СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc183573730)

[1. ОБЩАЯ ЧАСТЬ 4](#_Toc183573731)

[1.1. Цель разработки 4](#_Toc183573732)

[2. СПЕЦИАЛЬНАЯ ЧАСТЬ 6](#_Toc183573733)

[2.1. Постановка задачи 6](#_Toc183573734)

[2.1.1. Входные данные предметной области 6](#_Toc183573735)

[2.1.2. Требования к проекту 6](#_Toc183573736)

[2.2. Внешняя спецификация 8](#_Toc183573737)

[2.2.1. Описание задачи 8](#_Toc183573738)

[2.2.2. Входные и выходные данные 14](#_Toc183573739)

[2.2.3. Методы 19](#_Toc183573740)

[2.2.4. Тесты 22](#_Toc183573741)

[2.2.5. Контроль целостности данных 23](#_Toc183573742)

[2.3. Проектирование 24](#_Toc183573743)

[2.3.1. Схема архитектуры приложения 24](#_Toc183573744)

[2.3.2. Логическая схема данных 25](#_Toc183573745)

[2.3.3. Физическая схема данных 26](#_Toc183573746)

[2.3.4. Структурная схема 27](#_Toc183573747)

[2.3.5. Функциональная схема 28](#_Toc183573748)

[2.3.6. Диаграмма классов 28](#_Toc183573749)

[2.3.7. Схема тестирования 29](#_Toc183573750)

[2.3.8. Схема пользовательского интерфейса 29](#_Toc183573751)

[2.4. Результат работы программы 31](#_Toc183573752)

[3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 35](#_Toc183573753)

[3.1. Инструментальные средства 35](#_Toc183573754)

[3.2. Отладка программы 36](#_Toc183573755)

[3.3. Защитное программирование 36](#_Toc183573756)

[3.4. Характеристики программы 37](#_Toc183573757)

[ЗАКЛЮЧЕНИЕ 38](#_Toc183573758)

[СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ 40](#_Toc183573759)

# ВВЕДЕНИЕ

Раньше кулинарные рецепты передавались из поколения в поколение устно или записывались в тетрадях. С развитием технологий появились кулинарные книги и журналы, но с приходом интернета появилась возможность делиться рецептами мгновенно и на глобальном уровне. Однако, несмотря на разнообразие доступных ресурсов, многие люди сталкиваются с трудностями в поиске качественных рецептов, их классификации и взаимодействии с другими кулинарами.

В связи с этим в данной курсовой работе разработано WEB-приложение «РецептГид», которое предоставляет функционал для размещения и поиска рецептов блюд. Приложение будет включать в себя удобный интерфейс для добавления рецептов, возможность поиска по времени и категориям, что позволит пользователям легко находить и делиться любимыми блюдами.

# ОБЩАЯ ЧАСТЬ

## Цель разработки

Целью разработки является предоставление возможности поиска и размещение рецептов для ускоренного процесса приготовления блюда и удобной системы хранения собственных рецептов.

* 1. Средства разработки

Программные средства, используемые для проектирования, разработки и тестирования WEB-ресурса представлены в таблице 1.

Таблица 1 – Программные средства

|  |  |  |  |
| --- | --- | --- | --- |
| № | Тип средства | Название средства | Назначение |
| 1 | Операционная система | Windows 10 | Организация взаимодействия  программ и пользователя |
| 2 | Среда разработки | PyCharm Professional | Среда запуска |
| 3 | Текстовый редактор | Microsoft Word 2016 | Разработка документации,  формирование отчетных документов по шаблонам. |
| 4 | Сервер базы данных | sqlite | Хранение данных. |
| 5 | Графическая оболочка | PyCharm Professional | Разработка базы данных и функционала работы с ней. |

В качестве средств вычислительной техники при разработке ПО использовался ноутбук HP Laptop 14s-fq0111ur. Характеристики представлены в Таблице 2.

Таблица 2 - Технические характеристики ноутбука и мобильного устройства при разработке ПО

|  |  |  |
| --- | --- | --- |
| № | Тип оборудования | Наименование оборудования |
| 1 | 2 | 3 |
| Для разработки | | |
| HP Laptop 14s-fq0111ur | | |
| 1 | Процессор: | AMD Ryzen 3 3250U |
| 2 | Оперативная память: | 16 GB |
| 4 | Тип видеокарты: | Встроенная |
| 5 | Видеокарта: | AMD Radeon Graphics |
| 6 | Конфигурация накопителей: | SSD |
| 7 | Общий объем накопителей: | 256 ГБ |
| 8 | Компьютерная мышь: | Нет |
| 9 | Клавиатура: | Любая рабочая клавиатура с английской и русской раскладкой |
| 10 | Операционная система | Windows 10 Home |

# СПЕЦИАЛЬНАЯ ЧАСТЬ

## Постановка задачи

Разработать WEB-приложение «РецептГид» которое представляет из себя многостраничный сайт для размещения рецептов и их просмотра, вместе с сопутствующими данными пользователей.

## Входные данные предметной области

Входными данными WEB-приложения «РецептГид» являются следующие данные:

* Аутентификационные данные пользователя (Пароль, логин);
* Регистрационные данные пользователя (Пароль, пароль, логин, описание, фото, дата рождения);
* Создание ингредиента (Название ингредиента, каллории на 100 грамм, белки, жиры, углеводы, граммовка(внутри рецепта));
* Создание этапов рецепта (Номер этапа, фото, описание);
* Создание рецепта (Название, заголовок, количество порций и т.п.);
* Создание тега
* Написание причин удаления

Выходные данные предметной области

* Рецепты (Название, заголовок, количество порций и т.п.);
* Этапы рецепта (Фото, описание, номер);
* Ингредиенты рецепта (Название, граммовка, БЖУ);
* Теги рецепта (Название);
* Пользователи (Ник, описание, фото, дата регистрации, дата рождения);

## Требования к проекту

* Многопользовательская поддержка – данные о пользователях и их ролях должны хранится в БД;
* В приложении должно быть три роли: Администратор, Модератор и Пользователь;
* Пользователь может изменять свой профиль, просматривать чужие профили, просматривать рецепты, создавать свои рецепты и добавлять новые ингредиенты и этапы к ним, а также устанавливать теги к рецепту, писать рецензии к рецептам, добавлять пользователей в друзья, отправлять личные сообщения, добавлять рецепты в избранное, выполнять поиск рецептов, используя фильтры;
* Модератор может делать то же самое, но дополнительно имея возможность удалять рецензии, принимать рецепты или удалять их;
* Администратор же не может делать все вышеперечисленное, а может лишь добавлять теги, удалять пользователей и изменять их роли;
* Приложение должно иметь простой и интуитивный интерфейс;
* Приложение должно предоставлять возможность размещать и просматривать рецепты;
* Вход в приложение должен будет начинаться с ввода логина и пароля от аккаунта. Требования для логина: не короче 6 символов, только латинские буквы. Требования для пароля: не короче 8 символов, хотя бы один спецсимвол, хотя бы одна цифра, без пробелов, только латинские буквы;
* Приложение должно позволять формировать рейтинг рецепта и пользователя путем написания рецензии
* Приложение должно предоставлять возможность для удобного поиска рецептов по названию, используя определенные фильтры;
* После написания рецепта пользователем, его должен проверить модератор, предварительно проверив состав и сам рецепт на его безопасность и соответствие правилам площадки;
* Приложение должно предоставлять возможность поэтапно готовить блюдо по рецепту, у каждого этапа должен быть свое текстовое описание и фото;
* Приложение должно предоставлять возможность экспорта и импорта данных в формате csv и sql;
* В приложении должна быть статистика по определенному параметру и соответствующая диаграмма;
* Пароль должен храниться в виде хэша формата sha256;
* База данных должна быть приведена в 3 нормальную форму.

## Внешняя спецификация

## Описание задачи

Проект представляет из себя многостраничное WEB-приложение на фреймворке Django и БД sqlite.

В приложении должно быть три основные роли: пользователь, администратор, модератор.

У пользователя и модератора одна общая часть приложения, за исключением дополнительных возможностей у модератора.

У админа же отдельная админ-панель, с необходимыми функциями, такими как экспорт/импорт БД, блокировка пользователей, вывод статистики.

На рисунке 1 приведена контекстная диаграмма для модели бизнес-процессов деятельности интернет-ресурса по размещению и поиску рецептов. Контекстная диаграмма представляет собой самое общее описание системы и ее взаимодействие внутри себя.

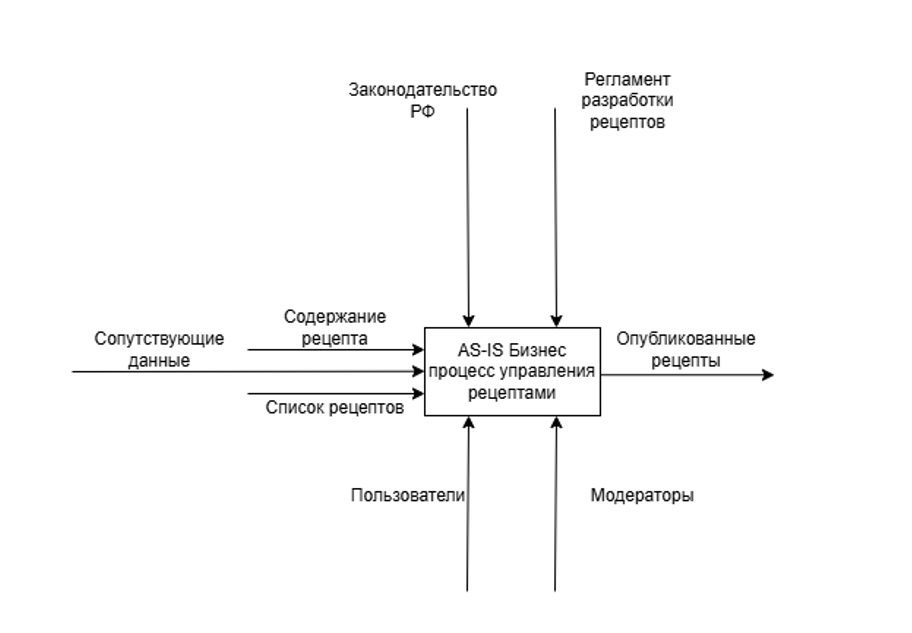


Рисунок 1 – Контекстная диаграмма

Для подробного рассмотрения деятельности магазина необходимо провести процесс декомпозиции контекстной диаграммы на рисунке 2.

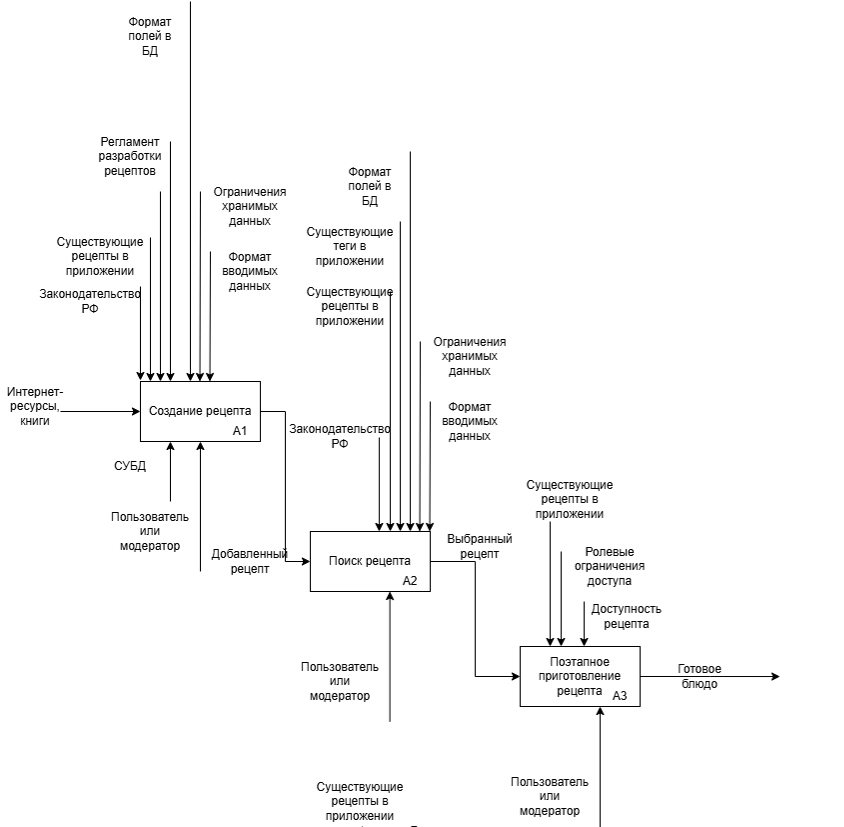


Рисунок 2 – Диаграмма декомпозиции

Ниже на рисунках 3-5 представлены модели IDEF0 для интернет-ресурса размещения рецептов.

