ФГБОУ ВО

«Сибирский Государственный Университет Телекоммуникаций и Информатики»

Кафедра Прикладной математики

и кибернетики

Курсовая работа часть 2

по дисциплине

«Визуальное программирование и человеко-машинное взаимодействие»

Выполнил: студент группы ИП-814

Краснов И.В.

Проверил: доцент кафедры ПМ и К

Мерзлякова Е.Ю.

Оглавление

[Часть 1 3](#_Toc62040119)

[Анализ задач и пользователей 3](#_Toc62040120)

[Выбор репрезентативных задач 4](#_Toc62040121)

[Заимствования 5](#_Toc62040122)

[Черновое описание дизайна 7](#_Toc62040123)

[Часть 2 9](#_Toc62040124)

[Описание программы 9](#_Toc62040125)

[Скриншоты работы программы 10](#_Toc62040126)

[Часть 3 20](#_Toc62040127)

[CWT-анализ 20](#_Toc62040128)

[Постановка задач 20](#_Toc62040129)

[Первая задача. Поиск по фильтрам 20](#_Toc62040130)

[Вторая задача. Просмотр рецептов 20](#_Toc62040131)

[Третья задача. Создание рецепта 21](#_Toc62040132)

[Итог CWT-анализа 22](#_Toc62040133)

[GOMS-анализ 23](#_Toc62040134)

[Описание анализа 23](#_Toc62040135)

[Первая цель 23](#_Toc62040136)

[Вторая цель 24](#_Toc62040137)

[Оценка интерфейса в соответствии с принципами Нильсена-Молиха 27](#_Toc62040138)

[Соответствие правилам организации графического интерфейса 29](#_Toc62040139)

[Список выявленных и исправленных проблем 31](#_Toc62040140)

[Выводы 32](#_Toc62040141)

[Ссылки 33](#_Toc62040142)

# **Часть 1**

## **Анализ задач и пользователей**

**Цель:** найти двух человек, которые могут быть заинтересованы в решении предложенной задачи. Дайте их краткое описание (возраст, образование, профессия, навыки в выбранной сфере, навыки владения компьютером). Люди должны быть заинтересованы в решении предложенной задачи.

**Решение:** программа для просмотра рецептов может заинтересовать как семейную пару, так и человека вне брака, а также других людей, живущих и готовящих пищу самостоятельно. В качестве заинтересованных лиц были выбраны 2 близких человека:

Мама: возраст – 49 лет, образование – среднее-специальное текстильное, профессия – ведущий финансовый консультант в банке, навыки в выбранной сфере – способности в приготовлении разных национальных блюд, навыки владения компьютером – средние.

Друг: возраст – 20 лет, образование – незаконченное высшее техническое,

профессия – не работает / студент, навыки в выбранной сфере – высокие кулинарные способности, навыки владения компьютером – высокие.

## **Выбор репрезентативных задач**

**Цель:** перечислить репрезентативные задачи, затем все задачи, решение которых будет поддерживать разрабатываемая программа.

**Решение:**

Репрезентативные задачи:

1. Просмотр рецептов: программа должна предоставлять удобный интерфейс для просмотра информации о рецептах.
2. Разделение на категории: программа должна разделять рецепты на разные категории в соответствии с типом блюда.
3. Хранение рецептов: программа должны сохранять все рецепты в базе данных.

Второстепенные задачи:

1. Поиск по фильтрам. Пользователь должен иметь возможность осуществлять поиск по названиям рецептов, ингредиентам и рецепту приготовления.
2. Сортировка блюд по категориям. Пользователь должен иметь возможность выбирать категорию, которая будет отображаться в приложении.
3. Добавление в избранное. Пользователь должен иметь возможность помечать рецепты как избранные. Программа должна относить избранные рецепты в отдельную вкладку.
4. Изменение рецептов. Программа должна иметь возможность добавления новых рецептов, редактирования рецептов и удаления рецептов.
5. Выход из приложения.

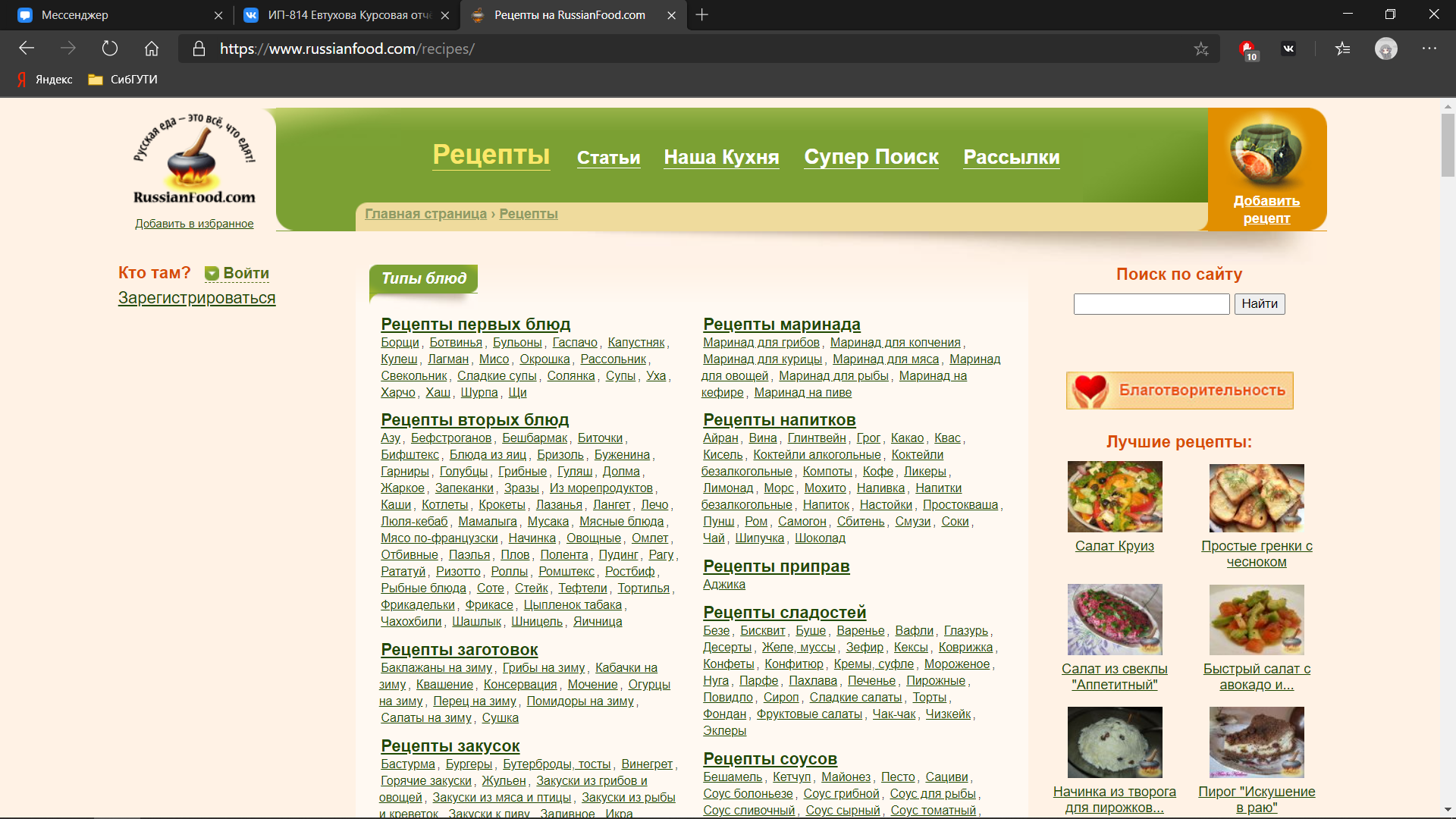
## **Заимствования**

**Задача:** найти приложения или сайты, с которых можно заимствовать какие-либо решения интерфейса, приведите ссылку на источник. Эти приложения не обязательно должны выполнять точно такие же задачи. Заимствовать можно что угодно, даже расположение кнопок. Выберите и напишите, что именно Вы будете заимствовать из данных приложений и зачем.

**Решение:**

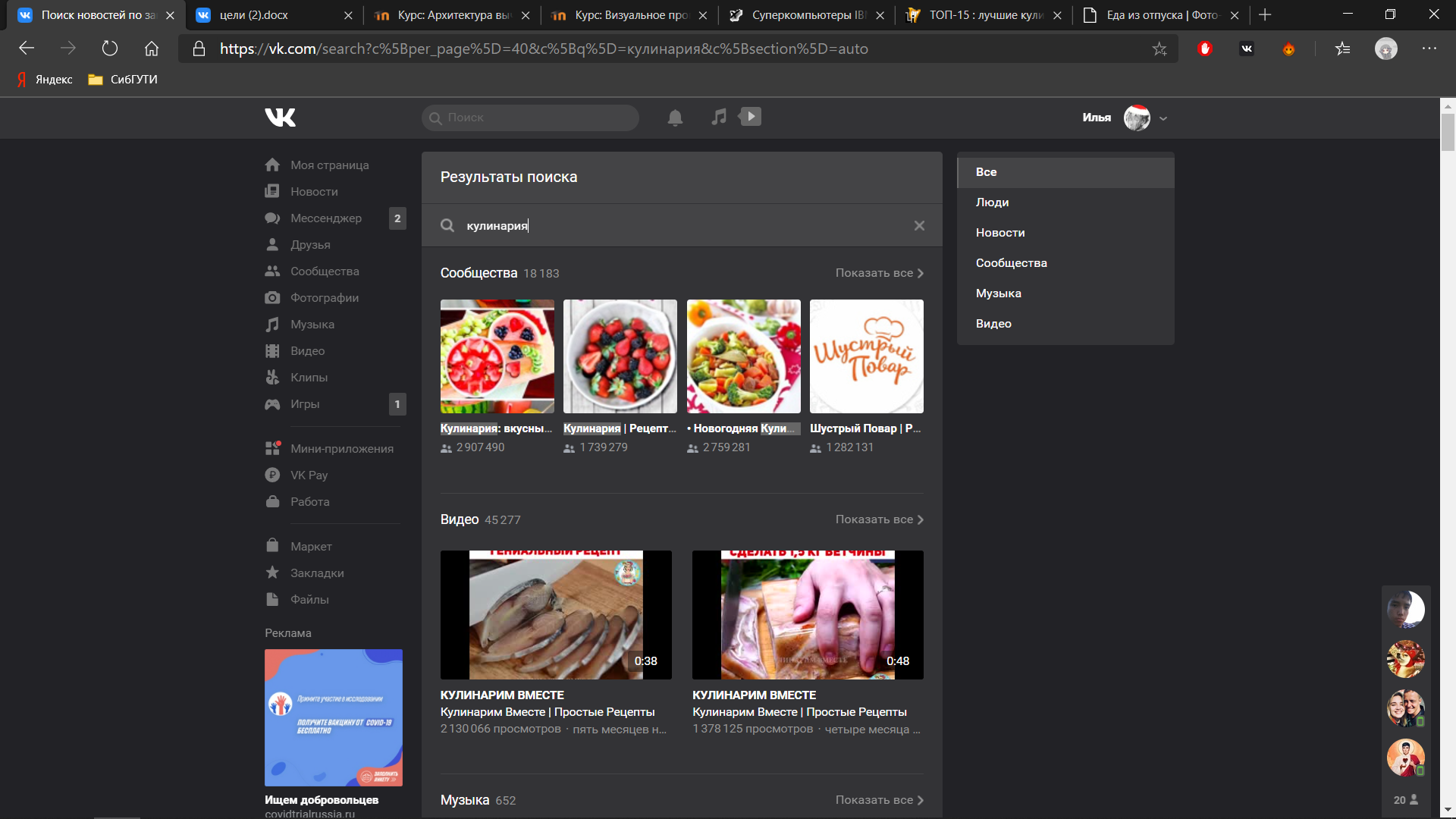
1. **RussianFood**

С данного портала будет заимствовано разделение блюд по категориям. С сайта взяты 10 основных категорий блюд: первое, второе, гарнир, закуски, соусы, напитки, заготовки, сладости, изделия из теста, маринады. В программе будет отображено 11 категорий, к списку категорий с данного портала добавляется категория «салаты». Также с данного ресурса взято несколько рецептов, которые будут заранее находиться в бае данных при первом запуске приложения.



