МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет-изданий)

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированное программирование и стандарты проектирования»

Тема Программное средство «Grandma pizza»

Исполнитель

Студентка 2 курса группы 5 Кубик Елизавета Валентиновна

(Ф.И.О.)

Руководитель работы ассистент Северинчик Никита Александрович

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Пацей Н. В.

(подпись)

Минск 2020

# **Содержание**

[Содержание 2](#_Toc41278715)

[Введение 3](#_Toc41278716)

[1. Аналитический обзор литературы 4](#_Toc41278717)

[1.1. Анализ прототипов 4](#_Toc41278718)

[2. Моделирование предметной области и разработка функциональных требований 6](#_Toc41278720)

[2.1. Описание инструментов разработки программного средства 6](#_Toc41278721)

[2.2. Описание функций, реализуемых программой 7](#_Toc41278722)

[3. Проектирование программного средства 11](#_Toc41278723)

[3.1. Проектирование базы данных 11](#_Toc41278727)

[3.2. Определение последовательности передачи данных 13](#_Toc41278728)

[3.3 Диаграмма действий 14](#_Toc41278729)

[4. Создание (реализация) программного средства 15](#_Toc41278730)

[4.1. Реализация библиотеки классов 15](#_Toc41278731)

[4.2. Реализация серверной части 15](#_Toc41278732)

[4.3. Реализация клиентской части приложения 17](#_Toc41278733)

[5. Тестирование программного средства 19](#_Toc41278734)

[Список использованных источников 22](#_Toc41278735)

[Приложение А 23](#_Toc41278736)

[Приложение Б 24](#_Toc41278737)

**Введение**

Программное средство «Grandma pizza» реализует работу с пиццерией. «Grandma pizza» имеет несколько режимов доступа: пользователь, менеджер и администратор.

Целью курсового проекта является разработка программного обеспечения для быстрого заказа продуктов, предлагаемых пиццерией, со стороны клиента и удобного редактирования списка продуктов со стороны администратора и менеджера.

В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс — Windows Presentation Foundation (WPF), предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом.

Для работы с WPF использовался объектно-ориентированный язык программирования с С-подобным синтаксисом — С#, разработанный для создания приложений на платформе Microsoft .NET Framework.

Кроме того, используемая в WPF модель разделения кода и дизайна предполагала работу с декларативным языком описания интерфейса — eXtensible Application Markup Language (XAML).

Технологии, используемые в работе:

1. Объектно-ориентированный язык программирования C#;
2. Система для построения клиентских приложений WPF;
3. Технология для работы с базами данных ADO.NET;
4. Шаблон проектирования MVVM;
5. Среда разработки Visual Studio 2017;
6. Система управления реляционными базами данных Microsoft SQL Server;
7. Язык T-SQL для написания запросов к БД;
8. Клиент-серверное взаимодействие по протоколу TCP.

**1. Аналитический обзор литературы**

* 1. Анализ прототипов

В данном обзоре я буду сравнивать наиболее популярные из них с аналогичной тематикой.

* «Domino’s pizza»

«Domino’s pizza» на данный момент является одной из наиболее популярных сетей пиццерий в мире. Сайт данной компании достаточно удобный для заказа пиццы.

Достоинства:

* Несколько категорий пицц.
* Удобное меню.
* Кастомизация пицц.
* Круглосуточная доставка.
* Красивый внешний вид сайта.

Недостатки:

* Отсутствие фильтрации пицц.

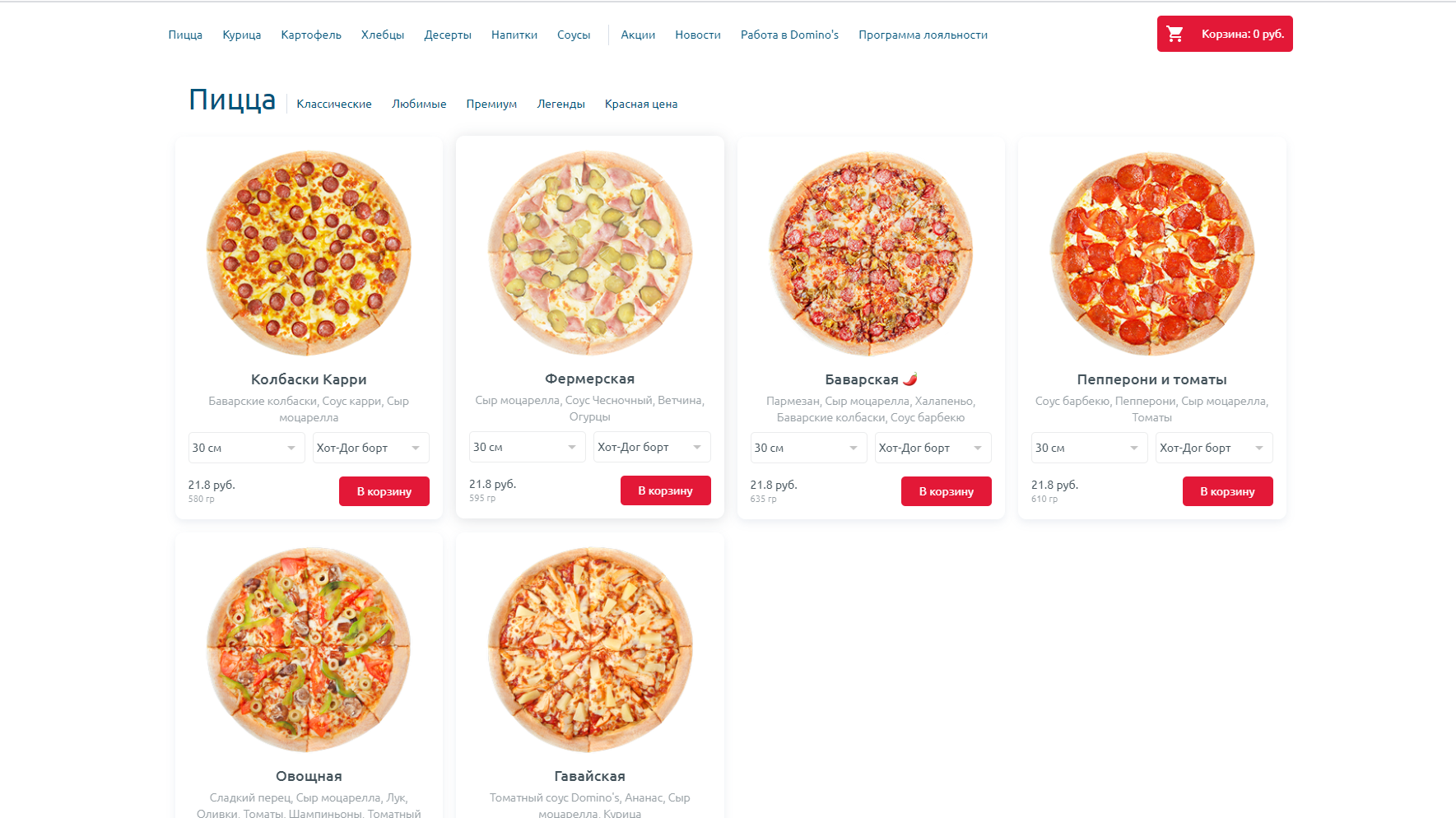


Рисунок 1.1 – Сайт сети пиццерий «Domino’s pizza»

* «Пицца-лисица»

«Пицца-лисица» является одной из наиболее популярных сетей пиццерий в Беларуси. Сайт данной компании достаточно удобный для заказа пиццы.

