recId, int vote]* | *массив объектов, каждый из которых представляет собой пару: идентификатор рецепта recId (см. поле idRecipe объекта* **recipe***) и результат голосования пользователя по этому рецепту vote:*  *-1 - против,*  *0 – не голосовал,*  *1 – за.* |

#### Создание нового аккаунта

POST /accounts

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | POST |
| **Ресурс** | /accounts |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: «application/json») |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 201. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/accounts/[id]", где [id] – присвоенный идентификатор пользователя |

**Описание полей тела запроса**:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| **account** | object | представляет собой набор данных пользователя |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **account** | object | представляет собой набор данных пользователя |

**Пример:**

**Тело запроса**:

{

    "account": {

        "name": "Вильям",

        "surname": "Похлебкин",

        "midname": "Васильевич",

        "photo": "https://clck.ru/VKaWr",

        "email": "phb@ya.ru",

        "phone": "8(911)1234567",

        "DOB": "22/02/2001",

        "pwd": "d2UzZUsh"

    }

}

**Тело ответа**:

{

**"Version": "1.0.1.10",**

**"RetStatus": "Success",**

    "account": {

**"idUser": 2,**

        "name": "Вильям",

        "surname": "Похлебкин",

        "midname": "Васильевич",

        "photo": "https://clck.ru/VKaWr",

        "email": "phb@ya.ru",

        "phone": "8(911)1234567",

        "DOB": "22/02/2001",

        "pwd": "d2UzZUsh",

**"status": 1,**

**"lastSin": 0,**

**"blockTillDate": "",**

**"favorits": [],**

**"votes": []**

    }

}

#### Изменение существующего аккаунта

PATCH /accounts[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | PATCH |
| **Ресурс** | /accounts/[id], где [id] – идентификатор пользователя |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: "application/json") |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/accounts/[id]", где [id] – присвоенный идентификатор пользователя |

**Описание полей тела запроса:**

В теле запроса записываются только те поля, значения которых нужно изменить.

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **account** | object | представляет собой набор измененных данных пользователя |

**Пример**:

**Тело запроса**:

{

    "account": {

        "idUser": 2,

        "status": **2**,

        "favorits": [

            {

                "recId": 14,

                "note": "**Заменить кинзу на тархун."**

            },

**{**

**"recId": 19,**

**"note": ""**

**}**

        ],

        "votes": [

            {

**"recId": 14,**

**"vote": 1**

            },

            {

**"recId": 19,**

**"vote": 0**

            }

        ]

    }

}

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "account": {

        "idUser": 2,

        "name": "Вильям",

        "surname": "Похлебкин",

        "midname": "Васильевич",

        "photo": "https://clck.ru/VKaWr",

        "email": "phb@ya.ru",

        "phone": "8(911)1234567",

        "DOB": "22/02/2001",

        "pwd": "d2UzZUsh",

**"status": 2,**

        "lastSin": 0,

        "blockTillDate": "",

        "favorits": [

            {

                "recId": 11,

                "note": "Специй в 2 раза меньше."

            },

            {

                "recId": 14,

                "note": **"Заменить кинзу на тархун."**

            },

            {

                "recId": 7,

                "note": "Лучше сходить в \"Саперави.\""

            },

**{**

**"recId": 19,**

**"note": ""**

**}**

        ],

        "votes": [

            {

**"recId": 14,**

**"vote": 1**

            },

            {

**"recId": 19,**

**"vote": 0**

            }

        ]

    }

}

#### Получение данных аккаунта

GET /accounts/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /accounts/[id], где [id] – идентификатор пользователя |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **account** | object | представляет собой набор данных пользователя |

**Пример**

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "account": {

        "idUser": 2,

        "name": "Вильям",

        "surname": "Похлебкин",

        "midname": "Васильевич",

        "photo": "https://clck.ru/VKaWr",

        "email": "phb@ya.ru",

        "phone": "8(911)1234567",

        "DOB": "22/02/2001",

        "pwd": "d2UzZUsh",

        "status": 1,

        "lastSin": 0,

        "blockTillDate": "",

        "favorits": [

            {

                "recId": 11,

                "note": "Специй в 2 раза меньше."

            },

            {

                "recId": 14,

                "note": ""

            },

            {

                "recId": 7,

                "note": "Лучше сходить в \"Саперави.\""

            }

        ],

        "votes": [

            {

                "recId": 14,

                "vote": 1

            },

            {

                "recId": 19,

                "vote": 0

            }

        ]

    }

}

#### Получение списка аккаунтов с указанными параметрами

GET /accounts

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /accounts |
| **Параметры запроса** | *psize=n* – количество записей, возвращаемых  в запросе, в режиме постраничной выдачи;  формат: целое положительное число;  опциональный, при отсутствии выдаются все записи за один запрос. |
| *offset=m* – элемент, с которого начинается  постраничная выдача;  формат: целое положительное число;  опциональный, работает только в паре с psize, при отсутствии выдача начинается с первого элемента. |
| *name=”xxx” –* имя пользователя начинается  на *xxx*;  опциональный. |
| *surname=”xxx” – фамилия* пользователя  начинается на *xxx*;  опциональный. |
| *phone=”xxx” –* номер мобильного телефона  пользователя начинается на *xxx*;  опциональный. |
| *email=”xxx” –* адрес электронной почты  пользователя;  опциональный. |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| RecNum | int | число записей в массиве Results; формат: целое число. |
| Results | array[**account** ] | список аккаунтов; массив объектов, каждый из которых представляет собой набор данных аккаунта(объект типа ***account*** ). |

