|  |  |  |  |
| --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН  СЕВЕРО-КАЗАХСТАНСКИЙ УНИВЕРСИТЕТ ИМ. М. КОЗЫБАЕВА | | | |
|  | | | |
|  | | |  |
| СЕМЕСТРОВЫЙ ПРОЕКТ  На тему: «Разработка информационной системы  «Книга рецептов» | | | |
| Выполнили студенты  группы АПО-20-2 |  | Ставская Е.А.  Толстов В.Ю. | |
| Доктор PHD,  доцент кафедры ИКТ |  | Астапенко Н.В. | |
|  |  |  | |

Петропавловск, 2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc133490884)

[Аналитическая часть: 4](#_Toc133490885)

[1.1 Характеристика/описание объекта исследования. 4](#_Toc133490886)

[1.2 Обзор аналогичных проектов. 4](#_Toc133490887)

[1.3 Техническое задание на разработку системы «Книга рецептов» на VBA. 7](#_Toc133490888)

[1.4 Функциональное моделирование (IDEF0, IDEF3). 9](#_Toc133490889)

[2 Проектная часть 15](#_Toc133490890)

[2.1 Проектирование информационных потоков (DFD). 15](#_Toc133490891)

[2.2 Проектирование средствами UML (6 видов диаграмм). 17](#_Toc133490892)

[2.3 Проектирование БД (инфологическое проектирование – ERD; даталогическое проектирование – описание таблиц и схемы данных). 28](#_Toc133490893)

[2.4 Проектирование пользовательских интерфейсов 30](#_Toc133490894)

[ЗАКЛЮЧЕНИЕ 40](#_Toc133490895)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 41](#_Toc133490896)

# ВВЕДЕНИЕ

Современный образ жизни требует от людей быстрого и эффективного выполнения различных задач, включая приготовление пищи. Однако, часто возникает проблема выбора блюда для приготовления на основе доступных продуктов. В связи с этим, разработка приложения для подбора рецептов из базы данных является актуальной задачей. В рамках данного проекта разработано приложение на VBA Excel под названием «Книга рецептов», которое позволяет пользователю подбирать рецепты по заданным продуктам, а также добавлять, изменять и удалять рецепты.

Целью исследования является разработка приложения «Книга рецептов», которое позволит пользователям быстро и легко находить подходящие рецепты на основе введенных ими продуктов, а также добавлять свои собственные рецепты в базу данных. В приложении предусмотрена возможность пометить рецепт как приготовленный, а также добавить его в список избранных. Для каждого рецепта установлен уровень сложности, который используется для расчета ранга пользователя. Имя пользователя автоматически определяется по имени ПК. Кроме того, в приложении реализована статистика и возможность получения сертификата со своим рангом.

Для достижения данной цели были поставлены следующие задачи:

* Разработать базу данных рецептов в Excel;
* Создать интерфейс приложения, который позволяет пользователю вводить продукты и получать список подходящих рецептов;
* Реализовать функционал добавления, изменения и удаления рецептов в базе данных;
* Разработать систему ранжирования пользователей на основе уровня сложности рецептов, приготовленных ими;
* Разработать возможность добавления рецептов в избранное;
* Разработать статистику приготовленных рецептов и остаток очков до следующего ранга, возможность получения сертификата со своим рангом.

Объектом исследования данного проекта является программа на VBA Excel для автоматизации подбора рецептов из базы данных. Предметом исследования являются функциональные возможности программы, ее эффективность и удобство использования для конечного пользователя.

В процессе работы были использованы следующие методы исследования: анализ литературы по теме, изучение аналогов, разработка базы данных, программирование на языке VBA Excel.

Практическая значимость данной работы заключается в создании удобного и функционального приложения для подбора рецептов, что поможет пользователям быстро и легко выбрать рецепт на основе доступных продуктов, а также управлять своей базой данных рецептов.

# Аналитическая часть:

# 1.1 Характеристика/описание объекта исследования.

Целью проекта является разработка приложения на VBA Excel. Приложение имеет простой и интуитивно понятный интерфейс, который позволяет пользователям быстро и легко находить нужные рецепты.

Приложение имеет следующие особенности:

* + - * Подбор рецептов из базы данных (листы Excel) по введенным пользователем продуктам;
      * Возможность добавления, изменения и удаления рецептов в базе данных;
      * Пометка рецепта как приготовленного;
      * Уровень сложности рецепта, который влияет на ранг пользователя;
      * Возможность добавления рецепта в избранное;
      * Имя пользователя задается автоматически по имени ПК;
      * Просмотр статистики и получение сертификата со своим рангом;
      * Приложение предназначено для использования в домашних условиях. Оно позволяет пользователю не только найти подходящие рецепты на основе имеющихся продуктов, но и добавлять новые рецепты, изменять уже существующие или удалять ненужные рецепты.

Каждый рецепт имеет уровень сложности, который определяет его ранг в системе. Чем выше уровень сложности рецепта, тем больше очков опыта пользователь получит за его приготовление.

Приложение также позволяет пользователю добавлять рецепты в избранное, чтобы быстро находить их в будущем. Имя пользователя автоматически устанавливается по имени ПК, что упрощает использование приложения для всех пользователей.

Просмотр статистики и получение сертификата со своим рангом позволяет пользователям отслеживать свой прогресс.

Таким образом, приложение "Книга рецептов" на VBA Excel позволяет пользователям быстро находить и сохранять рецепты, повышать свой ранг в системе и отслеживать свой прогресс.

# 1.2 Обзор аналогичных проектов.

Был проведен анализ аналогов

* Yummly

Yummly – это приложение для поиска и подбора рецептов, которое было разработано с учетом различных параметров, таких как ингредиенты, тип блюда, время приготовления и диетические ограничения. Пользователи могут искать рецепты по категориям, добавлять их в список «любимых», а также делиться ими с другими пользователями. В приложении также есть функция создания покупочного списка, позволяющая пользователям выбирать необходимые продукты для приготовления блюда.

* Cookpad

Cookpad – это приложение для поиска, создания и хранения рецептов, а также для общения и обмена рецептами с другими пользователями. Пользователи могут создавать свой профиль, настраивать свои предпочтения в питании, добавлять рецепты, сохранять любимые рецепты, отмечать их как попробованные, а также делиться своими рецептами и опытом с другими пользователями. Кроме того, приложение предлагает пользователю возможность подбора рецептов в зависимости от определенных параметров, например, доступных ингредиентов или времени приготовления. Cookpad имеет большую базу данных рецептов, которые постоянно обновляются и дополняются пользователями, что позволяет найти рецепты для любого вкуса и предпочтения в питании.

* Allrecipes

Allrecipes – это онлайн-ресурс, предназначенный для поиска, создания и обмена рецептами. Он предоставляет пользователям множество функций, позволяющих легко найти рецепты, которые соответствуют их предпочтениям и диетическим ограничениям. В Allrecipes есть функция поиска, где можно искать рецепты по ключевым словам, типу блюда, кухне, времени приготовления, ингредиентам и другим параметрам.

Пользователи также могут создавать свои собственные рецепты и делиться ими с другими пользователями. Allrecipes также имеет функцию "сохранить в избранное", что позволяет пользователям сохранять любимые рецепты для будущего использования.

В приложении Allrecipes также есть сообщество пользователей, где пользователи могут общаться, делиться своими рецептами, задавать вопросы и получать советы от других участников сообщества. Кроме того, Allrecipes предоставляет пользоваться рекомендательной системой, которая на основе предпочтений пользователя рекомендует новые рецепты.

Allrecipes имеет современный и простой в использовании интерфейс, что позволяет пользователям быстро и легко находить и создавать рецепты.

* Tasty

Tasty – это приложение, предоставляющее видео-рецепты на различные блюда. Оно позволяет пользователям искать рецепты по категориям, таким как «завтрак», «обед» или «ужин», а также по типу блюда, например, «супы» или «десерты». Кроме того, пользователи могут сохранять понравившиеся рецепты в избранное и делиться ими со своими друзьями через социальные сети. Одной из особенностей приложения является возможность просмотра видео-инструкций по готовке блюд, что делает процесс приготовления более наглядным и понятным. Также приложение предоставляет подробную информацию о каждом рецепте, включая список необходимых ингредиентов и пошаговую инструкцию по приготовлению.

* SideChef

SideChef – это приложение для поиска, создания и хранения рецептов, которое предлагает пользователям подробные инструкции для приготовления блюд. Одной из ключевых особенностей SideChef является возможность автоматического добавления ингредиентов в список покупок на основе выбранных рецептов.

Пользователи могут искать рецепты по категориям, таким как тип блюда, тип кухни, ингредиенты, диетические ограничения и т.д. Также возможен поиск по названию рецепта или ключевому слову.

Приложение предлагает видео-уроки и подробные фотоинструкции для приготовления блюд. Кроме того, SideChef обладает функционалом голосового управления и подключается к умным домашним устройствам, таким как Amazon Echo и Google Home.

С помощью SideChef пользователи также могут создавать свои собственные рецепты и делиться ими с другими пользователями. Кроме того, приложение позволяет пользователям добавлять заметки и отмечать рецепты как избранные для более удобного доступа.

* Paprika

Приложение Paprika представляет собой полнофункциональный менеджер рецептов, который позволяет пользователям искать, сохранять и создавать свои собственные рецепты. В Paprika можно организовывать свою коллекцию рецептов и сортировать их по категориям, а также создавать списки покупок на основе ингредиентов, необходимых для приготовления выбранных блюд. Приложение поддерживает импорт рецептов из различных источников, таких как сайты, блоги и даже сканы рецептов из бумажных книг. Оно также синхронизируется между различными устройствами, что позволяет пользователям иметь доступ к своим рецептам и спискам покупок в любое время и в любом месте.

Каждое из этих приложений имеет свои преимущества и недостатки в зависимости от потребностей и предпочтений пользователя. Некоторые из них предоставляют больше функций для социального взаимодействия, а другие – больше возможностей для персонализации рецептов. После исследования аналогов был разработан концепт, согласно которому приложение должно быть максимально простым, и содержать простые рецепты, с целью охватить больший сегмент пользователей.

# 1.3 Техническое задание на разработку системы «Книга рецептов» на VBA.

Общая информация:

Система «Книга рецептов» представляет собой приложение на языке VBA, которое предназначено для хранения и организации рецептов. Система призвана предоставить пользователям удобный и эффективный способ хранения и поиска рецептов.

Назначение и цели создания (развития) системы:

Цель системы – предоставить пользователям простой и удобный способ хранения и поиска рецептов. Предоставление возможности создания, редактирования и удаления рецептов. Предоставление возможности поиска рецептов по различным критериям. Оптимизация процесса хранения и поиска рецептов.

Характеристики объектов автоматизации:

К объектам автоматизации системы относятся:

* Список рецептов;
* Название рецепта;
* Ингредиенты рецепта;
* Порядок приготовления рецепта.

Требования к системе:

4.1. Базовые принципы разработки подсистем:

Система должна удовлетворять следующим требованиям:

* Пользовательский интерфейс системы должен быть интуитивно понятен для пользователей;
* Система должна быть легко управляемой и настраиваемой;
* Система должна быть безопасной;
* Система должна предоставлять возможность создания, редактирования и удаления рецептов;
* Система должна быть способна обрабатывать большой объем данных.
* Система должна содержать:
* Средства хранения рецептов;
* Средства поиска рецептов по различным критериям.

4.2. Требования к архитектуре системы:

* В качестве основного языка программирования используется VBA (Visual Basic for Applications);
* Интерфейс пользователя реализуется с помощью пользовательских форм;
* Данные хранятся в листах Excel;

4.3. Требования к аппаратному обеспечению:

* Компьютер с операционной системой Windows;
* Версия Microsoft Excel, поддерживающая VBA;
* Минимальное разрешение экрана 1024x768 пикселей.

4.4. Требования к сети:

* Система не требует подключения к сети.

4.5. Дизайн интерфейса:

* Интерфейс приложения должен быть интуитивно понятным и простым для пользователя;
* Кнопки и элементы управления должны быть наглядными и легко обозримыми;
* Цветовая гамма приложения должна быть спокойной и не вызывать раздражения у пользователя.

4.6. Схема базы данных:

* База данных представлена в виде листов Excel;
* Каждый рецепт представлен отдельной строкой в таблице, с указанием его названия, ингредиентов, способа приготовления и уровня сложности;
* Для каждого пользователя сохраняются данные о рецептах, добавленных в избранное, помеченных как приготовленные и о ранге пользователя.

4.6. Требования к режимам функционирования:

* Разрабатываемая система должна функционировать 24 часа в сутки, 365 дней в году.

4.7. Пользователи:

* Система должна иметь один тип пользователей: обычные пользователи.

4.8. Функционал, выполняемый пользователями:

* Подбор рецептов из базы данных (листы Excel) по введенным продуктам;
* Добавление, изменение и удаление рецептов в базе данных;
* Пометка рецепта как приготовленного;
* Добавление рецепта в избранное;
* Просмотр статистики и получение сертификата со своим рангом;

4.9. Функционал, выполняемый системой:

* Расчет ранга пользователя на основе сложности приготовления рецептов;
* Автоматическое определение имени пользователя по имени ПК.

5. Состав и содержание работ по созданию системы:

* Разработать модель базы данных для хранения и обработки всех необходимых данных для книги рецептов на VBA. База данных должна иметь возможность хранить информацию о рецептах, пользователях;
* Разработать пользовательский интерфейс, который позволит пользователям легко добавлять и просматривать рецепты, а также фильтровать их по категориям;
* Реализовать возможность сохранения любимых рецептов в избранное и просмотра списка избранных рецептов.