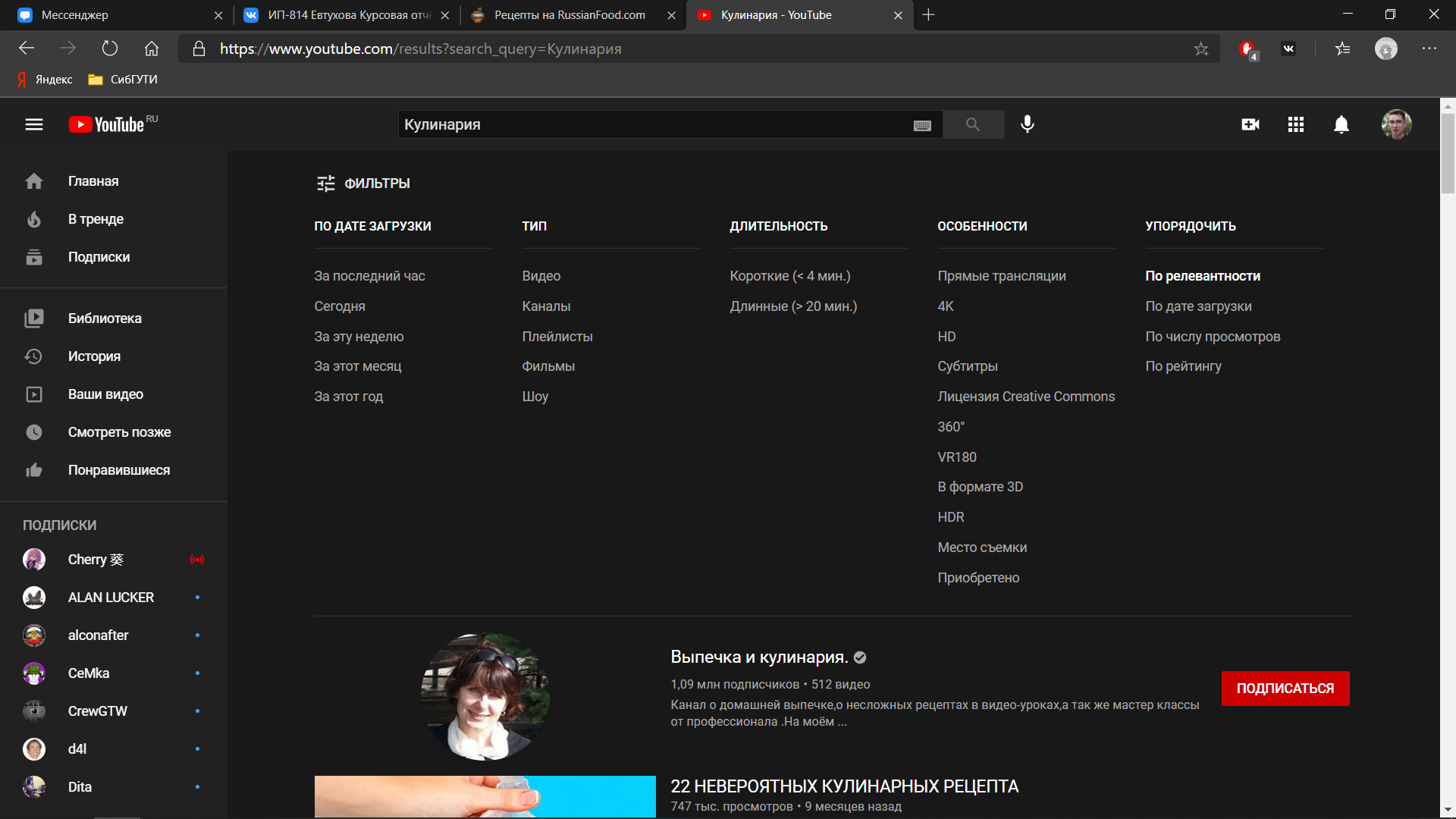
1. **ВКонтакте:**

С данного онлайн-сервиса будет заимствован поиск. Поиск в приложении будет осуществляться через текстовое поле. Результаты поиска будут делиться по категориям, и пользователь сам сможет выбрать, какую категорию следует отобразить



1. **YouTube:**

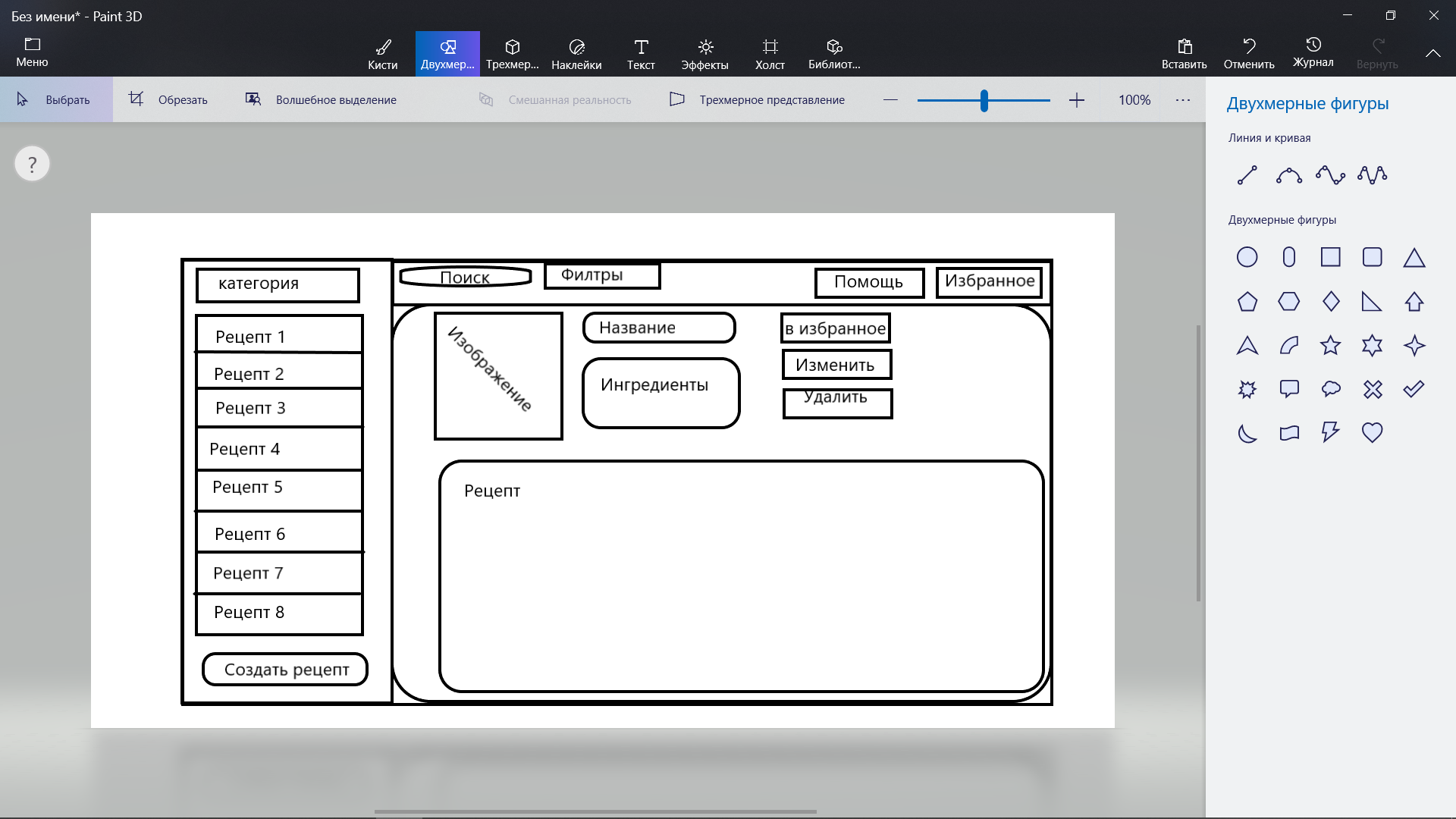
С данного ресурса заимствована идея поиска по фильтрам. Пользователь должен выбрать один фильтр в соответствии с которым будет осуществляться поиск.



## **Черновое описание дизайна**

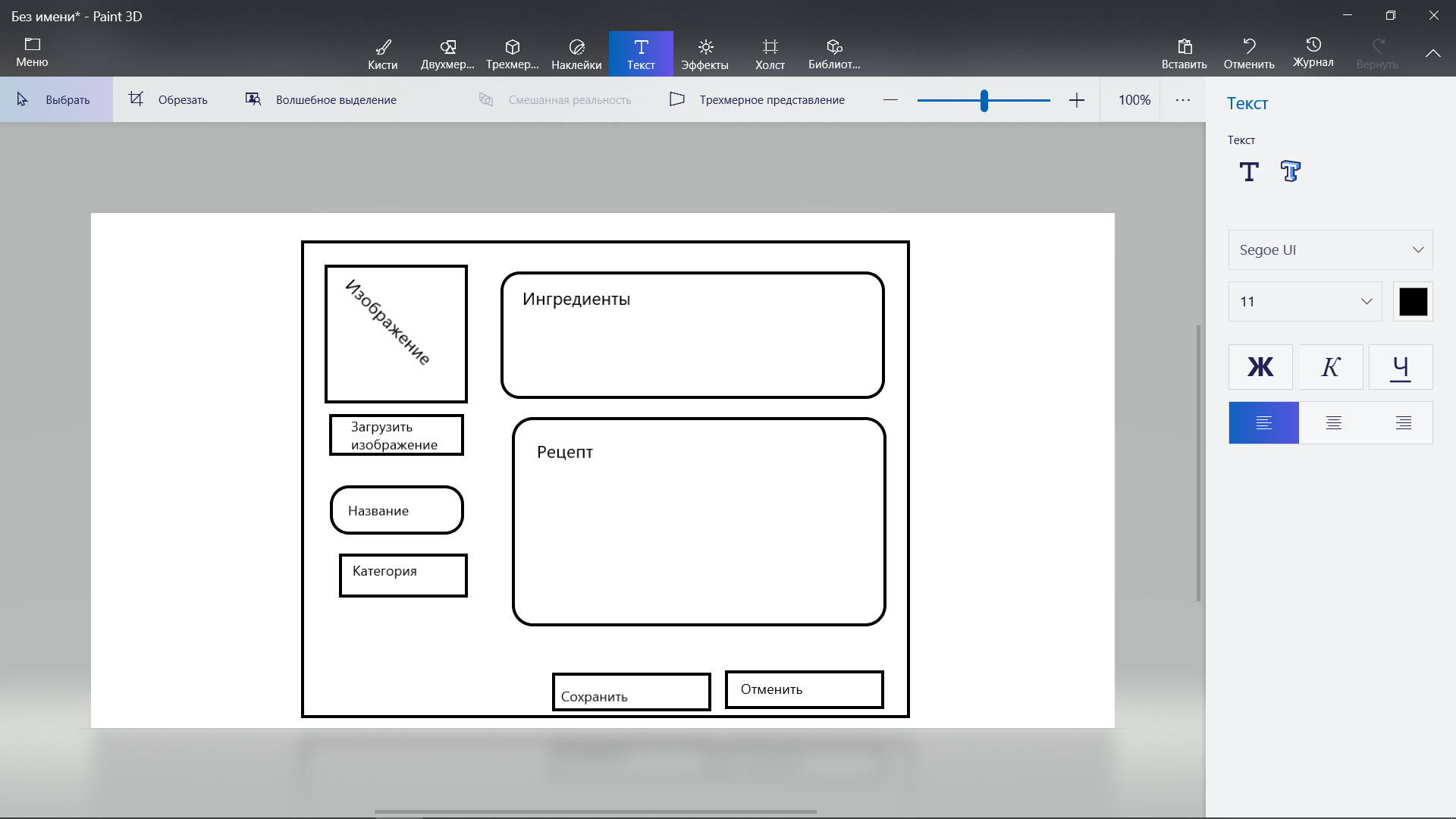
**Задача:** описать черновой вариант дизайна словами и графически (иллюстрации с пояснениями). Черновой вариант должен отражать все внешние элементы интерфейса и их назначение.

**Решение:**



(рис. 1) Главное окно приложения

В данном приложении будет производиться распределение по категориям, доступные категории будут представлены в меню, которое появляется при нажатии на кнопку «Категория». После выбора категории в поле с рецептами будут отображаться только те рецепты, которые подходят по категории. Сверху будет поле для поиска. После поиска в поле с рецептами появляются рецепты, которые соответствуют критериям поиска. При нажатии на кнопку «фильтры» должно появляться меню с выбором фильтра, по которому будет проходить поиск. В правом верхнем углу будет кнопка «избранное». При нажатии на неё в поле с рецептами появятся рецепты, помеченные как избранные. При нажатии на кнопку «помощь» должен появиться навигатор помощи, открывающийся в отдельном окне. При выборе рецепта в поле с рецептами, появится информация о нём. В поле «изображение» появится сопровождающее изображение для рецепта. В поле «название» будет отображено название рецепта. Избранные рецепты должны иметь в названии звёздочку (\*). В поле «ингредиенты» будут отображены ингредиенты данного рецепта. В поле «рецепт» должен быть описан рецепт приготовления. При нажатии на кнопку «в избранное» рецепт должен помечаться как избранный. Если рецепт уже помечен как избранный, кнопка должна удалять его из избранного. На ней соответственно должно быть написано «удалить из избранного». При нажатии на кнопку «изменить» должно открыться окно изменения рецепта. При нажатии на кнопку удалить рецепт должен удаляться из базы. При нажатии на кнопку «создать рецепт» должно открываться окно создания нового рецепта.



(рис. 2) Окно редактирования\создания рецепта

При создании нового рецепта пользователь должен обязательно заполнить все поля. В поле «название» должно быть введено название. В поле «ингредиенты» должны быть введены ингредиенты. В поле «рецепт» нужно написать рецепт приготовления блюда. При нажатии на кнопку «категория» должно появляться меню с 11 доступными категориями, одну из которых пользователь обязательно должен выбрать. При нажатии на кнопку «загрузить изображение» должно открываться окно выбора изображения. При нажатии на кнопку сохранить при условии, что заполнены все поля, рецепт должен добавляться в базу данных. Если одно из полей не заполнено, должно появиться уведомление, которое не даст сохранить рецепт. При нажатии на кнопку «отменить» окно создания рецепта закроется без сохранения написанных в нём данных.

Окно редактирования практически не отличается от окна создания рецептов. Единственное отличие – все данные введены заранее. При нажатии на кнопку «отменить» изменения не будут внесены в базу данных.

# **Часть 2**

## **Описание программы**

Программа по теме «Книга рецептов» реализована на Qt 6.0.0. Итоговый продукт получил название «Avocado».

В программе реализован следующий функционал:

1. Взаимодействие с рецептами – просмотр, добавление рецептов, удаление рецептов, редактирование рецептов.
2. Хранение рецептов в базе данных.
3. Разделение всех рецептов на категории с возможностью отображение конкретной категории или всех рецептов сразу. Доступно 11 категорий.
4. Добавление рецептов во вкладку избранное и возможность просмотра избранных рецептов.
5. Поиск рецептов по 4 фильтрам: поиск по имени, поиск по ингредиентам, поиск по рецепту и глобальный поиск (по имени, ингредиентам и рецепту одновременно).
6. Навигатор помощи с подробным описанием функционала приложения.

Для каждого функционального элемента создан свой класс и своё окно. В программе реализованы 5 классов: класс основного окна приложения, класс окна создания рецепта, класс окна редактирования рецепта, класс окна навигатора помощи, класс обработки базы данных, недоступный для пользователя.