Ниже указан бизнес-процесс «Создание рецепта» модели A-1. Выходом блока «Создание рецепта» является созданный рецепт в БД в процессе выполнения данного бизнес-процесса.

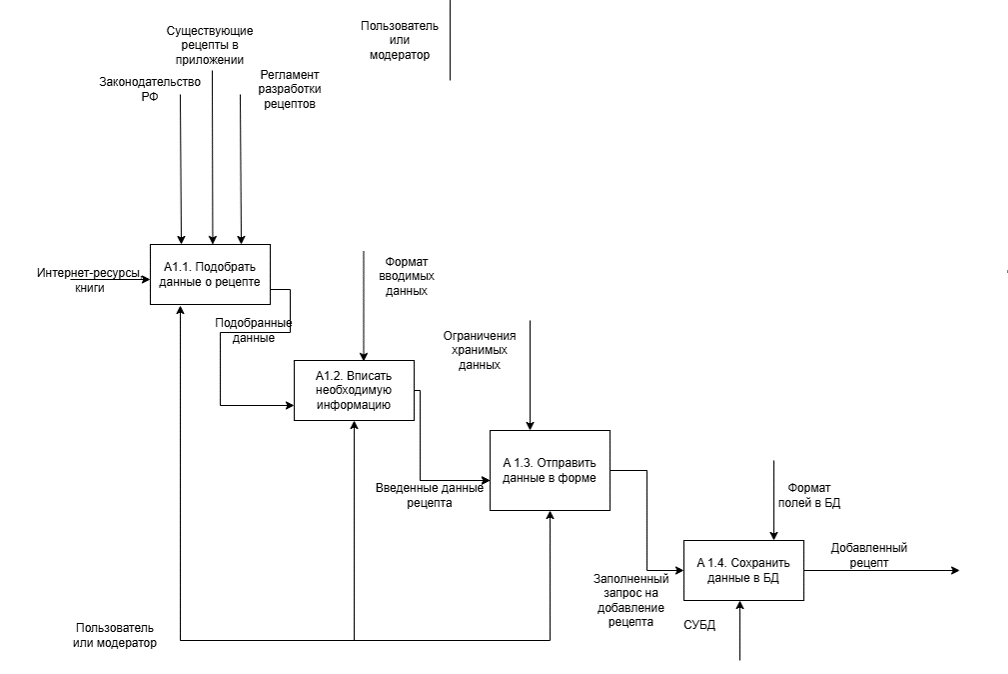


Рисунок 3 – Модель А1, создание рецепта

Ниже представлена модель А2, которая отвечает за поиск рецепта.

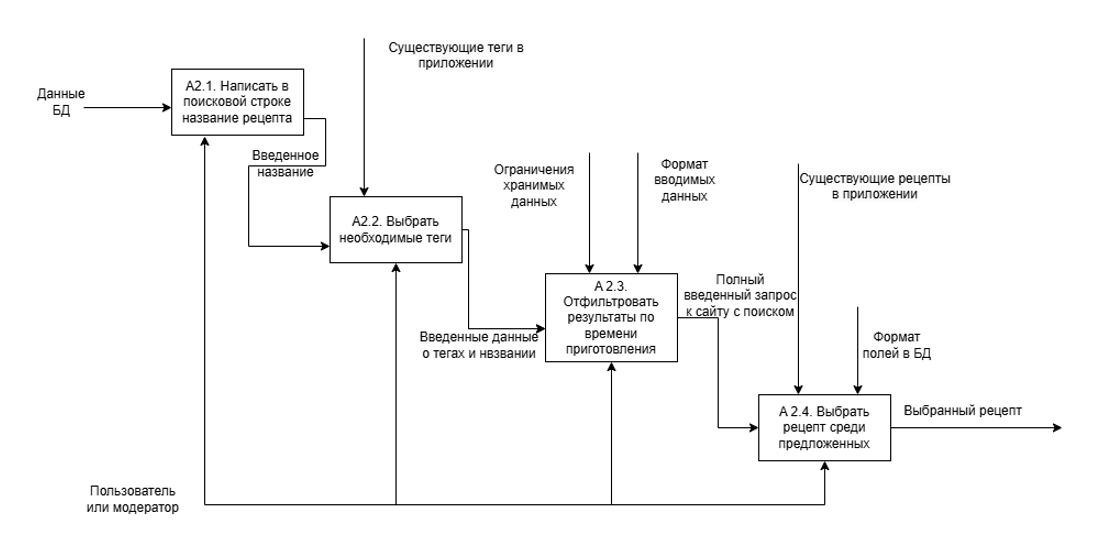


Рисунок 4 – Модель А2, поиск рецепта

Ниже представлена модель А3, которая отвечает за поэтапное приготовление рецепта.

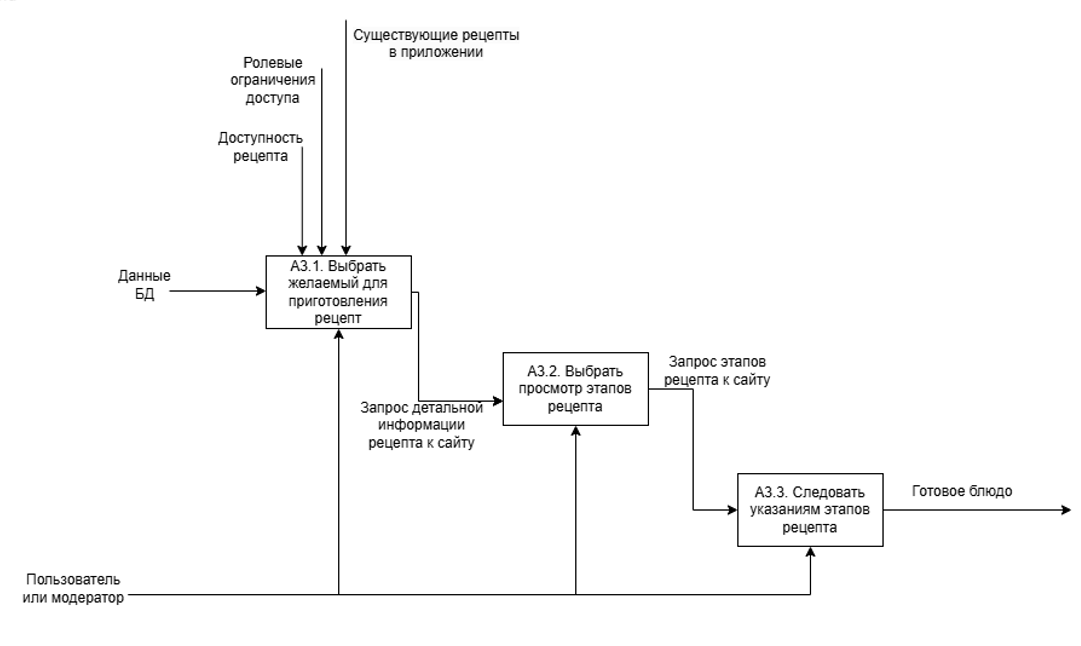


Рисунок 5 – Модель А3, поэтапное приготовление рецепта

На рисунке 6 представлена диаграмма прецедентов для WEB-приложения хранения и поиска рецептов.

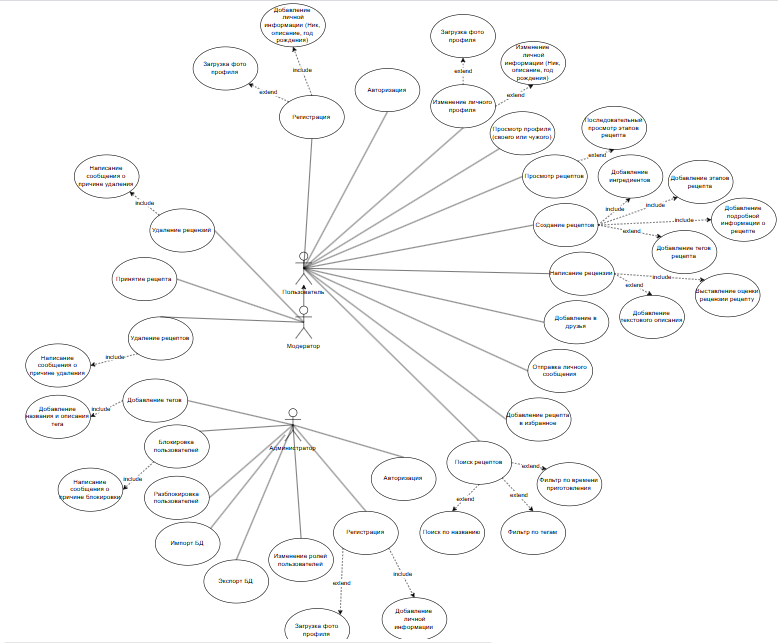


Рисунок 6 – Диаграмма прецедентов

На рисунке 7 представлен алгоритм авторизации внутри программы.

В алгоритме «Авторизация» пользователь открывает страницу авторизации, вводит данные для авторизации, далее данные передаются в логику сайта, где она через ORM сравнивается с имеющимися данными из БД, и возвращает результат: найден ли пользователь с таким логином и паролем или же нет. Если результат не удовлетворителен, то пользователя переносит обратно на этап ввода данных для авторизации. Если же пользователь найден, то для начала проверяется наличие у него причины удаления, что является признаком блокировки аккаунта. Если она есть, то пользователя возвращает на этап ввода данных для авторизации, если нет, то данные пользователя сохраняются в локальных статических переменных, создается запись сессии пользователя, в результате чего он авторизуется и переходит в основную часть приложения.

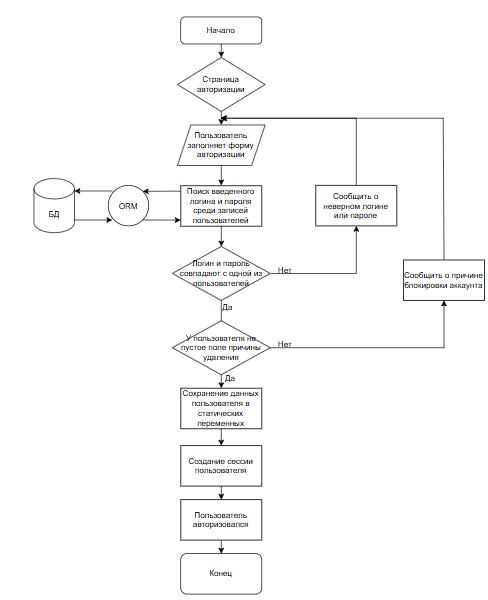


Рисунок 7 – Алгоритм авторизации

## Входные и выходные данные

На таблице 3 вы можете лицезреть входные данные WEB-приложения хранения и размещения рецептов.