**Пример**

GET /accounts?email=phb@ya.ru – возврат аккаунта с адресом электронной почты “phb@ya.ru”

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "Results": [

        {

            "idUser": 2,

            "name": "Вильям",

            "surname": "Похлебкин",

            "midname": "Васильевич",

            "photo": "https://clck.ru/VKaWr",

            "email": "phb@ya.ru",

            "phone": "8(911)1234567",

            "DOB": "22/02/2001",

            "pwd": "d2UzZUsh",

            "status": 1,

            "lastSin": 0,

            "blockTillDate": "",

            "favorits": [

                {

                    "recId": 11,

                    "note": "Специй в 2 раза меньше."

                },

                {

                    "recId": 14,

                    "note": ""

                },

                {

                    "recId": 7,

                    "note": "Лучше сходить в \"Саперави.\""

                }

            ],

            "votes": [

                {

                    "recId": 14,

                 "vote": 1

                },

                {

                 "recId": 19,

                 "vote": 0

                }

            ]

        }

    ]

}

#### Удаление аккаунта

*В данный момент не используется.*

DELETE /accounts/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | DELETE |
| **Ресурс** | /accounts/[id], где [id] – идентификатор пользователя |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** |  |
| **Заголовки ответа** |  |

### Ресурс комментария

URL: /comments

**Объект** **comment:**

***Примечание****: поля объекта comment, выделенные курсивом, в тело запроса на создание не включаются.*

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| *idComment* | *int* | *идентификатор комментария* |
| idRecipe | int | идентификаторрецепта, к которому написан комментарий |
| idUser | int | идентификаторпользователя, написавшего комментарий |
| text | string | текст комментария |
| pictureList | array[string] | список путей приложенных к комментарию картинок |
| parentComment | int | идентификатор комментария-предка |
| timeCreated | datetime | дата создания комментария |

#### Создание нового комментария

POST /comments

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | POST |
| **Ресурс** | /comments |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: «application/json») |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 201. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/comments/[ id]", где [id] – присвоенный идентификатор комментария |

**Описание полей тела запроса**:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| **comment** | object | представляет собой набор данных комментария |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **comment** | object | представляет собой набор данных комментария |

**Пример:**

**Тело запроса**:

{

    "comment": {

        "idUser": 504,

"idRecipe": 798,

        "parentComment": 1205,

        "timeCreated": "09/06/2021 12:30:00",

        "text": "Не кладите туда лимон! Испортите только. Прикладываю свой результат в профиль и анфас :)",

        "pictureList": [

            "https://clck.ru/VLjEN",

            "https://clck.ru/VPKiM"

        ]

    }

}

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "comment": {

        "idComment": 1403,

"idRecipe": 798,

        "idUser": 504,

        "parentComment": 1205,

        "timeCreated": "09/06/2021 12:30:00",

        "text": "Не кладите туда лимон! Испортите только. Прикладываю свой результат в профиль и анфас :)",

        "pictureList": [

            "https://clck.ru/VLjEN",

            "https://clck.ru/VPKiM"

        ]

    }

}

#### Изменение данных комментария

PATCH /comments/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | POST |
| **Ресурс** | /comments/[id], где [id] – идентификатор комментария |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: "application/json") |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/comments/[id]", где [id] – присвоенный идентификатор комментария |

**Описание полей тела запроса**:

В теле запроса записываются только те поля, значения которых нужно изменить.

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **comment** | object | представляет собой набор данных комментария |

**Пример**:

**Тело запроса**:

{

    "comment": {

        "pictureList": [

            "https://clck.ru/VLjEN",

            "https://clck.ru/VPKiM",

**"https://clck.ru/PiVmK"**

        ]

    }

}

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "comment": {

        "idComment": 1403,

"idRecipe": 798,

        "idUser": 504,

        "parentComment": 1205,

        "timeCreated": "09/06/2021 12:30:00",

        "text": "Не кладите туда лимон! Испортите только. Прикладываю свой результат в профиль и анфас :)",

        "pictureList": [

            "https://clck.ru/VLjEN",

            "https://clck.ru/VPKiM",

**"https://clck.ru/PiVmK"**

        ]

    }

}

#### Получение данных комментария

GET /comments/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /comments/[id], где [id] – идентификатор комментария |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **comment** | object | представляет собой набор данных комментария |

**Пример**

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "comment": {

        "idComment": 1403,

"idRecipe": 798,

        "idUser": 504,

        "parentComment": 1205,

        "timeCreated": "09/06/2021 12:30:00",

        "text": "Не кладите туда лимон! Испортите только. Прикладываю свой результат в профиль и анфас :)",