База данных реализована на языке SQLite, взаимодействующим с функционалом библиотеки QSql. В классе-обработчике базы данных содержится функционал для:

1. Добавление элемента в базу данных

query.prepare("INSERT INTO (…) VALUES (?, ?, ?, ?, ?, ?)");

query.exec();

1. Редактирование элемента базы данных

query.prepare("UPDATE (…) SET (…) WHERE (…)");

query.exec();

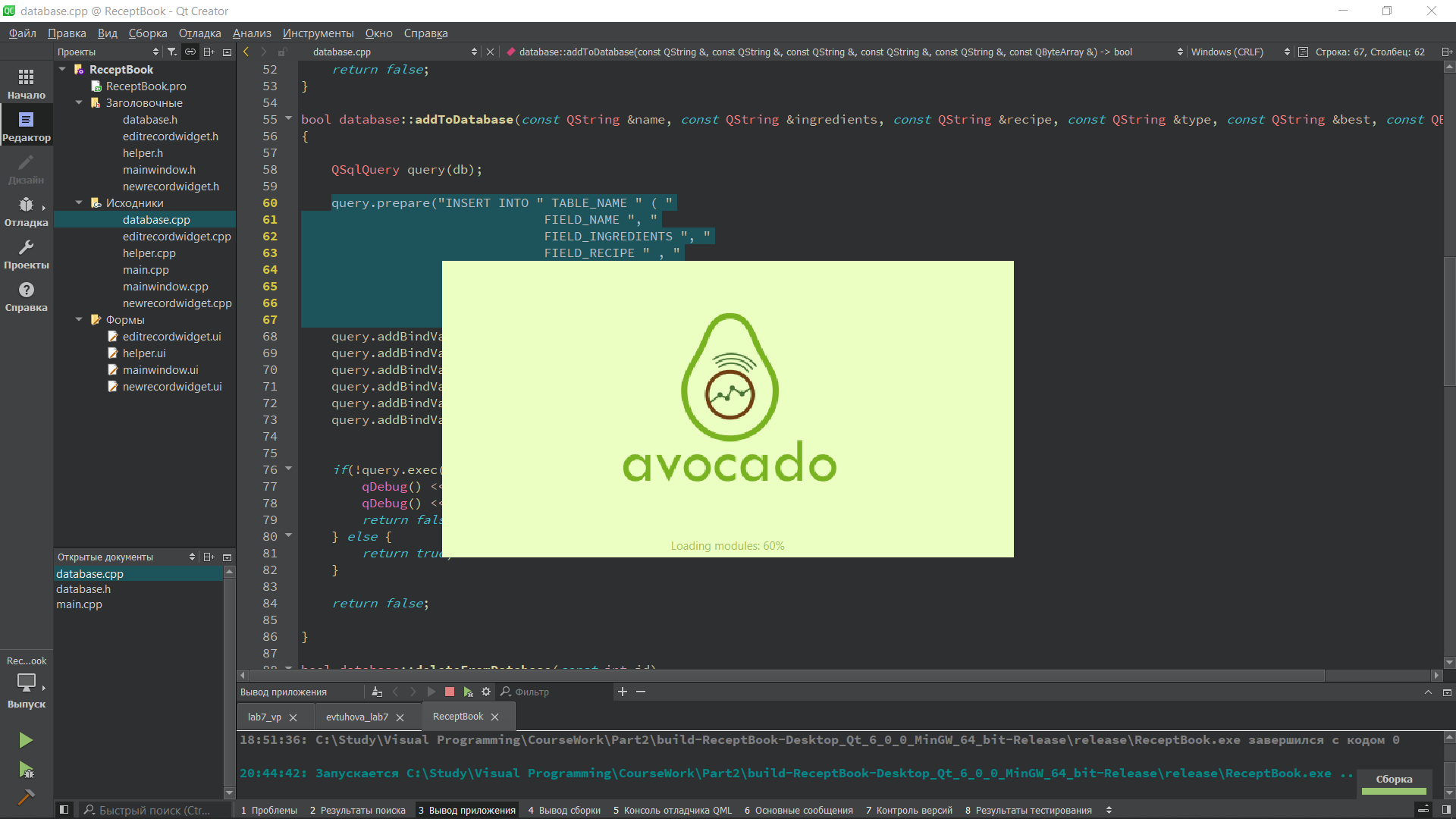
1. Удаление элемента из базы

query.prepare("DELETE FROM (…) WHERE (…)");

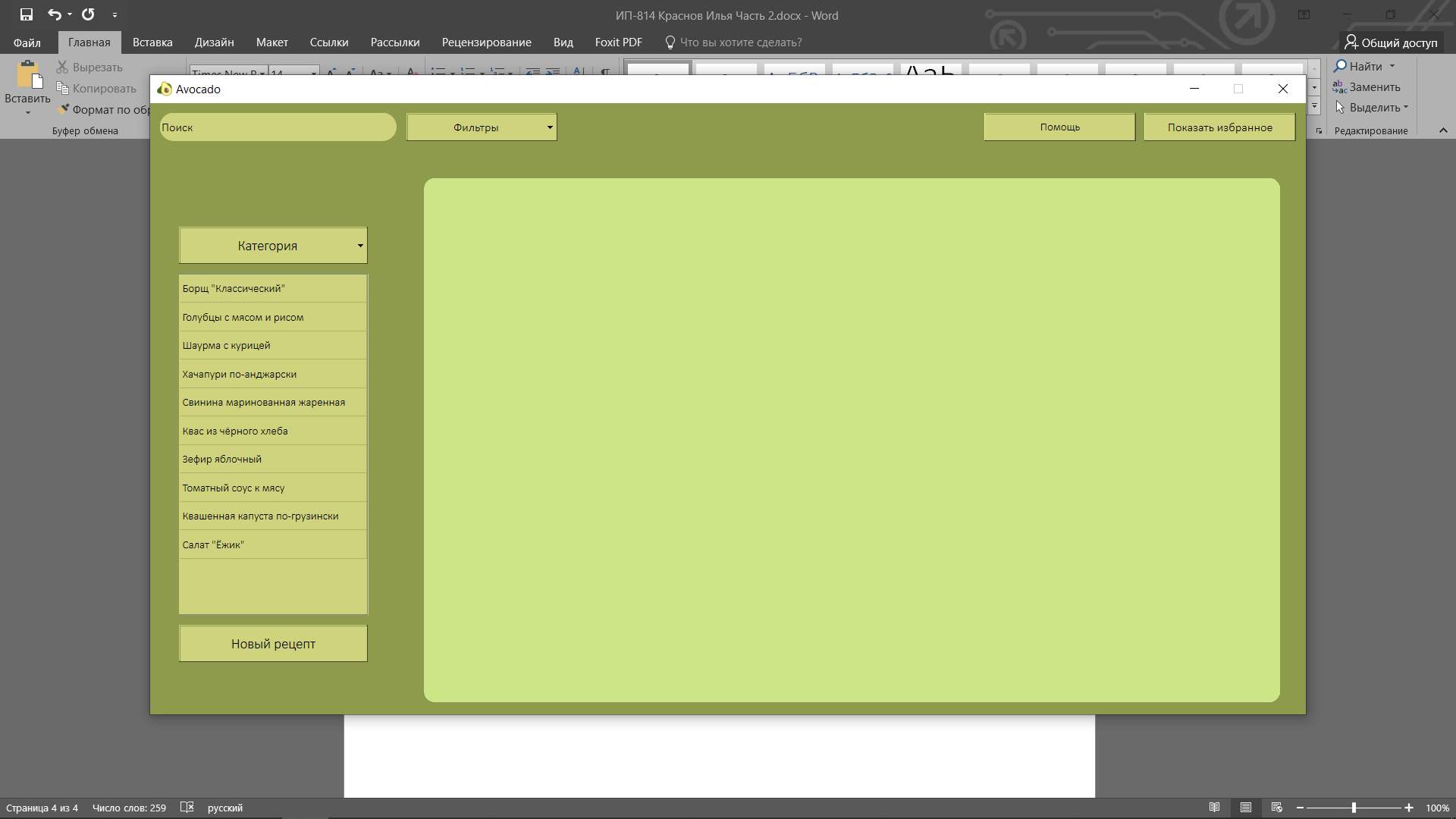
query.exec();

Навигатор помощи реализован с помощью файлов .html, содержимое которых выводится в элемент класса QTextBrowser.