Достоинства:

* Удобное меню.
* Фильтрация пицц.
* Круглосуточная доставка.
* Красивый внешний вид сайта.

Недостатки:

* Отсутствие кастомизации пицц.

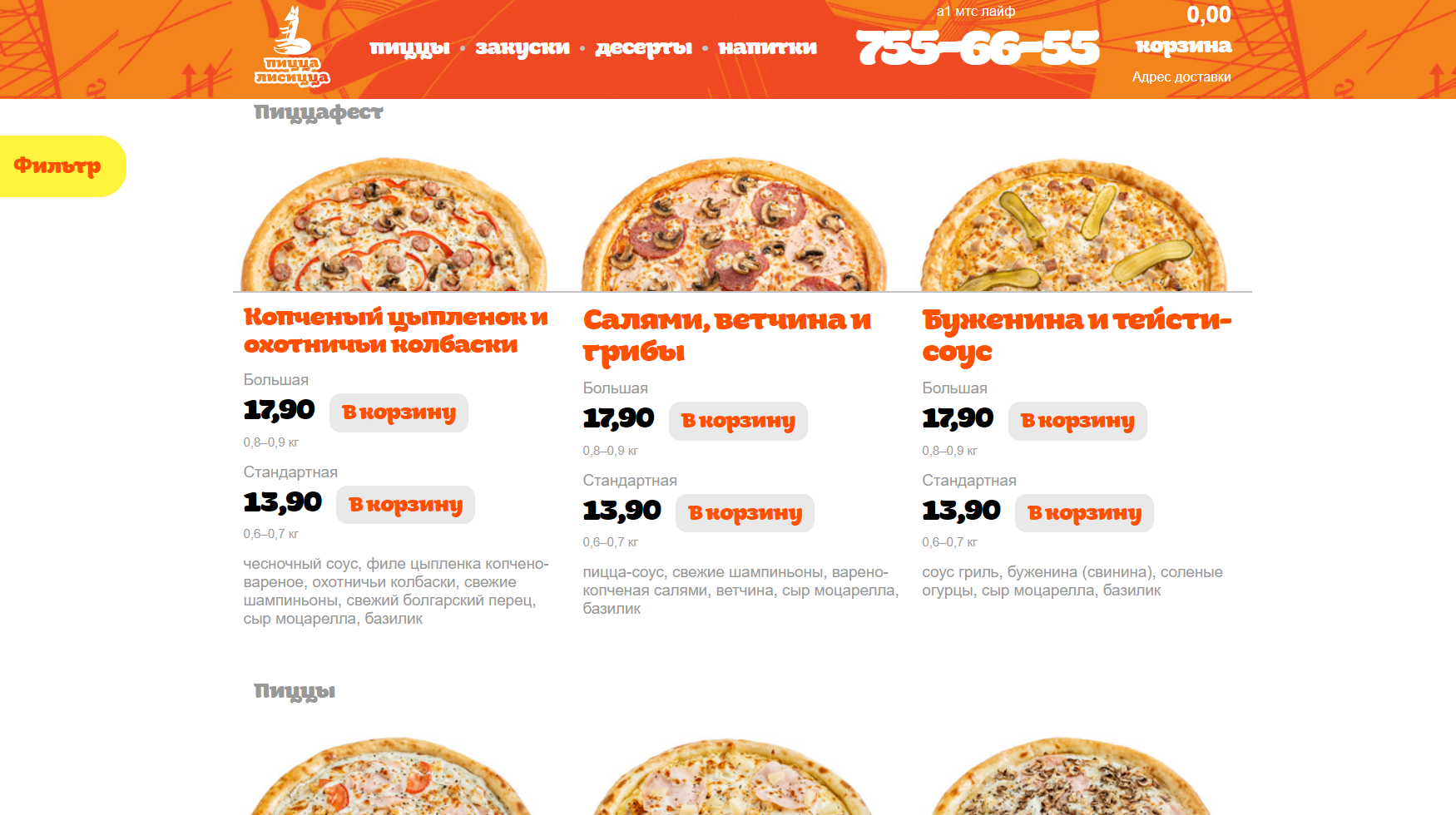


Рисунок 1.2 – Сайт сети пиццерий «Пицца-лисица»



# **Моделирование предметной области и разработка функциональных требований**

## 2.1. Описание инструментов разработки программного средства

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2019;
* программная платформа .NET Framework 4.7.2;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* ADO.NET
* Material Design Library;
* Протокол TCP.

*C#* — объектно-ориентированный язык программирования. Разработан в 1998 — 2001 под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework.

XAML — это декларативный язык разметки. С точки зрения модели программирования .NET Framework язык XAML упрощает создание пользовательского интерфейса для приложения .NET Framework.

WPF (Windows Presentation Foundation) — система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML.

ADO.NET — предоставляет собой технологию работы с данными, которая основана на платформе .NET Framework. Эта технология представляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Причем важно отметить, что систем управления баз данных может быть множество. В своей сущности они могут различаться. MS SQL Server, например, для создания запросов использует язык T-SQL, а MySQL и Oracle применяют язык PL-SQL. Разные системы баз данных могут иметь разные типы данных. Также могут различаться какие-то другие моменты. Однако функционал ADO.NET построен таким образом, чтобы предоставить разработчикам унифицированный интерфейс для работы с самыми различными СУБД.

Material Design Library — это исчерпывающая кросс-платформенная концепция визуального оформления приложений, затрагивающая правила отображения анимации, общий стиль приложения и правила взаимодействия с управляющими элементами.

TCP-протокол — огромное количество трафика и взаимодействия в сети сейчас происходит по протоколу TCP (Transmission Control Protocol). Этот протокол гарантирует доставку сообщений и широко используется в различных существующих на сегодняшний день программах.

Для работы с протоколом TCP в .NET предназначены классы TcpClient и TcpListener. Эти классы строятся поверх класса System.Net.Sockets.Socket. На основе этого протокола реализовано общение клиент-серверное общение.

## 2.2. Описание функций, реализуемых программой

Варианты использования программного средства показаны на диаграмме на рисунке 2.1.