        "pictureList": [

            "https://clck.ru/VLjEN",

            "https://clck.ru/VPKiM"

        ]

    }

}

#### Получение списка комментариев к рецепту с указанными параметрами

GET /comments

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /comments |
| **Параметры запроса** | *psize=n* – количество записей, возвращаемых  в запросе, в режиме постраничной выдачи;  формат: целое положительное число;  опциональный, при отсутствии выдаются все записи за один запрос. |
| *offset=m* – элемент, с которого начинается  постраничная выдача;  формат: целое положительное число;  опциональный, работает только в паре с psize, при отсутствии выдача начинается с первого элемента. |
| *recId=NN* – идентификатор рецепта блюда;  формат: целое положительное число;  обязательный. |
| *parentId=XX* – идентификатор родительского  комментария, для корневых комментариев = 0;  формат: целое положительное число;  опциональный. |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

***Примечание***: возвращаемый список отсортирован по возрастанию даты создания комментария (поле *timeCreated* объекта**comment**).

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| RecNum | int | число записей в массиве Results; формат: целое число. |
| Results | array[**comment**] | список комментариев; массив объектов, каждый из которых представляет собой набор данных комментария (объект типа *comment*). |

**Пример**

GET /comments?recId=798?&parentId=0 – возврат всех корневых комментариев к рецепту с идентификатором 798.

GET /comments?recId=798 – возврат всех комментариев к рецепту с идентификатором 798

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "RecNum": 1,

    "Results": [

        {

            "idComment": 1403,

            "idRecipe": 798,

            "idUser": 504,

            "parentComment": 1205,

            "timeCreated": "09/06/2021 12:30:00",

            "text": "Не кладите туда лимон! Испортите только. Прикладываю свой результат в профиль и анфас :)",

            "pictureList": [

                "https://clck.ru/VLjEN",

                "https://clck.ru/VPKiM"

            ]

        }

    ]

}

#### Удаление комментария

DELETE /comments/[id]

*Примечание: комментарий удаляется вместе со всем своим поддеревом комментариев.*

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | DELETE |
| **Ресурс** | /comments/[id], где [id] – идентификатор комментария |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** |  |
| **Заголовки ответа** |  |

### Ресурс кухни страны

URL: /countries

**Объект** **country:**

***Примечание****: поля объекта country, выделенные курсивом, в тело запроса на создание не включаются.*

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| *idCountry* | *int* | *идентификатор страны* |
| name | string | название страны |
| description | string | html-страница с описанием кухни страны |
| flag | string | ссылка на картинку с флагом страны |

#### Создание новой кухни

POST /countries

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | POST |
| **Ресурс** | /countries |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: «application/json») |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 201. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/countries /[id]", где [id] – присвоенный идентификатор кухни страны |

**Описание полей тела запроса**:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| **country** | object | представляет собой набор данных кухни страны |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **country** | object | представляет собой набор данных кухни страны |

**Пример:**

**Тело запроса**:

{

    "country": {

        "name": "Греция",

        "description": "https://clck.ru/9LEJs",

        "flag": "https://clck.ru/VLjEN"

    }

}

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "country": {

**"idCountry": 32,**

        "name": "Греция",

        "description": "https://clck.ru/9LEJs",

        "flag": "https://clck.ru/VLjEN"

    }

}

#### Изменение данных кухни страны

PATCH/countries/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | POST |
| **Ресурс** | /countries/[id], где [id] – идентификатор кухни страны |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: "application/json") |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/countries/[id]", где [id] – присвоенный идентификатор кухни страны |

**Описание полей тела запроса**:

В теле запроса записываются только те поля, значения которых нужно изменить.

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **country** | object | представляет собой набор данных кухни страны |

**Пример**:

**Тело запроса**:

{

    "country": {

        "description": "https://clck.ru/VLtvZ"

    }

}

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "country": {

        "idCountry": 32,

        "name": "Греция",

**"description": "https://clck.ru/VLtvZ",**

        "flag": "https://clck.ru/VLjEN"

    }

}

#### Получение данных кухни страны

GET /countries/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /countries/[id], где [id] – идентификатор кухни страны |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **country** | object | представляет собой набор данных кухни страны |

**Пример**

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "country": {

        "idCountry": 32,

        "name": "Греция",

        "description": "https://clck.ru/9LEJs",

        "flag": "https://clck.ru/VLjEN"

    }

}

#### Получение списка кухонь стран с указанными параметрами

GET /countries

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /countries |
| **Параметры запроса** | *psize=n* – количество записей, возвращаемых  в запросе, в режиме постраничной выдачи;  формат: целое положительное число;  опциональный, при отсутствии выдаются все записи за один запрос. |
| *offset=m* – элемент, с которого начинается  постраничная выдача;  формат: целое положительное число;  опциональный, работает только в паре с psize, при отсутствии выдача начинается с первого элемента. |
| *sortBy={name, id} –* сортировка по названию в  алфавитном порядке (*name)* или по идентификатору *(id);*  опциональный. |
| *startWith="X"* – название начинается с *X,*  от регистра не зависит;  формат: одиночный символ;  опциональный. |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| RecNum | int | число записей в массиве Results; формат: целое число. |
| Results | array[**country**] | список стран; массив объектов, каждый из которых представляет собой набор данных кухни страны (объект типа *country*). |