1. Порядок контроля и приемки системы:

Для проверки выполнения указанных функций Системы, определения и проверки соответствия требованиям ТЗ количественных и (или) качественных характеристик Системы, выявления и устранения недостатков в действиях Системы и в разрабатываемой документации, поэтапный контроль за ходом разработки, должны быть проведены следующие виды испытаний:

* Предварительное тестирование: проводятся на этапе разработки для выявления дефектов и обеспечения соответствия системы первоначальным требованиям. Эти тесты включают модульное тестирование, интеграционное тестирование и системное тестирование;
* Приемочное тестирование: проводится после этапа разработки для подтверждения того, что система соответствует критериям приемки и выполняет цели, указанные в требованиях. Этот тип тестирования можно разделить на альфа- и бета-тестирование, которые проводятся конечными пользователями и ограниченным числом внешних тестировщиков соответственно;
* Приемочное тестирование пользователей: проводится, чтобы убедиться, что система соответствует потребностям и ожиданиям конечных пользователей. Этот тип тестирования включает тестирование системы конечными пользователями и предоставление отзывов команде разработчиков.

# 1.4 Функциональное моделирование (IDEF0, IDEF3).

IDEF0 – методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами.

IDEF0 использует нотации и символы, которые позволяют описывать и моделировать функциональные процессы на разных уровнях детализации. Он позволяет создавать диаграммы на основе дерева функций, в котором корневой узел представляет собой основную функцию системы, а последующие узлы представляют подфункции и процессы, связанные с основной функцией.

IDEF0 – это метод, который позволяет моделировать бизнес-процессы и функции с помощью графических символов, показывающих входы, выходы и управляющие связи между функциями. IDEF0 может быть использован для описания и проектирования различных систем, включая бизнес-процессы, производственные системы, системы управления, компьютерные сети и др.

Для функционального моделирования книги рецептов на VBA использовалась методология IDEF0, которая состоит из контекстной диаграммы и диаграммы 1-уровня (декомпозиции).

Контекстная диаграмма показывает взаимодействие системы с внешней средой и состоит из одной функции, которая обозначает саму систему. Она показывает, что система взаимодействует с одним типом пользователей (механизм): пользователь. На входе: данные пользователя, рецепт, список продуктов, выбор действия. На выходе: избранные рецепты, статистика, сертификат, рецепты, измененная база данных. Контекстная диаграмма IDEF0 книги рецептов на VBA представлена в соответствии с рисунком 1.1.

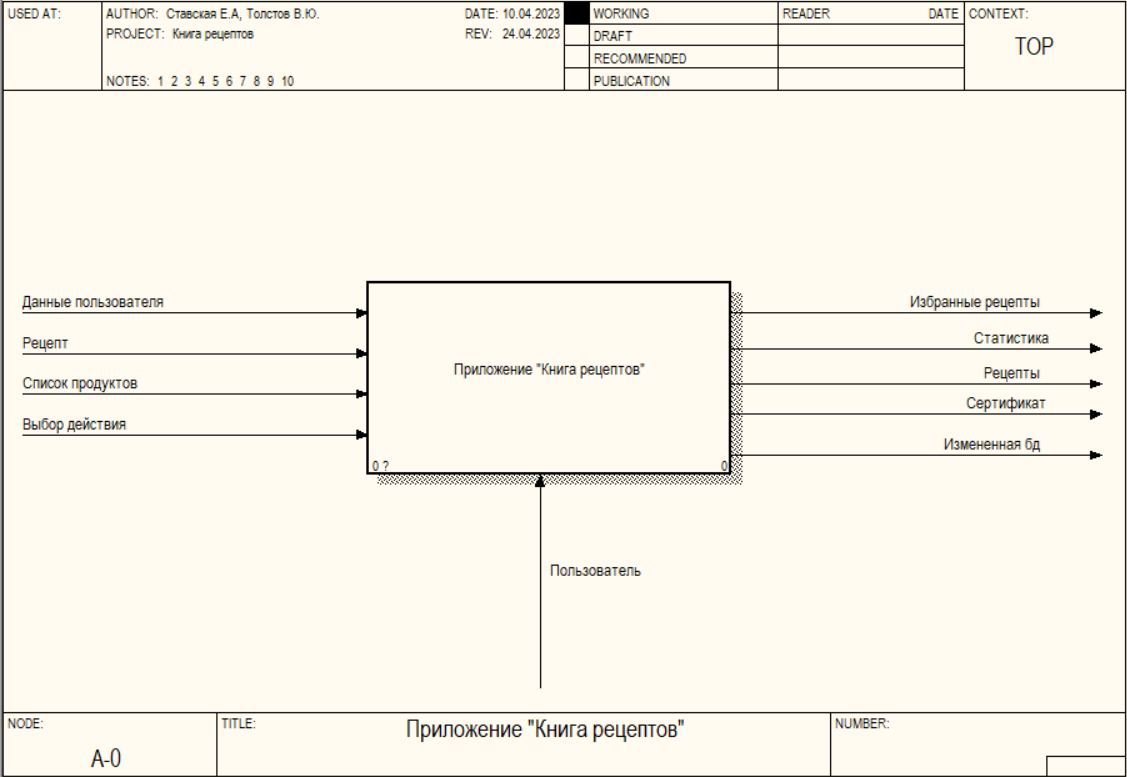


Рисунок 1.1 Контекстная диаграмма IDEF0

Диаграмма декомпозиции разбивает систему на более мелкие подсистемы, каждая из которых выполняет определенные функции. Она состоит из трех функций: подбор рецептов, профиль, редактирование базы данных. Каждая из этих функций разбивается на более детализированные функции, позволяя более точно определить задачи, которые должна выполнять каждая из подсистем. Диаграмма декомпозиции 1-уровня IDEF0 представлена в соответствии с рисунком 1.2.

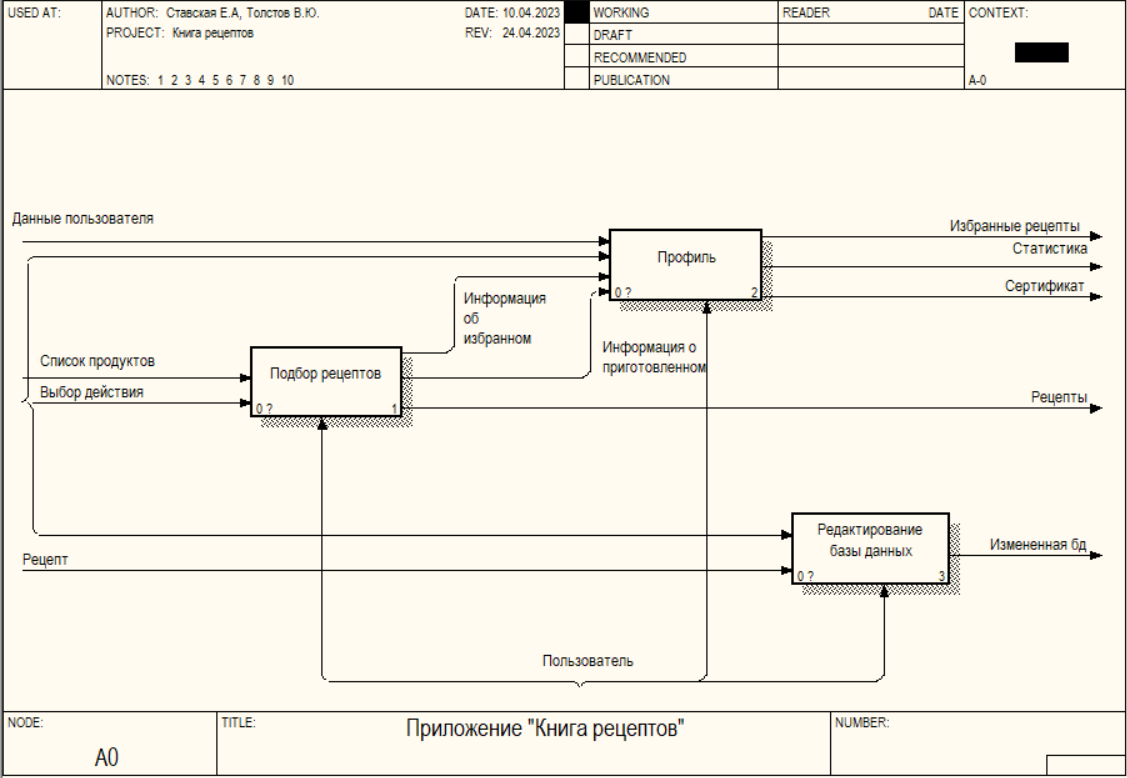


Рисунок 1.2 Диаграмма декомпозиции 1-уровня IDEF0

IDEF3 (Integrated Definition for Process Description Capture Method) является методом, который позволяет описать более детально процессы, которые определены в IDEF0. IDEF3 может быть использован для создания более подробных диаграмм действий, которые необходимы для выполнения каждой функции.

Например, для функции «подбор рецептов» можно определить действия, такие как ввод продуктов, подобрать рецепты, очистить поле ввода, добавить в избранное, добавить в приготовленное. На входе: список продуктов, выбор действия (подбор рецепта). На выходе: информация о приготовленном, информация об избранном, рецепты. Диаграмма декомпозиции 2-уровня IDEF3 «подбор рецептов» представлена в соответствии с рисунком 1.3.

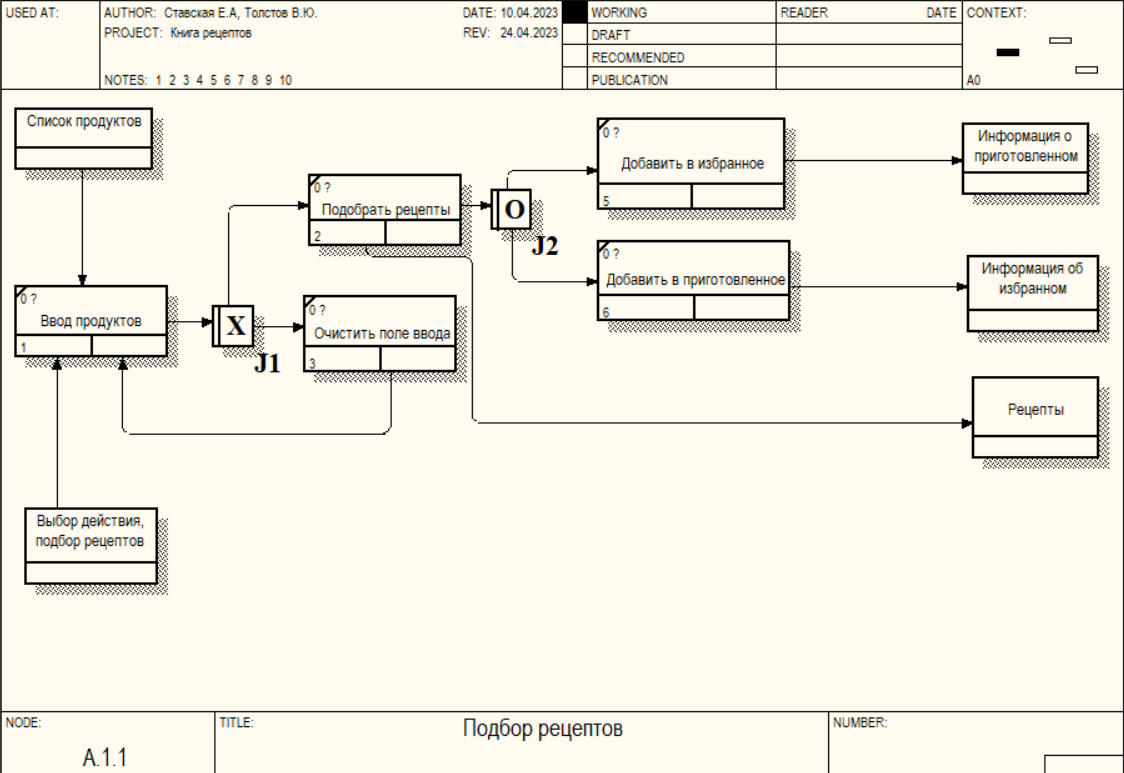


Рисунок 1.3 Диаграмма декомпозиции 2-уровня IDEF3 «подбор рецептов»

Для функции «профиль» можно определить действия, такие как просмотреть статистику, получить сертификат, просмотреть избранное, удалить из избранного. На входе: данные пользователя, информация о приготовленном, информация об избранном, выбор действия. На выходе: статистика, сертификат, избранные рецепты. Диаграмма декомпозиции 2-уровня IDEF3 «профиль» представлена в соответствии с рисунком 1.4.