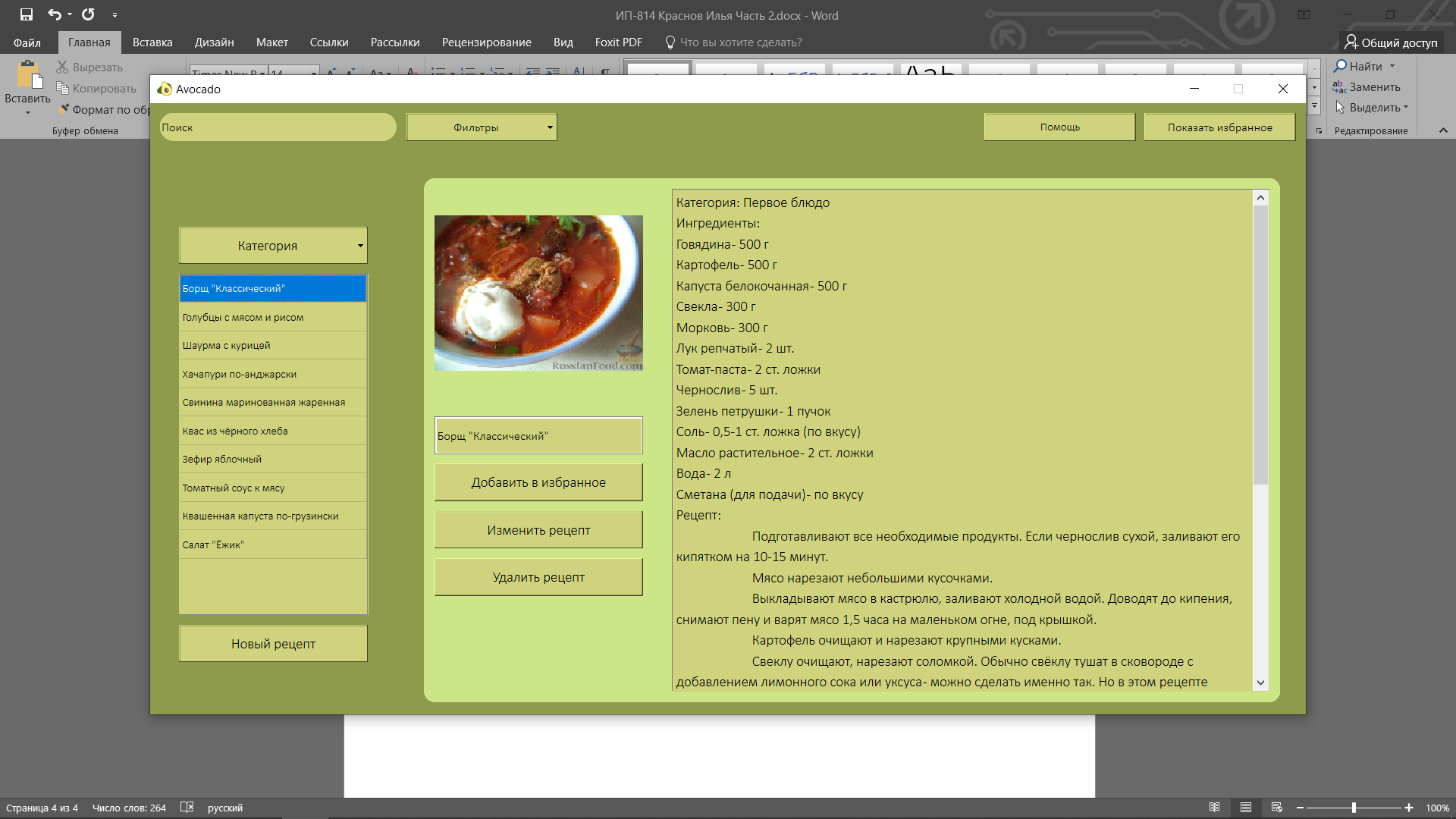
## **Скриншоты работы программы**



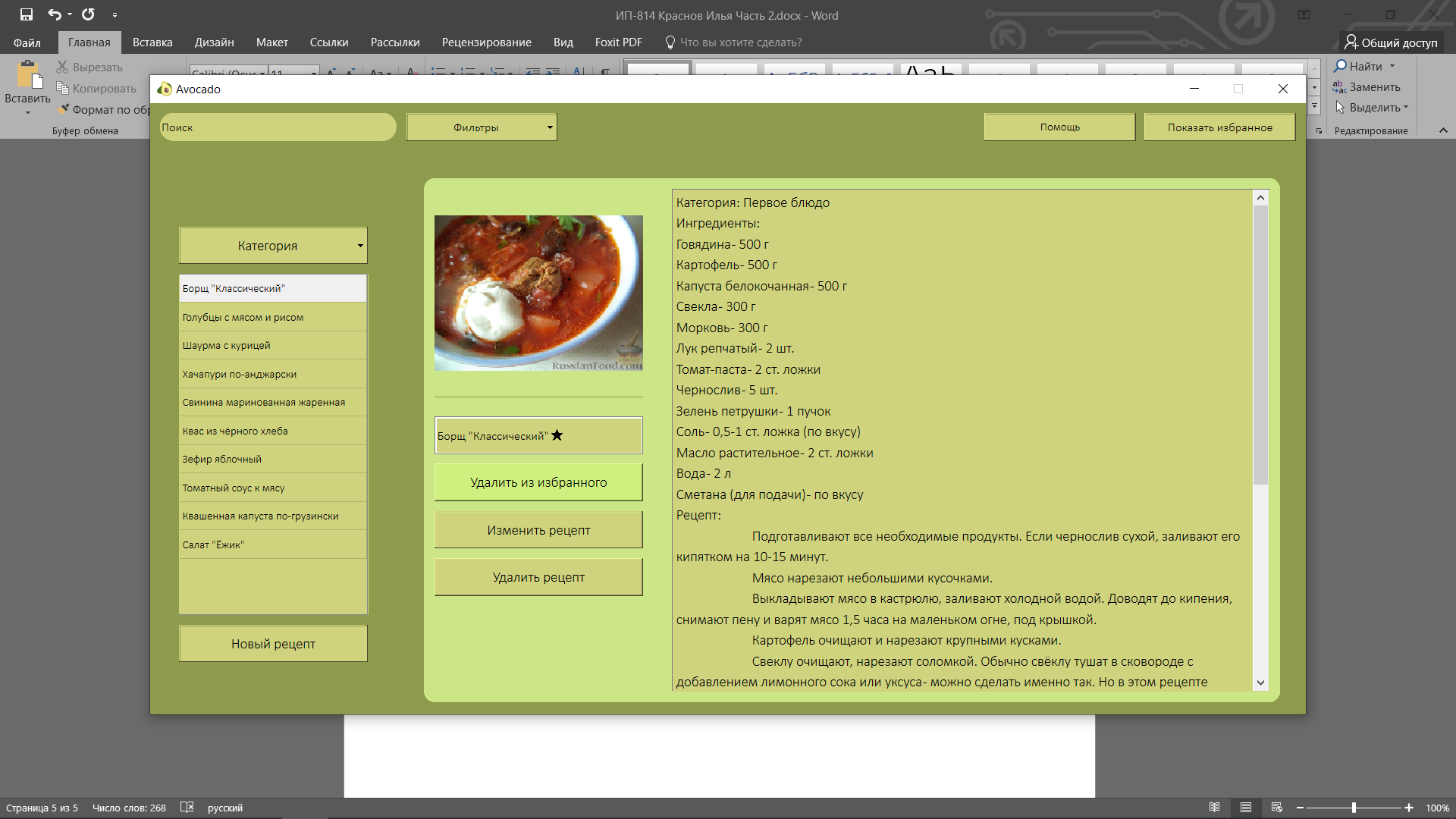
(рис. 1) Окно заставки



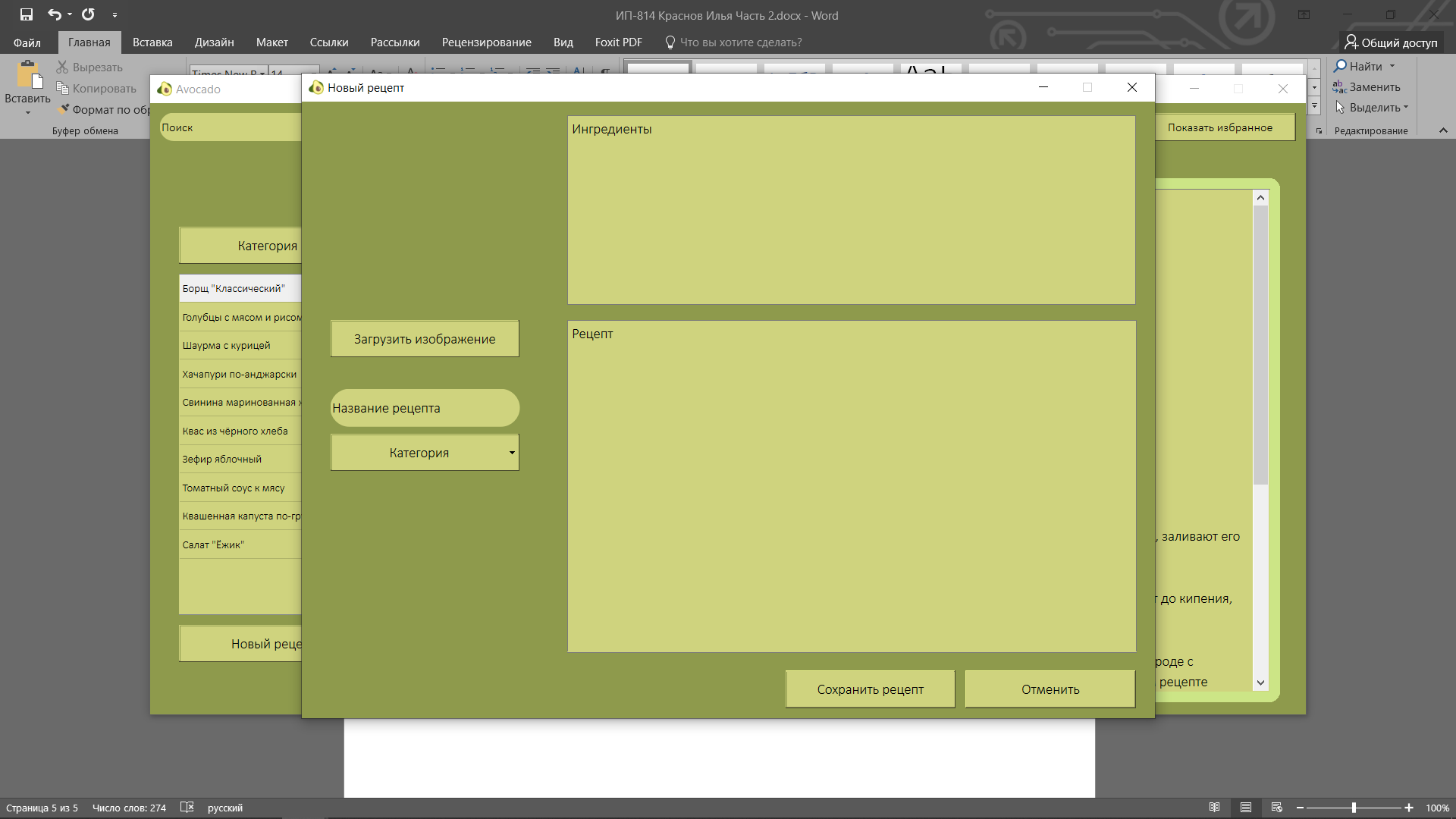
(рис. 2) Главное окно приложения



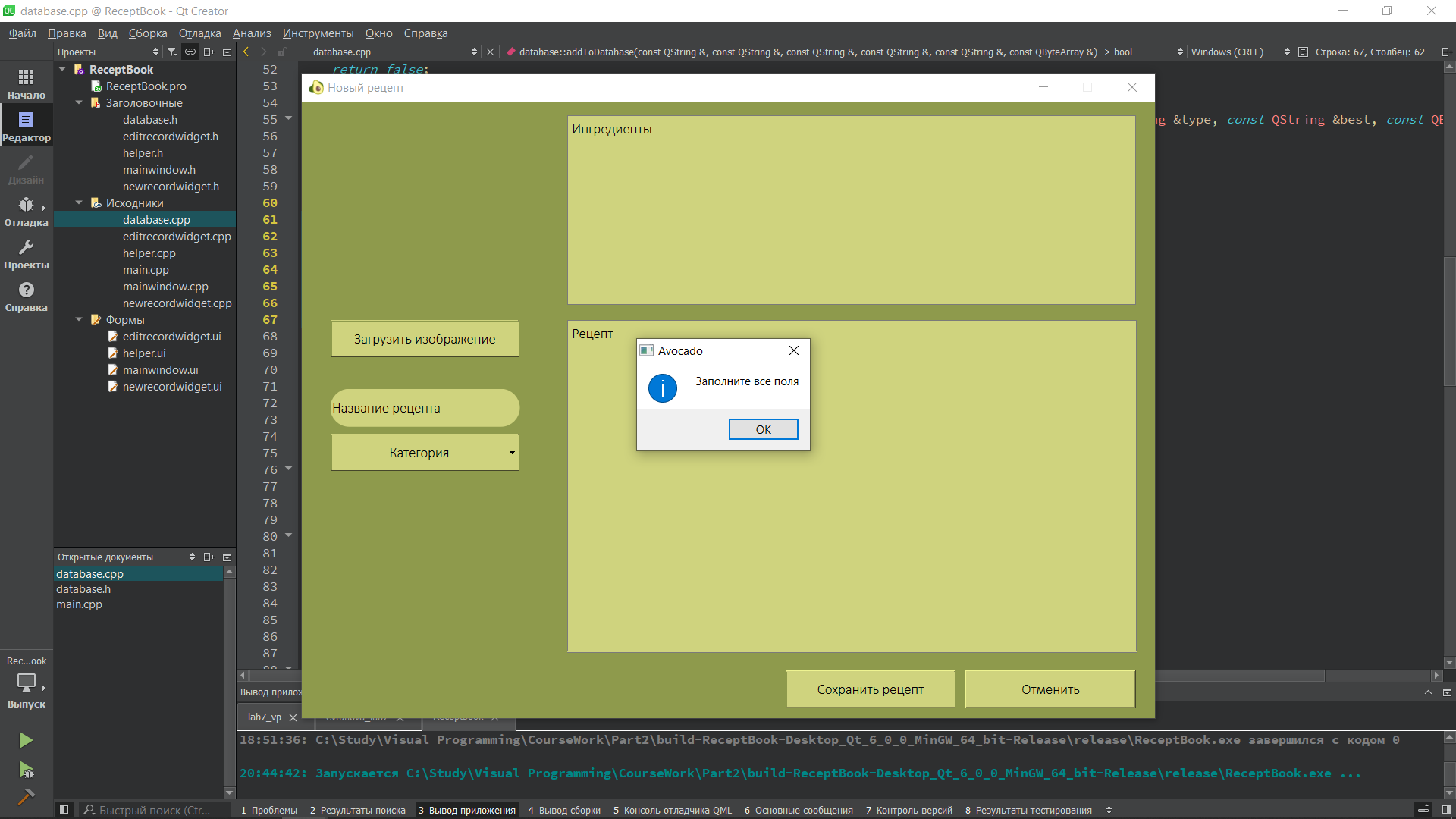
(рис. 3) Просмотр рецепта



(рис. 4) Добавление рецепта в избранное