**Пример**

GET /countries?sortBy=name&startWith=Г – возврат отсортированного по алфавиту списка стран, с названием, начинающимся на букву ‘Г’

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "RecNum": 3,

    "Results": [

        {

            "idCountry": 61,

            "name": "Голландия",

            "description": "https://clck.ru/0E3Ds",

            "flag": "https://clck.ru/R2GEM"

        },

        {

            "idCountry": 32,

            "name": "Греция",

            "description": "https://clck.ru/9LEJs",

         "flag": "https://clck.ru/VLjEN"

        },

        {

            "idCountry": 38,

            "name": "Грузия",

            "description": "https://clck.ru/89YJs",

            "flag": "https://clck.ru/mLjEp"

        }

    ]

}

#### Удаление кухни страны

*В данный момент не используется, но перечислено на всякий случай.*

DELETE /countries/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | DELETE |
| **Ресурс** | /countries/[id], где [id] – идентификатор кухни страны |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** |  |
| **Заголовки ответа** |  |

### Ресурс точки питания

URL: /restos

**Объект** **resto:**

***Примечание****: поля объекта resto, выделенные курсивом, в тело запроса на создание не включаются.*

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| *idResto* | *int* | *идентификатор* точки питания |
| name | string | название точки питания |
| address | string | адрес точки питания |
| url | string | адрес сайта точки питания |
| rate | int | рейтинг точки питания |

#### Создание новой точки питания

POST /restos

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | POST |
| **Ресурс** | /restos |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: «application/json») |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 201. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/restos/[ id]", где [id] – присвоенный идентификатор точки питания |

**Описание полей тела запроса**:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| **resto** | object | представляет собой набор данных точки питания |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **resto** | object | представляет собой набор данных точки питания |

**Пример:**

**Тело запроса**:

{

    "resto": {

        "name": "Forest Lounge",

        "address": "Ленинградское ш., 51, стр. 4",

        "site": "https://forest-lounge.ru/",

        "rate": 55

    }

}

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "resto": {

**"idResto": 6**,

        "name": "Forest Lounge",

        "address": "Ленинградское ш., 51, стр. 4",

        "site": "https://forest-lounge.ru/",

        "rate": 55

    }

}

#### Изменение данных точки питания

PATCH/restos/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | POST |
| **Ресурс** | /restos/[id], где [id] – идентификатор точки питания |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: "application/json") |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/restos/[id]", где [id] – присвоенный идентификатор точки питания |

**Описание полей тела запроса**:

В теле запроса записываются только те поля, значения которых нужно изменить.

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **resto** | object | представляет собой набор данных точки питания |

**Пример**:

**Тело запроса**:

{

    "resto": {

        "rate": **106**

    }

}

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "resto": {

        "idResto": 6,

        "name": "Forest Lounge",

        "address": "Ленинградское ш., 51, стр. 4",

        "site": "https://forest-lounge.ru/",

**"rate": 106**

    }

}

#### Получение данных точки питания

GET /restos/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /restos/[id], где [id] – идентификатор точки питания |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **resto** | object | представляет собой набор данных точки питания |

**Пример**

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "resto": {

        "idResto": 6,

        "name": "Forest Lounge",

        "address": "Ленинградское ш., 51, стр. 4",

        "site": "https://forest-lounge.ru/",

        "rate": 55

    }

}

#### Получение списка точек питания с указанными параметрами

GET /restos

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /restos |
| **Параметры запроса** | *psize=n* – количество записей, возвращаемых  в запросе, в режиме постраничной выдачи;  формат: целое положительное число;  опциональный, при отсутствии выдаются все записи за один запрос. |
| *offset=m* – элемент, с которого начинается  постраничная выдача;  формат: целое положительное число;  опциональный, работает только в паре с psize, при отсутствии выдача начинается с первого элемента. |
| *sortBy={name, rate} –* сортировка по  названию в алфавитном порядке (*name)* или по убыванию рейтинга *(rate);*  опциональный. |
| *startWith="X"* – название начинается с ‘*X’,* от  регистра не зависит;  формат: одиночный символ;  опциональный. |
| *recipeID=N* – возврат всех точек питания, в  которых подают блюдо по рецепту с идентификатором *N*  формат: целое положительное число;  опциональный. |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| RecNum | int | число записей в массиве Results; формат: целое число. |
| Results | array[**resto**] | список точек питания; массив объектов, каждый из которых представляет собой набор данных точки питания (объект типа ***resto***). |

**Пример**

GET /restos?sortBy=name&recipeID=16 – возврат списка точек питания, упорядоченный по названию в алфавитном порядке, в которых подают блюдо по рецепту с идентификатором 16