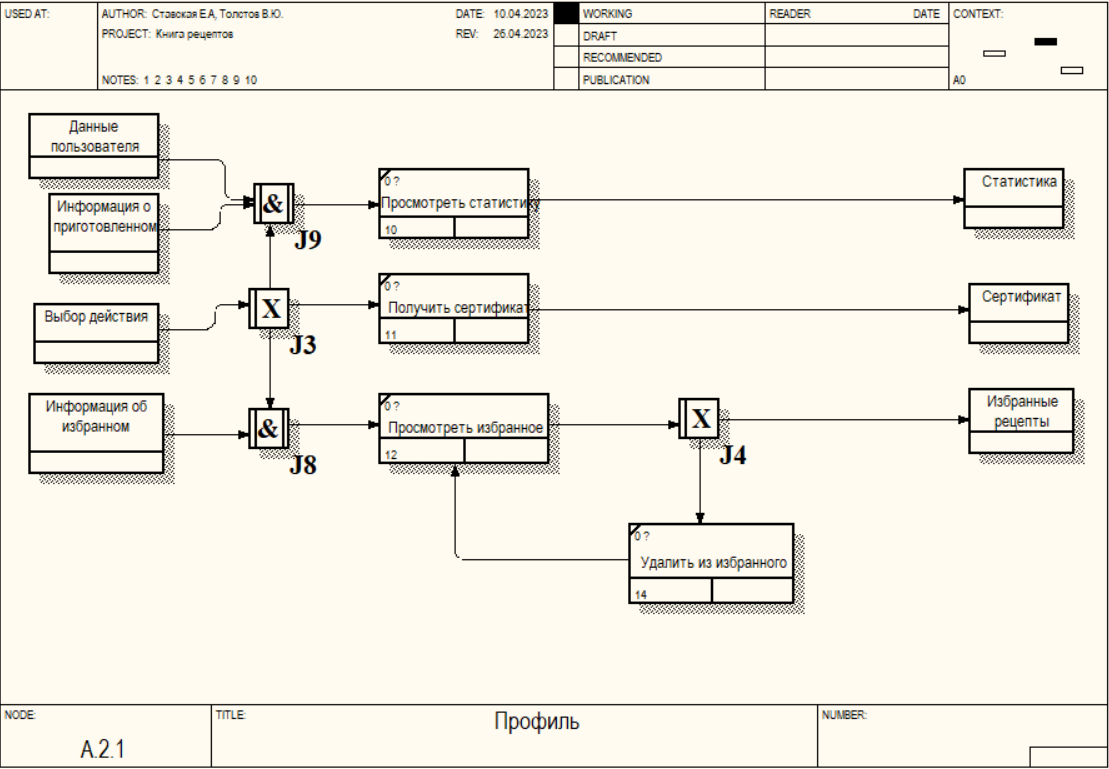


Рисунок 1.4 Диаграмма декомпозиции 2-уровня IDEF3 «профиль»

Для функции «редактирование базы данных» можно определить действия, такие как просмотреть добавить рецепт, редактировать рецепт, удалить рецепт. На входе: выбор действия, рецепт. На выходе: измененная база данных. Диаграмма декомпозиции 2-уровня IDEF3 «редактирование базы данных» представлена в соответствии с рисунком 1.5.

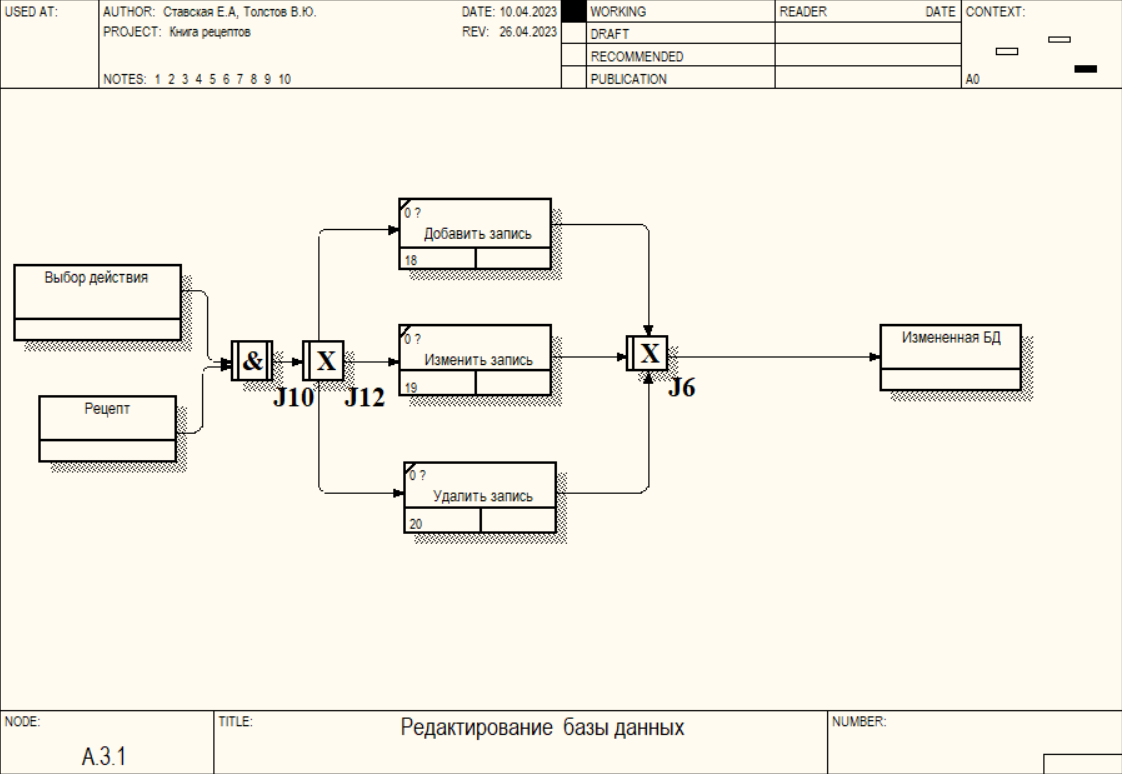


Рисунок 1.5 Диаграмма декомпозиции 2-уровня IDEF3 «редактирование базы данных»

Таким образом, использование методологии IDEF0 и IDEF3 позволяет более детально определить функции и процессы, которые необходимы для создания книги рецептов на VBA. Это поможет создать более эффективный и простой в использовании программный продукт.

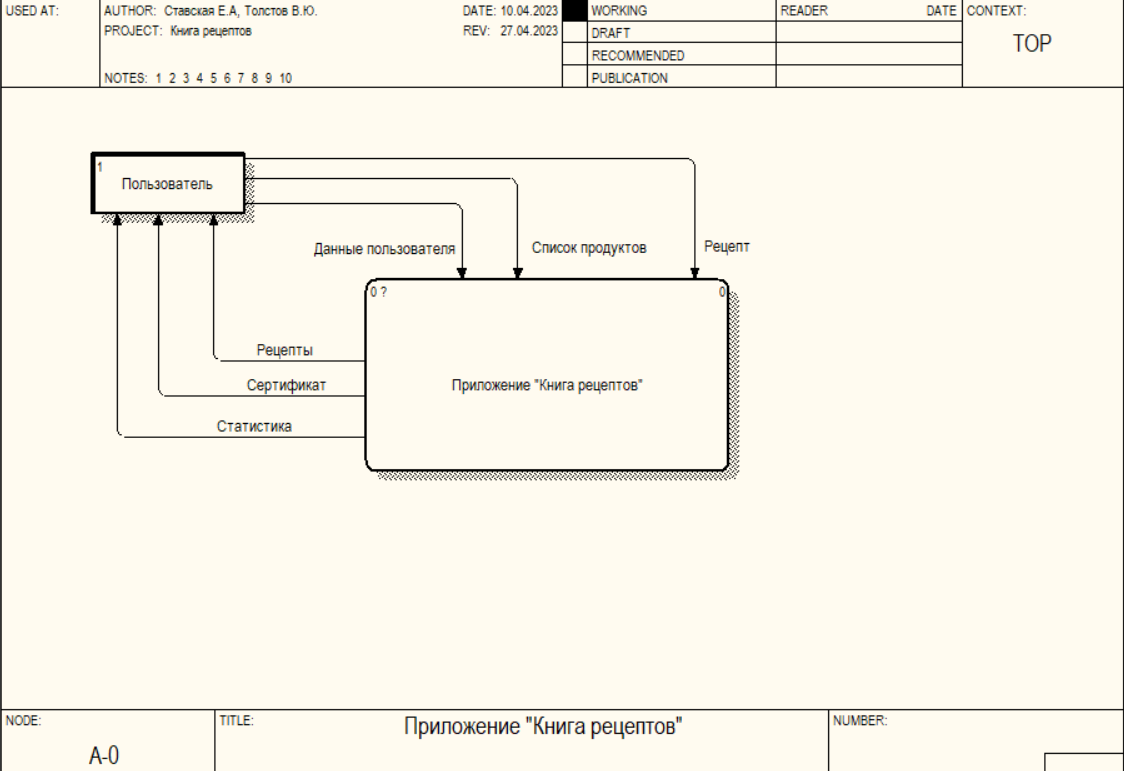
# 2 Проектная часть

# 2.1 Проектирование информационных потоков (DFD).

Data Flow Diagram (DFD) – это диаграмма, которая показывает поток данных и процессы, которые преобразуют эти данные. В контексте книги рецептов на VBA, DFD поможет понять, как информация проходит через систему и какие процессы нужны для обработки этой информации.

Диаграмма DFD наглядно отображает течение информации в пределах процесса или системы. Для изображения входных и выходных данных, точек хранения информации и путей ее передвижения между источниками и пунктами доставки в таких диаграммах применяются стандартные фигуры, такие как прямоугольники и круги, а также стрелки и краткие текстовые метки.

На первом уровне DFD, находится контекстная диаграмма, которая показывает взаимодействие между пользователем и системой. Контекстная диаграмма DFD представлена в соответствии с рисунком 2.1.

  
Рисунок 2.1 Контекстная диаграмма DFD

На втором уровне DFD, происходит более подробное описание процессов, которые преобразуют данные.

Подчиняясь правилам и инструкциям выбранной системы, символы отображают четыре компонента диаграммы DFD.

Внешние сущности – внешние системы, из которых поступает или куда направляется информация в результате взаимодействия с изображаемой системой. Иными словами, это источники и пункты доставки информации, которая приходит или уходит из системы. Как правило, располагаются по краям схемы.

Процессы – любые процессы, которые ведут к изменению информации и созданию выходных данных.

Хранилища данных – файлы или репозитории, где хранится информация для последующего использования, например, базы данных.

Потоки данных – маршруты, по которым информация перемещается между внешними сущностями, процессами и хранилищами данных. Потоки данных иллюстрируют взаимодействие между другими компонентами и отображаются в виде стрелок, как правило, с краткими метками. Диаграмма декомпозиции 1-уровня DFD представлена в соответствии с рисунком 2.2.

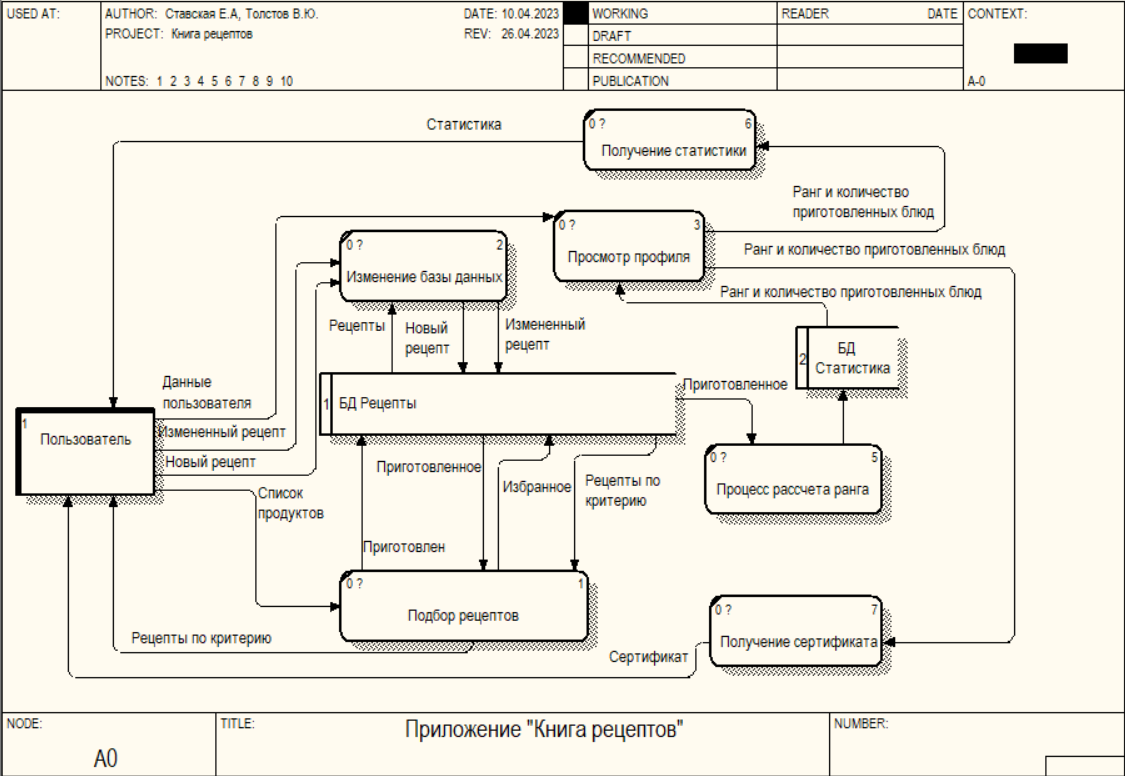


Рисунок 2.2 Диаграмма декомпозиции 1-уровня DFD

Для проекта «Книга рецептов» на диаграмме декомпозиции 1-уровня DFD:

Внешние сущности: пользователь.

Процессы: изменение базы данных, просмотр профиля, получение статистики, процесс расчета ранга, получение сертификата, подбор рецептов.

Базы данных: БД рецепты, БД статистика.

Таким образом, DFD поможет понять, как информация проходит через систему, какие процессы нужны для обработки этой информации и как они связаны между собой.

# 2.2 Проектирование средствами UML (6 видов диаграмм).

Диаграмма вариантов использования (Use Case Diagram) – это диаграмма, которая позволяет описать функциональность системы с точки зрения ее пользователей. Диаграмма вариантов использования состоит из актеров (Actor) – пользователей системы, и вариантов использования (Use Case) – функциональных возможностей системы, которые могут быть использованы пользователями. Варианты использования описываются в виде эллипсов и связаны с актерами линиями.