Таблица 3 – Входные данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Ограничения | Обязательное | Структура, формат | Форма ввода |
| 1 | 2 | 3 | 4 | 5 |
| Входные данные | | | | |
| Логин | Максимальная длина строки 255 символов, не менее 6 символов | Да | Строка | Текстовое поле |
| Пароль | Максимальная длина строки 255 символов, не менее 8 символов, только латинские буквы, по крайней мере одна цифра, по крайней мере один спецсимвол | Да | Строка | Текстовое поле |
| Фотография пользователя | Формат медиа-файла | Да | Файл | Проводник |
| Описание пользователя | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Дата рождения пользователя | Формат даты | Да | Строка | Текстовое поле |
| Название блюда | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Заголовок рецепта | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Время приготовления |  | Да | Число | Текстовое поле |
| Кол-во порций |  | Да | Число | Текстовое поле |
| Ккал на 100г |  | Да | Число | Текстовое поле |
| Белка на 100г |  | Да | Число | Текстовое поле |
| Жира на 100г |  | Да | Число | Текстовое поле |
| Углеводов на 100г |  | Да | Число | Текстовое поле |
| Описание | Максимальная длина строки 1024 символов | Да | Строка | Текстовое поле |
| Фото блюда | Формат медиа-файла | Да | Файл | Проводник |
| Фото этапа рецепта | Формат медиа-файла | Да | Файл | Проводник |
| Номер этапа рецепта |  | Да | Число | Текстовое поле |
| Описание этапа рецепта | Максимальная длина строки 1024 символов | Да | Строка | Текстовое поле |
| Название тега | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Описание тега | Максимальная длина строки 1024 символов | Да | Строка | Текстовое поле |
| Название ингредиента | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Ккал на 100г |  | Да | Число | Текстовое поле |
| Белка на 100г |  | Да | Число | Текстовое поле |
| Жира на 100г |  | Да | Число | Текстовое поле |
| Углеводов на 100г |  | Да | Число | Текстовое поле |

На таблице ниже расположены выходные данные WEB-приложения.

Таблица 4 – Выходные данные

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Ограничения | Обязательное | Структура, формат | Форма ввода |
| 1 | 2 | 3 | 4 | 5 |
| Входные данные | | | | |
| Логин | Максимальная длина строки 255 символов, не менее 6 символов | Да | Строка | Текстовое поле |
| Пароль | Максимальная длина строки 255 символов, не менее 8 символов, только латинские буквы, по крайней мере одна цифра, по крайней мере один спецсимвол | Да | Строка | Текстовое поле |
| Фотография пользователя | Формат медиа-файла | Да | Файл | Проводник |
| Описание пользователя | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Дата рождения пользователя | Формат даты | Да | Строка | Текстовое поле |
| Название блюда | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Заголовок рецепта | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Время приготовления |  | Да | Число | Текстовое поле |
| Кол-во порций |  | Да | Число | Текстовое поле |
| Ккал на 100г |  | Да | Число | Текстовое поле |
| Белка на 100г |  | Да | Число | Текстовое поле |
| Жира на 100г |  | Да | Число | Текстовое поле |
| Углеводов на 100г |  | Да | Число | Текстовое поле |
| Описание | Максимальная длина строки 1024 символов | Да | Строка | Текстовое поле |
| Фото блюда | Формат медиа-файла | Да | Файл | Проводник |
| Фото этапа рецепта | Формат медиа-файла | Да | Файл | Проводник |
| Номер этапа рецепта |  | Да | Число | Текстовое поле |
| Описание этапа рецепта | Максимальная длина строки 1024 символов | Да | Строка | Текстовое поле |
| Название тега | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Описание тега | Максимальная длина строки 1024 символов | Да | Строка | Текстовое поле |
| Название ингредиента | Максимальная длина строки 255 символов | Да | Строка | Текстовое поле |
| Ккал на 100г |  | Да | Число | Текстовое поле |
| Белка на 100г |  | Да | Число | Текстовое поле |
| Жира на 100г |  | Да | Число | Текстовое поле |
| Углеводов на 100г |  | Да | Число | Текстовое поле |

## Методы

Возможность объектов различных классов реагировать на одни и те же запросы с использованием общего интерфейса. Это позволяет использовать один и тот же метод для разных объектов, что упрощает кодирование и повышает его гибкость.

Использована инкапсуляция, которая позволяет защитить данные от нежелательного доступа и изменений, а также облегчить поддержку кода и его повторное использование с помощью скрытии определённых данных программы от других данных.

В приложении использовался рефакторинг и оптимизация. Рефакторинг - процесс изменения кода приложения с целью улучшения его качества, устранения технического долга и облегчения дальнейшей поддержки кода. Рефакторинг направлен на внутреннюю структуру приложения.

Оптимизация— это процесс улучшения эффективности работы программы путем оптимизации её алгоритмов, структур данных и использования ресурсов. Цель оптимизации состоит в повышении скорости выполнения программы, сокращении использования оперативной памяти и ресурсов процессора.

На рисунке 8 представлен пример без оптимизации. Данный код нужен для обработки и загрузки данных из БД в формате sql для страхового резервного копирования. В данном коде для загрузки SQL файла имеются серьезные проблемы с безопасностью и оптимизацией. Во-первых, отсутствует авторизация и проверка ролей, что позволяет любому пользователю выполнять потенциально опасные операции с базой данных без надлежащих прав. Во-вторых, код не обрабатывает ошибки при выполнении SQL скрипта, что может привести к падению приложения или некорректному состоянию базы данных. Это также делает систему уязвимой к SQL инъекциям, так как использование метода executescript без предварительной обработки входных данных может позволить злоумышленнику выполнить произвольные команды SQL. Кроме того, управление соединениями с базой данных не оптимизировано: соединение открывается, но не закрывается в случае возникновения ошибки, что может привести к утечкам ресурсов и исчерпанию лимитов соединений. Также код записывает временный файл без проверки его размера или типа, что может создать дополнительные риски безопасности, включая возможность загрузки вредоносных файлов.

В оптимизированном коде, который обновляет количество товара в базе данных. Он сделан более безопасным за счет использования параметров в запросах, чтобы избежать ошибок. Код стал более безопасным и эффективным, с улучшенной обработкой ошибок, управлением соединениями и читаемостью. Это повышает производительность и облегчает поддержку, что значительно улучшает качество приложения.

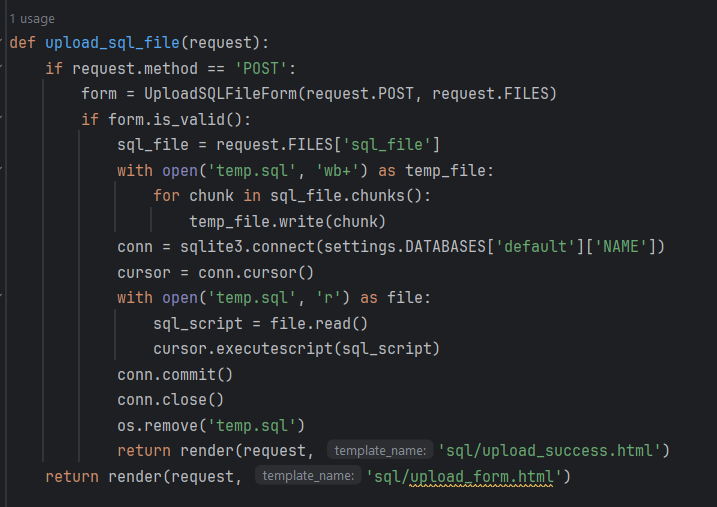


Рисунок 8 – Неоптимизированный код

На рисунке 9 представлен оптимизированный код функционала.

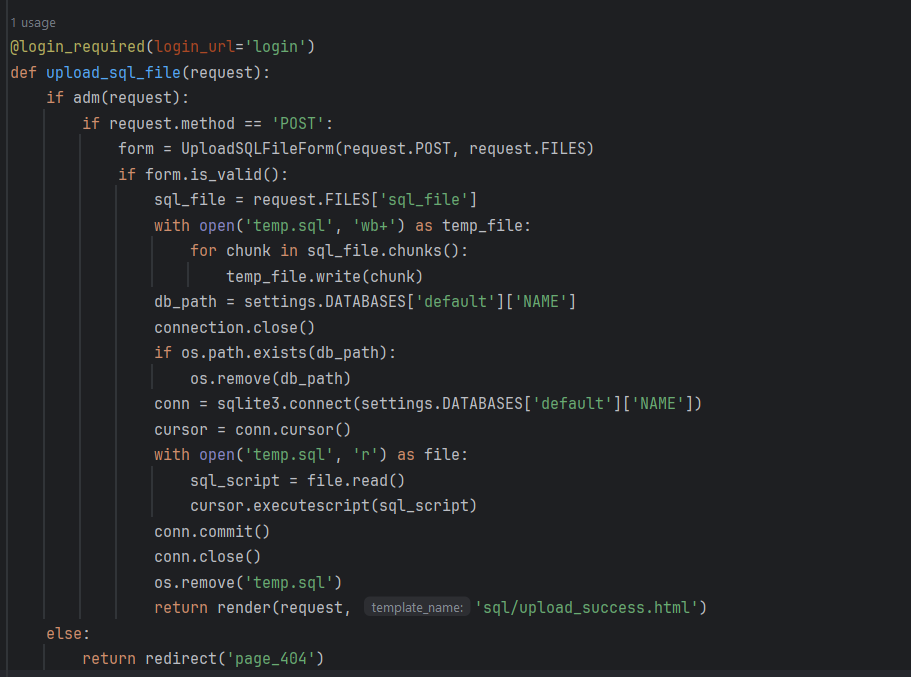


Рисунок 9 – Оптимизированный код

## Тесты

1. По формальности тестирования.

Тестирование по тестам направлено на проверку функций приложения на соответствие написанным тест-кейсам. Используется для того, чтобы с помощью некоторых шагов, убедиться в правильной работе программы.

2. По уровню тестирования.

Системное тестирование направлено на проверку работы всей системы на соответствие заявленным требованиям к программному продукту. Позволяет в разработанном приложении выявить ошибки, связанные с взаимодействием различных компонентов системы.

3. По целям.

Функциональное тестирование направлено на проверку реализации функций программного продукта и их точности, что позволяет убедиться в правильной работе разработанного приложения в соответствии с ожиданиями пользователя.

4. По степени автоматизации.

Ручное тестирование направлено на проверку программы без использования дополнительных программных средств, которая позволяет оценить работу разработанного приложения с точки зрения пользователя.

5. По позитивности сценария.

Позитивное тестирование направлено на проверку соответствия функций приложения ожидаемому поведению. Негативное тестирование направлено на проверку работы программы в случае, когда поведение пользователя отличается от ожидаемого, что позволяет в разработанном приложении

## Контроль целостности данных

Таблица 5 – Контроль целостности данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Ситуация | Аномалия | Реакция | Примечание |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Регистрация | Введён существующий логин и/или пароль и/или введены некорректные, согласно ограничениям, данные в соответствующие поля | Приложение обработает ошибку с базы данных и выведет пользователю диалоговое окно с содержанием  ошибки | Регистрация возможна только с валидными полями, соответствующим ограничениям полей |
| 2 | Добавление ингредиента | Не введено название ингредиента | Приложение возвращает фокус на пустое поле, данные не добавляются | Данные при добавлении рецепта не могут быть пустыми |
| Введены некорректные, согласно ограничениям, данные в соответствующие поля | Приложение обработает ошибку с базы данных и выведет пользователю диалоговое окно с содержанием  ошибки | Добавление товара возможно только с валидными полями, соответствующим ограничениям полей |
| 3 | Добавление рецепта | Не введено название рецепта | Приложение возвращает фокус на пустое поле, данные не добавляются | Данные при добавлении рецепта не могут быть пустыми |
| Введены некорректные, согласно ограничениям, данные в соответствующие поля | Приложение обработает ошибку с базы данных и выведет пользователю диалоговое окно с содержанием  ошибки | Добавление товара возможно только с валидными полями, соответствующим ограничениям полей |
| 4 | Изменение профиля пользователя | Вставлен файл, не подлежащий обработке по какой-либо технической причине | Приложение обработает ошибку с базы данных и выведет пользователю диалоговое окно с содержанием  ошибки | Изображения при изменении профиля должны иметь определенный подходящий тип и структуру, спокойно обрабатываться |

## Проектирование

## Схема архитектуры приложения

Компонент пользовательский интерфейс виден пользователю и взаимодействует с ним. В зависимости от действий пользователя осуществляются запросы на котором хранятся данные, который возвращает ответы на запрос в виде данных.