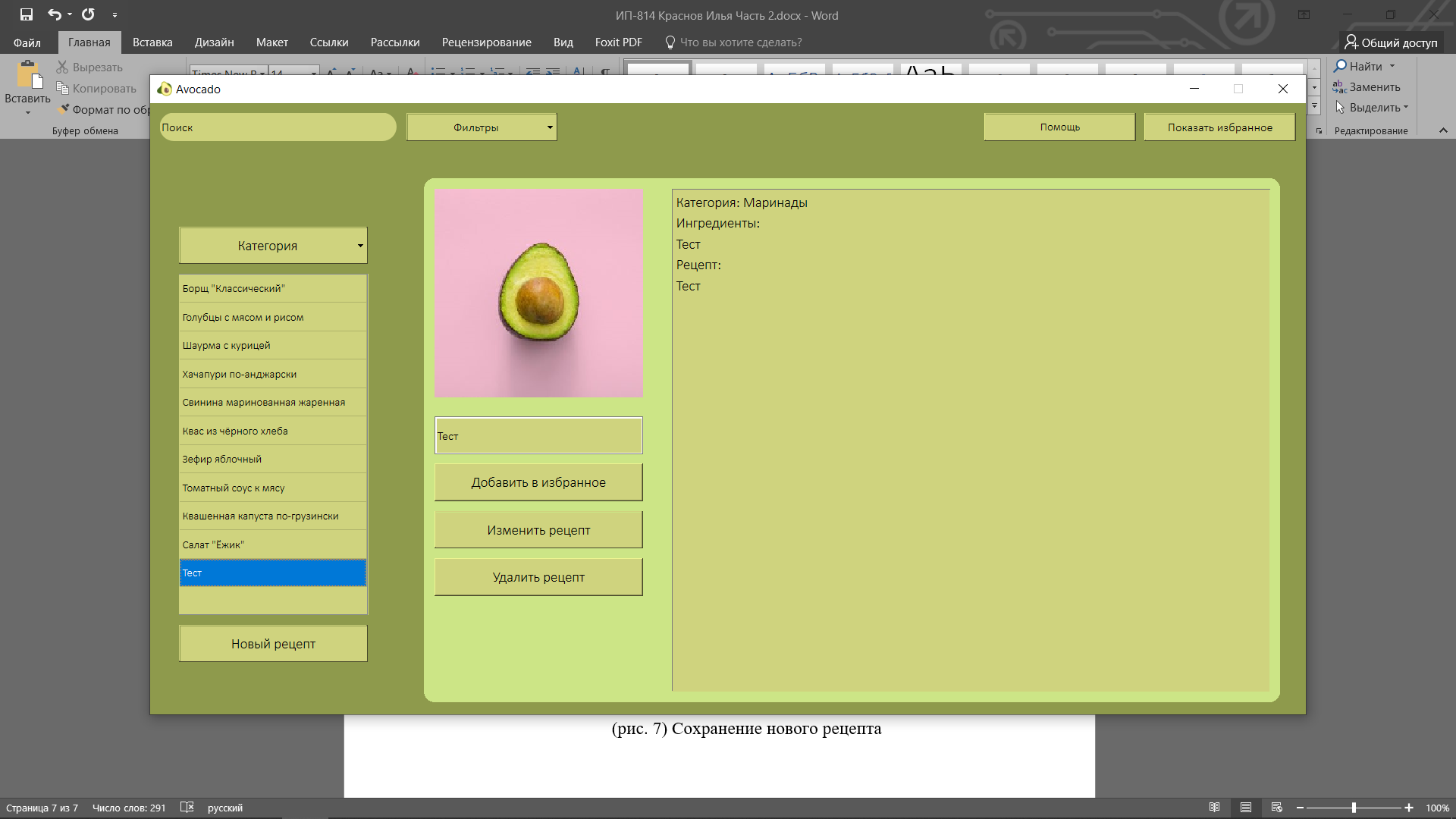
(рис. 5) Окно создания нового рецепта



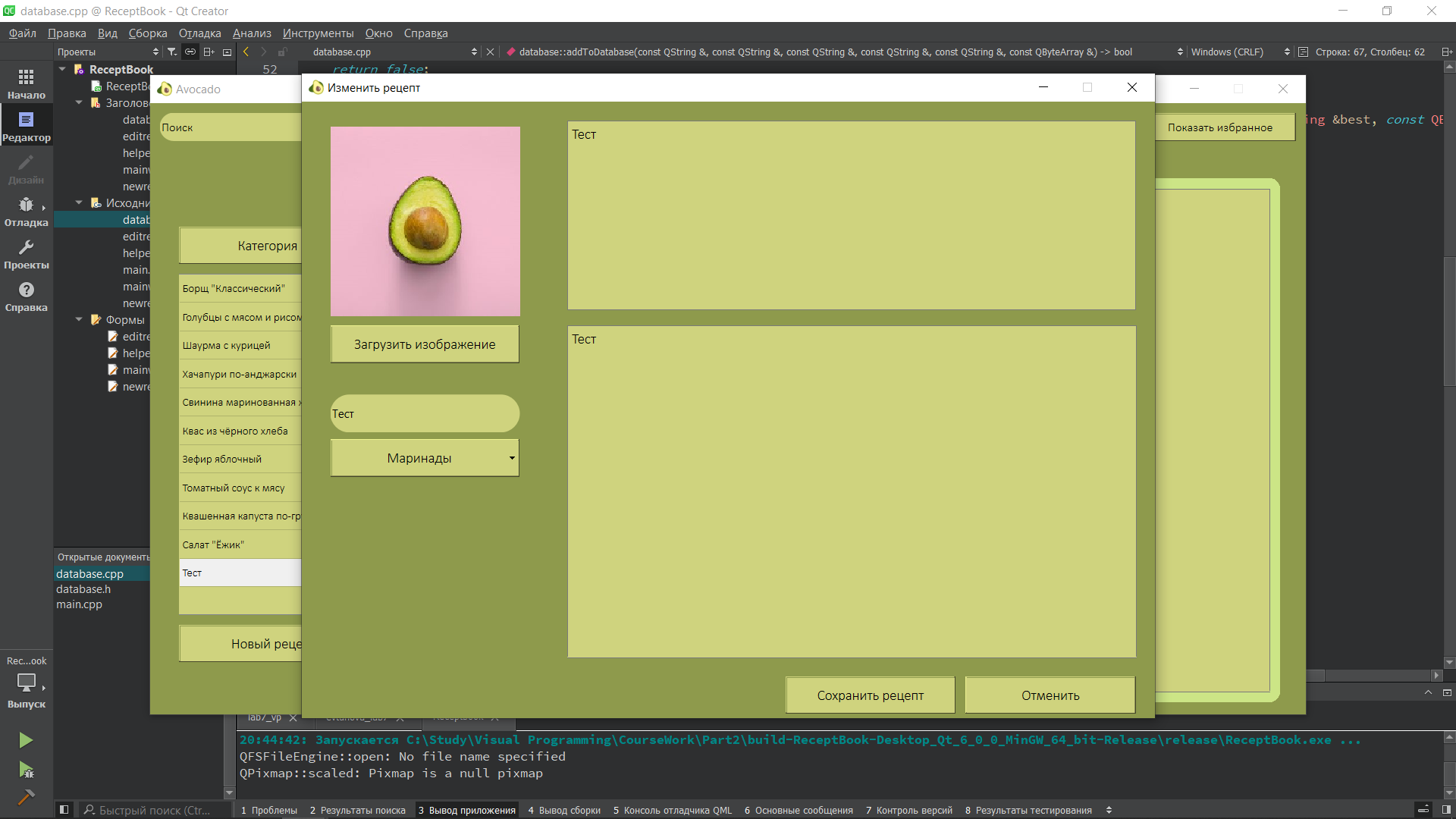
(рис. 6) Отказ сохранить пустой рецепт



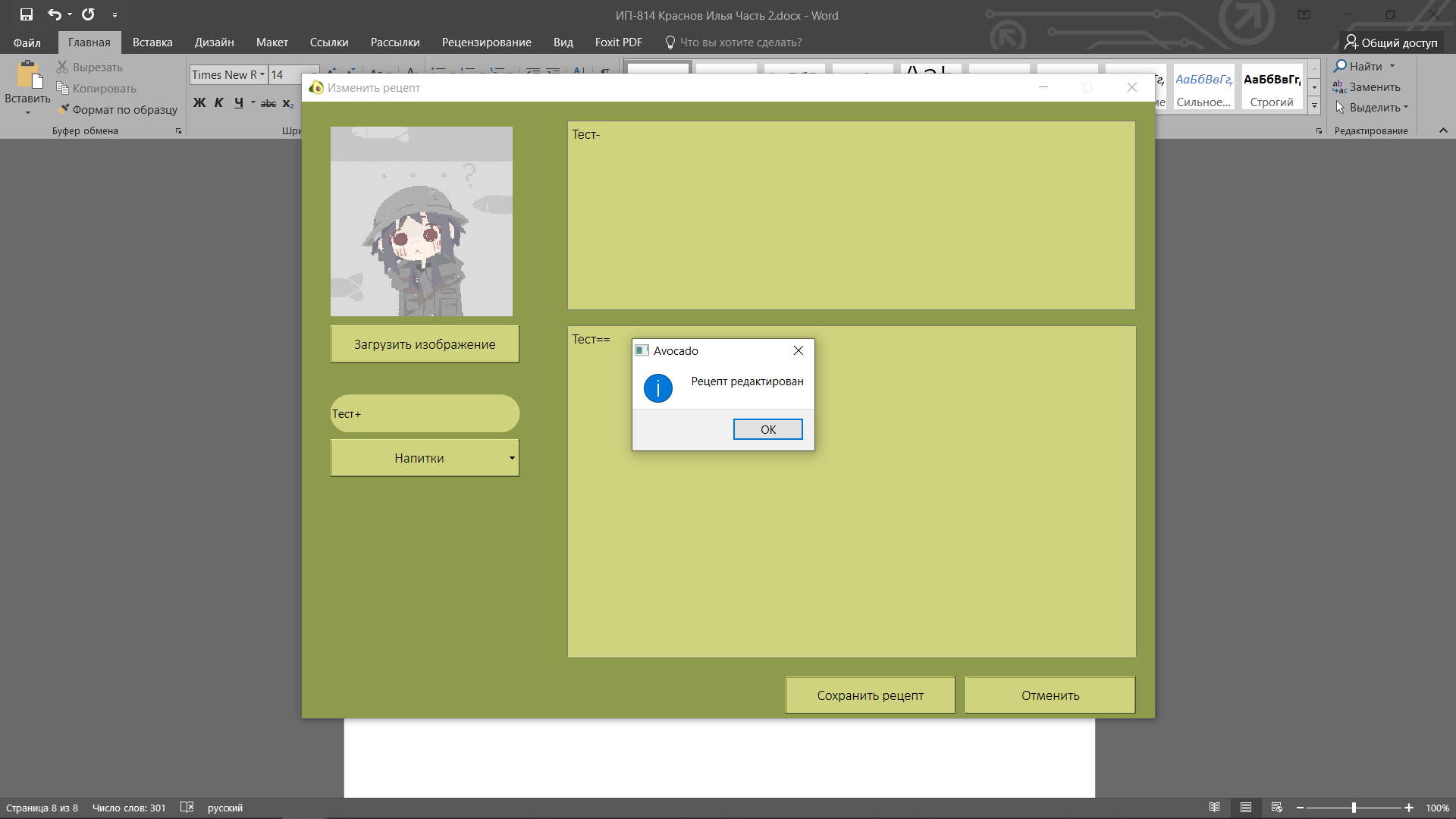
(рис. 7) Сохранение нового рецепта



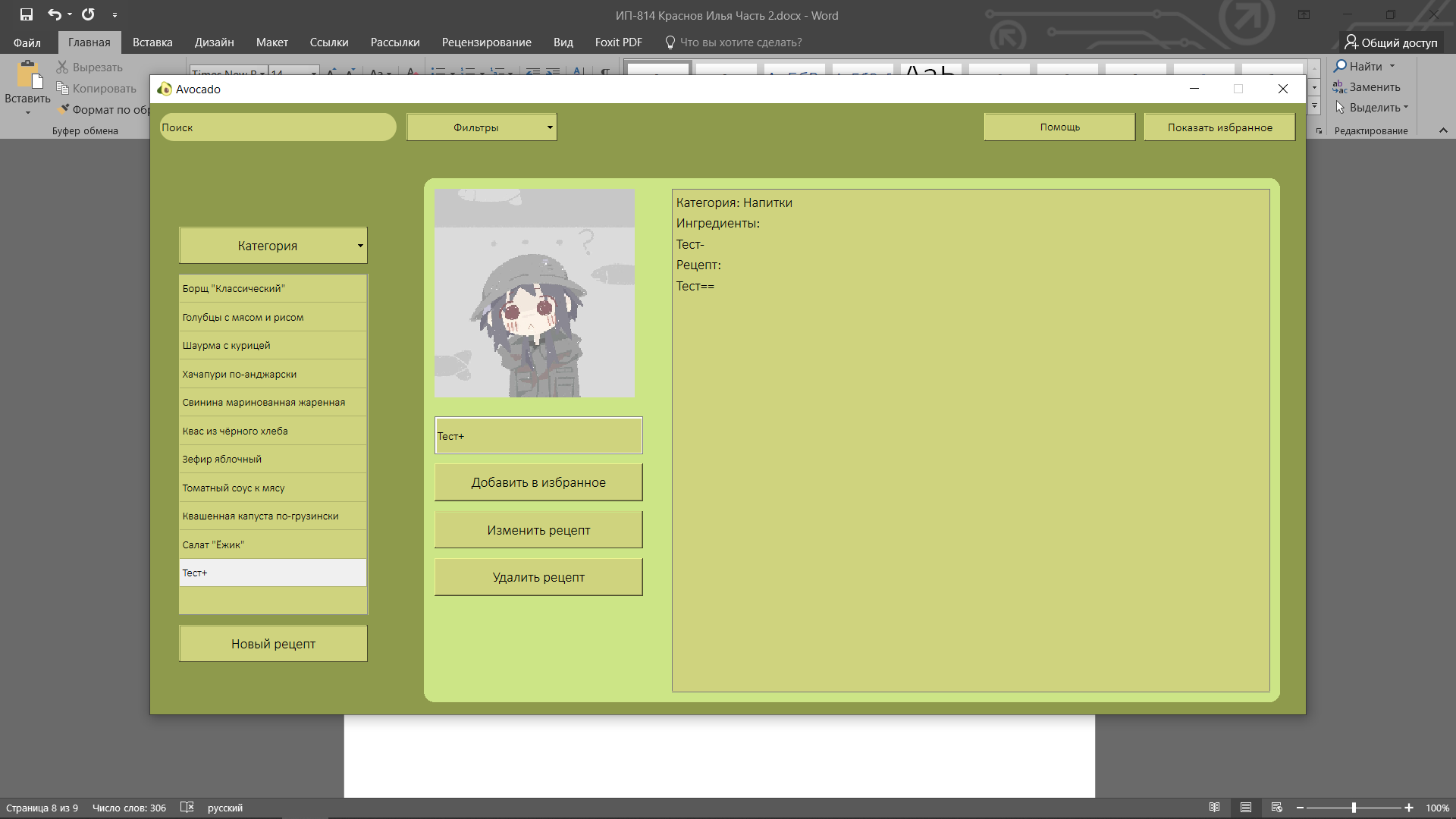
(рис. 8) Просмотр нового рецепта