Для выражения отношений между актерами и вариантами использования применяются стандартные виды отношений.

Отношение ассоциации применительно к диаграммам вариантов использования служит для обозначения специфической роли актера в отдельном варианте использования. Другими словами, ассоциация определяет семантические особенности взаимодействия актеров и вариантов использования в графической модели системы. Таким образом, это отношение устанавливает, какую конкретную роль играет актер при взаимодействии с экземпляром варианта использования.

Отношение расширения между вариантами использования обозначается пунктирной линией со стрелкой (вариант отношения зависимости), направленной от того варианта использования, который является расширением для исходного варианта использования. Данная линия со стрелкой помечается ключевым словом «extend» («расширяет»).

Отношение включения между двумя вариантами использования указывает, что поведение одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования. Графически данное отношение обозначается пунктирной линией со стрелкой (вариант отношения зависимости), направленной от базового варианта использования к включаемому. При этом данная линия со стрелкой помечается ключевым словом «include» («включает»).

Для проекта «Книга рецептов» диаграмма вариантов использования включает:

Актеры: пользователь.

Варианты использования: подбор рецепта с расширением добавление в приготовленное, добавление в избранное; редактирование базы данных с расширением добавление рецепта, удаление рецепта, редактирование рецепта; личный кабинет с расширением просмотр ранга, просмотр статистики, просмотр избранного, получение сертификата с расширением ввод имени пользователя.

Диаграмма вариантов использования представлена в соответствии с рисунком 2.3.



Рисунок 2.3 Диаграмма вариантов использования

Диаграммы классов при моделировании объектно-ориентированных систем встречаются чаще других. На таких диаграммах отображается множество классов, интерфейсов, коопераций и отношений между ними. Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Кроме того, диаграммы классов составляют основу еще двух диаграмм – компонентов и развертывания.

Диаграмма классов может отражать различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений.

Основными элементами являются классы и связи между ними. Классы характеризуются при помощи атрибутов и операций.

Классом называется описание совокупности объектов с общими атрибутами, операциями, отношениями и семантикой. Атрибут – это именованное свойство класса, включающее описание множества значений, которые могут принимать экземпляры этого класса. Операция – это некоторый сервис, который предоставляет экземпляр или объект класса по требованию своих клиентов (других объектов, в том числе и экземпляров данного класса). Класс может содержать любое число операций или не содержать их вовсе.

Связи:

Ассоциация (association) – представляет собой отношения между экземплярами классов.

Агрегация (aggregation) – это ассоциация типа «целое-часть». Агрегация в UML представляется в виде прямой с ромбом на конце.Ромб на связи указывает, какой класс является агрегирующим (т.е. «состоящим из»); класс с противоположного конца – агрегированным (т.е. те самые «части»).

Композиция (composition) – это такая агрегация, где объекты-части не могут существовать сами по себе и уничтожаются при уничтожении объекта агрегирующего класса. Композиция изображается так же, как ассоциация, только ромбик закрашен.

Наследование (inheritance) – это отношение типа «общее-частное». Позволяет определить такое отношение между классами, когда один класс обладает поведением и структурой ряда других классов. При создании производного класса на основе базового (одного или нескольких) возникает иерархия наследования.

Для проекта «Книга рецептов» диаграмма классов включает:

Класс «пользователь» с функциями: подобрать рецепт, редактировать записи в базе данных, добавить в избранное, добавить в приготовленное, получить статистику, получить сертификат.

Класс «рецепты» с атрибутами: название, описание, продукты, уровень, приготовлен, избранное. И функциями, такими как: вывести рецепты, изменить записи.

Класс «статистика» с атрибутами: набрано опыта, осталось опыта, приготовлено блюд, ранг. И функциями, такими как: сформировать сертификат, рассчитать ранг, сформировать статистику.

Связь между классами «пользователь» и «рецепты» один ко многим означает, что каждому пользователю доступны многие рецепты.

Связь между классами «пользователь» и «статистика» один ко многим означает, что для каждого пользователя формируется множество видов статистик.

Диаграмма классов представлена в соответствии с рисунком 2.4.

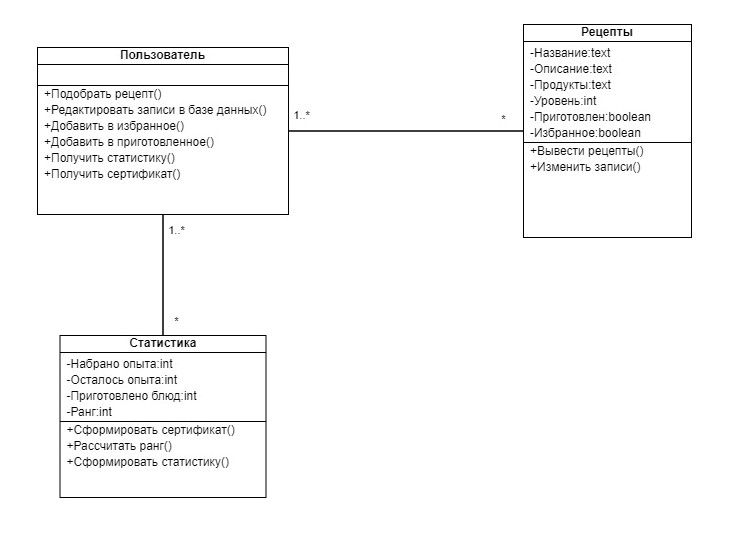
  
Рисунок 2.4 Диаграмма классов

Диаграмма активностей (видов деятельности) – как и диаграмма состояний, отражает динамические аспекты поведения системы. По существу, эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.

Активности на диаграмме «разбросаны» по беговым дорожкам, каждая из которых соответствует поведению одного из объектов. Благодаря этому легко определить, каким из объектов выполняется каждая из активностей. Дорожка – часть области диаграммы деятельности, на которой отображаются только те активности, за которые отвечает конкретный объект. Предназначены дорожки для разбиения диаграммы в соответствии с распределением ответственности за действия. Имя дорожки может означать роль или объект, которому она соответствует.

Для создания диаграммы деятельности используются следующие узлы:

Узел управления (control node) – это абстрактный узел действия, которое координирует потоки действий.

Начальный узел деятельности (или начальное состояние деятельности) (activity initial node) является узлом управления, в котором начинается поток (или потоки) при вызове данной деятельности извне.

Конечный узел деятельности (или конечное состояние деятельности) (activity final node) является узлом управления, который останавливает (stop) все потоки данной диаграммы деятельности. На диаграмме может быть более одного конечного узла.

Конечный узел потока (или конечное состояние потока) (flow final node) является узлом управления, который завершает данный поток. На другие потоки и деятельность данной диаграммы это не влияет.

Объект, над которым выполняются действия. Это не обязательный элемент диаграммы, но в некоторых случаях необходимо показать объект, инициирующий выполнение действий, или являющийся результатом его.

Узел решения предназначен для отображения расширений сценария на диаграмме деятельности. Для определения правила ветвления и различных вариантов дальнейшего развития сценария.

Узел объединения имеет два и более входящих узла и один исходящий.

Для проекта «Книга рецептов» диаграмма деятельности «Поиск рецепта»:

Начало процесса;

Пользователь открывает окно "Подбор рецепта";

Пользователь вводит в поле ввода продукты;

Приложение делает запрос к базе данных;

База данных возвращает значения;

Приложение выводит рецепты;

Пользователь просматривает рецепты, далее узел объединения: рецепт можно добавить в избранное и приготовленное, или добавить что-то одно, или не добавлять и завершить процесс;

Пользователем выбрано добавить в избранное: программа делает запрос к базе данных, база данных изменяет значение, конец процесса;

Пользователем выбрано отметить приготовленным: узел объединения. Запрос к базе данных, смена значения, конец процесса. Рассчитать статистику, запрос программы к базе данных, смена значений в базе данных, конец процесса.

Диаграмма деятельности «Поиск рецепта» представлена в соответствии с рисунком 2.5.

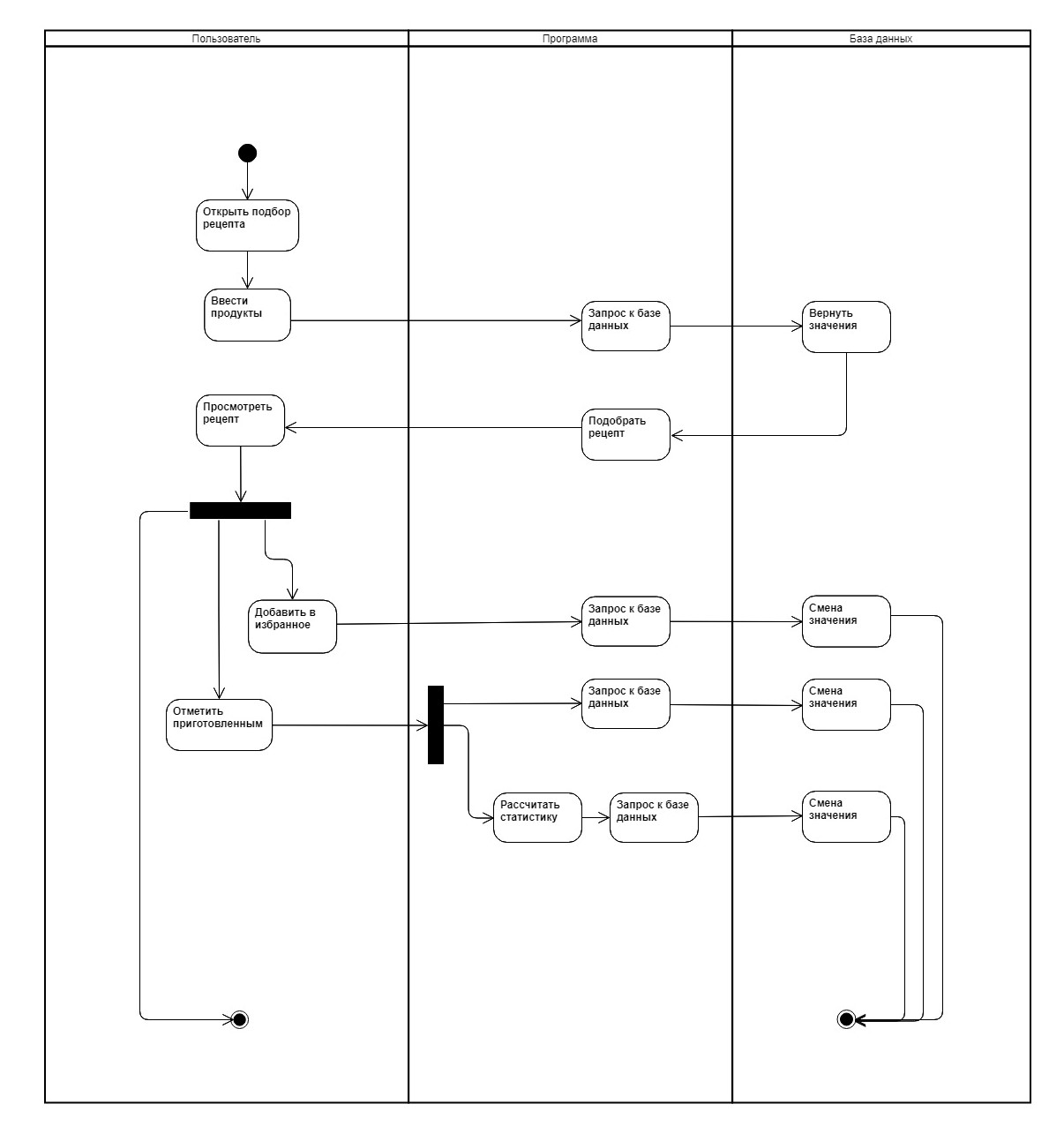


Рисунок 2.5 Диаграмма деятельности «Поиск рецепта»

Для проекта «Книга рецептов» диаграмма деятельности «Профиль»:

Начало процесса;

Пользователь открывает профиль. Узел выбора: избранное, статистика, сертификат;

Пользователь выбирает избранное. Программа делает запрос к базе данных, база возвращает значения. Программа формирует избранное. Пользователь просматривает избранное, конец процесса;

Пользователь выбирает получить статистику. Программа делает запрос к базе данных, база возвращает значения. Программа формирует статистику. Пользователь просматривает статистику, конец процесса;

Пользователь выбирает получить сертификат. Программа делает запрос к базе данных, база возвращает значения. Программа формирует сертификат. Пользователь просматривает сертификат, конец процесса.

Диаграмма деятельности «Профиль» представлена в соответствии с рисунком 2.6.



Рисунок 2.6 Диаграмма деятельности «Профиль»

Для проекта «Книга рецептов» диаграмма деятельности «Редактирование базы»:

Начало процесса;

Пользователь открывает редактирование базы. Узел выбора: добавить запись, удалить запись, редактировать запись;

Пользователь выбирает добавить запись. Программа делает запрос к базе данных, база изменяет значения, конец процесса;

Пользователь выбирает удалить рецепт. Программа делает запрос к базе данных, база изменяет значения, конец процесса;

Пользователь редактировать рецепт. Программа делает запрос к базе данных, база изменяет значения, конец процесса.

Диаграмма деятельности «Редактирование базы» представлена в соответствии с рисунком 2.7.