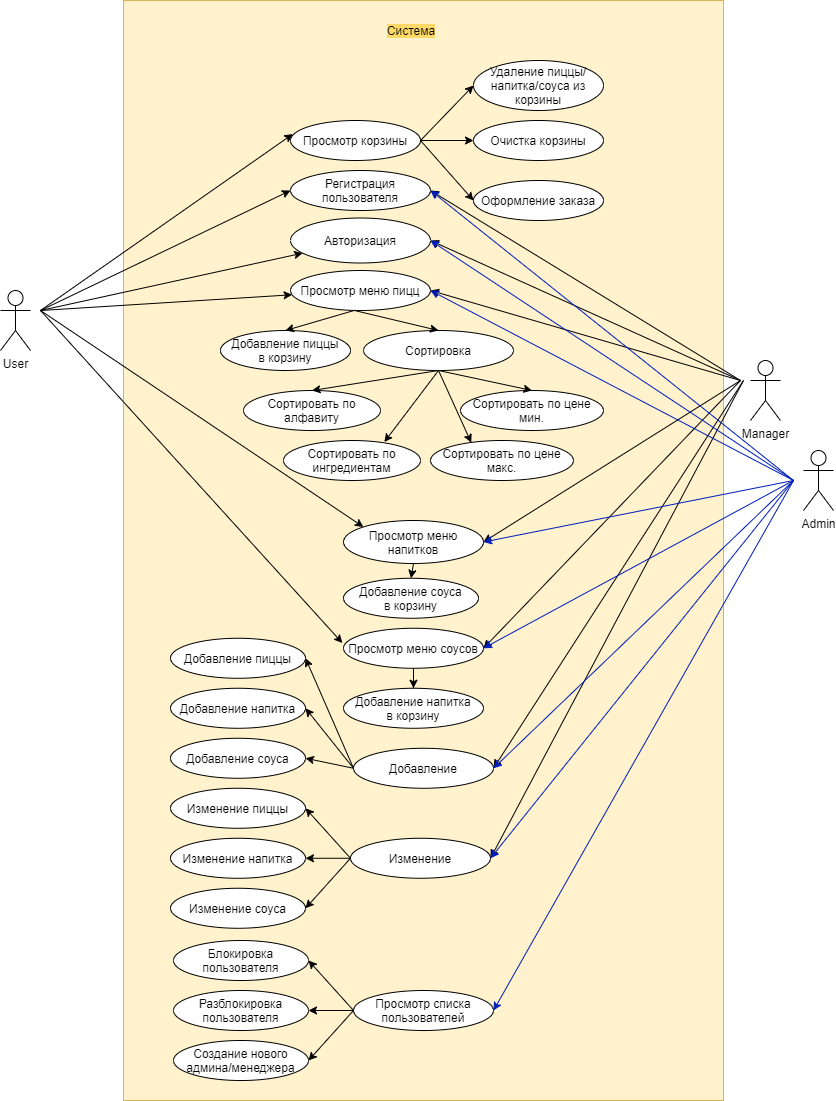


Рисунок 2.1 – Диаграмма вариантов использования (UML Case)

Сейчас же хотелось бы рассмотреть более подробно каждый из прецедентов.

Прецедент «Регистрация пользователя» заключается в регистрации нового пользователя с режимом доступа «User».

Прецедент «Авторизация» заключается в авторизации пользователя в системе.

Прецедент «Просмотр корзины» заключается в просмотре содержимого корзины пользователя. Доступ к ее просмотрю имеют только пользователи с режимом доступа «User».

Прецедент «Удаление пиццы/напитка/соуса из корзины» заключается в удалении выбранного продукта из корзины.

Прецедент «Очистка корзины» заключается в очистке корзины от всех имеющихся в ней продуктов.

Прецедент «Оформление заказа» заключается оформлении пользователем заказа. Перед пользователем появляется список всех продуктов из корзины и итоговая стоимость заказа.

Прецедент «Просмотр меню пицц» заключается в просмотре всех имеющихся в ассортименте пицц.

Прецедент «Добавление пиццы в корзину» заключается в добавлении выбранной пиццы в корзину.

Прецедент «Сортировка» заключается в сортировке списка пицц по заданному критерию.

Прецедент «Сортировка по алфавиту» заключается в сортировке списка пицц в алфавитном порядке. Фильтр можно сбросить.

Прецедент «Сортировка по цене мин» заключается в сортировке списка пицц в по возрастанию цены. Фильтр можно сбросить.

Прецедент «Сортировка по цене макс» заключается в сортировке списка пицц по убыванию цены. Фильтр можно сбросить.

Прецедент «Сортировка по ингредиенту» заключается в сортировке списка пицц по выбранному ингредиенту. Пользователь может выбрать ингредиент, который будет обязательно присутствовать или отсутствовать в пицце. Фильтр можно сбросить.

Прецедент «Просмотр меню напитков» заключается в просмотре всех имеющихся в ассортименте напитков.

Прецедент «Добавление напитка в корзину» заключается в добавлении выбранного напитка в корзину.

Прецедент «Просмотр меню соусов» заключается в просмотре всех имеющихся в ассортименте соусов.

Прецедент «Добавление соуса в корзину» заключается в добавлении выбранного соуса в корзину.

Прецедент «Добавление» заключается в добавлении нового продукта. Доступно только пользователям с режимом доступа «Admin» и «Manager».

Прецедент «Добавление пиццы» заключается в добавлении новой пиццы.

Прецедент «Добавление напитка» заключается в добавлении нового напитка.

Прецедент «Добавление соуса» заключается в добавлении нового соуса.

Прецедент «Изменение» заключается в редактировании продуктов. Доступно только пользователям с режимом доступа «Admin» и «Manager».

Прецедент «Изменение пиццы» заключается в изменении выбранной пиццы.

Прецедент «Изменение напитка» заключается в изменении выбранного напитка.

Прецедент «Изменение соуса» заключается в изменении выбранного соуса.

Прецедент «Просмотр списка пользователей» заключается в просмотре всех зарегистрированных в системе пользователей. Доступно только пользователям с режимом доступа «Admin».

Прецедент «Блокировка пользователя» заключается в блокировке выбранного пользователя.

Прецедент «Разблокировка пользователя» заключается в разблокировке выбранного пользователя.

Прецедент «Создание выбранного пользователя» заключается регистрации нового пользователя с режимом доступа «User», «Manager» или «Admin»

Из всего вышесказанного следует, что программное средство работает следующим образом: изначально запускается главное окно MainWindow, в котором необходимо авторизоваться либо нажать на кнопку «Регистрация» и зарегистрироваться, после чего новый пользователь автоматически авторизируется. После авторизации, в зависимости от режима доступа пользователя, ему станет доступен основной функционал, предоставляемый программным средством. Итак, рассмотрим более подробно каждый из сценариев: администратора, менеджер и пользователя.

* Администратор

Если пользователь вошёл в систему в качестве администратора, то на главном окне MainWindow страница со списком пицц, появляются кнопки «Пицца», «Напитки», «Соусы» со списками соответствующих продуктов, а также кнопки «Добавить», «Редактор» и «Аккаунты». В окне с меню пицц может провести сортировку пицц по заданному критерию. Помимо этого, присутствует кнопка «Корзина», к которой администратор не имеет доступа, так как предполагается, что админ со своей учётной записи не может делать заказы, поскольку имеет возможность редактировать цены продуктов.

В окне добавления новых продуктов администратор может добавить новую пиццу, её описание, цену для трёх размеров (22, 30 и 36 см) и изображение самой пиццы, новый напиток, его цену для 0,5 и 1л и изображение, и соус, его цену и изображение.

В окне изменения продуктов администратор может изменить все характеристики выбранной пиццы, напитка или соуса.

В окне «Аккаунты» администратор может просмотреть список всех зарегистрированных в системе пользователей, их имя, фамилию, логин, режим доступа и статус блокировки. Также может заблокировать или разблокировать любого пользователя и создать нового пользователя с любым режимом доступа.

* Менеджер

Если пользователь вошёл в систему в качестве менеджера, то на главном окне MainWindow страница со списком пицц, появляются кнопки «Пицца», «Напитки», «Соусы» со списками соответствующих продуктов, а также кнопки «Добавить», «Редактор» и «Аккаунты». В окне с меню пицц может провести сортировку пицц по заданному критерию. Помимо этого, присутствует кнопка «Корзина», к которой менеджер не имеет доступа, так как предполагается, что менеджер со своей учётной записи не может делать заказы, поскольку имеет возможность редактировать цены продуктов.

В окне добавления новых продуктов администратор может добавить новую пиццу, её описание, цену для трёх размеров (22, 30 и 36 см) и изображение самой пиццы, новый напиток, его цену для 0,5 и 1л и изображение, и соус, его цену и изображение.

В окне изменения продуктов администратор может изменить все характеристики выбранной пиццы, напитка или соуса.