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "RecNum": 5,

    "Results": [

        {

            "idResto": 8,

            "name": "Björn",

            "address": "Пятницкая, 3/4, стр. 1",

            "site": "http://bjorn.rest/",

            "rate": 23

        },

        {

            "idResto": 26,

            "name": "Hygge",

            "address": "Кронштадтский б-р, 6, корп. 5",

            "site": "http://hyggefood.ru/",

            "rate": 6

        },

        {

            "idResto": 40,

            "name": "Kaffebröd",

            "address": "Крымская наб., 10, корп. 1",

            "site": "https://www.facebook.com/kaffebrod",

            "rate": 19

        },

        {

            "idResto": 15,

            "name": "Møs",

            "address": "Трубецкая, 10",

            "site": "http://mosnordic.ru/",

            "rate": 8

        },

        {

            "idResto": 87,

            "name": "Стокгольм",

            "address": "Шлюзовая наб., 6, стр. 1, гостиница «Катерина-сити»",

            "site": "http://www.katerinahotels.com/",

            "rate": 2

        }

    ]

}

#### Удаление точки питания

*В данный момент не используется, но перечислено на всякий случай.*

DELETE /restos/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | DELETE |
| **Ресурс** | /restos/[id], где [id] – идентификатор точки питания |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** |  |
| **Заголовки ответа** |  |

### Ресурс рецепта блюда

URL: /recipes

**Объект** **recipe:**

***Примечание****: поля объекта recipe, выделенные курсивом, в тело запроса на создание не включаются.*

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| *idRecipe* | *int* | *идентификатор рецепта блюда* |
| name | string | название рецепта блюда |
| picture | string | путь к основной иллюстрации рецепта блюда |
| descr | string | описание рецепта блюда |
| type | int | тип блюда – см.enumeration «Тип блюда» на диаграмме «Логическая модель данных» |
| idCountry | int | идентификатор кухни страны |
| rate | int | рейтинг рецепта блюда |
| kKal | float | энергетическая ценность блюда |
| time | float | время приготовления, мин |
| ingredients | array[string *ingredient*, string *quantity*] | список ингредиентов: название ингредиента (*ingredient*) и его количество (*quantity*) |
| steps | array[int *step*, string *picture*, string *stepDescr*] | список шагов по приготовлению блюда - массив объектов, каждый из которых представляет собой набор: номер шага рецепта (*step*), путь к иллюстрации шага (*picture)* и его описание (*stepDescr)*. |
| restoList | array[int idResto] | список точек питания (*см. поле idResto объекта* **resto**), в которых подают это блюдо |

**Объект** **recipeConcise** (краткий набор данных рецепта)**:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| *idRecipe* | *int* | *идентификатор рецепта блюда* |
| name | string | название рецепта блюда |
| picture | string | путь к основной иллюстрации рецепта блюда |
| descr | string | описание рецепта блюда |
| rate | int | рейтинг рецепта блюда |

#### Создание нового рецепта блюда

POST /recipes

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | POST |
| **Ресурс** | /recipes |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: «application/json») |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 201. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/recipes/[ id]", где [id] – присвоенный идентификатор рецепта блюда |

**Описание полей тела запроса**:

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| recipe | object | представляет собой набор данных рецепта блюда |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| recipe | object | представляет собой набор данных рецепта блюда |

**Пример:**

**Тело запроса**:

{

    "recipe": {

        "name": "Норвежский омлет",

        "descr": "Рецепт сытного и очень вкусного завтрака из омлета с зеленью и соленым лососем.",

        "type": 0,

        "idCountry": 55,

"picture": "https://clck.ru/VNPcn",

        "rate": 0,

        "kKal": 210,

        "time": 10,

        "ingredients": [

            {

                "ingredient": "Яйцо куриное",

                "quantity": "3-4 шт."

            },

            {

                "ingredient": "Лосось соленый",

                "quantity": "50 г"

            },

            {

                "ingredient": "Сыр твердый",

                "quantity": "30 г"

            },

            {

                "ingredient": "Лук зеленый",

                "quantity": "1 шт."

            },

            {

                "ingredient": "Петрушка",

                "quantity": "пару веточек"

            },

            {

                "ingredient": "Масло постное",

                "quantity": "1 ст.л."

            },

            {

                "ingredient": "Cпеции и приправы",

                "quantity": "по вкусу"

            }

        ],

        "steps": [

            {

                "step": 1,

                "picture": "https://clck.ru/VNPDb",

                "stepDescr": "Подготовьте все необходимые продукты для норвежского омлета. Зелень помойте, обсушите, натрите сыр на крупной терке."

            },

            {

                "step": 2,

                "picture": "https://clck.ru/VNPRK",

                "stepDescr": "Взбейте в глубокой миске яйца."

            },

            {

                "step": 3,

                "picture": "https://clck.ru/VNPTM",

                "stepDescr": "Разогрейте в сковороде немного постного масла, вылейте яичную массу. Жарьте омлет классическим способом, отодвигая готовый лопаткой к серединке. Когда только верхушка останется чуть жидковатой, посыпьте половину натертым сыром."

            },

            {

                "step": 4,

                "picture": "https://clck.ru/VNPV7",

                "stepDescr": "Сверху сыра выложите нарезанный зеленый лук (только зеленую часть) и свежую зелень."