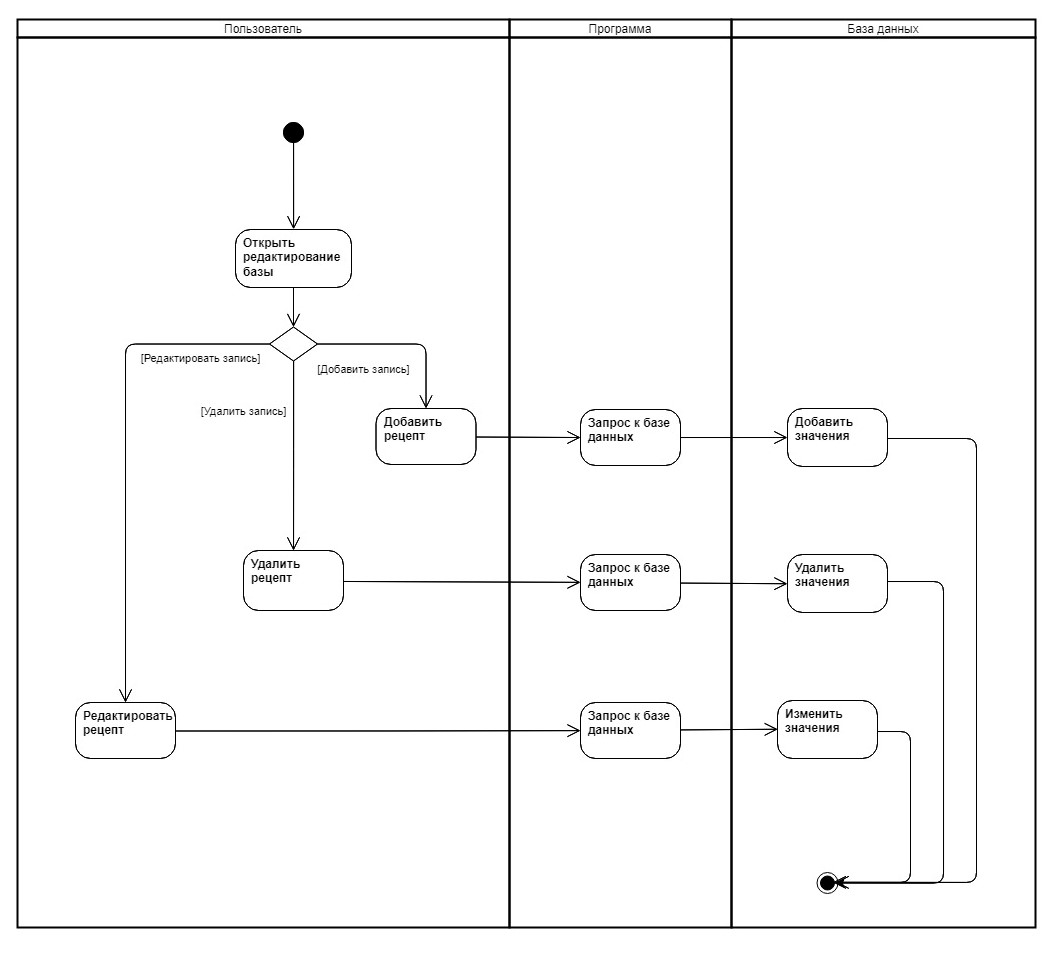


Рисунок 2.7 Диаграмма деятельности «Редактирование базы»

Диаграмма последовательности UML (Unified Modeling Language) – это вид диаграммы, используемый для описания взаимодействия между объектами или компонентами в системе. Она показывает последовательность сообщений, передаваемых между объектами или компонентами во времени.

Диаграмма последовательности включает в себя:

* Объекты или компоненты, между которыми происходит взаимодействие;
* Сообщения, передаваемые между объектами или компонентами, обозначенные стрелками.

Временные ограничения, обозначенные вертикальными пунктирными линиями, которые показывают длительность взаимодействия между объектами.

Диаграмма последовательности часто используется для описания взаимодействия между компонентами в рамках одного процесса, например, взаимодействия между компонентами веб-приложения или между классами в программе.

Для проекта «Книга рецептов» диаграмма последовательности содержит объекты такие, как: пользователь, программа, БД.

Диаграмма последовательности представлена в соответствии с рисунком 2.8.

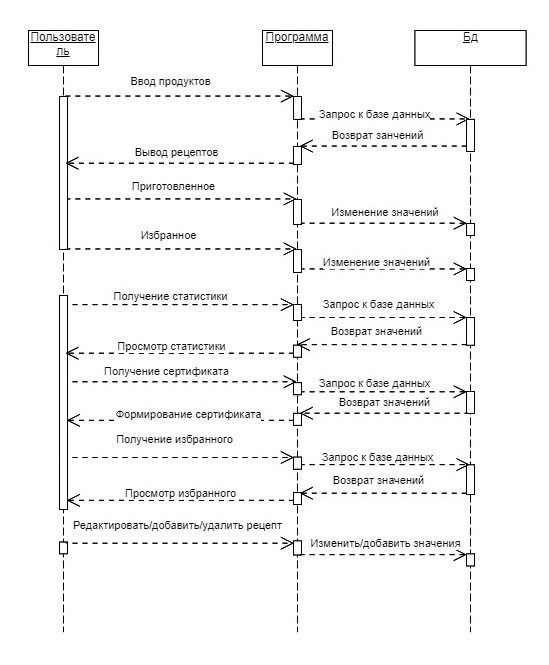


Рисунок 2.8 Диаграмма последовательности

Диаграмма компонентов – это структурная диаграмма, которая показывает, как компоненты приложения связаны друг с другом и с внешними компонентами. Она используется для описания архитектуры приложения, определяя его основные компоненты, их взаимодействие и зависимости.

Символы на диаграмме компонентов

Компонент: прямоугольник со значком компонента и стереотипом компонента.

Предоставляемый интерфейс и требуемый интерфейс: интерфейсы на компонентных схемах показывают, как компоненты соединены друг с другом и взаимодействуют друг с другом. Соединитель сборки позволяет соединить требуемый интерфейс компонента (представленный полукругом и сплошной линией) с предусмотренным интерфейсом (представленный окружностью и сплошной линией) другого компонента. Это показывает, что один компонент предоставляет услугу, которая требуется другому.

Диаграмма компонентов представлена в соответствии с рисунком 2.9.

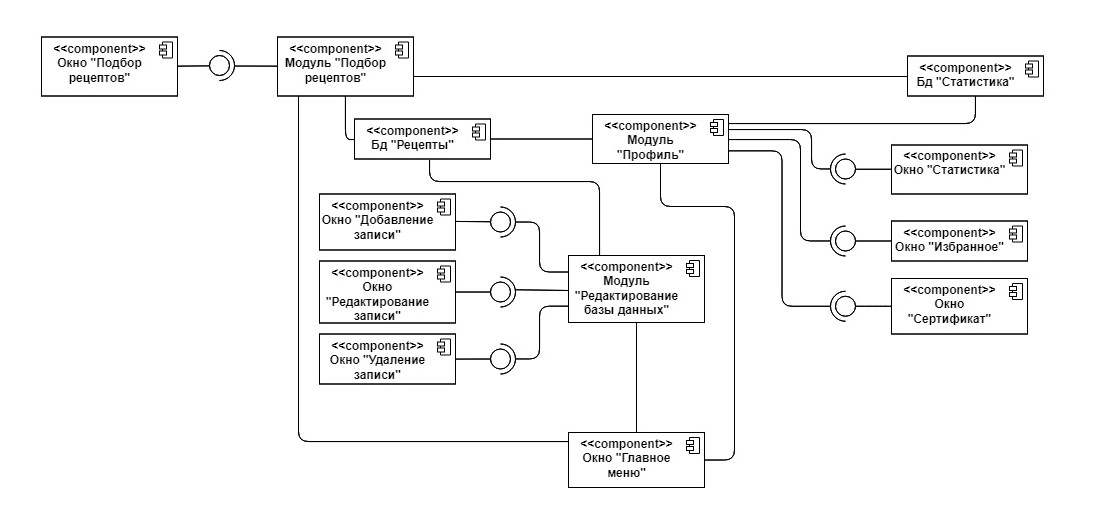


Рисунок 2.9 Диаграмма компонентов

Диаграмма состояний UML – это модель поведения, которая показывает, как объекты проходят через различные состояния во время своей жизненного цикла и как они реагируют на события, которые происходят в системе. Диаграмма состояний используется для описания поведения сложных систем и управления их состояниями.

Диаграмма состояний состоит из различных состояний объекта, переходов между этими состояниями и действий, которые происходят при переходе между состояниями. Диаграмма состояний имеет схожую семантику с диаграммой деятельности, только деятельность здесь заменена состоянием, переходы символизируют действия. Таким образом, если для диаграммы деятельности отличие между понятиями «Деятельность» и «Действие» заключается в возможности дальнейшей декомпозиции, то на диаграмме состояний деятельность символизирует состояние, в котором объект находится продолжительное количество времени, в то время как действие моментально.

Диаграмма состояний представлена в соответствии с рисунком 2.10.

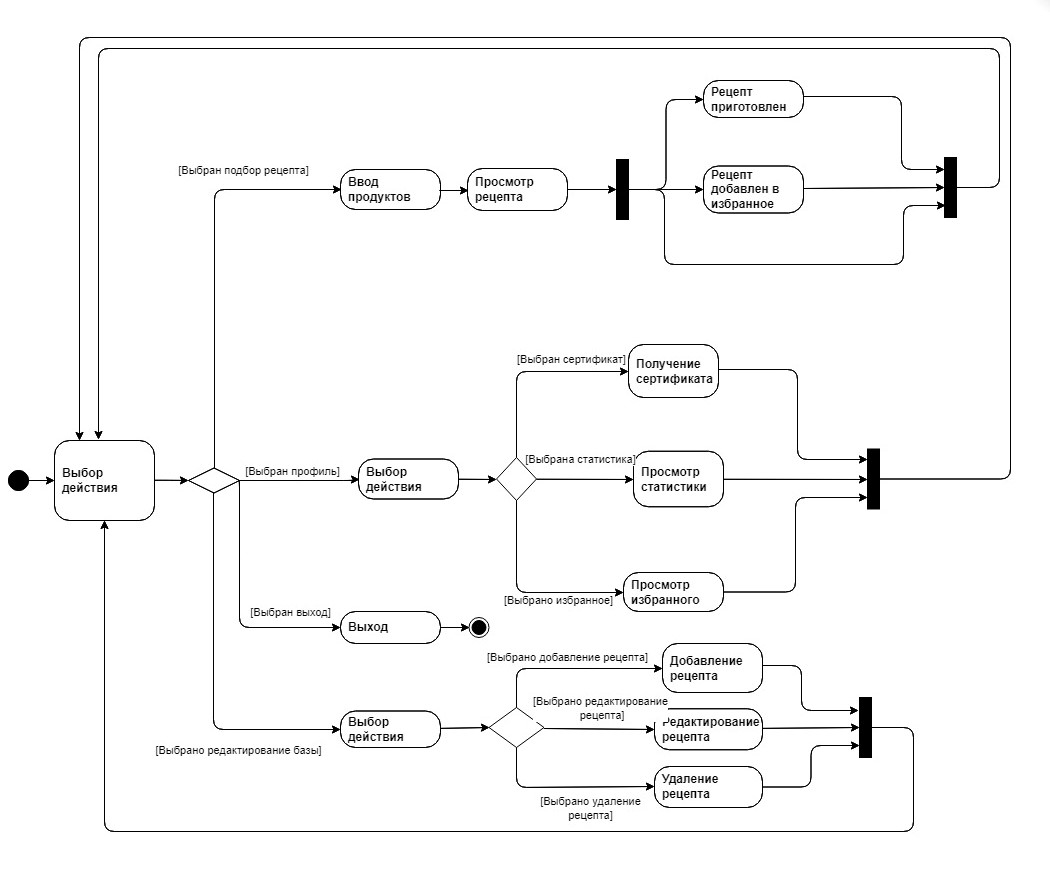


Рисунок 2.10 Диаграмма состояний

UML диаграммы представляют собой визуальное средство моделирования объектно-ориентированных систем. Использование UML диаграмм позволяет упростить процесс разработки и повысить эффективность коммуникации между различными участниками проекта.

Одна из ключевых особенностей UML диаграмм заключается в том, что они могут использоваться на разных стадиях жизненного цикла проекта. Так, на ранних стадиях проектирования UML диаграммы могут быть использованы для определения требований, анализа и проектирования системы, в то время как на более поздних стадиях – для реализации и тестирования системы.

UML диаграммы могут быть использованы для моделирования различных аспектов системы, таких как структура, поведение и взаимодействие между объектами. Каждый тип UML диаграммы имеет свои особенности и используется для определенных целей.

# 2.3 Проектирование БД (инфологическое проектирование – ERD; даталогическое проектирование – описание таблиц и схемы данных).

Схема «сущность-связь» (также ERD или ER-диаграмма) – это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы. ER-диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи – глаголов.

Символы ERD-атрибутов

Под понятием «сущности» подразумеваются объекты или понятия, несущие важную информацию. Обычно обозначаются прямоугольником.

Связи используются в схемах «сущность-связь» для обозначения взаимодействия между двумя сущностями. Грамматически связи, как правило, выражаются глаголами. Обычно обозначаются ромбом.

ERD-атрибуты характеризуют сущности, позволяя пользователям лучше разобраться в устройстве базы данных. Атрибуты содержат информацию о сущностях, выделенных в концептуальной ER-диаграмме. Обычно обозначаются овалом.

В ER-диаграмме для приложения «Книга рецептов» определим такие сущности: «первые блюда», «вторые блюда», «закуски», «салаты», «десерты», «напитки», «приготовлено рецептов», «остаток опыта».

Связи: «учитывает».

Диаграмма ER представлена в соответствии с рисунком 2.11.