* Пользователь

Если пользователь вошёл в систему в качестве пользователя, то на главном окне MainWindow страница со списком пицц, появляются кнопки «Пицца», «Напитки», «Соусы» со списками соответствующих продуктов. В окне с меню пицц может провести сортировку пицц по заданному критерию. Помимо этого, присутствует кнопка «Корзина». В окне корзины пользователь может просмотреть выбранные продукты, очистить корзину или сделать заказ, если корзина не пуста. Если пользователь нажмёт на кнопку «Оформить заказ», перед ним откроется список с продуктами из корзины и итоговая стоимость заказа. Далее у пользователя попросят его мобильный номер, адрес доставки и способ оплаты товара заказа при получении.

# **3. Проектирование программного средства**



## Проектирование базы данных

Для работы с базой дынных выбрана технология ADO.NET. Эта технология представляет нам набор классов, через которые мы можем отправлять запросы к базам данных, устанавливать подключения, получать ответ от базы данных и производить ряд других операций.

Я выбрала данную технологию, т.к. мне удобно самостоятельно прописывать с помощью кода создание базы данных и таблиц, что позволяет работать с базой на более низком уровне.

Ниже приведена схема взаимодействия приложения с пользователем.

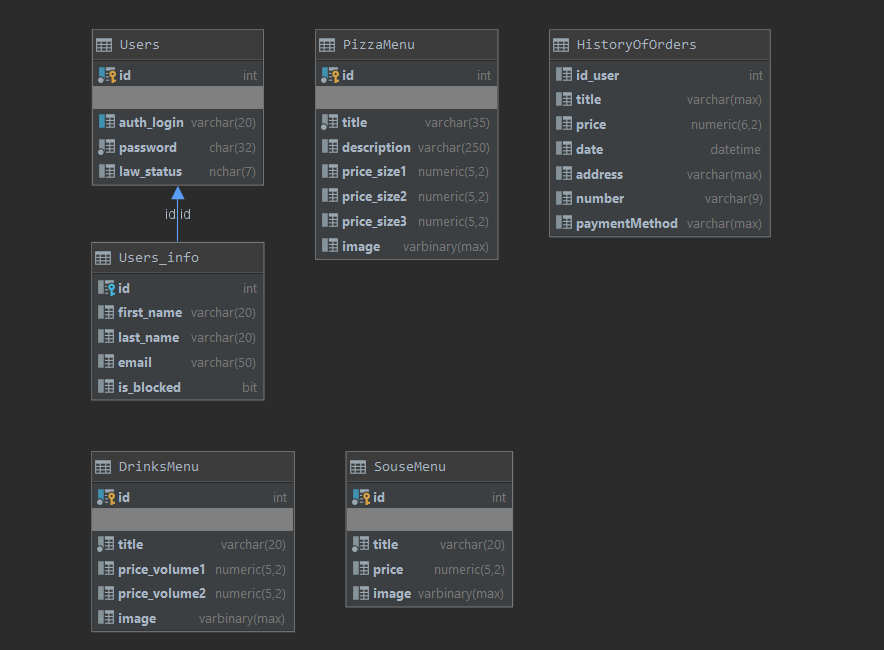


Рисунок. 3.1 – Диаграмма базы данных

Всего в базе данных 6 таблиц. В таблицах Users и Users\_info хранятся данные о всех пользователях, зарегистрированные в программе. Таблицы PizzaMenu, DrinksMenu и SouseMenu хранят себе информацию о пиццах, напитках и соусах соответственно. Таблица HistoryOfOrders содержит себе информацию о заказах.

Таблица Users состоит из 4 столбцов, иллюстрирующих основную информацию о пользователе:

* id – хранит идентификатор пользователя;
* auth\_login – хранит логин пользователя;
* password – хранит пароль пользователя в зашифрованном виде;
* law\_status – хранит режим доступа пользователя;

Таблица Users\_info состоит из 5 столбцов, иллюстрирующих дополнительную информацию о пользователе:

* id - хранит идентификатор пользователя;
* first\_name – хранит имя пользователя;
* last\_name – хранит фамилию пользователя;
* email – хранит почту пользователя;
* is\_blocked - хранит булевую переменную, показывающую состояние блокировки пользователя;

Таблица PizzaMenu состоит из 7 столбцов, иллюстрирующих полную информацию о списке пицц:

* id - хранит идентификатор пиццы;
* title– хранит название пиццы;
* description – хранит описание пиццы;
* price\_size1 – хранит цену пиццы для размера 22 см;
* price\_size2 – хранит цену пиццы для размера 22 см;
* price\_size3 – хранит цену пиццы для размера 22 см;
* image – хранит изображение пиццы;

Таблица DrinksMenu состоит из 5 столбцов, иллюстрирующих полную информацию о списке напитков:

* id - хранит идентификатор напитка;
* title– хранит название напитка;
* price\_volume1 – хранит цену напитка для объема 0,5 литра;
* price\_volume2 – хранит цену напитка для объема 1 литр;
* image – хранит изображение напитка;

Таблица SouseMenu состоит из 4 столбцов, иллюстрирующих полную информацию о списке соусов:

* id - хранит идентификатор соуса;
* title– хранит название соуса;
* price – хранит цену соуса;
* image – хранит изображение соуса;

Таблица HistoryOfOrders состоит из 7 столбцов, иллюстрирующих полную информацию о списке заказов:

* id\_user - хранит идентификатор пользователя;
* title– хранит название продукта или null;
* price – хранит стоимость отдельного продукта или полную стоимость заказа;
* data – хранит дату заказа;
* address – хранит адрес доставки;
* number – хранит номер пользователя;
* paymentMethod – хранит способ оплаты заказа;

## Определение последовательности передачи данных

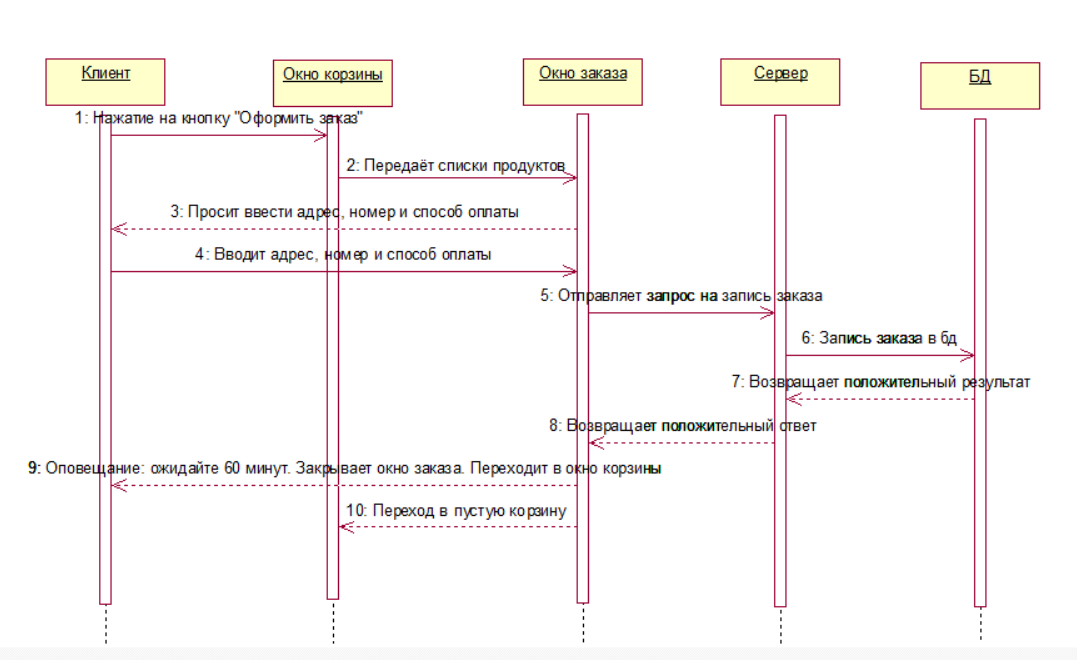
Также стоит определить, каким образом будет происходить взаимодействие между пользователем и данными в базе данных. Для этого рассмотрим пример оформления заказа, используя диаграмму последовательности (рис 3.2) 