            },

            {

                "step": 5,

                "picture": "https://clck.ru/VNPXF",

                "stepDescr": "Сверху выложите тонко нарезанные ломтики лосося, поперчите. Можно добавить немного морской соли."

            },

            {

                "step": 6,

                "picture": "https://clck.ru/VNPbK",

                "stepDescr": "Накройте омлет свободной половиной с помощью лопатки и подержите под крышкой на медленном огне пару минут, чтобы сыр расплавился."

            },

            {

                "step": 7,

                "picture": "https://clck.ru/VNPcn",

                "stepDescr": "Готовый норвежский омлет подавайте сразу же после приготовления."

            }

        ],

        "restoList": [

            26,

            15

        ]

    }

}

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "recipe": {

**"idRecipe": 710**,

        "name": "Норвежский омлет",

        "descr": "Рецепт сытного и очень вкусного завтрака из омлета с зеленью и соленым лососем.",

        "type": 0,

        "idCountry": 5,

        "rate": 0,

        "kKal": 210,

        "time": 10,

        "ingredients": [

            {

                "ingredient": "Яйцо куриное",

                "quantity": "3-4 шт."

            },

            {

                "ingredient": "Лосось соленый",

                "quantity": "50 г"

            },

            {

                "ingredient": "Сыр твердый",

                "quantity": "30 г"

            },

            {

                "ingredient": "Лук зеленый",

                "quantity": "1 шт."

            },

            {

                "ingredient": "Петрушка",

                "quantity": "пару веточек"

            },

            {

                "ingredient": "Масло постное",

                "quantity": "1 ст.л."

            },

            {

                "ingredient": "Cпеции и приправы",

                "quantity": "по вкусу"

            }

        ],

        "steps": [

            {

                "step": 1,

                "picture": "https://clck.ru/VNPDb",

                "stepDescr": "Подготовьте все необходимые продукты для норвежского омлета. Зелень помойте, обсушите, натрите сыр на крупной терке."

            },

            {

                "step": 2,

                "picture": "https://clck.ru/VNPRK",

                "stepDescr": "Взбейте в глубокой миске яйца."

            },

            {

                "step": 3,

                "picture": "https://clck.ru/VNPTM",

                "stepDescr": "Разогрейте в сковороде немного постного масла, вылейте яичную массу. Жарьте омлет классическим способом, отодвигая готовый лопаткой к серединке. Когда только верхушка останется чуть жидковатой, посыпьте половину натертым сыром."

            },

            {

                "step": 4,

                "picture": "https://clck.ru/VNPV7",

                "stepDescr": "Сверху сыра выложите нарезанный зеленый лук (только зеленую часть) и свежую зелень."

            },

            {

                "step": 5,

                "picture": "https://clck.ru/VNPXF",

                "stepDescr": "Сверху выложите тонко нарезанные ломтики лосося, поперчите. Можно добавить немного морской соли."

            },

            {

                "step": 6,

                "picture": "https://clck.ru/VNPbK",

                "stepDescr": "Накройте омлет свободной половиной с помощью лопатки и подержите под крышкой на медленном огне пару минут, чтобы сыр расплавился."

            },

            {

                "step": 7,

                "picture": "https://clck.ru/VNPcn",

                "stepDescr": "Готовый норвежский омлет подавайте сразу же после приготовления."

            }

        ],

        "restoList": [

            26,

            15

        ]

    }

}

#### Изменение данных рецепта блюда

PATCH/recipes/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | POST |
| **Ресурс** | /recipes/[id], где [id] – идентификатор точки питания |
| **Параметры запроса** |  |
| **Формат тела запроса** | json (хидер Content-Type: "application/json") |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** | добавляется Location: "https://rnr/recipes/[id]", где [id] – присвоенный идентификатор точки питания |

**Описание полей тела запроса**:

В теле запроса записываются только те поля, значения которых нужно изменить.

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **recipe** | object | представляет собой набор данных точки питания |

**Пример**:

**Тело запроса**:

{

    "recipe": {

        "rate": 13,

        "restoList": [

            26,

            15,

            87

        ]

    }

}

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "recipe": {

        "idRecipe": 710,

        "name": "Норвежский омлет",

        "descr": "Рецепт сытного и очень вкусного завтрака из омлета с зеленью и соленым лососем.",

        "type": 0,

        "idCountry": 5,

"picture": "https://clck.ru/VNPcn",

**"rate": 13**,

        "kKal": 210,

        "time": 10,

        "ingredients": [

            {

                "ingredient": "Яйцо куриное",

                "quantity": "3-4 шт."

            },

            {

                "ingredient": "Лосось соленый",

                "quantity": "50 г"

            },

            {

                "ingredient": "Сыр твердый",

                "quantity": "30 г"

            },

            {

                "ingredient": "Лук зеленый",

                "quantity": "1 шт."

            },

            {

                "ingredient": "Петрушка",

                "quantity": "пару веточек"

            },

            {

                "ingredient": "Масло постное",

                "quantity": "1 ст.л."