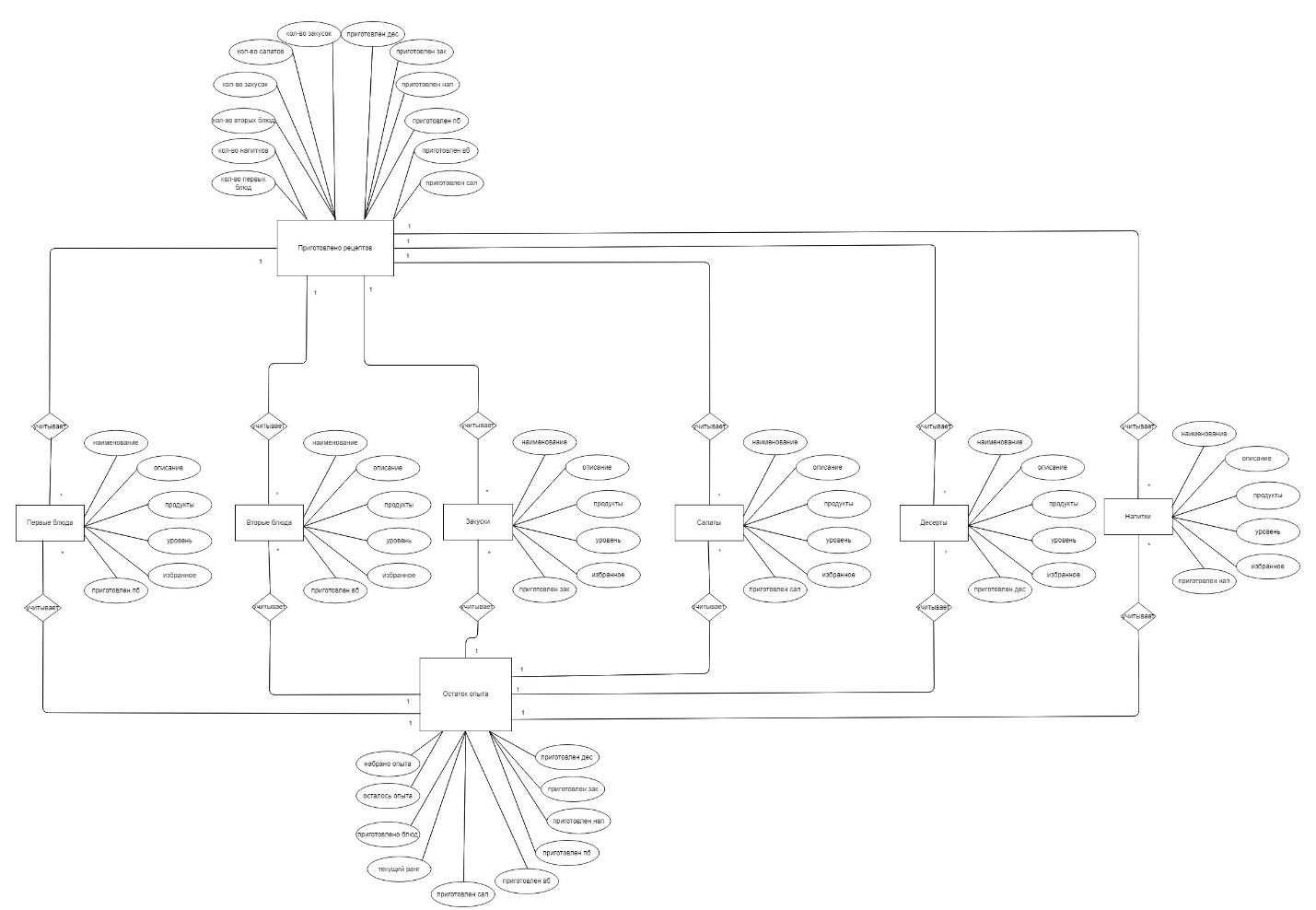


Рисунок 2.11 ER-диаграмма

Даталогическая модель – это модель базы данных, которая описывает реализацию сущностей и связей, определенных в инфологической модели, с учетом специфики используемой СУБД и ее ограничений.

Даталогическая модель описывает все таблицы, их поля и связи между ними, а также ограничения на эти таблицы и связи. Каждая таблица представляет собой отдельную сущность в системе и имеет свое имя и набор полей, которые описывают свойства этой сущности. Связи между таблицами определяют отношения между этими сущностями.

Даталогическая модель также учитывает ограничения на таблицы и связи, такие как уникальность значений в поле, ограничения на ввод значений, ограничения на удаление записей и т.д.

Кроме того, даталогическая модель включает определение типов данных для каждого поля и связи, а также определение индексов, которые используются для ускорения доступа к данным.

Для проекта «Книга рецептов», даталогическая модель будет содержать таблицы, такие как «первые блюда», «вторые блюда», «закуски», «салаты», «десерты», «напитки», «приготовлено рецептов», «остаток опыта». Каждая таблица будет содержать свои поля, ограничения и связи с другими таблицами.

Даталогическая модель представлена в соответствии с рисунком 2.12.

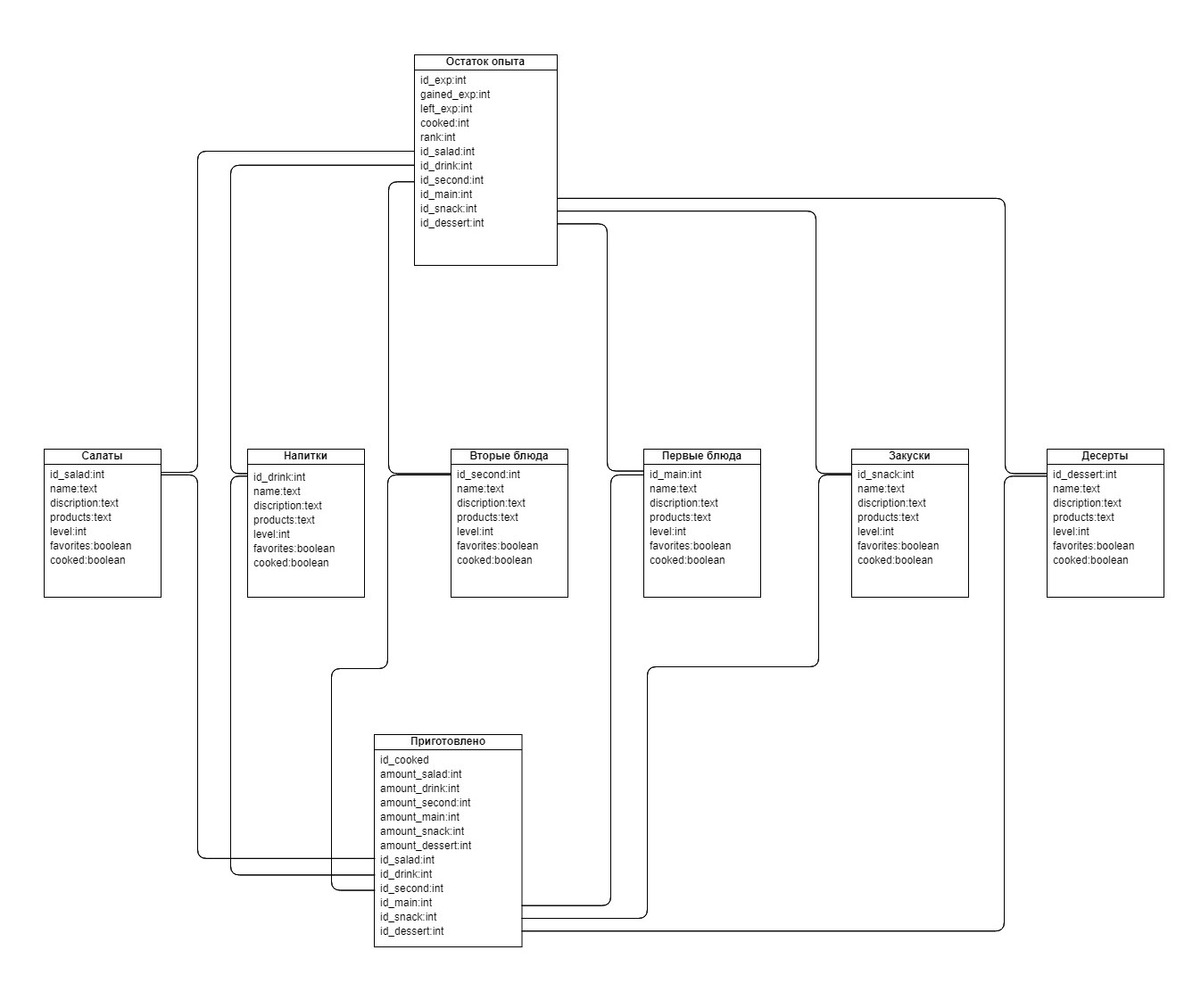


Рисунок 2.12 Даталогическая модель

Концептуальная схема базы данных, полученная в результате даталогического проектирования, является базовой структурой, на основе которой разрабатываются физические схемы базы данных, используемые конкретной СУБД. Даталогическая модель позволяет оптимизировать работу с данными и обеспечить целостность и надежность базы данных. Одним из главных преимуществ даталогической модели является то, что она позволяет более точно определить структуру базы данных и предотвратить возможные проблемы согласованности данных. Она также упрощает процесс создания физической модели базы данных и позволяет эффективнее работать с ней в дальнейшем.

# 2.4 Проектирование пользовательских интерфейсов

Окно «Главная» является основным интерфейсом пользовательского приложения, написанного на VBA. В этом окне имеются четыре кнопки:

«Найти рецепт» - при нажатии на данную кнопку пользователь переходит на страницу поиска рецептов, где ему предоставляется возможность осуществить поиск по имеющимся продуктам.

«Личный кабинет» - кнопка, предназначенная для перехода на страницу личного кабинета пользователя.

«Редактировать базу» - кнопка, которая открывает страницу редактирования базы данных рецептов. Здесь пользователь может добавлять новые рецепты, редактировать их описание, а также удалять ненужные записи из базы.

«Выход» - кнопка, позволяющая пользователю выйти из приложения.

Окно «Главная» представлено в соответствии с рисунком 2.13.



Рисунок 2.13 Окно «Главная»

Окно «Найти рецепт» предназначено для поиска рецептов в базе данных. Оно содержит следующие элементы интерфейса:

* Текстовое поле для ввода продуктов, по которым будет осуществляться поиск.
* Кнопку «Подобрать рецепты», при нажатии на которую будет запущен поиск рецептов с использованием введенных ключевых слов и переход в окно «Результат подбора».
* Кнопку «На главную», для возврата на предыдущую страницу.
* Кнопку «Очистить поле ввода», для очистки поля ввода.

Окно «Найти рецепт» представлено в соответствии с рисунком 2.14.

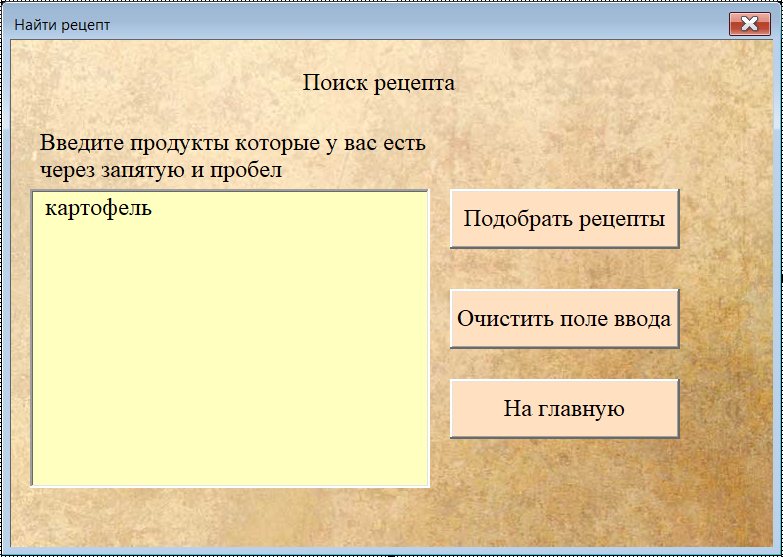


Рисунок 2.14 Окно «Найти рецепт»

Окно «Результат подбора» содержит следующие элементы:

* Кнопку «Добавить рецепт в избранное», при нажатии на которую рецепт будет добавлен в раздел избранное.
* Кнопку «Назад», при нажатии на которую произойдет возврат в окно «Найти рецепт».
* Кнопку «На главную», для возврата в окно «Главная».
* Поля для вывода описания рецепта, его название и его уровень.
* Выпадающий список для выбора рецепта из найденного по категориям.
* Checkbox для пометки рецепта как приготовленного.

Окно «Результат подбора» представлено в соответствии с рисунком 2.15.