На Рисунке 10 представлена архитектурная схема приложения.

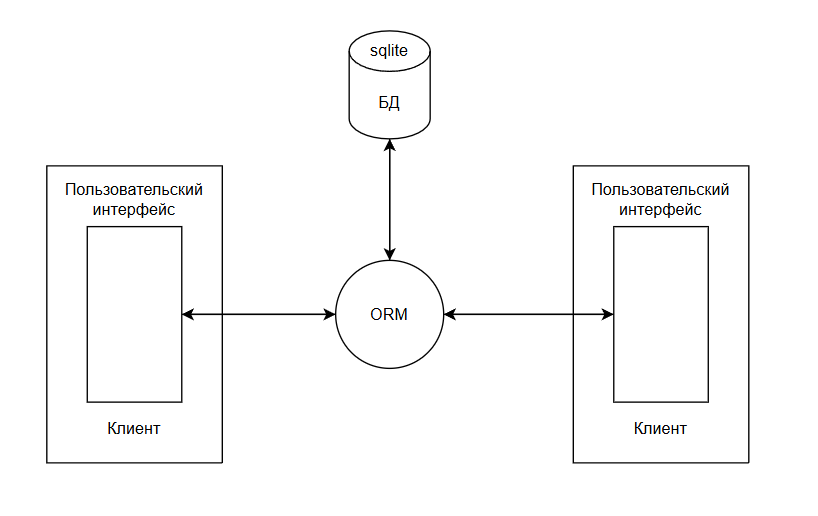


Рисунок 10 – Архитектурная схема приложения

## Логическая схема данных

Разработанная логическая модель базы данных, представленная на Рисунке 11, послужила основой для реализации логики манипуляции данными в проектируемой базе данных.

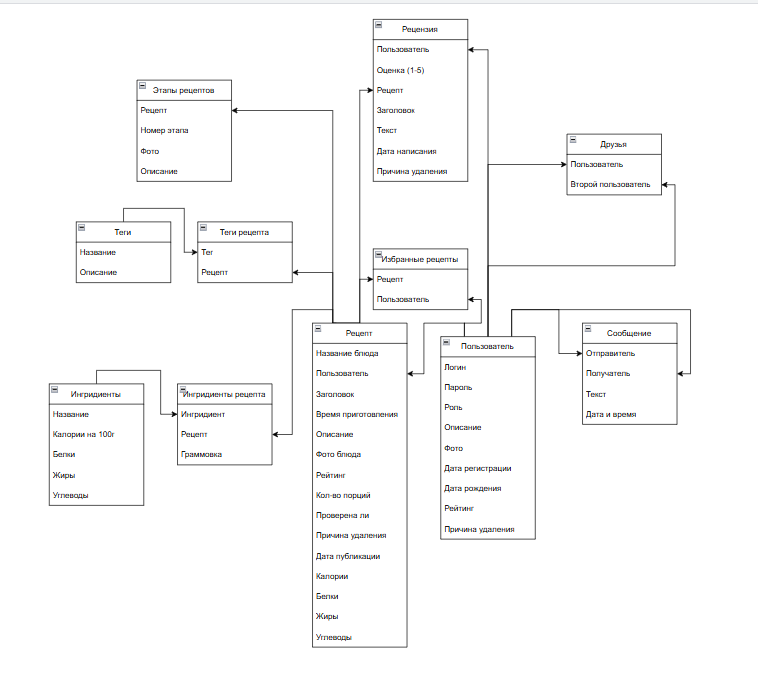


Рисунок 11 – Логическая схема данных

## Физическая схема данных

На Рисунке 12 представлена модель базы данных с фактическими названиями и типами данных в схематическом виде.

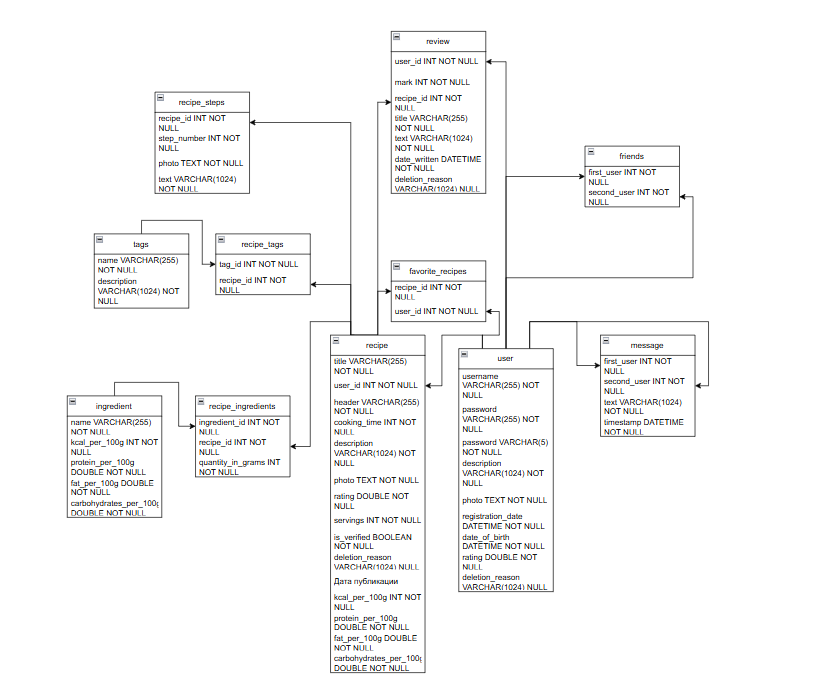


Рисунок 12 – Физическая схема данных

## Структурная схема

На Рисунке 13 представлен общий план структурной схемы, в которой графически показано взаимодействие всех компонентов и их переходы между друг другом.

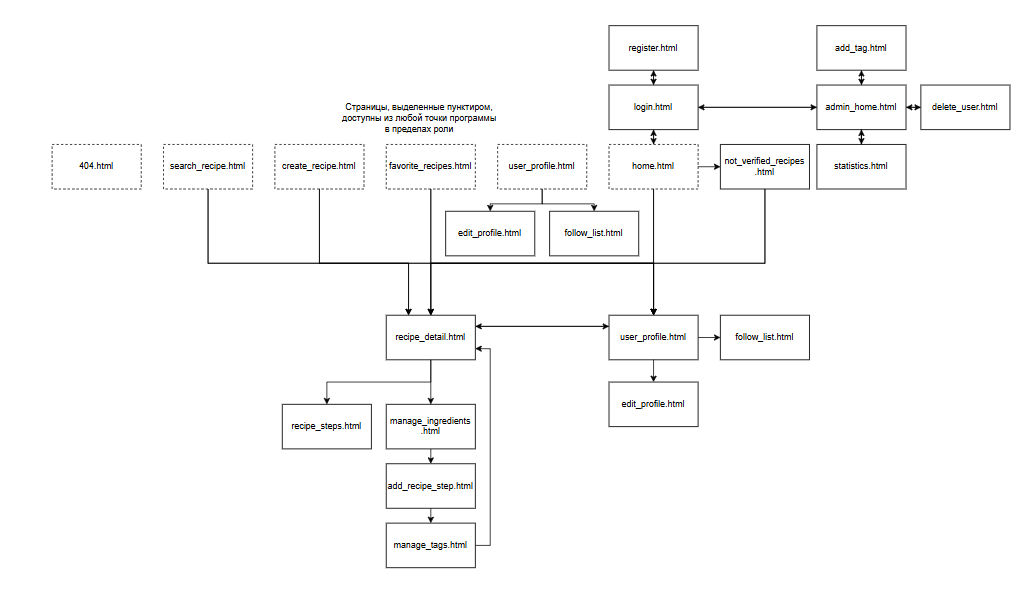


Рисунок 13 – Структурная схема

## Функциональная схема

На Рисунке 14 представлена функциональная схема, на которой графически изображены возможности пользователя в процессе эксплуатации.

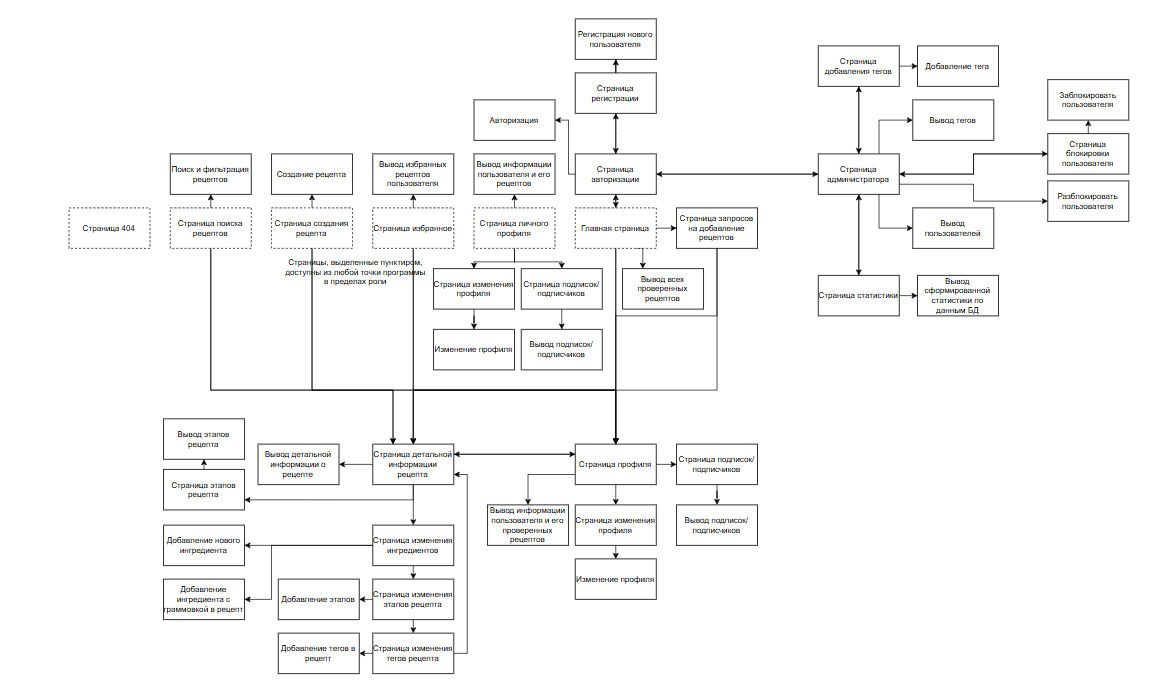


Рисунок 14 – Функциональная схема

## Диаграмма классов

Нет необходимости.

## Схема тестирования

На Рисунке 16 представлена схема тестирования WEB-приложения, на котором графически изображены возможности разных ролей в процессе эксплуатации.