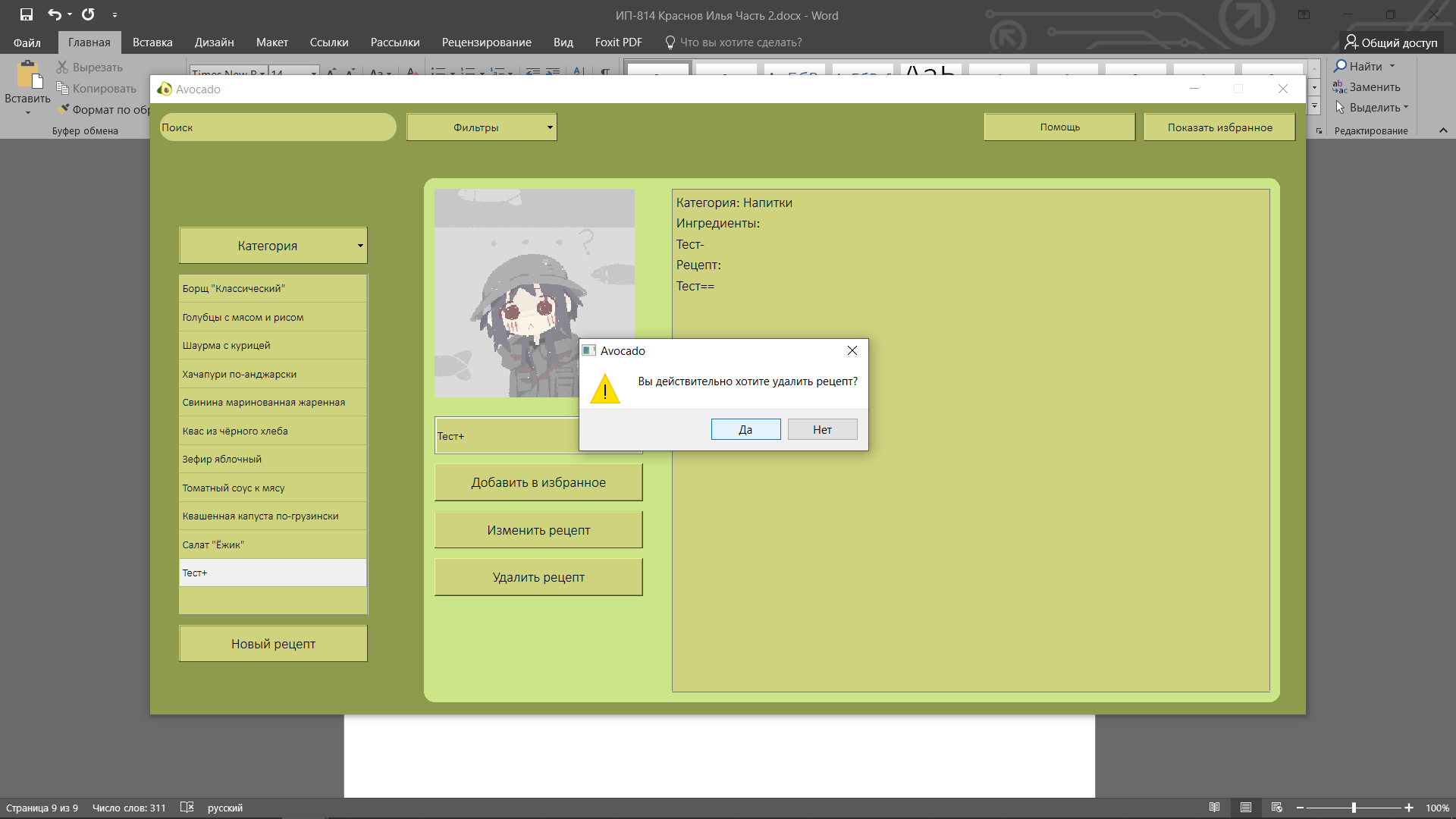
(рис. 9) Окно редактирования рецепта



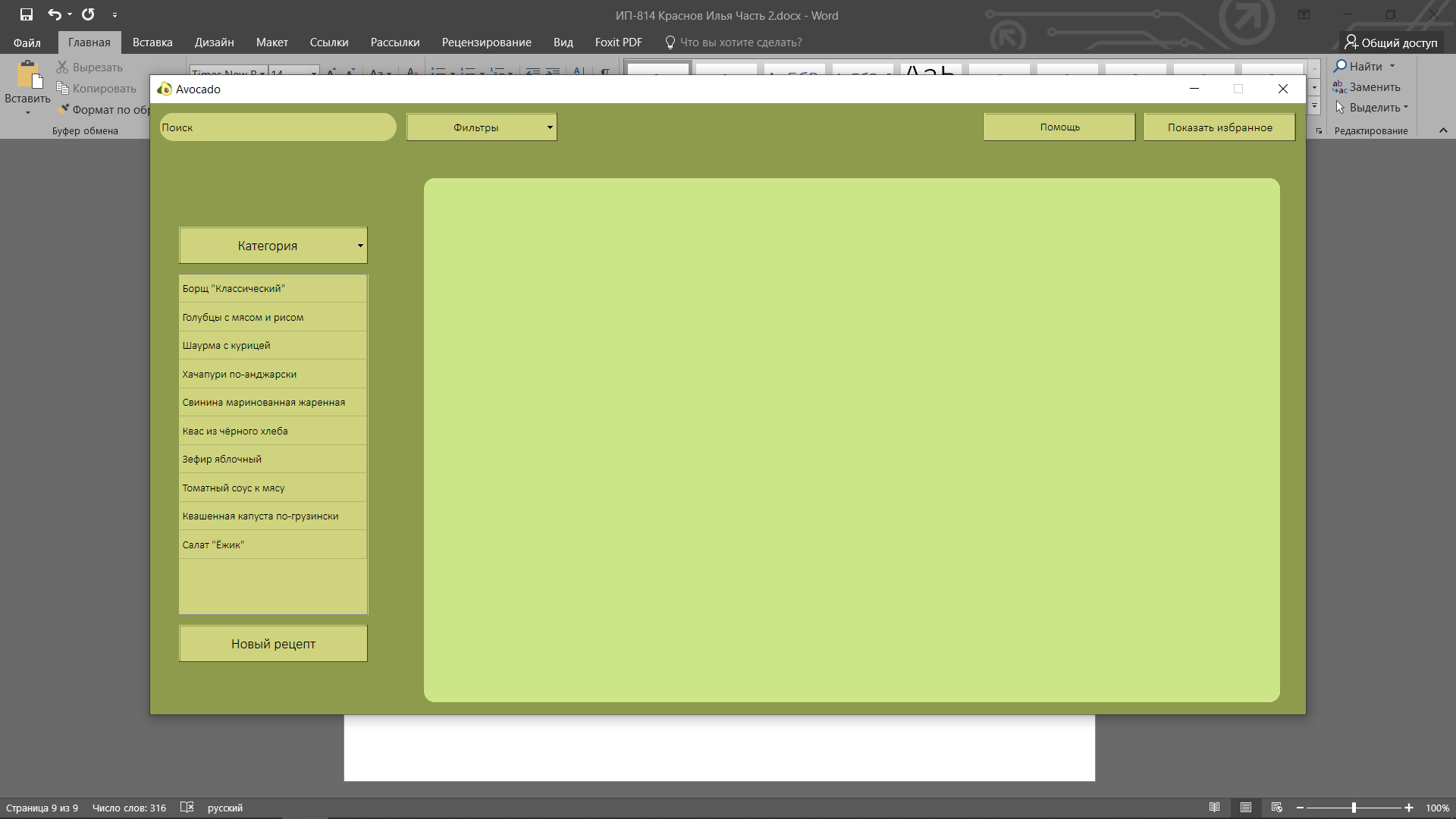
(рис. 10) Сохранение отредактированного рецепта



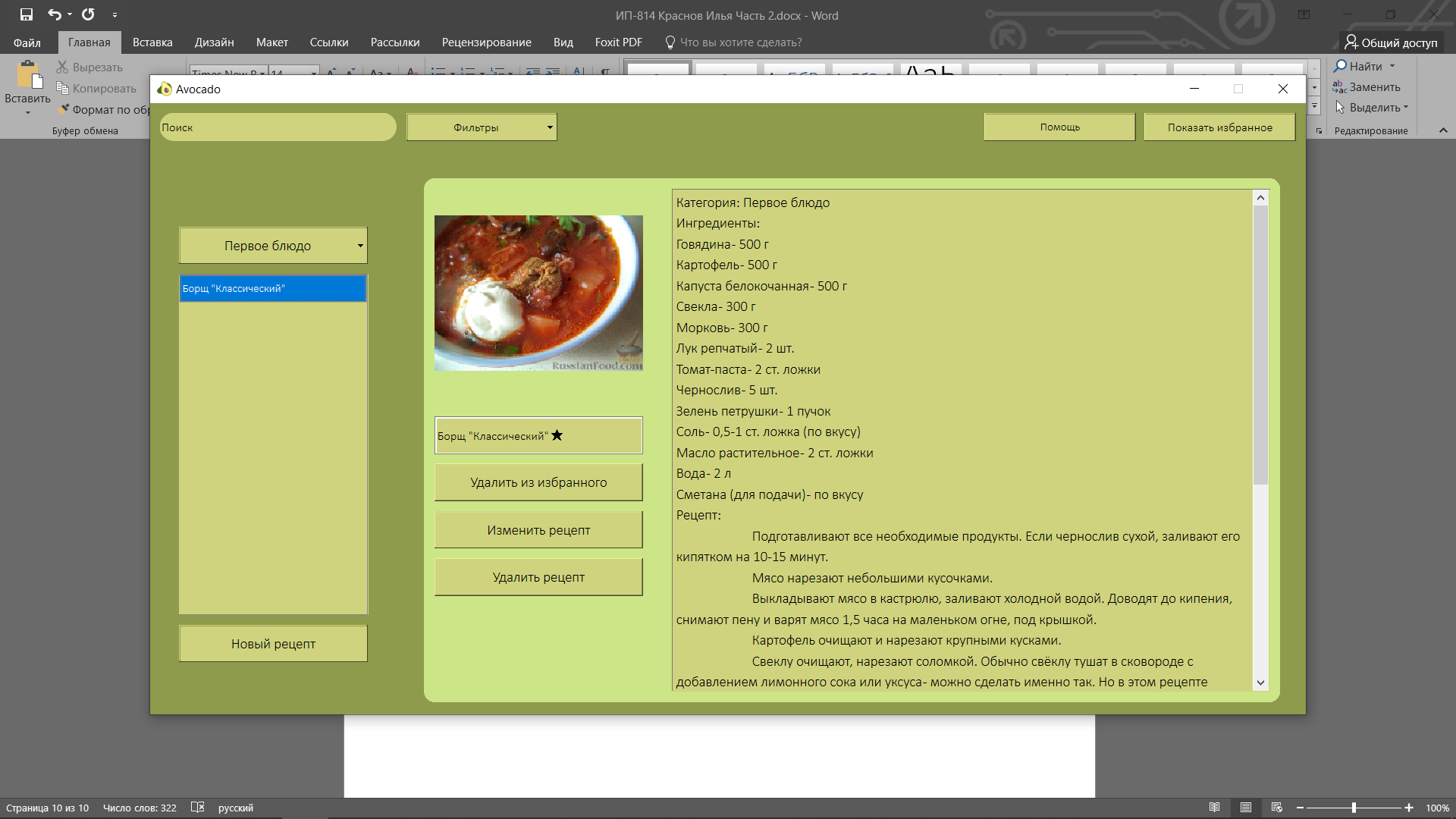
(рис. 11) Просмотр отредактированного рецепта



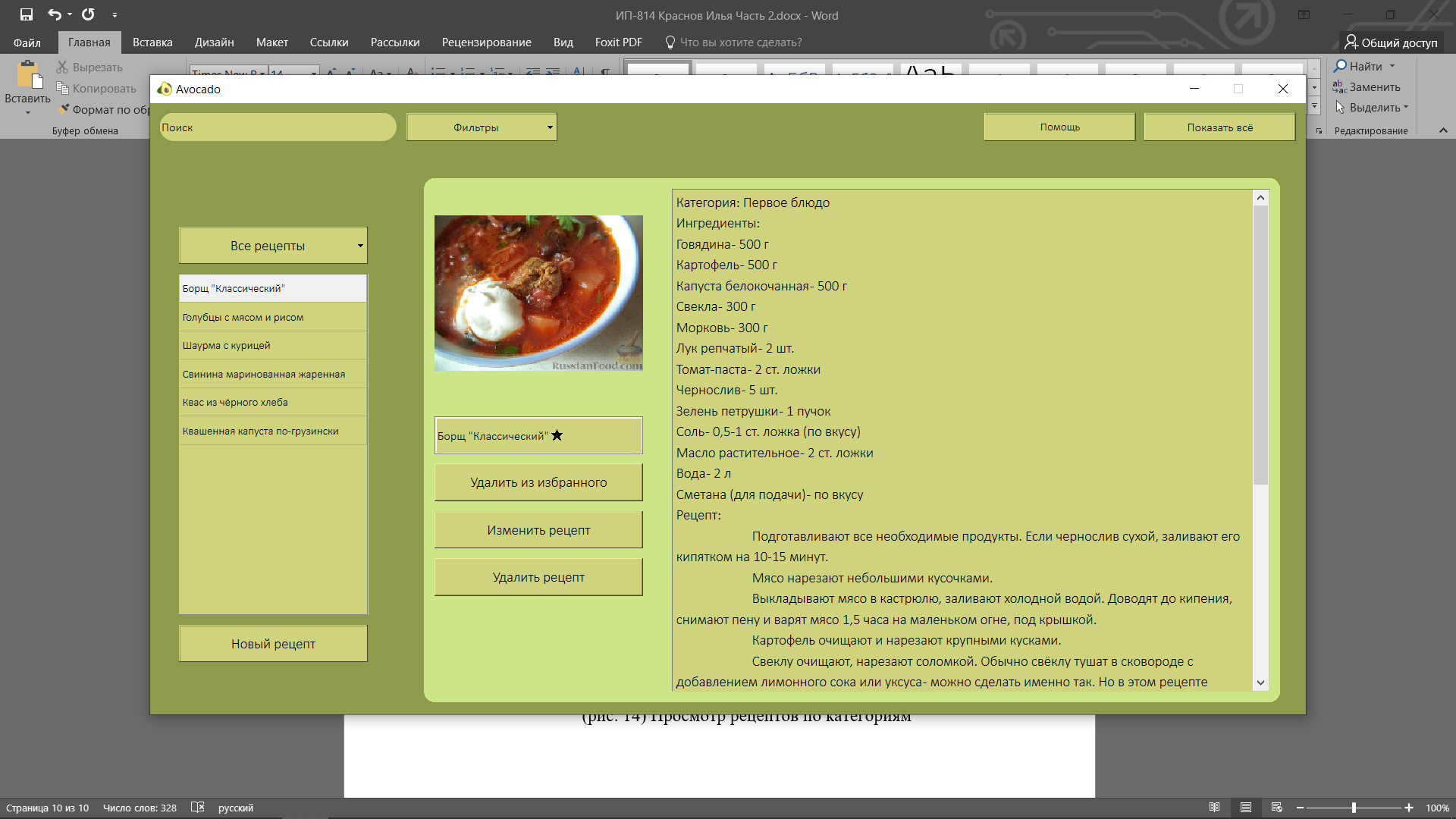
(рис. 12) Подтверждение удаления рецепта