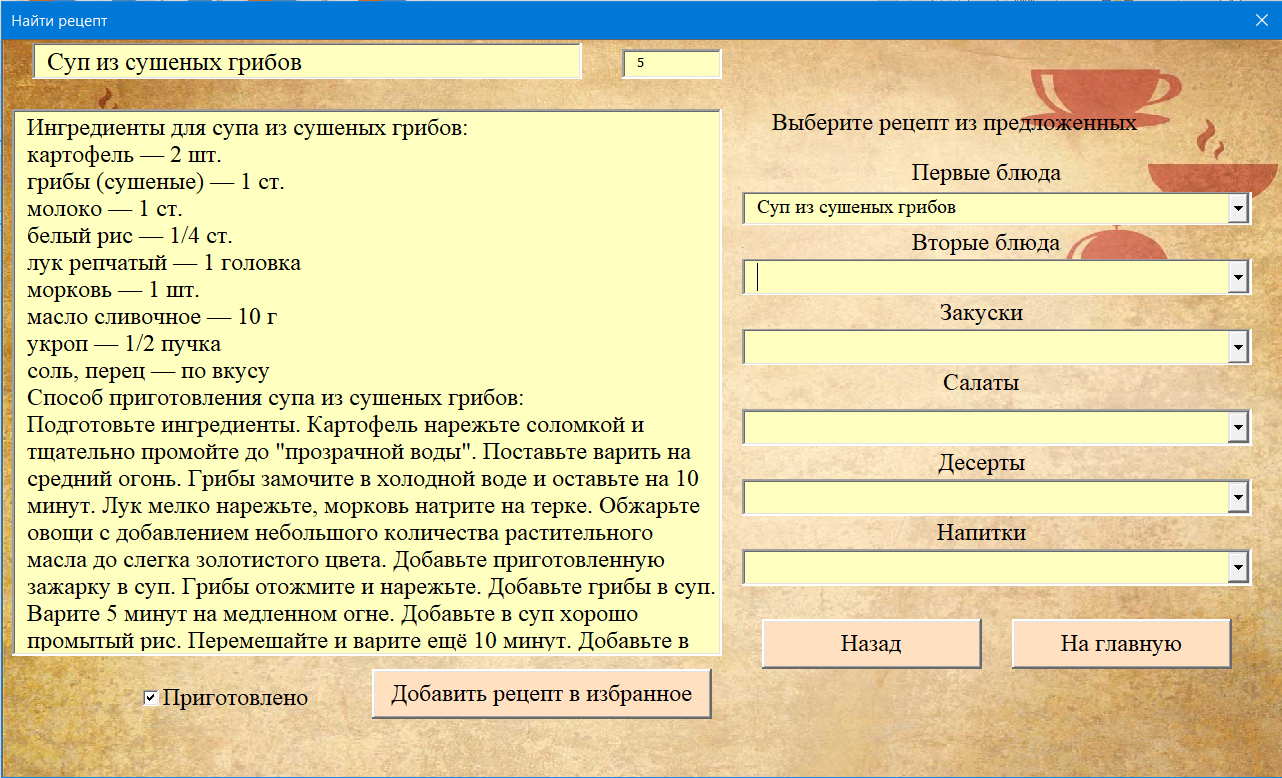


Рисунок 2.15 Окно «Результат подбора»

Окно «Профиль» содержит следующие элементы:

* Кнопку «Избранное», при нажатии на которую происходит переход в окно «Избранное».
* Кнопку «Сертификат», при нажатии на которую происходит переход в окно «Сертификат».
* Кнопку «Статистика», при нажатии на которую происходит переход в окно «Статистика».
* Кнопку «На главную», для возврата в окно «Главная».
* Информация о пользователе, такая как имя пользователя, количество приготовленных блюд, ранг пользователя и фото профиля.

Окно «Профиль» представлено в соответствии с рисунком 2.16.



Рисунок 2.16 Окно «Профиль»

Окно «Статистика» содержит следующие элементы:

* Кнопку «Назад», при нажатии на которую происходит переход в окно «Профиль».
* Поле с круговой диаграммой, отражающей оставшийся опыт до следующего уровня.
* Поле с гистограммой, отражающей статистику приготовленных рецептов по категориям.

Окно «Статистика» представлено в соответствии с рисунком 2.17.

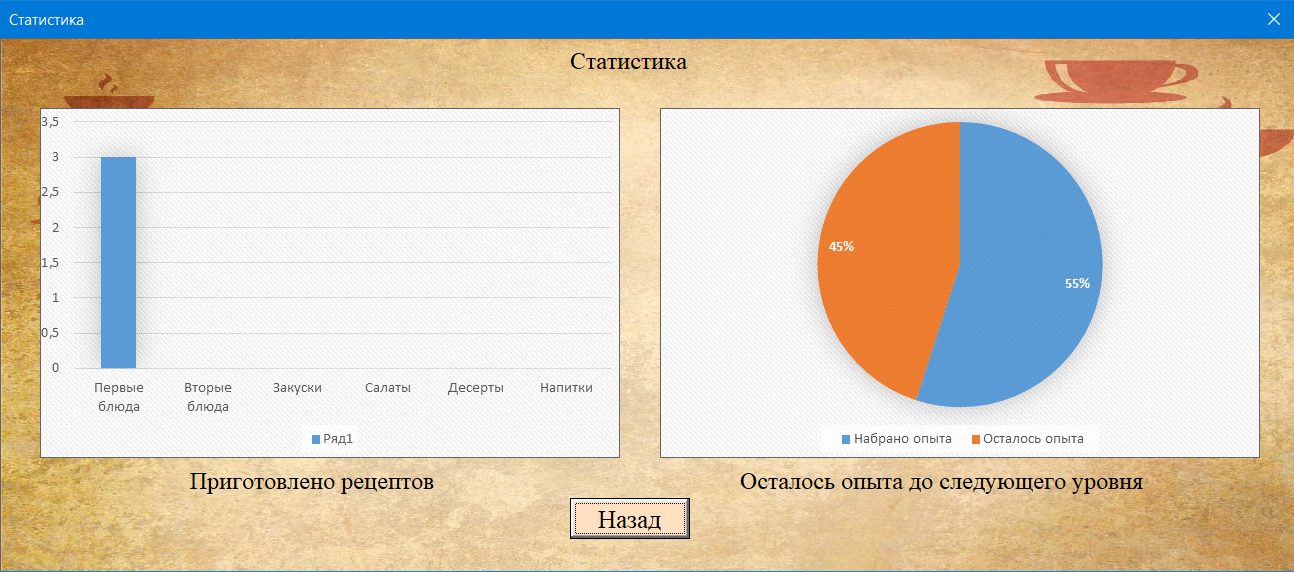


Рисунок 2.17 Окно «Статистика»

Окно «Избранное» содержит следующие элементы:

* Кнопку «Удалить рецепт в избранного», при нажатии на которую рецепт будет удален из раздела избранное.
* Кнопку «Назад», при нажатии на которую произойдет возврат в окно «Главная».
* Кнопку «На главную», для возврата в окно «Главная».
* Поля для вывода описания рецепта, его названия.
* Выпадающий список для выбора рецепта из избранного по категориям.

Окно «Избранное» представлено в соответствии с рисунком 2.18.

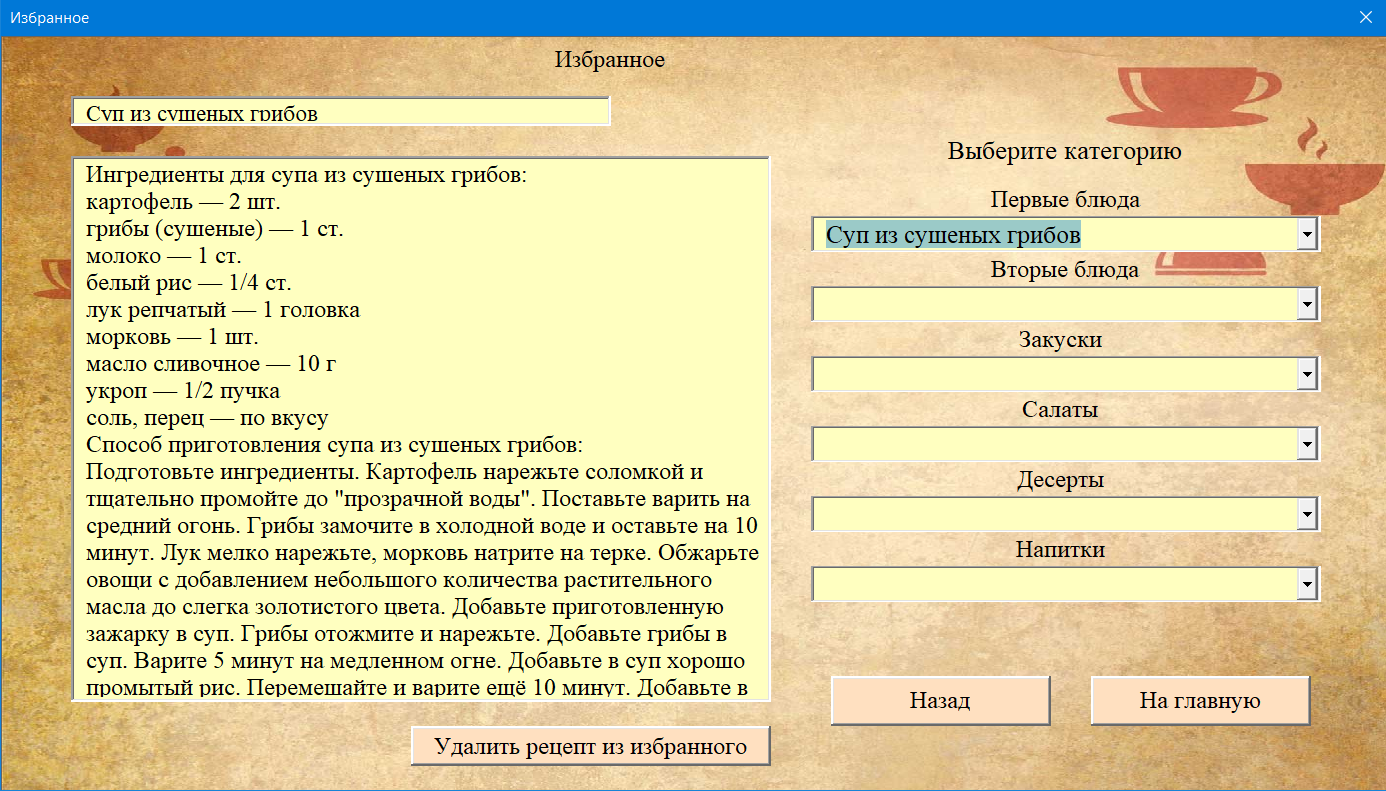


Рисунок 2.18. Окно «Избранное»

Окно «Сертификат» содержит следующие элементы:

* Кнопку «Получить сертификат», при нажатии на которую на устройство будет помещен сертификат, содержащий имя пользователя, его ранг.
* Кнопку «Назад», при нажатии на которую произойдет возврат в окно «Главная».
* Поле для ввода имени.

Окно «Сертификат» представлено в соответствии с рисунком 2.19.

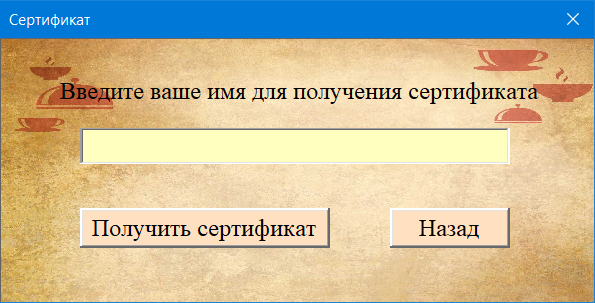


Рисунок 2.19 Окно «Сертификат»

Окно «Редактирование базы» содержит следующие элементы:

* Кнопку «Добавить запись», при нажатии на которую произойдет переход в окно «Добавление рецепта».
* Кнопку «Редактировать запись», при нажатии на которую произойдет переход в окно «Редактирование рецепта».
* Кнопку «Удалить запись», при нажатии на которую произойдет переход в окно «Удаление рецепта».
* Кнопку «Назад», для возврата в окно «Профиль».

Окно «Редактирование базы» представлено в соответствии с рисунком 2.20.

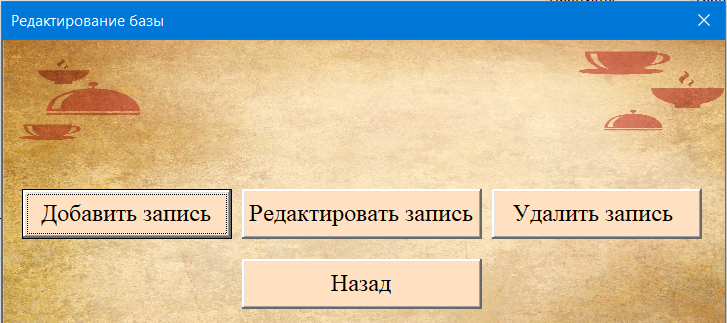


Рисунок 2.20 Окно «Редактирование базы

Окно «Добавления рецепта» содержит следующие элементы:

* Кнопку «Добавить рецепт», при нажатии на которую сохраняются введенные значения.
* Кнопку «Назад», для возврата в окно «Редактирование базы».
* Поле для ввода названия рецепта, поле для ввода описания рецепта, поле для ввода уровня рецепта, поле для ввода продуктов.
* Выпадающий список для выбора категории рецепта.

Окно «Добавление рецепта» представлено в соответствии с рисунком 2.21.