Рисунок. 3.2 – Диаграмма последовательности

Как можно увидеть, в процессе оформления заказа участвуют 5 объектов: клиент, окно корзины, окно заказа, сервер и база данных. Процесс запускает клиент, нажимая на кнопку оформления заказа в корзине. Это все осуществляется при помощи интерфейса, который, обработав событие, передаёт список продуктов в окно заказа, которое, в свою очередь отображает список продуктов и итоговую сумму клиенту, требуя от него ввести адрес доставки заказа, мобильный номер клиента и способ оплаты товара. Окно заказа, получив данные от клиента, отправляет заказ на сервер, который, в свою очередь, отправляет данные на запись в БД. После успешной записи сервер отправляет окну заказа пустой объект, свидетельствующий об успешной обработке заказа. Окно заказа уведомляет клиента, что заказ принят, закрывается и открывает окно уже пустой корзины.

## Диаграмма действий

Данная диаграмма отображает процесс деятельности пользователя в процессе работы с приложением.

Диаграмма деятельности позволяет любому, кто выполняет данный процесс, выбирать порядок действий. Другими словами, диаграмма только устанавливает правила обязательной последовательности действий, которым пользователь должен следовать. Это важно для моделирования бизнес-процессов, поскольку эти процессы часто выполняются параллельно.

На рисунке 3.3 представлена диаграмма деятельности приложения «Grandma pizza».

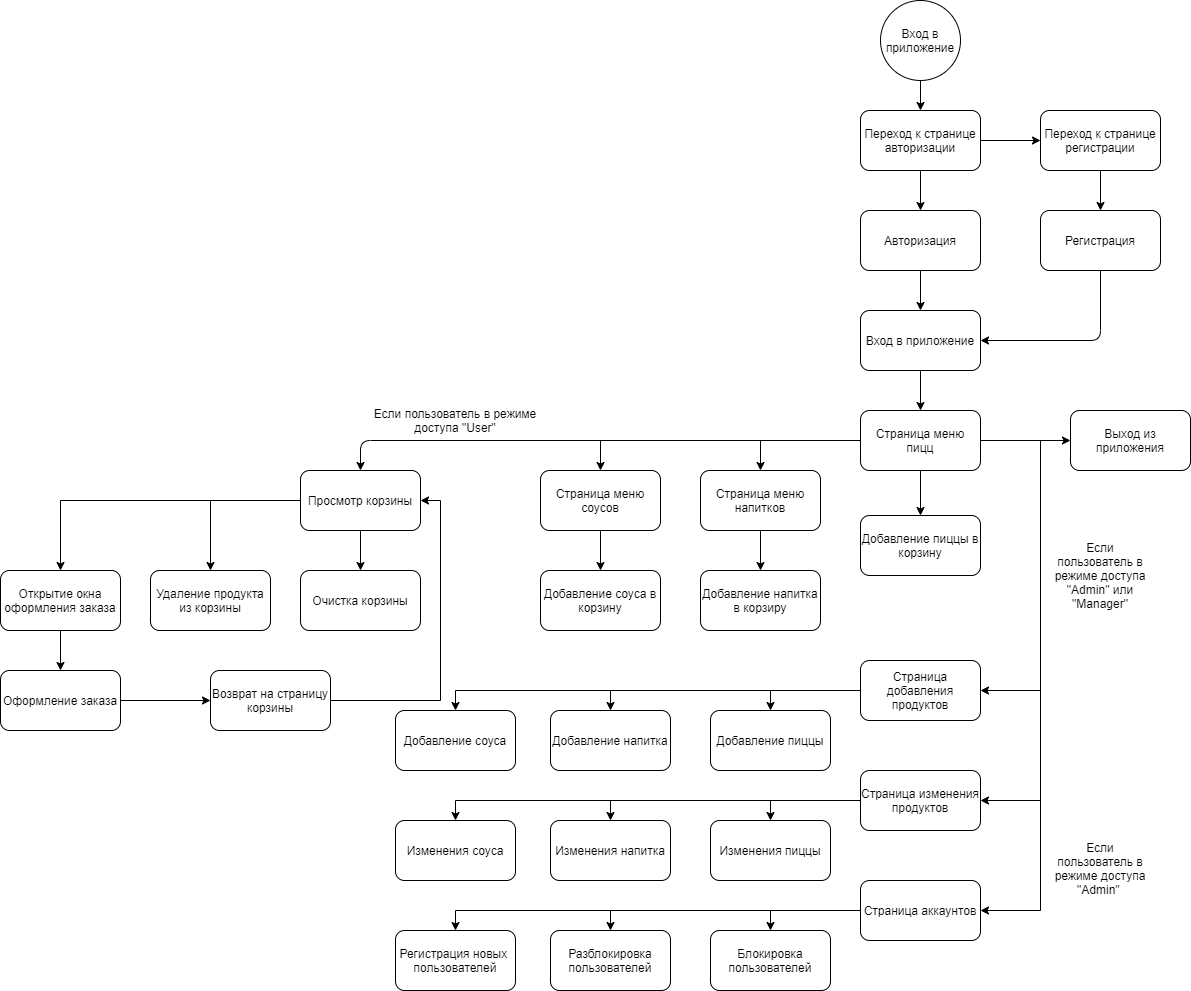


Рисунок 3.3 – Диаграмма деятельности

# **Создание (реализация) программного средства**

## 4.1. Реализация библиотеки классов

Библиотека классов представляет собой описание классов, на основе которых происходит клиент-серверное взаимодействие. Диаграмма классов изображена на рисунках 4.1.