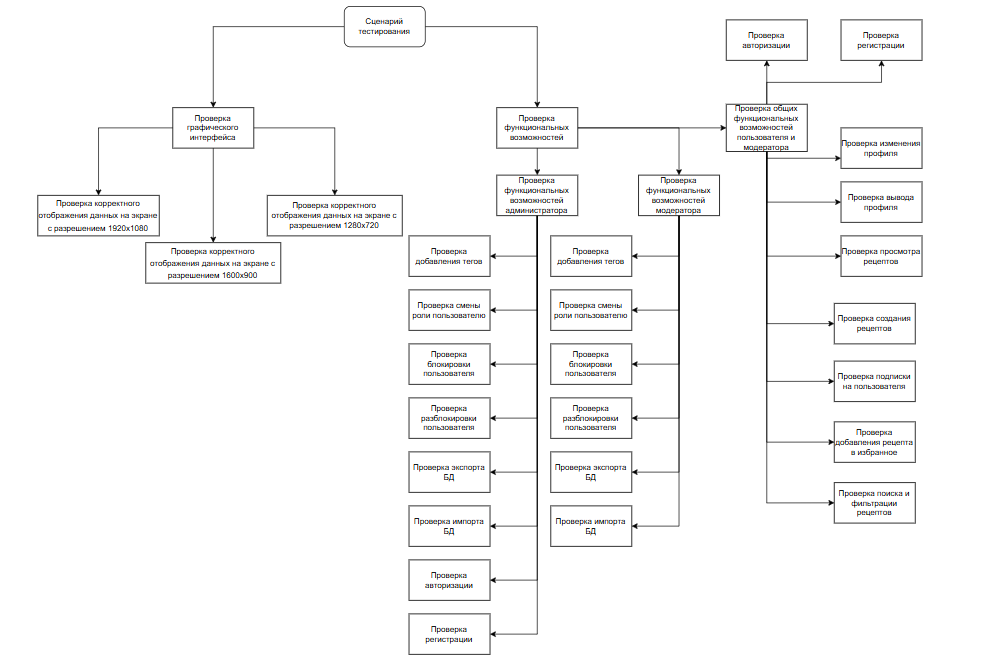


Рисунок 16 – Схема тестирования

## Схема пользовательского интерфейса

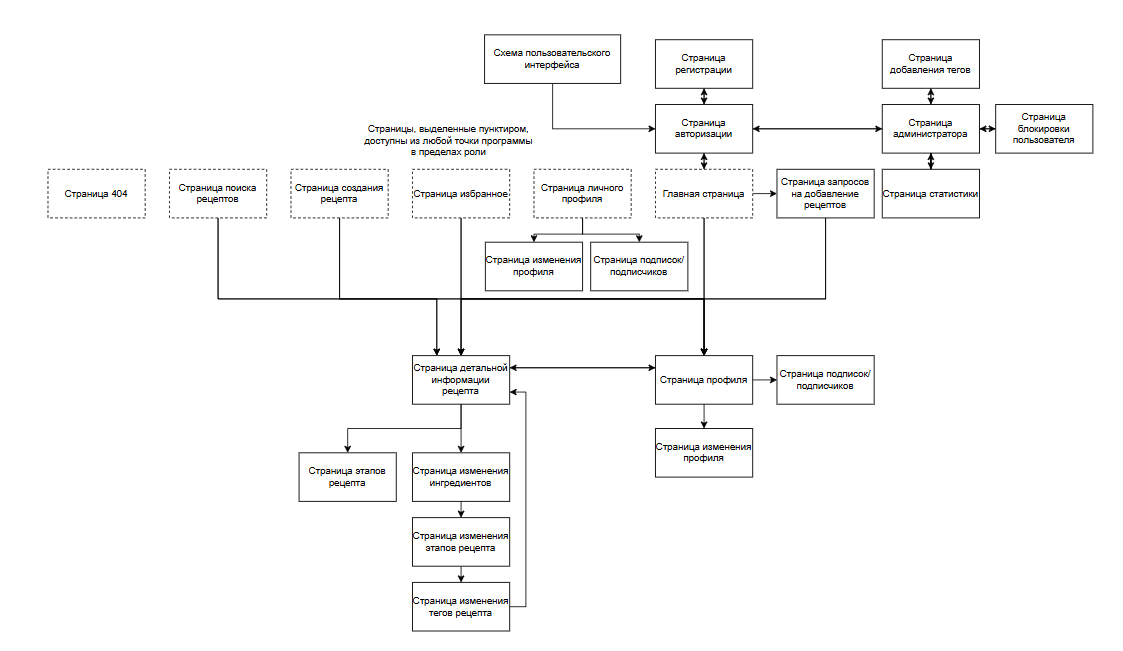


Рисунок 17 – Схема пользовательского интерфейса

Согласно схеме, на Рисунке 17, сначала пользователь попадает на страницу авторизации, где необходимо ввести данные для входа в свой аккаунт. Если же аккаунта нет, то можно зарегистрироваться на странице регистрации, перейдя по ссылке. Авторизовавшись, пользователь в зависимости от роли может попасть на две домашние страницы. В случае, если пользователь – Админ, то он попадает на админ-панель, где отображаются пользователи, теги, можно перейти на страницу добавления тегов, страницу блокировки пользователей, страницу статистики, а также можно экспортировать БД в формате sql или csv, разблокировать пользователя, или импортировать БД из файла. Если же пользователь не обладает ролью админа, то он переходит на основную страницу WEB-приложения. В этой части программы из-за статической шапки со ссылками всегда доступны следующие страницы: главная страница, страница «Избранное», страница создания рецепта, страница поиска рецептов. Также, при открытии страницы, на которую отсутствует доступ у пользователя, возвращается страница с ошибкой 404. На самой же главной странице отображается список всех проверенных рецептов, отсортированных по новизне. Если же пользователь обладает правами модератора, то здесь же он может перейти на страницу подтверждения новых рецептов, где можно просматривать рецепты и принимать или отклонять их. У каждого рецепта есть своя детальная информация, открыв которую можно увидеть всю информацию, включая рецепты и теги, где можно добавить рецепт в избранное. Также, при определенных условиях (если рецепт еще не опубликован) можно редактировать рецепт, добавляя туда ингредиенты, этапы и теги. Открыв страницу этапов рецепта, можно увидеть и их. Открыв страницу личного профиля (своего или чужого), можно увидеть подробную информацию о пользователе, перейти на страницы, где выведутся его подписчики и подписки, опубликованные рецепты и еще неопубликованные или заблокированные (если открытый профиль ваш). Открыв страницу избранного, можно увидеть все добавленные в избранное рецепты, с таким же функционалом что и на главной странице. На странице создания рецепта можно заполнить форму рецепта, создав его. После создания рецепта откроется страница добавления ингредиентов (новых, и непосредственно в рецепт). Далее откроется страница добавления этапов рецепта, где тоже можно добавлять их в рецепт, а после, страница тегов, где новые теги добавлять нельзя, но можно имеющиеся прикреплять к рецепту. После этого приложение возвращается на страницу детального рецепта. Открыв страницу поиска появится поле для ввода названия рецепта и фильтрация по времени приготовления и по тегам.

## Результат работы программы

В результате поставленной задачи было разработано требуемое программное решение в виде WEB-приложения

Более развёрнуто результат работы программы описан в ПРИЛОЖЕНИИ Г «Руководство пользователя».

На рисунке 18 представлен основной экран авторизации. На рисунке 19 представлено окно регистрации для пользователя.