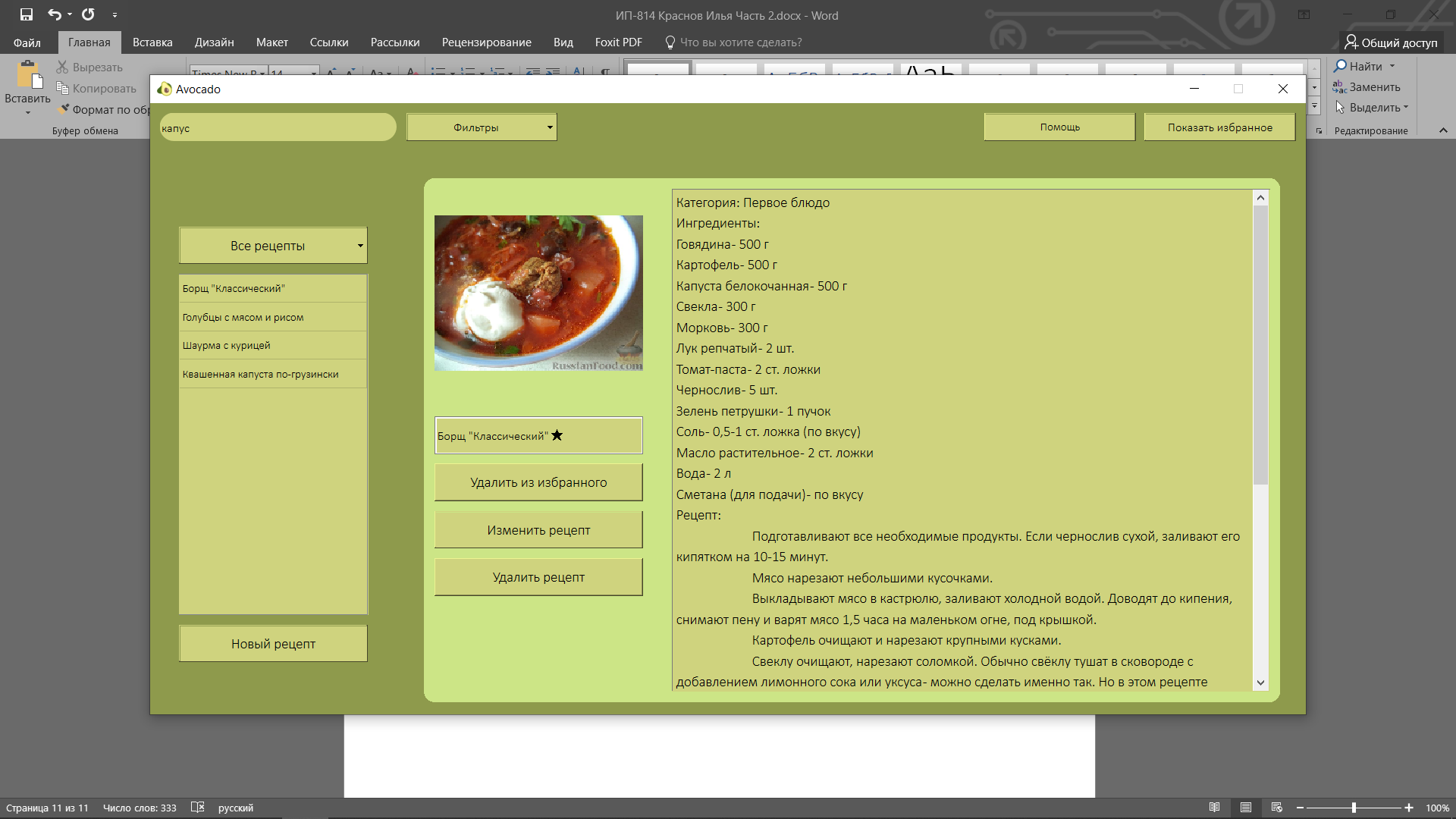
(рис. 13) Окно после удаления рецепта



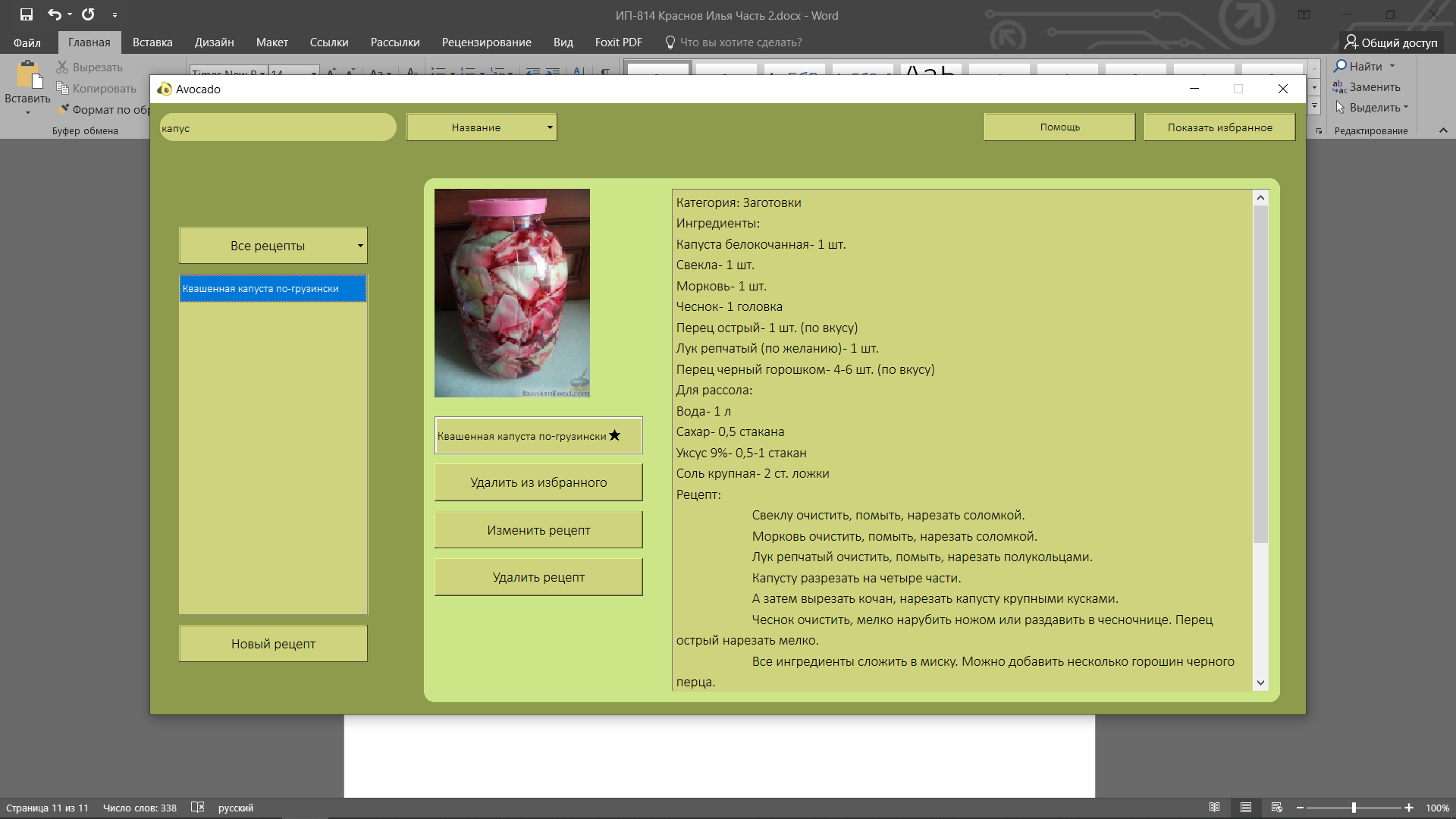
(рис. 14) Просмотр рецептов по категориям



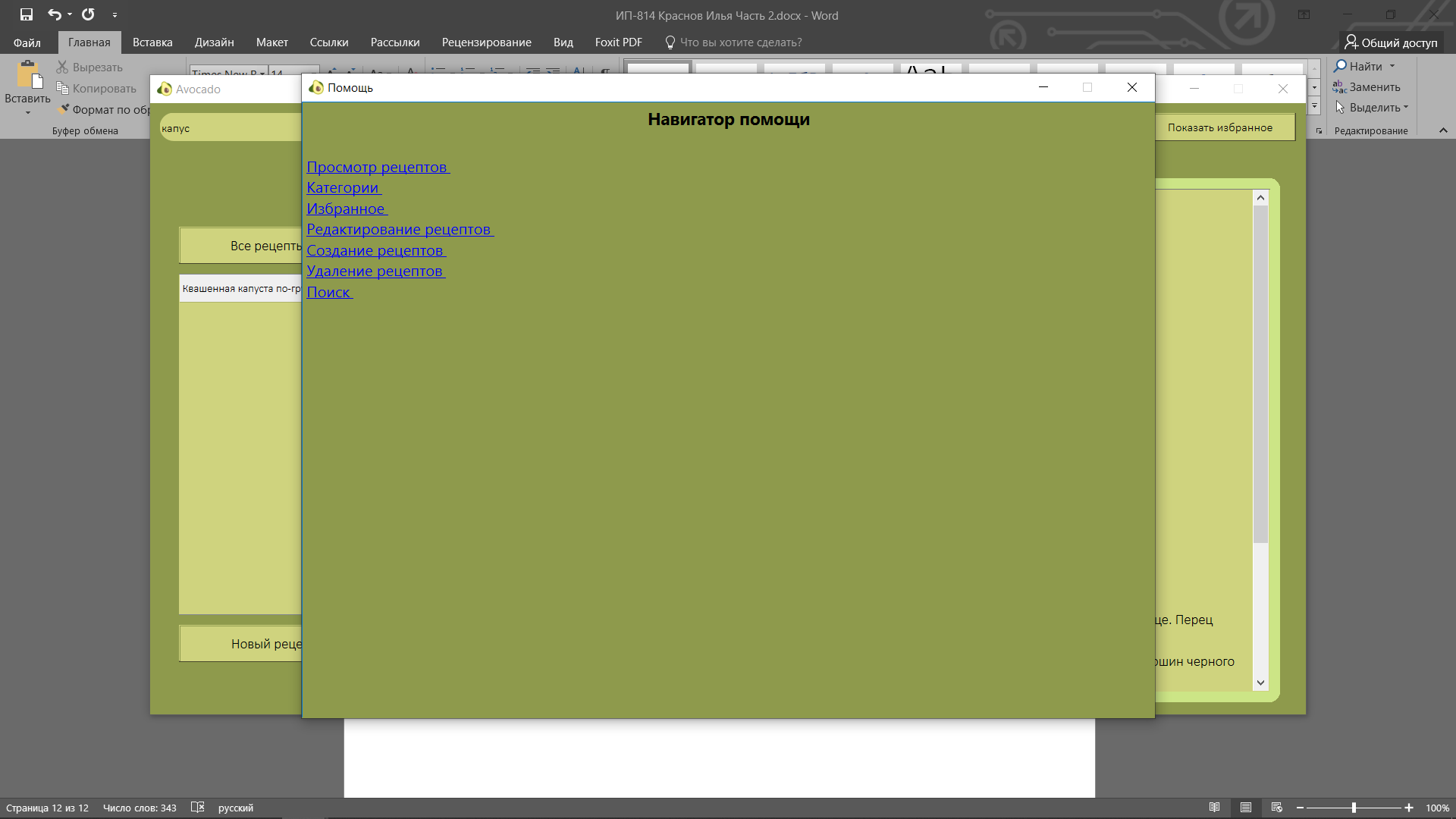
(рис. 15) Просмотр избранных рецептов



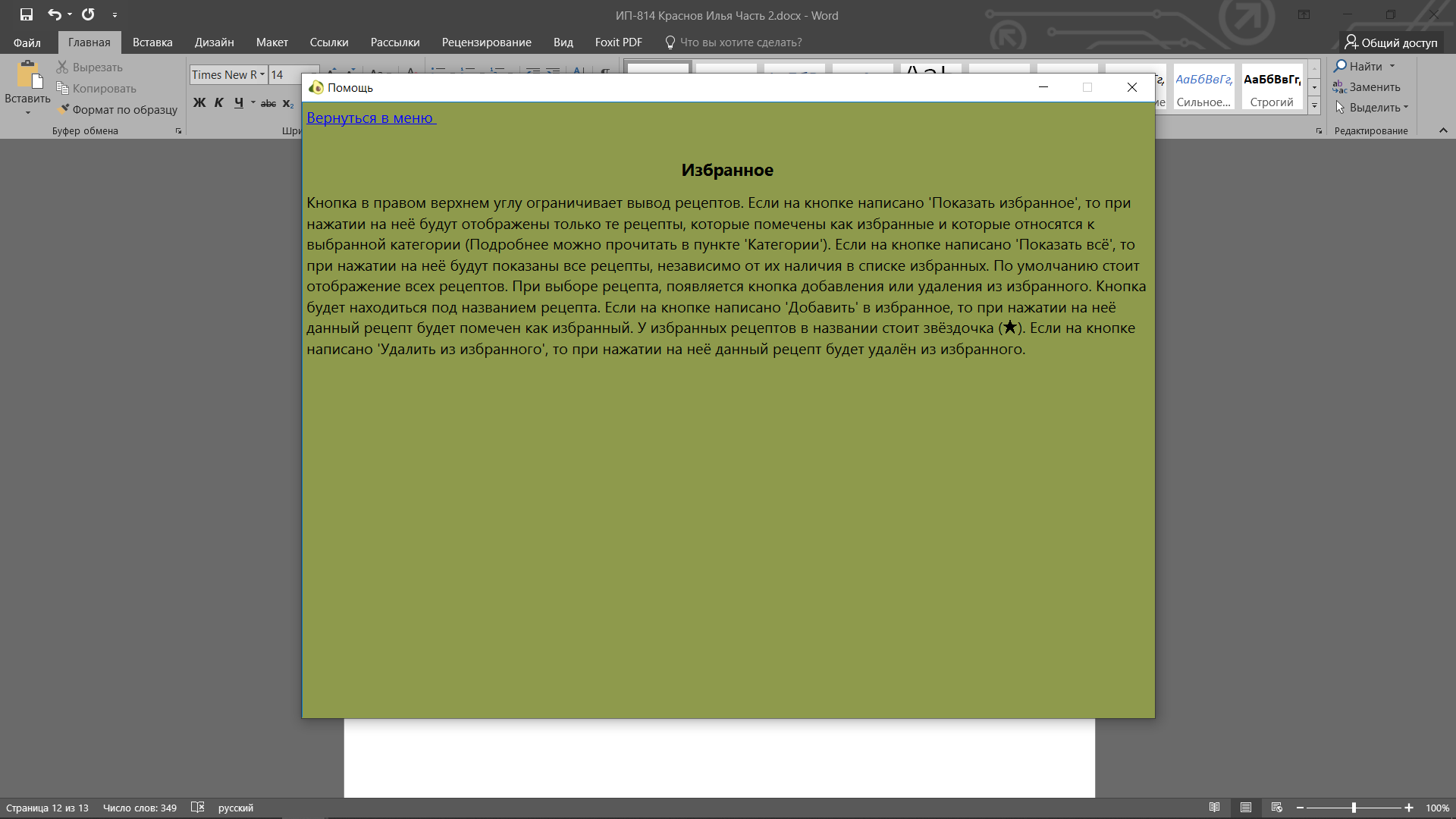
(рис. 16) Поиск без фильтров



(рис. 17) Поиск с фильтром



(рис. 18) Главное меню навигатора помощи



(рис. 19) Просмотр информации в навигаторе помощи

# **Часть 3**

## **CWT-анализ**

### **Постановка задач**

В данном анализе будет разобрано три задачи:

1. Поиск с применением фильтров.
2. Просмотр рецепта.
3. Добавление нового рецепта.