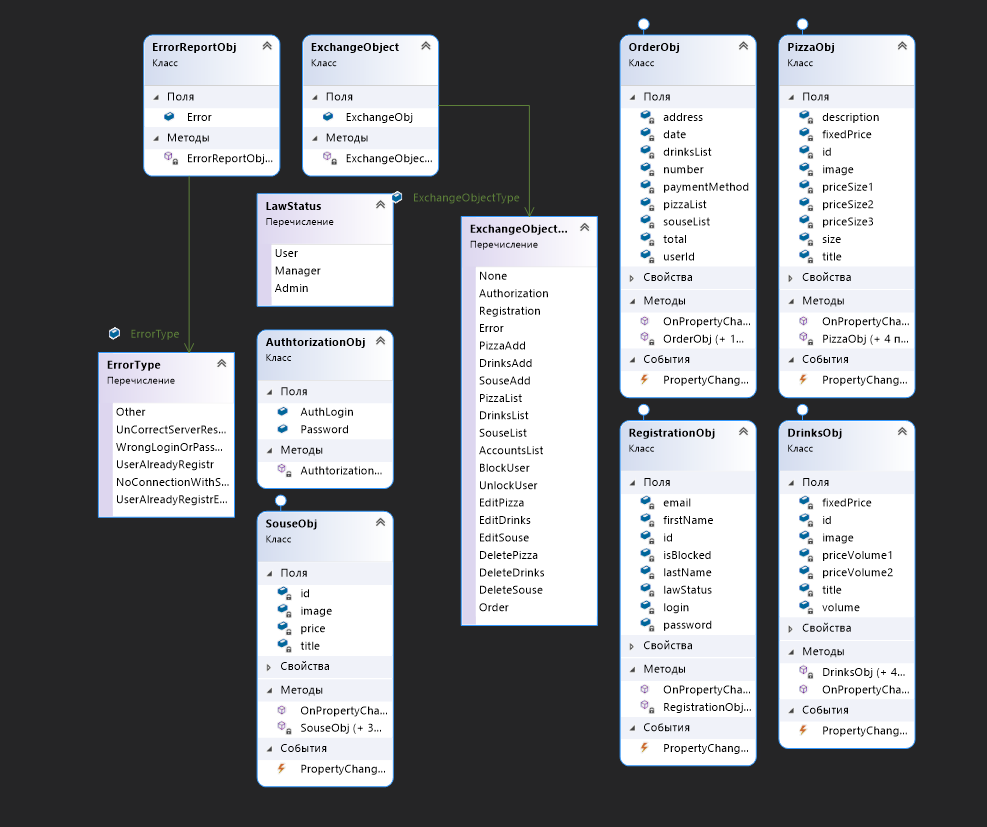


Рисунок 4.1. – Диаграмма классов библиотеки

Главным классом во взаимодействии выступает класс ExchangeObject. В поле ExchangeObjectType хранится значение перечисления, которое сигнализирует о типе объекта, упакованного в ExchangeObject.

Благодаря этому классу сервер и клиент могут взаимодействовать через одну сущность.

## Реализация серверной части

В роли сервера выступает консольное приложение .NET Framework. Диаграмма классов серверной части представлена в Приложении Б.

Сервер реализован на основе TCP-протокола. Он является многопоточным, при каждом запросе с клиентской части на сервер создается новый поток, в котором запрос обрабатывается и возвращается ответ. На рисунке 4.4 приведен пример кода, где реализовано создание потока под каждый запрос с клиентской части.

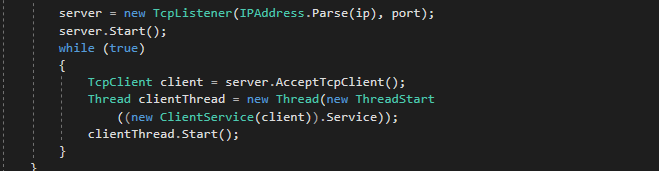


Рисунок 4.2. – Реализация многопоточного сервера

Функция AcceptTcpClient возвращает TCP-оболочку сокета пользователя. Пока на сервер не поступает никаких запросов он находится в режиме ожидания.

В функции Service объекта ClientService происходит обработка запроса клиента и отправка соответствующего ответа.

На рисунке 4.3 представлен пример кода получения данных из пользовательского потока и приведения их к нужному типу данных.

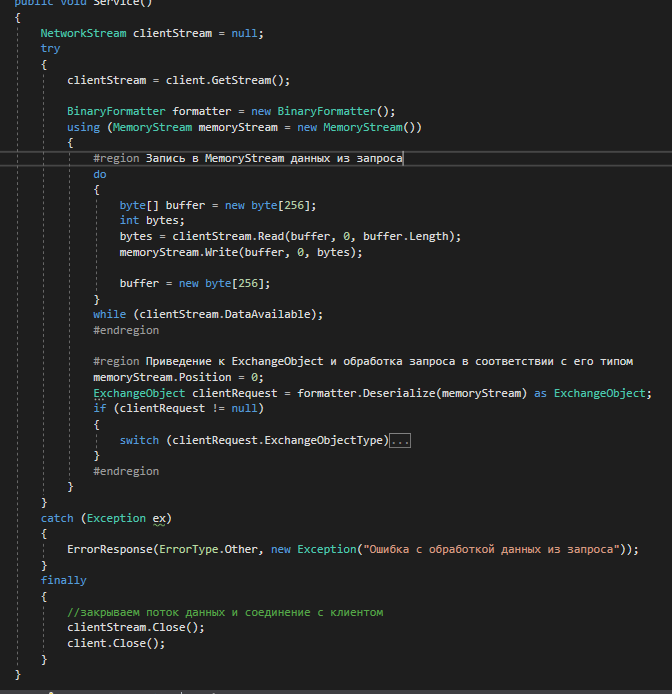


Рисунок 4.3 – Листинг функции Service

## Реализация клиентской части приложения

При создании приложения были реализованы такие паттерны как MVVM и Command.

Структура клиентской части приложения представлена на рисунке 4.7.

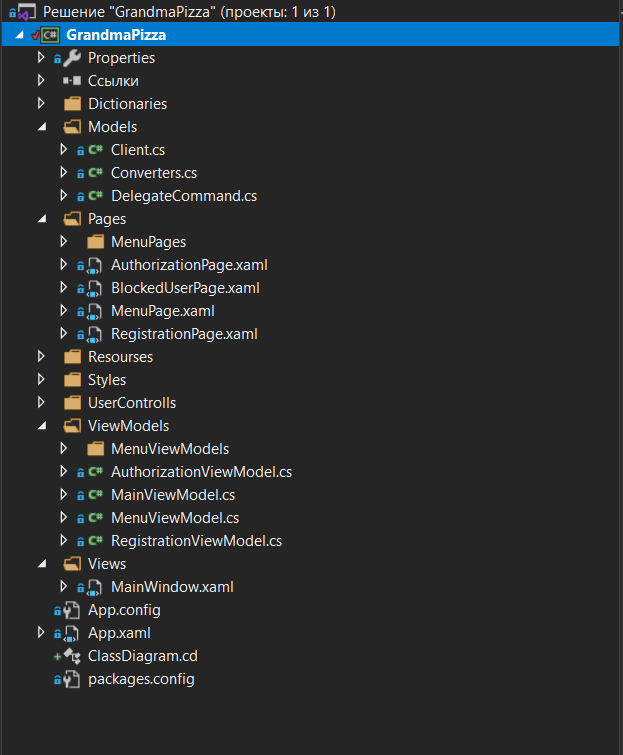


Рисунок 4.4. – Структура клиентской части приложения

Как показано на рисунке, все элементы проекта, согласно паттерну MVVM, рассредоточены в специальных папках для создания целостности и упрощения организованности приложения:

* Models - модель данных, описывает используемые в приложении данные. Модели содержат ту логику, непосредственно связанную с этими данными, но не содержат никакой логики, связанной с отображением данных и взаимодействием с визуальными элементами управления
* Pages – основные страницы окна приложения;
* Resources и Styles - содержат в себе словари ресурсов, необходимые для задания стилей для элементов управления и установления языка приложения;
* UserControlls – собственные элементы управления для предотвращения дублирования кода и сохранения единообразия стилей;
* ViewModels - модель представления, связывает модель и представление через механизм привязки данных;
* View - содержит в себе все представления, которое определяет визуальный интерфейс, через который пользователь взаимодействует с приложением.

Диаграмма классов клиентской части приложения представлена в Приложении А.

Класс Client – является статическим классом, с помощью которого отправляются единоразовые запросы на сервер. Для каждого нового запроса в нем создается новое подключение и после того, как ответ от сервера поступил, если он ожидается, то подключение закрывается.