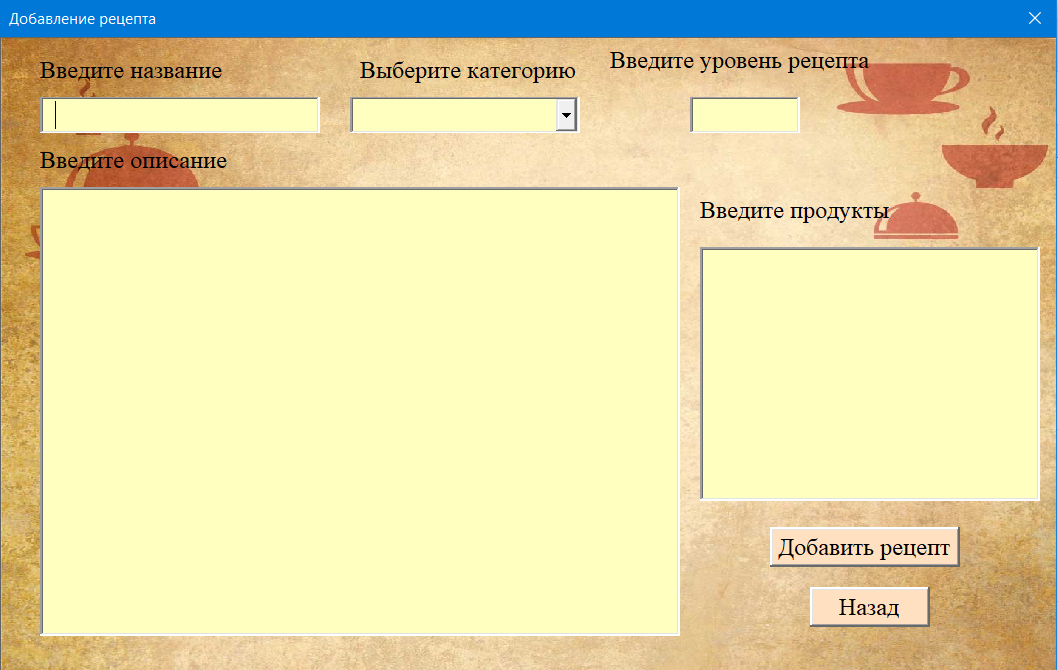


Рисунок 2.21 Окно «Добавление рецепта»

Окно «Редактирование рецепта» содержит следующие элементы:

* Выпадающий список для выбора категории рецепта.
* Поле вывода всех рецептов по выбранной категории таблицей.
* Поля для ввода названия рецепта, ввода описания рецепта, ввода уровня рецепта, ввода продуктов.
* Кнопку «Внести изменения», при нажатии на которую сохраняются новые значения рецепта.
* Кнопку «Назад», для возврата в окно «Редактирование базы».

Окно «Редактирование рецепта» представлено в соответствии с рисунком 2.22.

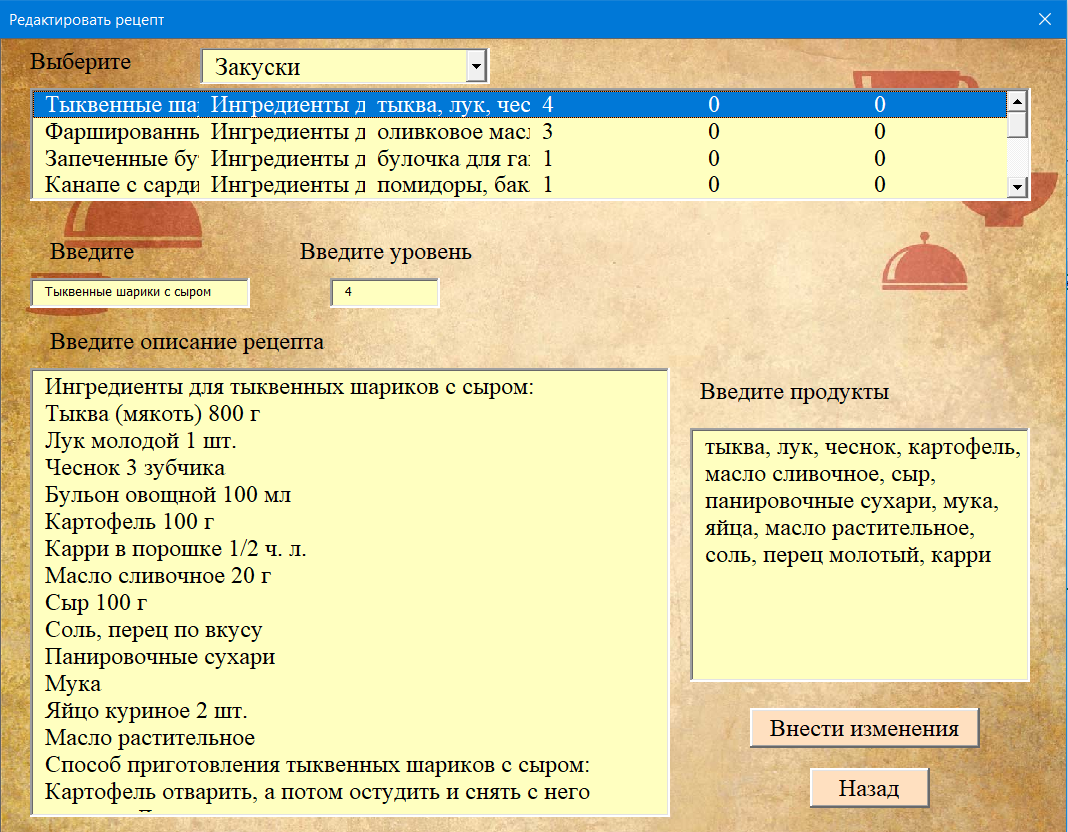


Рисунок 2.22 Окно «Редактирование рецепта»

Окно «Редактирование рецепта» содержит следующие элементы:

* Выпадающий список для выбора категории рецепта.
* Поле вывода всех рецептов по выбранной категории в виде таблицы.
* Кнопку «Удалить запись», при нажатии на которую выбранная запись удаляется из базы данных.
* Кнопку «Назад», для возврата в окно «Редактирование базы».

Окно «Удаление рецепта» представлено в соответствии с рисунком 2.23.

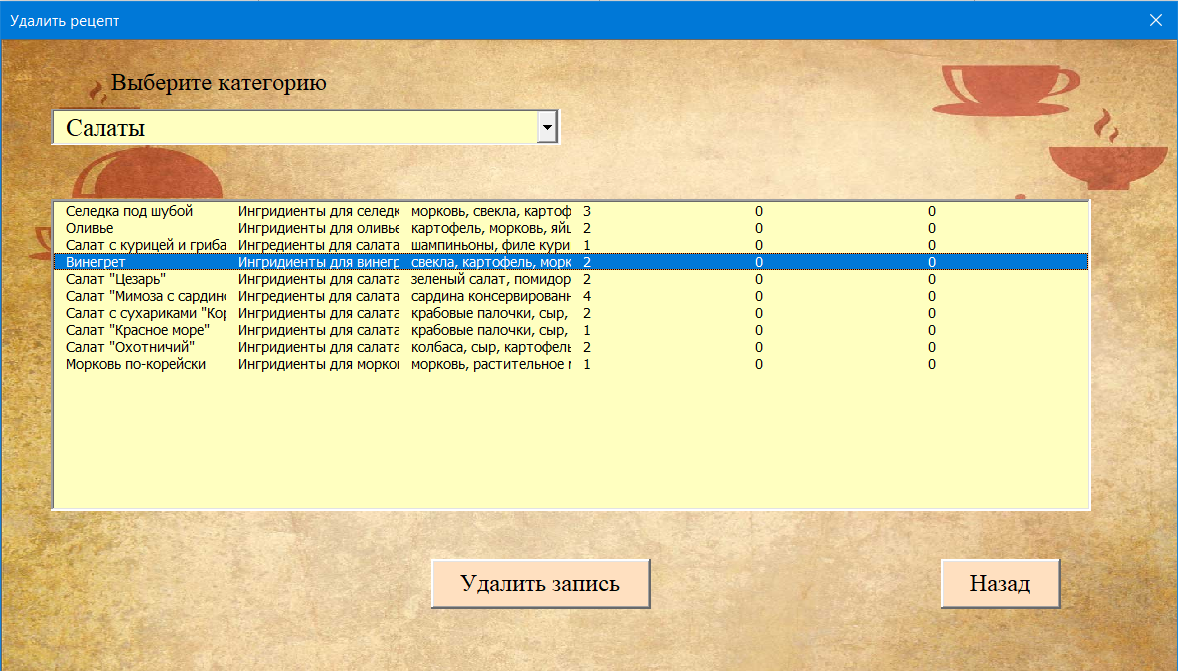


Рисунок 2.23 Окно «Удаление рецепта»

Проектирование пользовательских интерфейсов на VBA является важной задачей в разработке приложений на этой платформе. Цель создания удобного и понятного интерфейса заключается в том, чтобы обеспечить пользователям простой и интуитивно понятный доступ к функциям приложения.

Для достижения этой цели важно следовать некоторым общим принципам проектирования пользовательского интерфейса, таким как использование простых и понятных элементов управления, правильная организация размещения элементов на форме, создание логической структуры приложения и т.д.

При проектировании пользовательских интерфейсов на VBA также важно учитывать особенности платформы и целевой аудитории приложения.

В целом, проектирование пользовательских интерфейсов на VBA является важным аспектом разработки приложений на этой платформе, и его успешное выполнение может повысить удобство использования приложения для его пользователей и, следовательно, улучшить рейтинг приложения.

# ЗАКЛЮЧЕНИЕ

В результате выполнения данного проекта было разработано приложение на VBA Excel под названием «Книга рецептов», которое позволяет пользователям быстро и легко находить подходящие рецепты на основе введенных ими продуктов, а также добавлять свои собственные рецепты в базу данных.

Проект "Книга рецептов" представляет собой приложение на языке VBA Excel, которое предоставляет пользователям возможность быстрого и удобного поиска рецептов на основе имеющихся у них продуктов. Пользователь может ввести список продуктов, которые есть у него в наличии, и приложение автоматически найдет рецепты, которые можно приготовить на основе этих продуктов. Кроме того, в приложении предусмотрена возможность добавления своих собственных рецептов в базу данных, а также редактирования и удаления уже существующих рецептов.

В рамках проекта были выполнены все поставленные задачи, такие как разработка базы данных, создание интерфейса приложения, реализация функционала добавления, изменения и удаления рецептов в базе данных, разработка системы ранжирования пользователей на основе уровня сложности рецептов, разработка возможности добавления рецептов в избранное, разработка статистики приготовленных рецептов и возможность получения сертификата со своим рангом. Были использованы методы исследования, такие как анализ литературы по теме, изучение аналогов, разработка базы данных, программирование на языке VBA Excel.

Одной из ключевых особенностей приложения является система ранжирования пользователей на основе сложности рецептов, которые они приготавливают. Это позволяет стимулировать пользователей выбирать более сложные рецепты и повышать свой уровень в приложении. Кроме того, приложение ведет статистику приготовленных рецептов, что позволяет пользователям следить за своими достижениями и улучшать свои навыки в кулинарии.

Для реализации проекта были использованы различные методы исследования, включая анализ литературы и изучение аналогов на рынке. Также была проведена работа по программированию на языке VBA Excel, что позволило создать удобный и функциональный интерфейс приложения. В результате работы было получено приложение, которое помогает пользователям быстро и легко находить рецепты на основе имеющихся у них продуктов, а также управлять своей базой данных рецептов и повышать свой уровень в кулинарии.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Ушакова, Е.В. Методические указания по выполнению дипломных проектов/работ для бакалавриата направления информационно-коммуникационные технологии [Текст] / Е.В. Ушакова, Л.В. Долматова. -Петропавловск: СКГУ им. М.Козыбаева, 2020. – 99 с.;

2 Хант Р., Пратт Д. VBA для чайников [Текст] / Хант Р., Пратт Д. – СПб.: Питер, 2017. – 416 с.;

3 Леонтьева Т.А. Разработка приложений в Excel на VBA [Текст] / Леонтьева Т.А. – М.: Издательский дом «Экзамен», 2014. – 272 с.;

4 Арсентьев А.В. Microsoft Excel VBA [Текст] / Арсентьев А.В. – М.: ДМК Пресс, 2017. – 256 с.;

5 Панасенко Н.В. Программирование в Excel на VBA для начинающих [Текст] / Панасенко Н.В. – М.: Логос, 2018. – 384 с.;

6 Журавлев А.В. Excel 2016. Программирование на VBA [Текст] / Журавлев А.В. – М.: Лань, 2017. – 320 с.;

7 Бондарев А.А. VBA в Excel 2013: Программирование для пользователей [Текст] / Бондарев А.А. – М.: Эксмо, 2013. – 304 с.;

8 Киселев С.А. Программирование в Microsoft Excel с помощью VBA [Текст] / Киселев С.А. – М.: Логос, 2013. – 256 с.;

9 Серебряков И.А. Разработка приложений в Excel на VBA [Текст] / Серебряков И.А. – М.: Издательство "СамМедиа", 2020. – 416 с.;

10 Панасенко Н.В. VBA для Excel 2016: учебник [Текст] / Панасенко Н.В. – М.: Диалектика, 2016. – 320 с.;

11 Покровский А.В., Шевелева Е.А. Программирование в VBA для Excel 2013 [Текст]: учебное пособие / А.В. Покровский, Е.А. Шевелева. – СПб.: БХВ-Петербург, 2013. – 336 с.;

12 Ракитов А.А. Программирование на VBA в приложениях Microsoft Office 2013 [Текст]: учебное пособие / А.А. Ракитов. – М.: Юрайт, 2013. – 256 с.;

13 Черезова Е.В., Шашков А.В. Программирование в Excel 2013. Самоучитель [Текст] / Е.В. Черезова, А.В. Шашков. – М.: Эком, 2014. – 320 с.;

14 Соловьев В.В. Информационные технологии и Excel [Текст]: учебное пособие / В.В. Соловьев. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2013. – 223 с.;

15 Рогозин Д.В. Программирование на языке VBA в Microsoft Office [Текст]: учебное пособие / Д.В. Рогозин. – СПб.: Питер, 2013. – 336 с.

16 Козлов А.Г. Excel 2013. Практикум по программированию на VBA [Текст] / А.Г. Козлов. – М.: БХВ-Петербург, 2013. – 320 с.