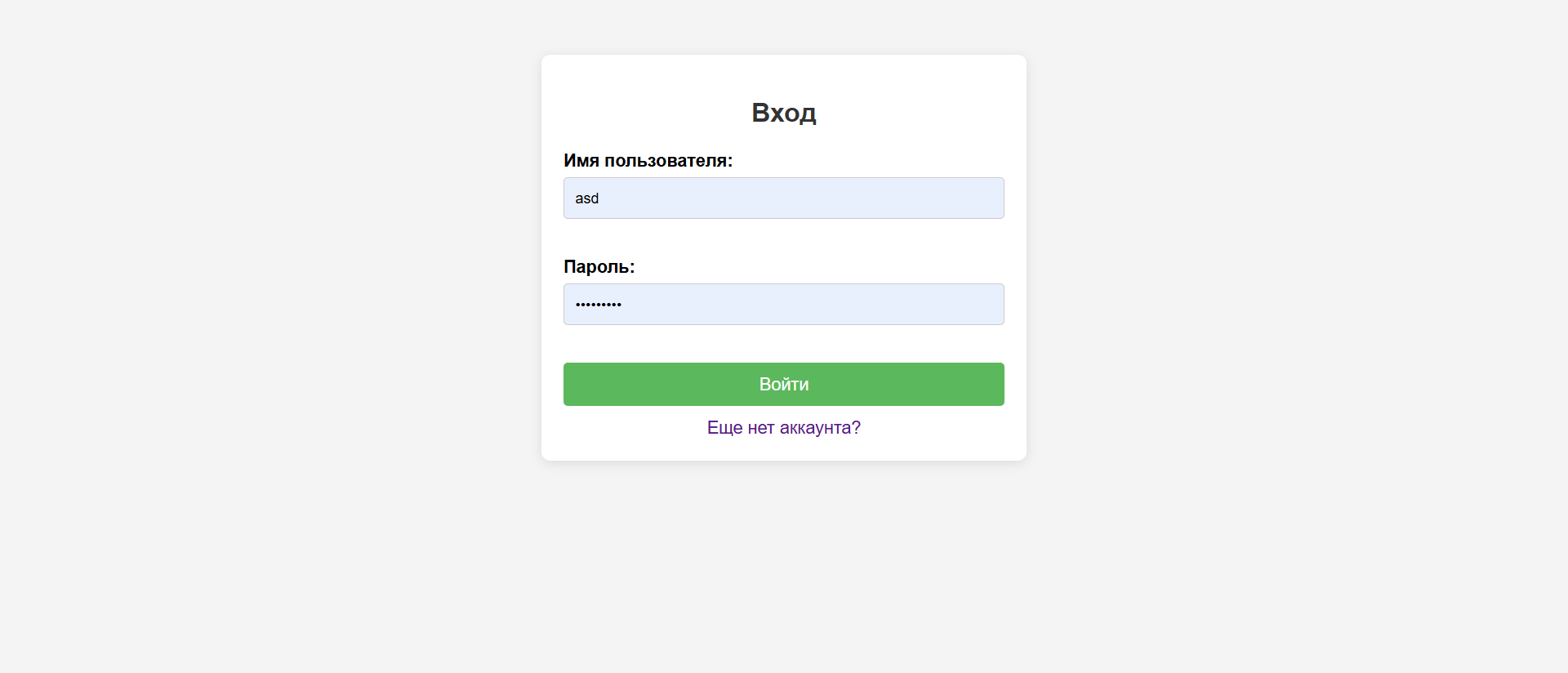


Рисунок 18 – Страница авторизации

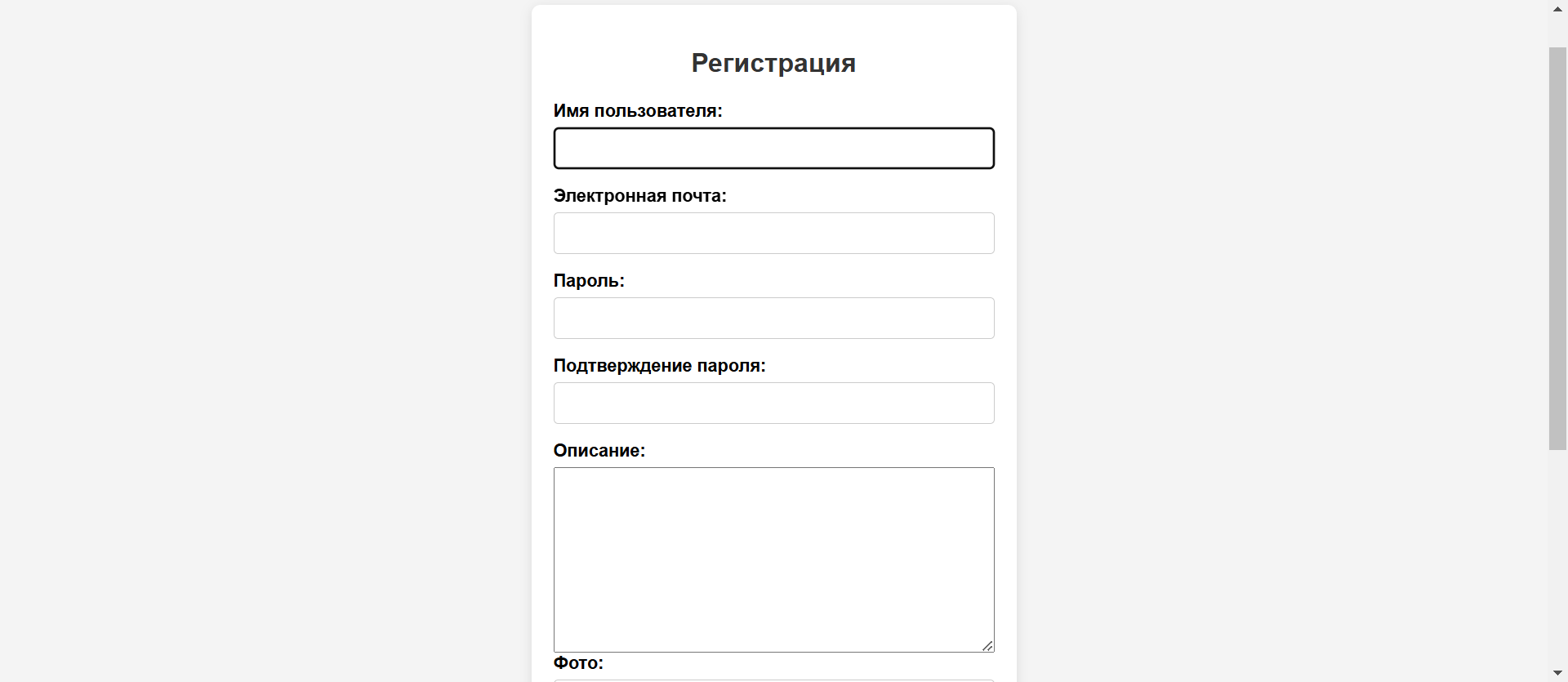


Рисунок 19 – Страница регистрации

На Рисунке 20 представлена главная страница.

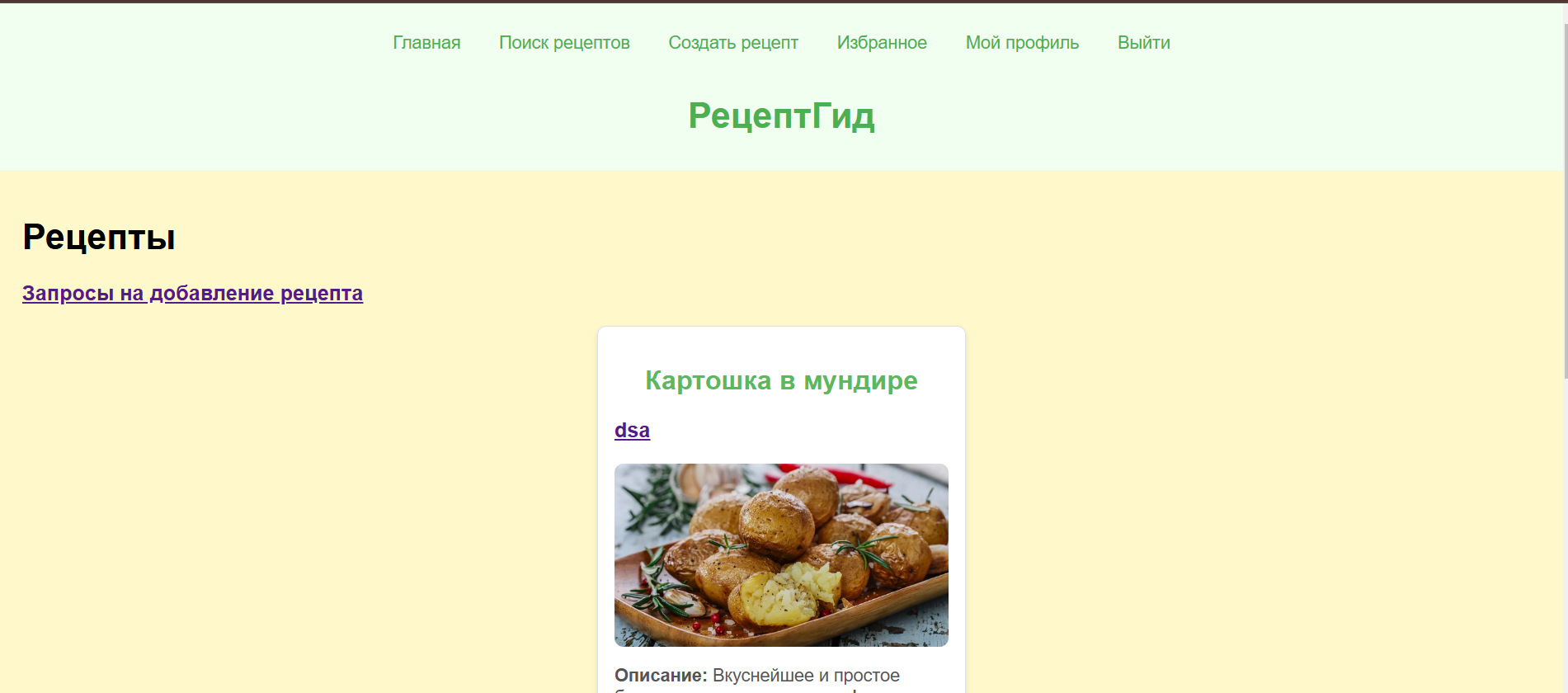


Рисунок 20 – Главная страница

На рисунке 21 представлена страница поиска рецепта.

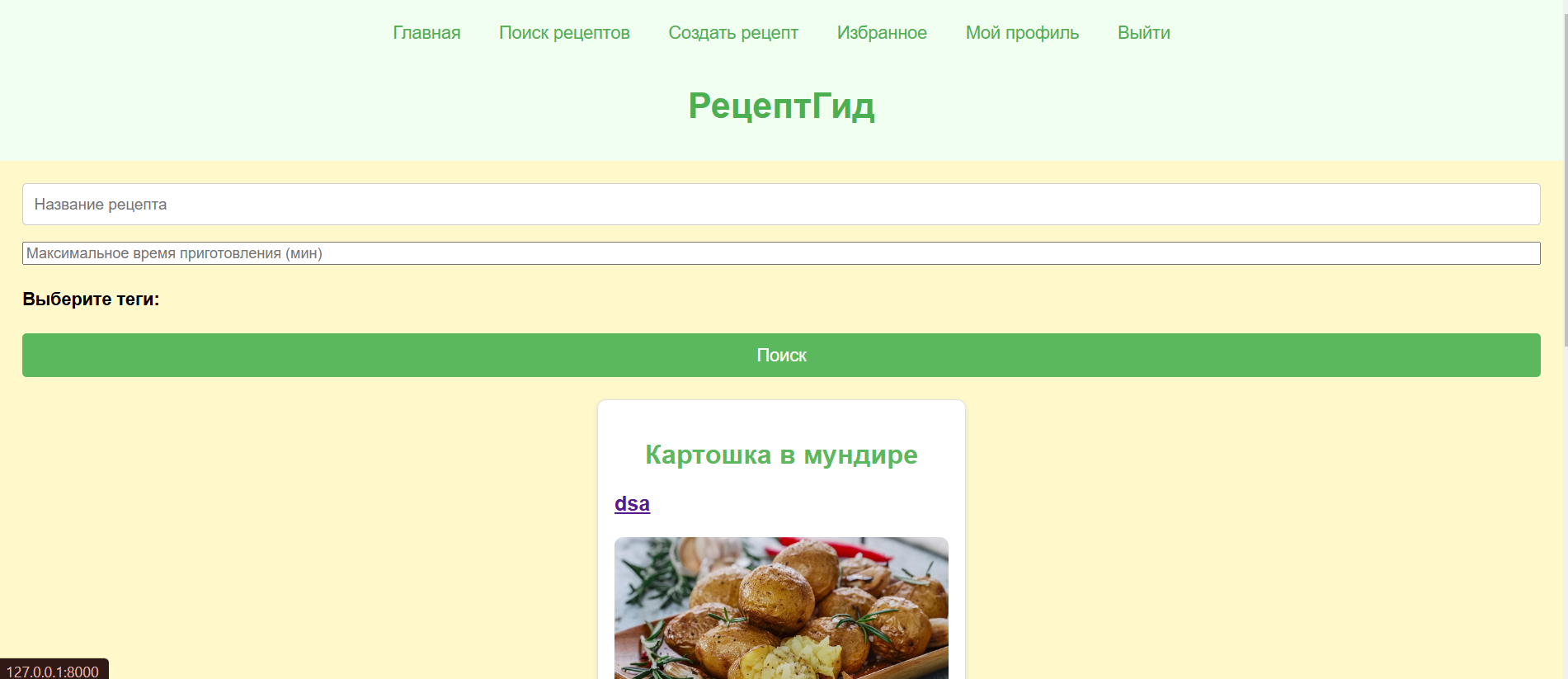


Рисунок 21 – Страница поиска рецептов

На рисунке 22 представлена страница создания рецепта.

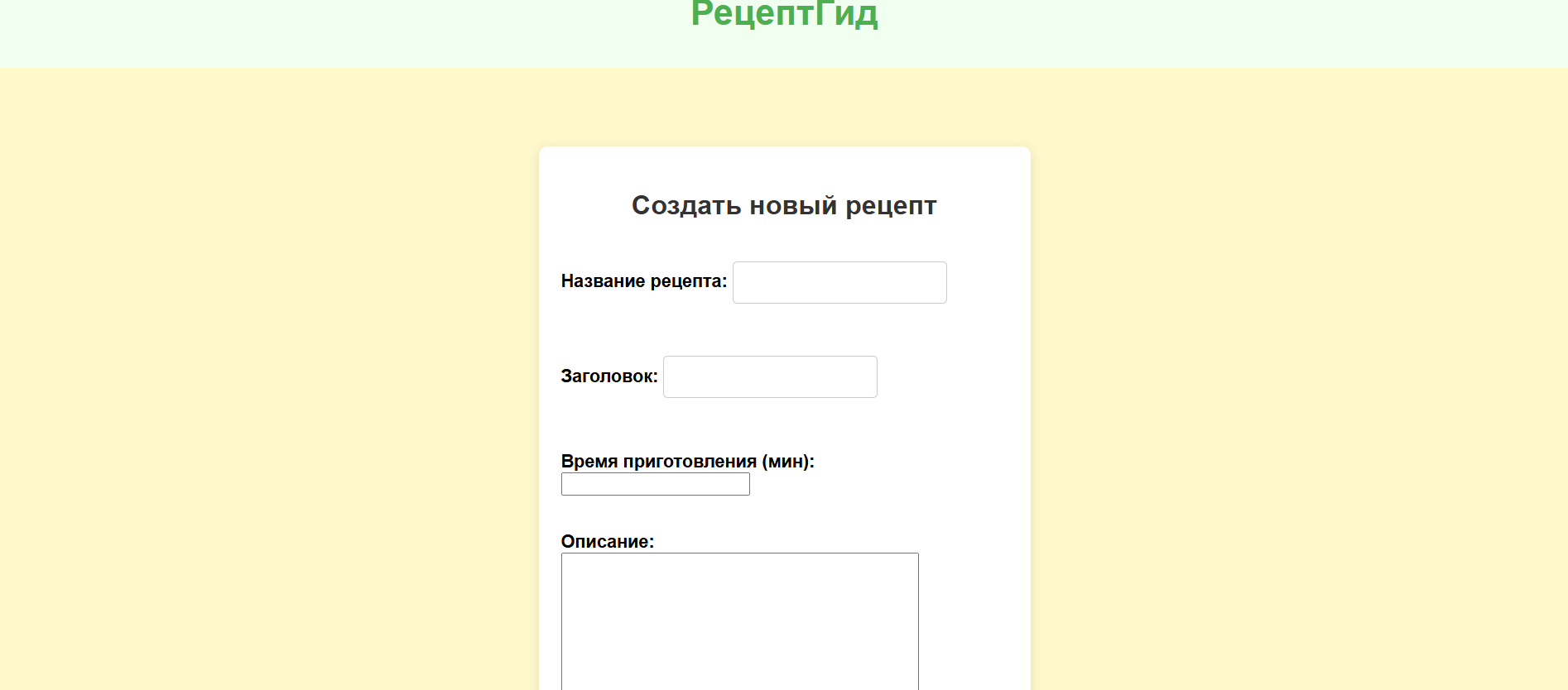


Рисунок 22 – Страница создания рецептов

На рисунке 23 представлена страница избранных рецептов.