### **Первая задача. Поиск по фильтрам**

Условие задачи: заранее известен только ингредиент. Пользователь хочет найти нужный ему рецепт с этим ингредиентом

В программе предусмотрен поиск по фильтрам, что облегчает выполнение задачи. Алгоритм решения задачи:

1. Установить фильтр «Ингредиенты».
2. Написать ингредиент в строке поиска.

При реализации алгоритма не возникло проблем. Был выставлен нужный фильтр и набран текст. После набора текста были отображены все рецепты, в составе которых есть нужные нам ингредиенты. Трудности возникают при достижении этой цели с немного изменённым алгоритмом. Новый алгоритм таков:

1. Написать ингредиент в строке поиска.
2. Установить фильтр «Ингредиенты».

После набора текста были отображены все совпадения, включая наличие заданных слов в названиях рецептов и в рецептах приготовления. Однако после установки фильтра результат не изменился. Программа обновляет результаты поиска только после изменений текста в строке поиска. Это является первой проблемой данного приложения.

### **Вторая задача. Просмотр рецептов**

Условие задачи: программа была только что открыта. Пользователь хочет открыть рецепт и прочитать ингредиенты, входящие в его состав. Для достижения цели нужно выполнить простой алгоритм:

1. Выбрать рецепт в списке.
2. Изучить информацию о нём и найти ингредиенты.

Первый пункт не вызывает сложности. Нажимаем на рецепт и в поле справа выводится вся информация о нём. Однако текст в поле вывода написан в одном стиле. Имеются пометки о том, какая информация представлена – перед названием категории блюда пишется «Категория», перед ингредиентами пишется «Ингредиент» и т.д. Однако различить эти пометки на общем фоне очень сложно. Будет намного удобнее, если они будут написаны полужирным начертанием или курсивом. Это вторая проблема данного приложения.

### **Третья задача. Создание рецепта**

Условие задачи: пользователь решил создать новый рецепт, написал информацию, однако допустил ошибку и случайно закрыл окно создания рецепта.

Для реализации данного алгоритма, имитирующего ошибку, допущенную пользователем, нужно:

1. Нажать на кнопку «Новый рецепт».
2. Написать информацию в любом из полей.
3. Нажать на кнопку «Отменить» или на крестик в верхнем правом углу окна, чтобы закрыть окно создания рецепта.
4. Нажать на кнопку «Новый рецепт».

Действуем по алгоритму. Нажимаем на кнопку «Новый рецепт». На экране появляется окно с 3 текстовыми полями, кнопкой выбора изображения, кнопкой выбора категории блюда и кнопками сохранения рецепта и отмены создания нового рецепта. Для примера напишем слово «авокадо» в поле ингредиентов. Нажимаем на кнопку «Отменить». Окно закрывается. Ещё раз нажимаем на кнопку «Новый рецепт» и видим, что текстовое поле очищено. Если допустить ошибку при написании большого рецепта, вся информация будет утеряна. Будет намного удобнее, если поля будут очищаться только при сохранении нового рецепта. А если закрыть окно без сохранения, то все данные сохранятся в оперативной памяти, чтобы пользователь мог вернуться к написанию рецепта. Это третья проблема данного приложения.

### **Итог CWT-анализа**

Было выявлено три недостатка и предложены варианты их исправления. Версия приложения на момент анализа – Avocado 1.03.

## **GOMS-анализ**

### **Описание анализа**

Практически все интерфейсные взаимодействия в можно описать следующими операциями:

K – нажатие клавиши;

B – клик кнопкой мыши;

P – наведение указателя мыши;

R – ожидание ответной реакции компьютера;

H – перенос руки с клавиатуры на мышь или наоборот

D – проведение с помощью мыши прямой линии (например, выделение или прокрутка текста);

M – мыслительная подготовка (к осуществлению одной из перечисленных операций).

Разные пользователи выполняют указанные операции за разное время. Однако, GOMS исследует работу опытного пользователя. Многочисленные исследования выявили средние значения времени операций, выполняемых опытными пользователями.

K 0.2 с

B 0.2 с

P 1.1 с

H 0.4 с

M 1.35 с

### **Первая цель**

**Поставим цель:** посмотреть избранный рецепт.

Для достижения цели нужно выполнить следующие подцели:

1. Нажать на кнопку «Показать избранное»
2. Нажать на один из отобразившихся рецептов

Опишем метода для каждой подцели:

1. Нажать на кнопку «Показать избранное»
   1. Кликнуть по кнопке «Показать избранное»

PBR(0,4 с.)

1. Нажать на один из отобразившихся рецептов

2.1) Кликнуть по названию рецепта

PBR(0,4 c.)

С учётом мыслительных операций, получаем следующую последовательность достижения поставленной цели:

MPBR(0,4 c.)

MPBR(0,4 c.)

Время, затраченное на достижение цели:

2 \* M + 2 \* P + 2 \* B + 2 \* R(0,4 c.) = 2,7 + 2,2 + 0,4 + 0,8 = 6,1 c.

Можно сделать так, чтобы имена избранных рецептов помечались в общем списке рецептов. Тогда не придётся нажимать на кнопку для отображения только избранных рецептов, можно будет сразу выбрать избранный рецепт. Тогда формулой для достижения цели будет:

MPBR(0,4 c.)

Время выполнения составит:

M + P + B + R(0,4 c.) = 3,05 c.

Время достижения цели сократится на 3,05 с., что в 2 раза быстрее.

Таким образом с помощью GOMS-анализа было установлено, что при внесении исправлений, цель может быть достигнута в 2 раза быстрее.

### **Вторая цель**

Поставим вторую цель: создать новый рецепт, заполнить все поля 1 символом, выбрать категорию и сохранить новый рецепт.

Для достижения цели необходимо выполнить подцели:

1. Нажать на кнопку «Новый рецепт»
2. Написать по 1 символу в каждом текстовом поле
3. Выбрать категорию
4. Нажать на кнопку «Сохранить рецепт»

Распишем методы для каждой подцели:

1. Нажать на кнопку «Новый рецепт»
   1. Кликнуть а кнопку «Новый рецепт»

PBR(0,4 c.)

1. Написать по одному символу в каждом текстовом поле

2.1) Нажать на текстовое поле имени

PB

2.2) Нажать на любой символ на клавиатуре

K

2.3) Нажать на текстовое поле ингредиентов

PB

2.4) Нажать на любой символ на клавиатуре

K

2.5) Нажать на текстовое поле рецепта

PB

2.6) Нажать на любую кнопку на клавиатуре

K

1. Выбрать категорию

3.1) Нажать на кнопку «Категория»

PB

3.2) Нажать на одну из категорий в выбранном меню

PB

1. Нажать на кнопку «Сохранить рецепт»

4.1) Кликнуть на кнопку «Сохранить рецепт»

PB

С учётом мыслительных операций, последовательность достижения цели примет вид:

MPBR(0,4 c.)

MPB

MK

MPB

MK

MPB

MK

MPB

MPB

MPB

Тогда итоговое время составит:

10 \* M + 7 \* P + 7 \* B + 3 \* K + R (0,4) = 23,6 c.

Если добавить комбинации клавиш для кнопок создания нового рецепта и сохранения, то такие действия как “Нажать на кнопку «Создать рецепт»”(MPBR(0,4 c.) – 3,05) и “Нажать на кнопку «Сохранить рецепт»”(MPB – 2.65 c.) можно будет заменить на «Нажать комбинацию Ctrl+N» (MKKR(0,4 c.) – 2,15 c.) и «Нажать комбинацию Ctrl+S» (MKK – 1,75 c.) соответственно. Тогда будет сэкономлено 1,8 с. и итоговое время составит 21.8 с.

Таким образом с помощью GOMS-анализа было установлено, что достичь цели «Создать новый рецепт с заполненными полями и сохранить его» сожно на 1,8 секунды быстрее.

## **Оценка интерфейса в соответствии с принципами Нильсена-Молиха**

**1.Простой и естественный диалог.** В интерфейсе нет лишних деталей. Пользователь может редактировать информацию только в окне редактирования рецепта и в окне создания нового рецепта, остальные элементы интерфейсы доступны только для нажатия на них.

Вся взаимосвязанная информация группируется и появляется в порядке, который соответствует ожиданию пользователя.

**2. Говорите на языке пользователя.** В программе не используются специфические термины. Всё написано простым языком, который будет понятен человеку, не имеющему опыт в работе с компьютером.

**3. Минимизируйте загрузку памяти пользователя.** Вся информация выводится только тогда, когда пользователь выбирает нужный ему рецепт.

**4. Будьте последовательны.** У пользователей имеется возможность изучить действия в одной части программы и применить их снова, чтобы получить похожие результаты в других местах.

**5. Обеспечьте обратную связь.** Пользователь видит эффект, оказываемый его действиями на систему. Выводятся предупреждающие и информирующие окна.

**6. Обеспечьте хорошо обозначенные выходы.** Пользователь имеет возможность закрыть окна создания и редактирования рецептов, нажав на кнопку «Отменить». Выйти из любого окна можно нажав на крестик на окне. Находясь в навигаторе помощи и просматривая информацию об одном из модулей, пользователь может прекратить просмотр, нажав на кнопку «Вернуться в меню».

**7. Обеспечьте быстрые клавиши ярлыки.** В GOMS-анализе была введена возможность создавать новый рецепты и сохранять изменения в рецептах с помощью комбинаций клавиш Ctrl+N и Ctrl+S соответственно.

**8. Хорошие сведения об ошибках.** При введении данных в формы добавления или редактирования, программа сообщает об отсутствии информации в полях ввода.

**9. Предотвращайте ошибки.** При отсутствии загруженной фотографии, чтобы избежать ошибки в соединении с базой устанавливается изображение по умолчанию.

**10.** **Снабдите программу системой помощи.** Для этого в главном окне приложения находится «Навигатор помощи», где описаны все принципы и нюансы работы программы. Для открытия навигатора нужно нажать на кнопку «Помощь».

## **Соответствие правилам организации графического интерфейса**

**1.Принцип кластеризации**

Все разделы имеют похожую функциональность и похожее расположение кнопок, которые отвечают за одинаковые действия. Окна создания и редактирования рецепта имеют совершенно одинаковый интерфейс, который в свою очередь похож на поле отображение информации из главного окна приложения: слева находится изображение, под ним наименование рецепта, справа от приложения описаны ингредиенты, а под ними рецепт приготовления.

**2. Принцип «видимость отражает полезность»**

В программе все кнопки имеют одинаковый приоритет, и все постоянно используются при ее работе. Поэтому выглядят одинаково. Чтобы программа соответствовала этому принципу, необходимо добавить в главном окне приложения картинки, которые будут отражать суть кнопок, чтобы визуально можно было понять, какая кнопка за что отвечает.

**3.Принцип интеллектуальной последовательности**

Разделы в программе имеют схожий интерфейс и отличаются от предупреждающих сообщений. Программа соответствует этому принципу.

**4.Принцип «цвет как приложение»**

Приложение выполнено в пастельных тонах зелёного и жёлтого, а текст написан чёрным цветом шрифтом Calibri Light, что хорошо выделяет текст ан фоне интерфейса. Цвет кнопок изменяется при наведении на них (кнопки становятся более яркими). Никакую информацию в себе цвет не несет. Поэтому приложение соответствует данному принципу.

**5. Принцип уменьшения беспорядка**

Весь интерфейс выполнен в одинаковом стиле. Вывод информации об рецепте ограничивается одной областью, представляющей из себя прямоугольник с закруглёнными углами. Информация в таблице с названиями рецептов вмещает в себя все названия. Приложении соответствует данному принципу.

## **Список выявленных и исправленных проблем**

По результатам CWT-анализа:

* Исправлена ошибка в модуле поиск, который начинал работать только при изменении текста в строке поиска. Теперь поиск идёт и при изменении фильтра.
* Добавлено выделение текста, обозначающего название раздела в поле с информацией о рецепте.
* Убрано удаление информации из полей в окне создания рецепта при закрытии окна

По результатам GOMS-анализа:

* Добавлены отметки для избранных рецептов в поле выбора рецепта
* Добавлены комбинации клавиш для создания нового рецепта и для сохранения рецепта

Принципы Нильсена-Молиха:

Не было выявлено недостатков.

Принципы организации графического интерфейса:

* Добавлены изображения для кнопок и поля поиска для визуализации задач, которые выполнят эти объекты

# **Выводы**

В данной курсовой работе были использованы практические и теоретические знания, полученные в процессе прохождения курса «Визуальное программирование и человеко-машинное взаимодействие». Приложение с графическим интерфейсом «Avocado» написано с помощью фреймворка Qt в среде разработки Qt Creator 6.0.0.

Данное приложение позволяет просматривать рецепты, создавать новые, редактировать и удалять их. В приложении предусмотрено разделение всех блюд на категории по их типу. В приложении есть возможность отмечать рецепты как избранные. Для быстрого доступа к рецепту в приложении предусмотрен поиск с использованием фильтров.

В процессе реализации курсового проекта были пройдены все этапы создания итогового продукта. Были поставлены задачи для реализации и описан черновой дизайн будущего приложения. Было реализовано приложение в соответствии с поставленными задачами. Были проведены анализы приложения, и исправлены выявленные недостатки.

В процессе реализации курсового проекта были пройдены все этапы создания итогового продукта. Были поставлены задачи для реализации и описан черновой дизайн будущего приложения. Было реализовано приложение в соответствии с поставленными задачами. Были проведены анализы приложения и исправлены выявленные недостатки.

Таким образом, мне удалось выполнить поставленные перед началом создания курсового проекта цели.

# **Ссылки**

* DB Browser for SQLite – использовалось для создания базы данных, её просмотра в процессе разработки проекта и ручного редактирования

URL: <https://sqlitebrowser.org/>

* Qt Documentation – документация по разработке в среде Qt

URL: <https://doc.qt.io/>

* Репозиторий проекта, в котором можно отследить процесс разработки и текущее состояние проекта

URL: <https://github.com/Krasnoffsky/Study/tree/master/Visual%20Programming/CourseWork>