            },

            {

                "ingredient": "Cпеции и приправы",

                "quantity": "по вкусу"

            }

        ],

        "steps": [

            {

                "step": 1,

                "picture": "https://clck.ru/VNPDb",

                "stepDescr": "Подготовьте все необходимые продукты для норвежского омлета. Зелень помойте, обсушите, натрите сыр на крупной терке."

            },

            {

                "step": 2,

                "picture": "https://clck.ru/VNPRK",

                "stepDescr": "Взбейте в глубокой миске яйца."

            },

            {

                "step": 3,

                "picture": "https://clck.ru/VNPTM",

                "stepDescr": "Разогрейте в сковороде немного постного масла, вылейте яичную массу. Жарьте омлет классическим способом, отодвигая готовый лопаткой к серединке. Когда только верхушка останется чуть жидковатой, посыпьте половину натертым сыром."

            },

            {

                "step": 4,

                "picture": "https://clck.ru/VNPV7",

                "stepDescr": "Сверху сыра выложите нарезанный зеленый лук (только зеленую часть) и свежую зелень."

            },

            {

                "step": 5,

                "picture": "https://clck.ru/VNPXF",

                "stepDescr": "Сверху выложите тонко нарезанные ломтики лосося, поперчите. Можно добавить немного морской соли."

            },

            {

                "step": 6,

                "picture": "https://clck.ru/VNPbK",

                "stepDescr": "Накройте омлет свободной половиной с помощью лопатки и подержите под крышкой на медленном огне пару минут, чтобы сыр расплавился."

            },

            {

                "step": 7,

                "picture": "https://clck.ru/VNPcn",

                "stepDescr": "Готовый норвежский омлет подавайте сразу же после приготовления."

            }

        ],

        "restoList": [

**26,**

**15,**

**87**

        ]

    }

}

#### Получение данных рецепта блюда

GET /recipes/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /recipes/[id], где [id] – идентификатор рецепта блюда |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| **recipe** | object | представляет собой набор данных рецепта блюда |

**Пример**

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "recipe": {

        "idRecipe": 710,

        "name": "Норвежский омлет",

        "descr": "Рецепт сытного и очень вкусного завтрака из омлета с зеленью и соленым лососем.",

        "type": 0,

        "idCountry": 5,

"picture": "https://clck.ru/VNPcn",

        "rate": 13,

        "kKal": 210,

        "time": 10,

        "ingredients": [

            {

                "ingredient": "Яйцо куриное",

                "quantity": "3-4 шт."

            },

            {

                "ingredient": "Лосось соленый",

                "quantity": "50 г"

            },

            {

                "ingredient": "Сыр твердый",

                "quantity": "30 г"

            },

            {

                "ingredient": "Лук зеленый",

                "quantity": "1 шт."

            },

            {

                "ingredient": "Петрушка",

                "quantity": "пару веточек"

            },

            {

                "ingredient": "Масло постное",

                "quantity": "1 ст.л."

            },

            {

                "ingredient": "Cпеции и приправы",

                "quantity": "по вкусу"

            }

        ],

        "steps": [

            {

                "step": 1,

                "picture": "https://clck.ru/VNPDb",

                "stepDescr": "Подготовьте все необходимые продукты для норвежского омлета. Зелень помойте, обсушите, натрите сыр на крупной терке."

            },

            {

                "step": 2,

                "picture": "https://clck.ru/VNPRK",

                "stepDescr": "Взбейте в глубокой миске яйца."

            },

            {

                "step": 3,

                "picture": "https://clck.ru/VNPTM",

                "stepDescr": "Разогрейте в сковороде немного постного масла, вылейте яичную массу. Жарьте омлет классическим способом, отодвигая готовый лопаткой к серединке. Когда только верхушка останется чуть жидковатой, посыпьте половину натертым сыром."

            },

            {

                "step": 4,

                "picture": "https://clck.ru/VNPV7",

                "stepDescr": "Сверху сыра выложите нарезанный зеленый лук (только зеленую часть) и свежую зелень."

            },

            {

                "step": 5,

                "picture": "https://clck.ru/VNPXF",

                "stepDescr": "Сверху выложите тонко нарезанные ломтики лосося, поперчите. Можно добавить немного морской соли."

            },

            {

                "step": 6,

                "picture": "https://clck.ru/VNPbK",

                "stepDescr": "Накройте омлет свободной половиной с помощью лопатки и подержите под крышкой на медленном огне пару минут, чтобы сыр расплавился."

            },

            {

                "step": 7,

                "picture": "https://clck.ru/VNPcn",

                "stepDescr": "Готовый норвежский омлет подавайте сразу же после приготовления."