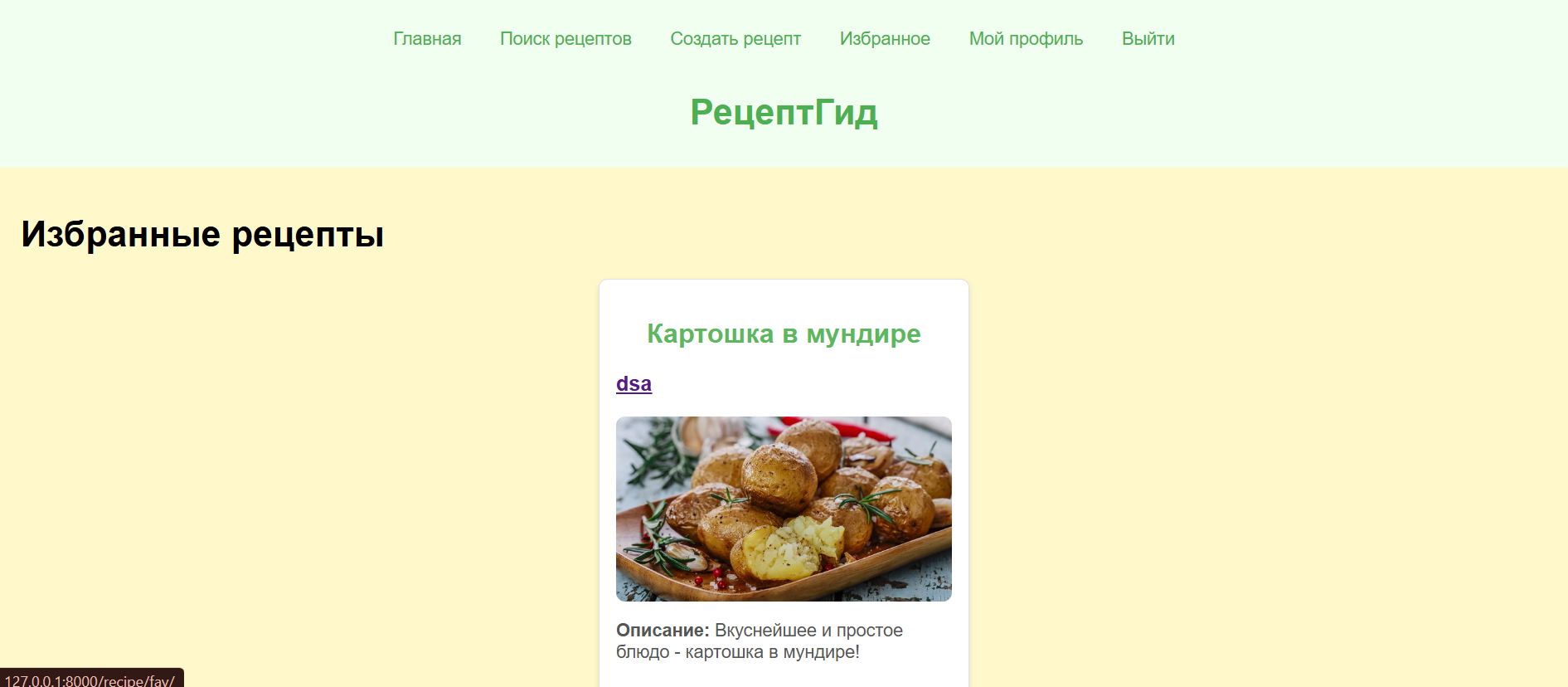


Рисунок 23 – Страница избранных рецептов

На рисунке 24 представлена страница профиля пользователя.

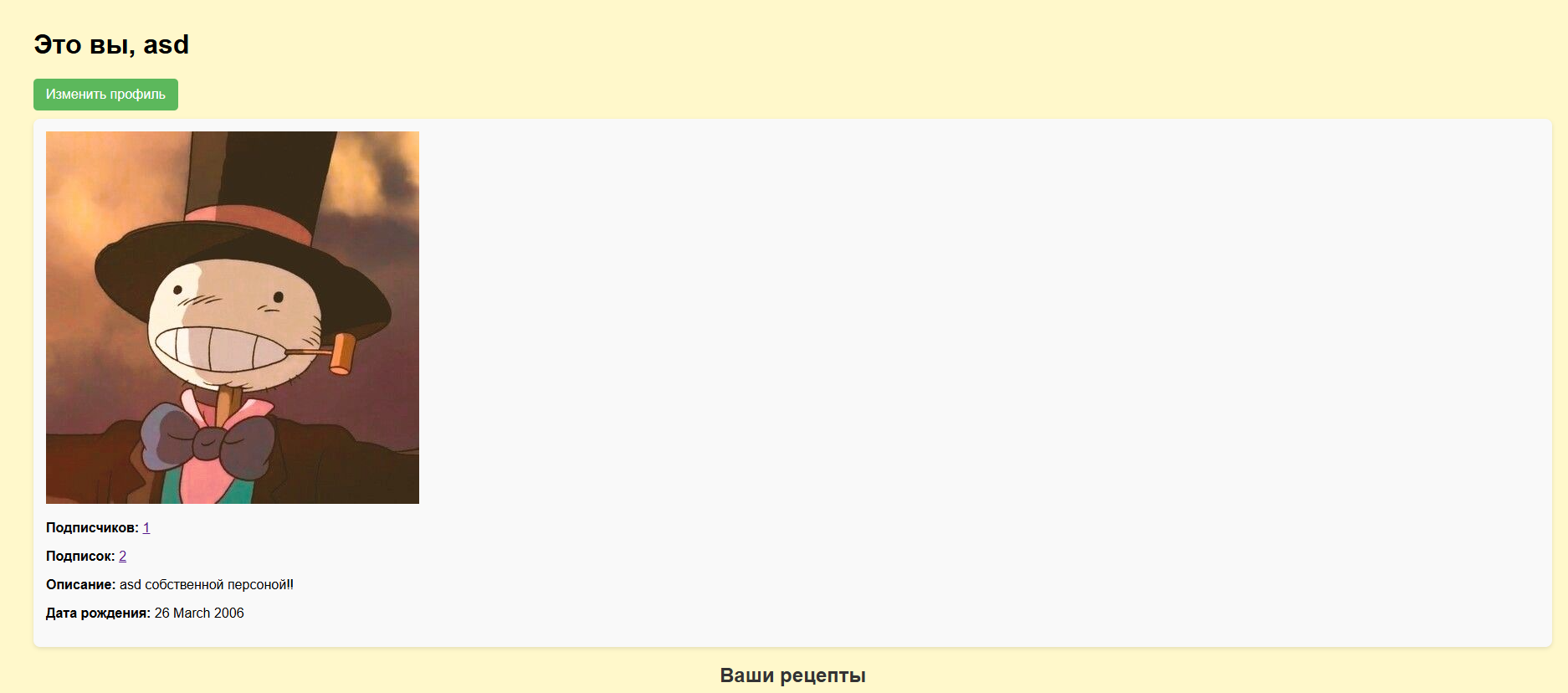


Рисунок 24 – Страница профиля

На рисунке 25 представлена страница подробной информации о рецепте.

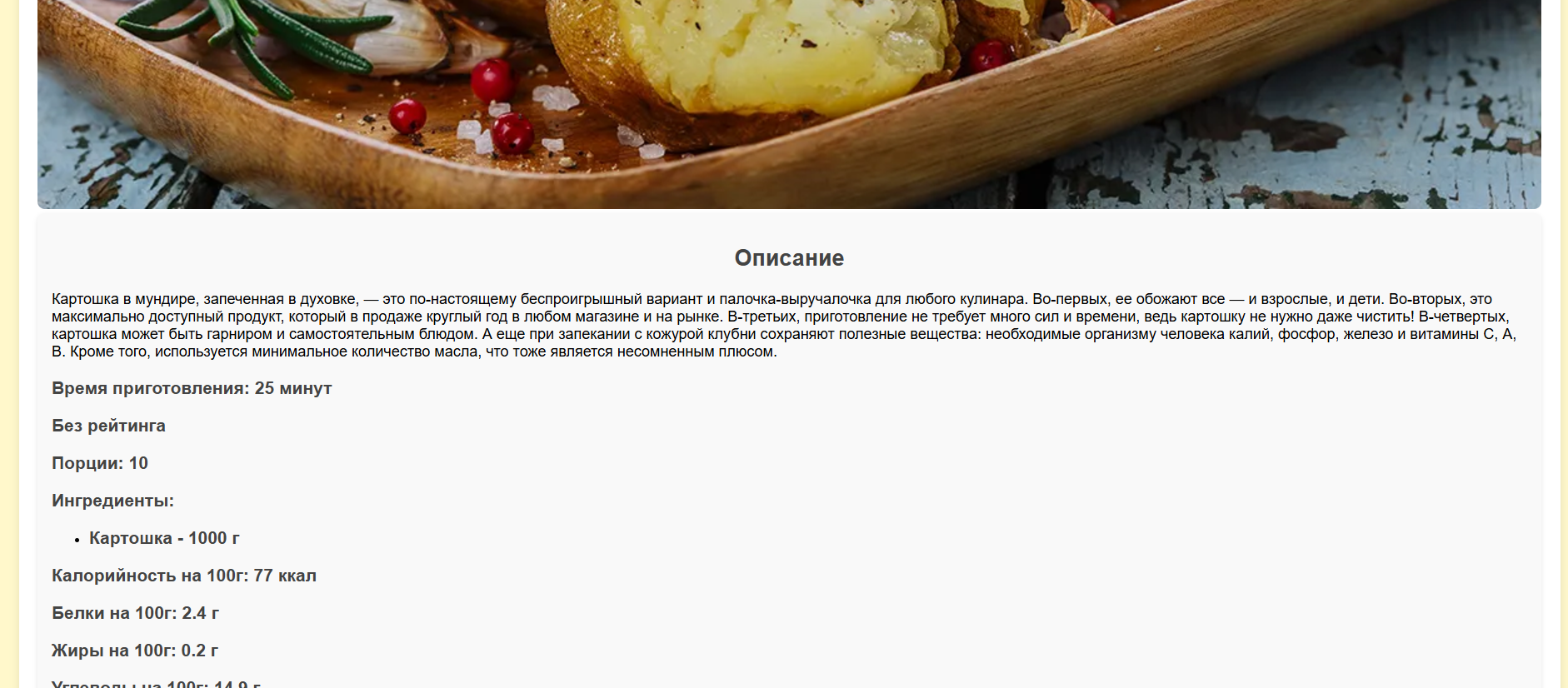


Рисунок 25 – Подробная информация о рецепте

На рисунке 26 представлена домашняя страница администратора.

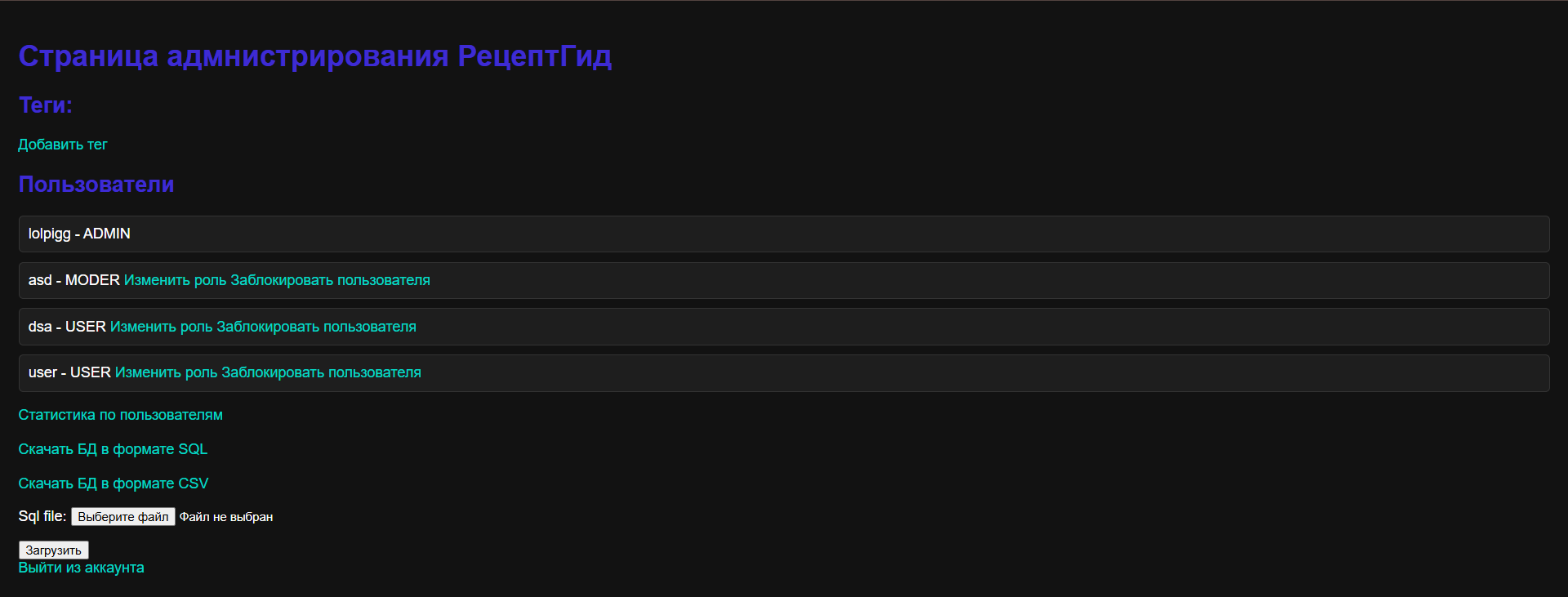


Рисунок 26 – Домашняя страница администратора

На рисунке 27 представлена страница статистики пользователей и рецептов.

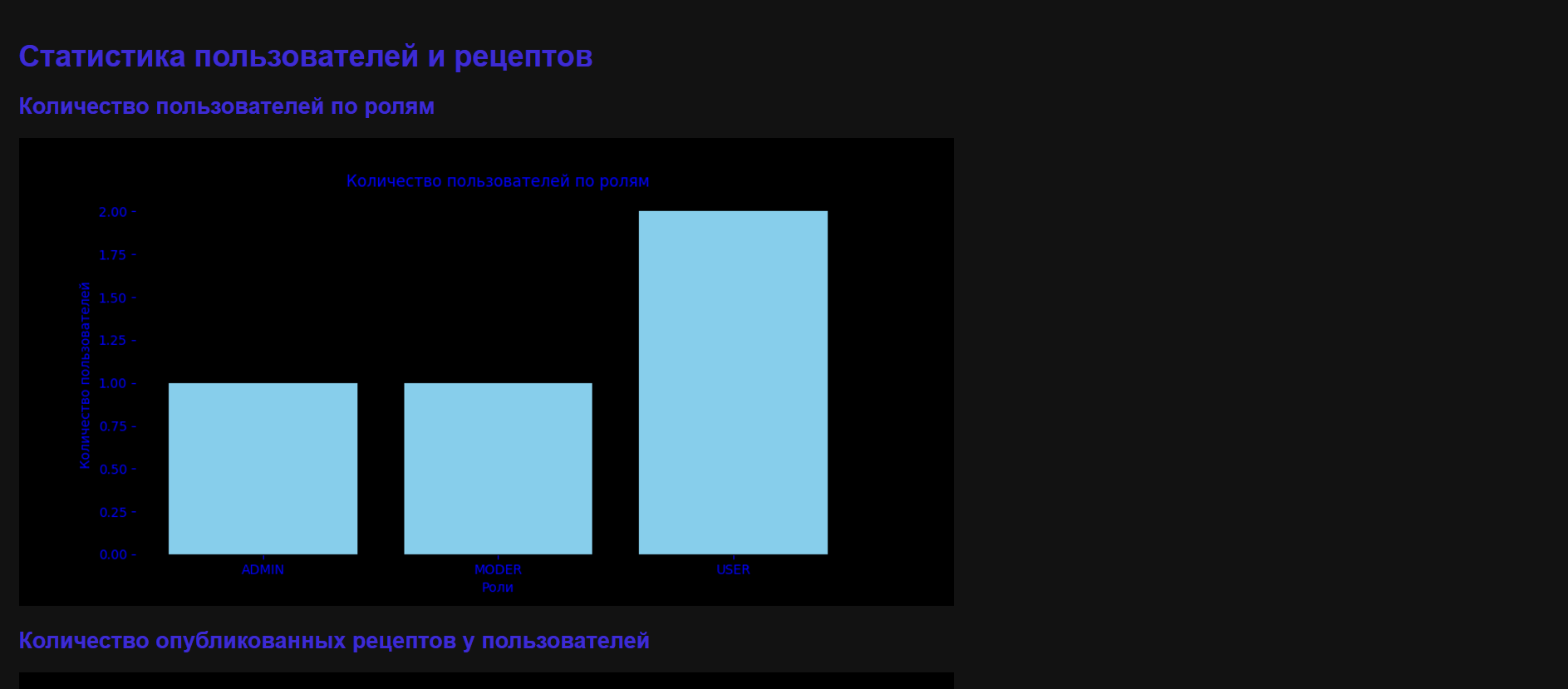


Рисунок 27 – Статистика пользователей и рецептов

# ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

## Инструментальные средства

Для разработки WEB-приложения создания и поиска рецептов была использована интегрированная среда разработки PyCharm. Она предлагает мощные инструменты для написания, отладки и тестирования приложений на Python. PyCharm поддерживает множество библиотек и фреймворков, что делает разработку более гибкой и продуктивной. Удобный интерфейс и встроенные средства отладки помогают разработчикам эффективно работать с кодом.

В качестве системы управления базами данных была выбрана SQLite — легковесная реляционная база данных, которая обеспечивает быстрый доступ к данным и надежное хранение. SQLite идеально подходит для небольших и средних приложений, благодаря своей простоте в использовании и отсутствию необходимости в отдельном сервере. Это позволяет легко интегрировать базу данных в приложение без лишних затрат на настройку.

Python был выбран в качестве основного языка разработки благодаря своей простоте и читаемости, а также мощным возможностям для работы с базами данных. Он поддерживает объектно-ориентированный подход и множество современных библиотек, таких как Django для создания веб-приложений, что делает его идеальным для разработки WEB-приложения создания и поиска рецептов.

## Отладка программы

Отладка — это важный этап разработки программного обеспечения, на котором выявляются и исправляются ошибки в коде. В рамках разработки WEB-приложения создания и поиска рецептов отладка помогает обеспечить корректную работу всех функций приложения, таких как управление рецептами, личный профиль и взаимодействие с базой данных.

PyCharm: интегрированная среда разработки, которая предоставляет мощные инструменты для отладки, включая точки останова, просмотр переменных и трассировку стека.

Логирование: использование встроенного логирования Django для записи информации о работе программы, что позволяет отслеживать выполнение кода и выявлять проблемные места.

## Защитное программирование

На Рисунке 28 представлена обработка ошибок при авторизации пользователя в случае неверных данных или блокировки.



Рисунок 28 - Код программы

## Характеристики программы

Характеристики WEB-приложения представлены в приложении А «Текст программы» в таблице 1 «Модули».

# ЗАКЛЮЧЕНИЕ

В рамках курсового проекта было создано WEB-приложение размещения и поиска рецепта «РецептГид».

В процессе разработки была реализована вся функциональность, предусмотренная техническим заданием. Выполнены ключевые этапы проектирования, включая разработку архитектуры приложения, создание моделей данных, диаграмм взаимодействия компонентов и описание входных и выходных данных.

На подготовительном этапе проведена настройка среды разработки PyCharm и установка сервера базы данных sqlite. Эти действия обеспечили надежное взаимодействие интерфейса приложения с базой данных, а также организовали хранение мультимедийных файлов, таких как изображения рецептов и профили.

В ходе разработки системы были реализованы функции добавления и разграничения доступа рецептов, создание личного профиля, обработки возвратов и управления учетными записями пользователей. Приложение создано с использованием языка программирования Python и языка разметки html, что позволило разработать производительное, функциональное и удобное для пользователя приложение.

На этапе тестирования системы были проверены все реализованные функции, выявлены и устранены недочеты. Благодаря использованию методов защитного программирования система продемонстрировала стабильную работу и отсутствие критических ошибок.

Реализация проекта позволила закрепить знания в проектировании баз данных на языке sqlite, а также в разработке WEB-приложений с использованием PyCharm. Итоговый результат полностью соответствует поставленным задачам и создает возможности для дальнейшего развития системы.

Несмотря на достигнутые результаты, система обладает потенциалом для совершенствования. Среди направлений для дальнейшего улучшения можно выделить:

Добавление системы рецензий рецептов.

Реализацию функции обмена сообщениями.

Адаптацию интерфейса под мобильные устройства.

Внедрение автоматического формирования логирования запросов и данных внутри приложения.

Разработанное WEB-Приложение уже является эффективным инструментом для автоматизации процессов размещения и поиска рецептов и закладывает основу для дальнейшего расширения функциональности.

# СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

1. Документация Django: сайт <https://docs.djangoproject.com/en/stable/>. Дата обращения:11.09.2024.

2. Документация Django Auth: сайт - <https://docs.djangoproject.com/en/stable/topics/auth/>. Дата обращения:15.09.24.

3. Создание макета для приложения: сайт - <https://www.sqlite.org/docs.html>. Дата обращения: 11.09.2024.

4. Пример сайта рецептов: сайт - <https://www.russianfood.com>. Дата обращения: 20.10.2024.

5. Исправление ошибок: сайт - <https://stackoverflow.com/questions/48737478/how-to-load-sql-file-with-manage-py>. Дата обращения: 15.10.2024