            }

        ],

        "restoList": [

            26,

            15,

            87

        ]

    }

}

#### Получение списка рецептов с указанными параметрами

GET /recipes

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | GET |
| **Ресурс** | /recipes |
| **Параметры запроса** | *psize=n* – количество записей, возвращаемых  в запросе, в режиме постраничной выдачи;  формат: целое положительное число;  опциональный, при отсутствии выдаются все записи за один запрос. |
| *offset=m* – элемент, с которого начинается  постраничная выдача;  формат: целое положительное число;  опциональный, работает только в паре с psize, при отсутствии выдача начинается с первого элемента. |
| *sortBy={name, rate, type} –* сортировка по  названию в алфавитном порядке (*name),* по убыванию рейтинга *(rate)* или по категории блюда *(type);*  опциональный. |
| *country=x* – рецепт принадлежит кухне с  идентификатором *x*;  формат: целое положительное число;  опциональный. |
| *type=N* – возврат всех рецептов с типом  блюда *N*,– см.enumeration «Тип блюда» на диаграмме «Логическая модель данных»  формат: целое положительное число;  опциональный |
| *contain* – фильтр по составу блюд:  компоненты, хотя бы один из которых должен входить в состав;  формат: строка с разделителем, разделитель компонентов - “||”;  опциональный, при отсутствии возвращаются все блюда.  При одновременном включении какого-либо компонента в списки **contains** и **nocontains** компонент  игнорируется;  опциональный. |
| *nocontain* – фильтр по составу блюд:  компоненты, которые не должны входить в состав;  формат: строка с разделителем, разделитель - “||”;  опциональный, при отсутствии возвращаются все блюда.  При одновременном включении какого-либо компонента в списки **contains** и **nocontains** компонент игнорируется;  опциональный. |
| *out={full, concise} –* формат тела ответа;  обязательный. |
| **Формат тела запроса** |  |
| **Авторизация** |  |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** | json (хидер Content-Type: "application/json") |
| **Заголовки ответа** |  |

**Описание полей тела ответа:**

|  |  |  |
| --- | --- | --- |
| **Название** | **Тип** | **Описание** |
| Version | string | версия сервера в формате хх.хх.хх.хх |
| RetStatus | int | статус результата запроса |
| RecNum | int | число записей в массиве Results; формат: целое число. |
| Results | array[**recipeConcise**] | список точек питания; массив объектов, каждый из которых представляет собой краткий набор данных точки питания (объект типа **recipeConcise**). |

**Примеры запросов:**

GET /recipe?country=5&sortBy=type&out=concise

- возврат всех рецептов кухни с idCountry=5, упорядоченных по типу блюда

GET /recipe?country=5&type=0&sortBy=rate&out=full

- возврат всех рецептов завтраков (type=0) кухни с idCountry=5, упорядоченных по рейтингу

GET /recipe?contain=свекла||морковь&nocontain=перец&sortBy=name&out=concise

- возврат всех рецептов, упорядоченных по названию, в состав которых входят свекла и морковь, но не входит перец:

**Тело ответа**:

{

    "Version": "1.0.1.10",

    "RetStatus": "Success",

    "Results": [

        {

            "idRecipe": 4,

            "name": "Винегрет",

            "descr": "Винегрет - универсальный салат, который можно приготовить в любой день и даже на праздник.",

            "picture": "https://clck.ru/VPKsc",

            "rate": 54

        },

        {

            "idRecipe": 71,

            "name": "Салат \"Гранатовый браслет\"",

            "descr": "Красивый и вкусный слоеный праздничный салат из вареный свёклы, свиного языка и граната с грецкими орехами.",

            "picture": "https://clck.ru/VPKpw",

            "rate": 29

        },

        {

            "idRecipe": 45,

            "name": "Свекольник",

            "descr": " Этот холодный суп особенно хорош в жаркие дни, он освежает и придает бодрости всем от мала до велика.",

            "picture": "https://clck.ru/VPKoJ",

            "rate": 4

        },

        {

            "idRecipe": 88,

            "name": "Свекольный салат с изюмом",

            "descr": "Замена традиционной селедке под шубой, отличный вариант для новогоднего стола.",

            "picture": "https://clck.ru/VPKkD",

            "rate": -2

        },

        {

            "idRecipe": 39,

            "name": "Селедка под шубой",

            "descr": "Одним из самых главных блюд любого новогоднего праздничного стола является слоеный салат \"Селедка под шубой.\"",

            "picture": "https://clck.ru/VPKiM",

            "rate": 403

        }

    ]

}

#### Удаление рецепта блюда

DELETE /restos/[id]

|  |  |
| --- | --- |
| **Параметры** | |
| **Метод запроса** | DELETE |
| **Ресурс** | /recipes/[id], где [id] – идентификатор рецепта блюда |
| **Параметры запроса** |  |
| **Формат тела запроса** |  |
| **Авторизация** | Bearer token (например: authorization: "Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IjI4Y") |
| **Код возврата при успешном завершении вызова** | 200. В противном случае возвращается HTTP-код ошибки на сервере. |
| **Формат тела ответа** |  |
| **Заголовки ответа** |  |

## В качестве идеи для будущего развития:

1. Добавить ритейл: в профиле выбирается поставщик из списка предложенных, ингредиенты заинтересовавшего рецепта добавляются в корзину в минимально необходимом для готовки, но с учетом фасовки, количестве.
2. Добавить туротрасль: для указанной кухни страны выдается список гастротуров по данному направлению.

Плата за переходы + % от заказов => профит.