# **Тестирование программного средства**

При тестировании данного приложения были проведены следующие тесты на ситуации, которые могли бы привести к ошибке:

* тесты авторизации и регистрации;
* тест некорректно введённых данных при добавлении нового продукта;
* Попытка введения неверного пароля:

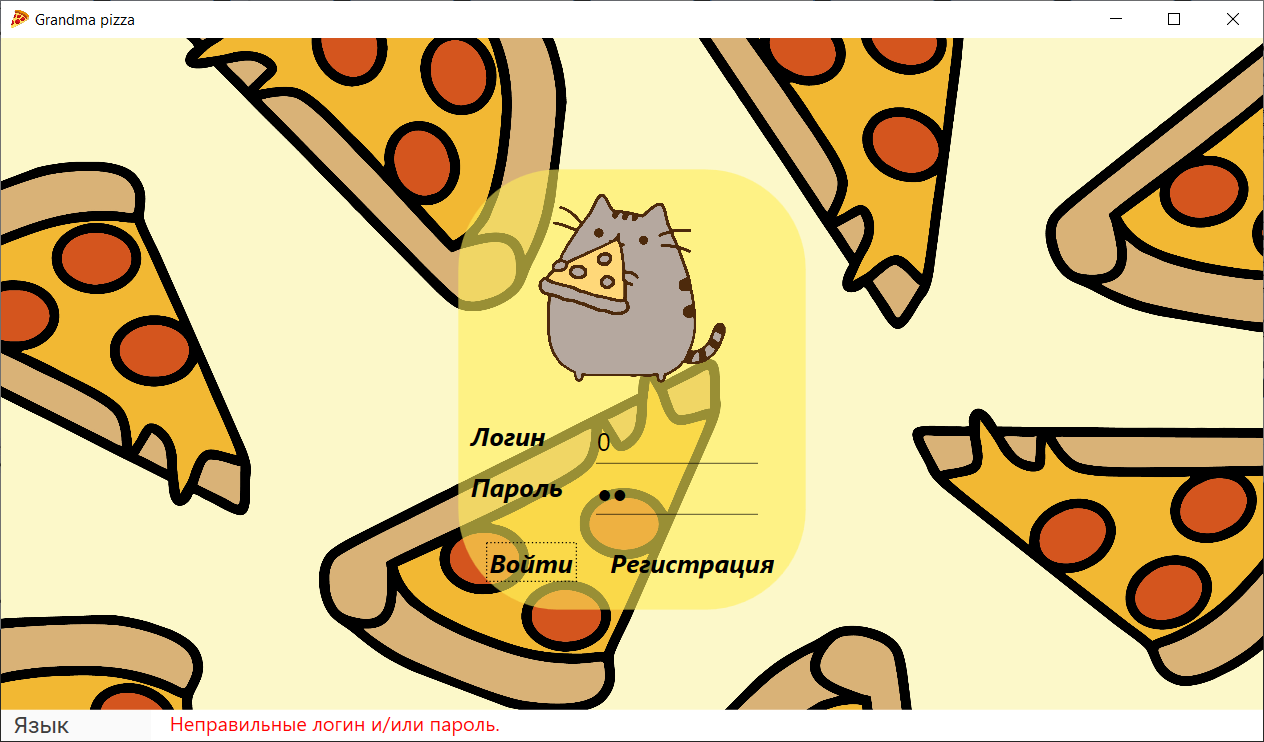


Рисунок 5.1 – Пользователь ввёл неверный пароль

– Попытка ввода некорректных данных при регистрации:

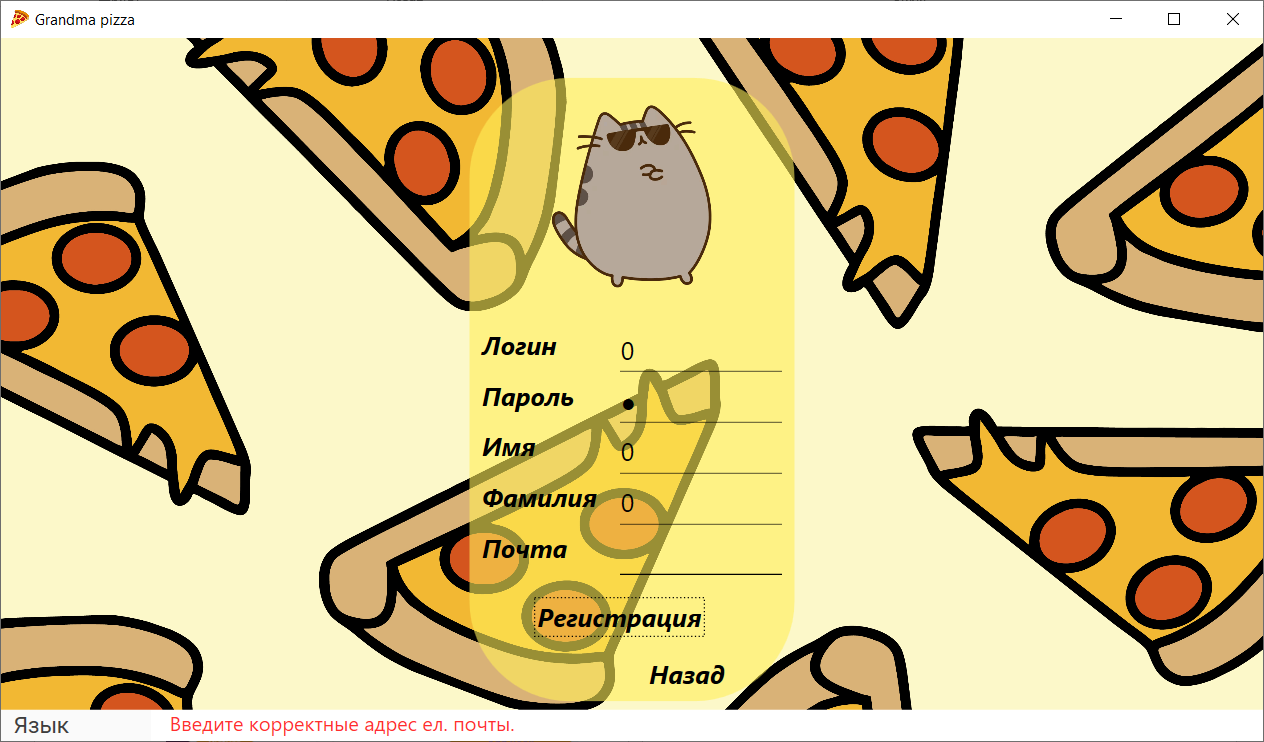


Рисунок 5.2 – Пользователь ввёл некорректные данные

– Попытка ввода некорректных данных при добавлении новой пиццы:

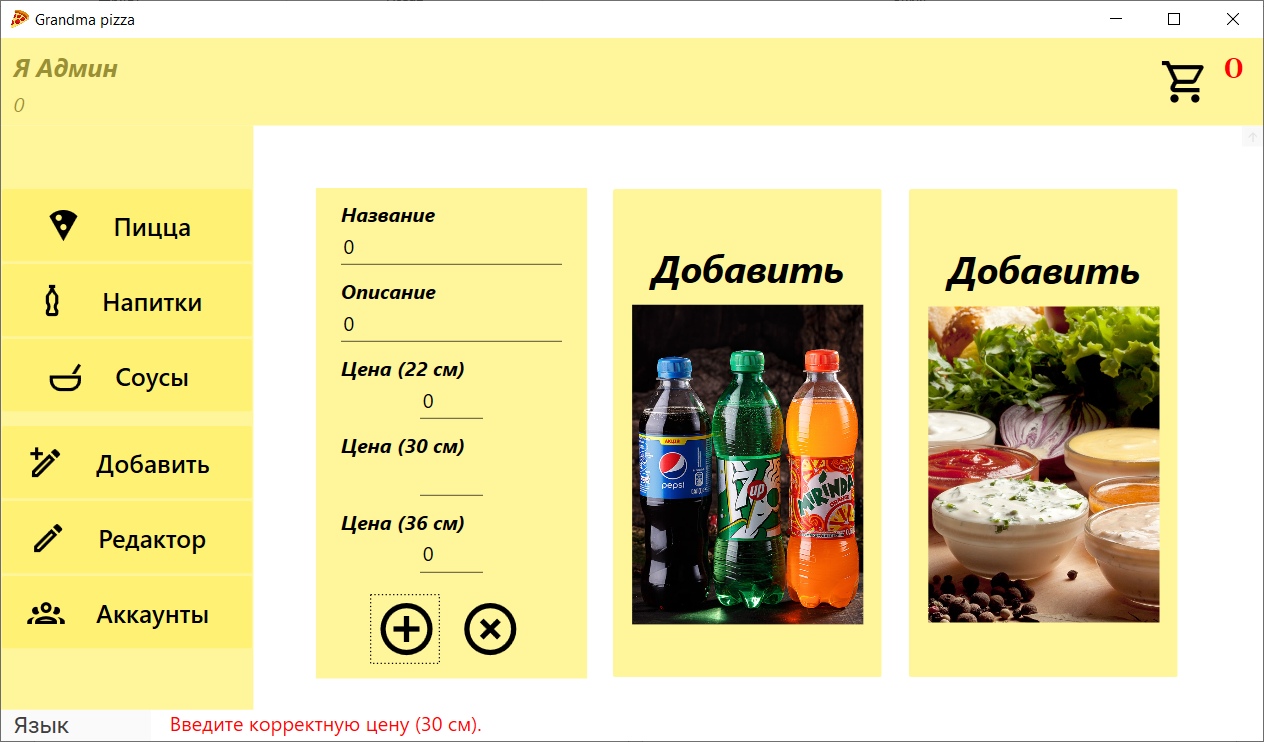


Рисунок 5.3 – Администратор ввёл некорректные данные

– Попытка ввода некорректных данных при редактировании пиццы:

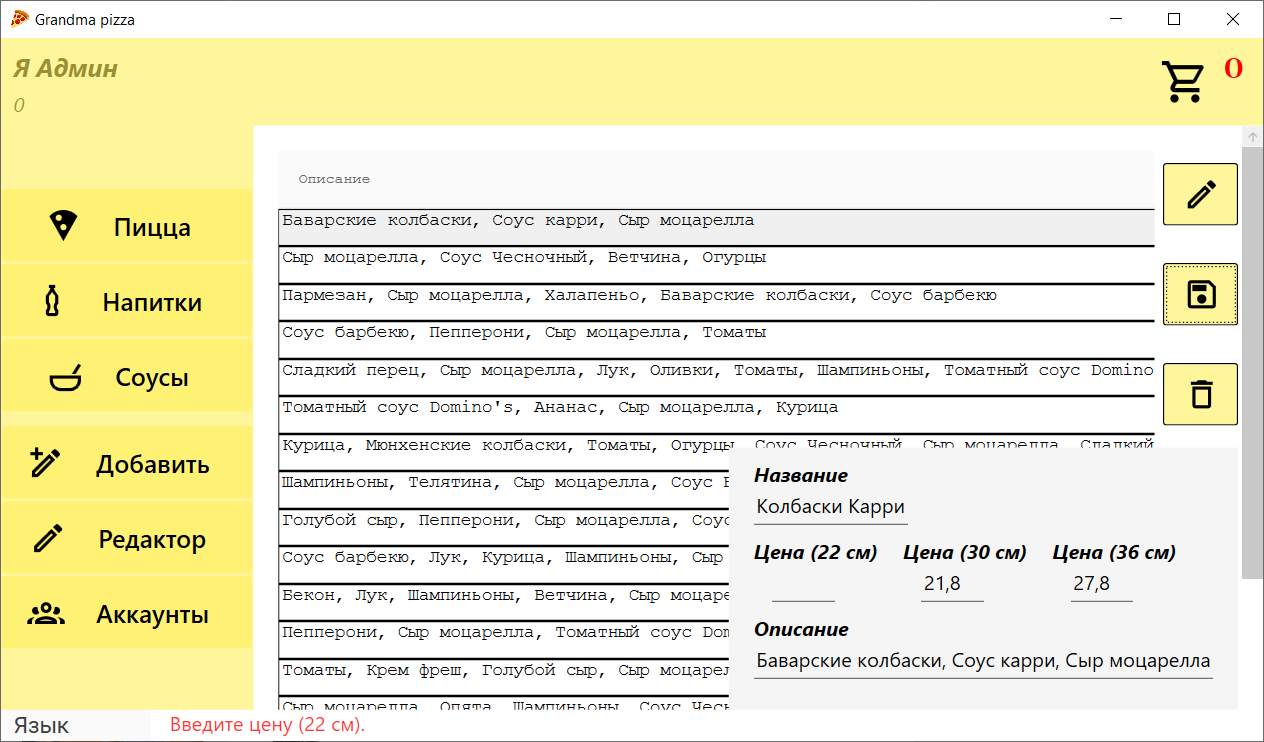


Рисунок 5.4 – Администратор ввёл некорректные данные

**Заключение**

В процессе решения поставленной задачи была достигнута поставленная цель по созданию программного средства «Grandma pizza». При разработке были выполнены все пункты из указанного списка предполагаемого основного функционала приложения.

В программном средстве были реализованы следующие функции:

Для любого режима доступа:

* Смена языка приложения с русского на английский и наоборот;
* Просмотр меню пицц, напитков и соусов;
* Сортировка списка пицц по критерию;

Пользователь:

* Выполнение регистрации и авторизации
* Добавление выбранных продуктов в корзину и удаление из нее;
* Просмотр содержимого корзины, очистка корзины;
* Возможность оформления заказа;

Менеджер:

* Добавление новых продуктов в меню пицц, напитков и соусов;
* Редактирование продуктов в меню пицц, напитков и соусов;

Менеджер:

* Поддерживает функционал менеджера;
* Блокировка / разблокировка пользователе;
* Регистрация новых пользователей с любым режимом доступа.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает верно, а требования технического задания выполнены в полном объеме.

**Список использованных источников**

1. Пацей, Н.В. Курс лекций по языку программирования С# / Н.В. Пацей. – Минск: БГТУ, 2016. – 175 с.
2. Пацей, Н.В. Технология разработки программного обеспечения / Н.В. Пацей. – Минск: БГТУ, 2016. – 129 с.
3. MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Режим доступа: <http://msdn.microsoft.com/library/rus/>. Дата доступа: 20.04.2018
4. METANIT.COM Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com>. Дата доступа: 20.05.2018
5. ProfessorWeb .NET & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru> Дата доступа: 13.04.2018
6. Youtube.com C# WPF Material Design UI Menu [Электронный ресурс]/ Режим доступа: <https://www.youtube.com/watch?v=YQ1EJJZBHyE&list=LLJYz_lF4d1mOUKPGhpU53AQ&index=2&t=584s>

**Приложение А**

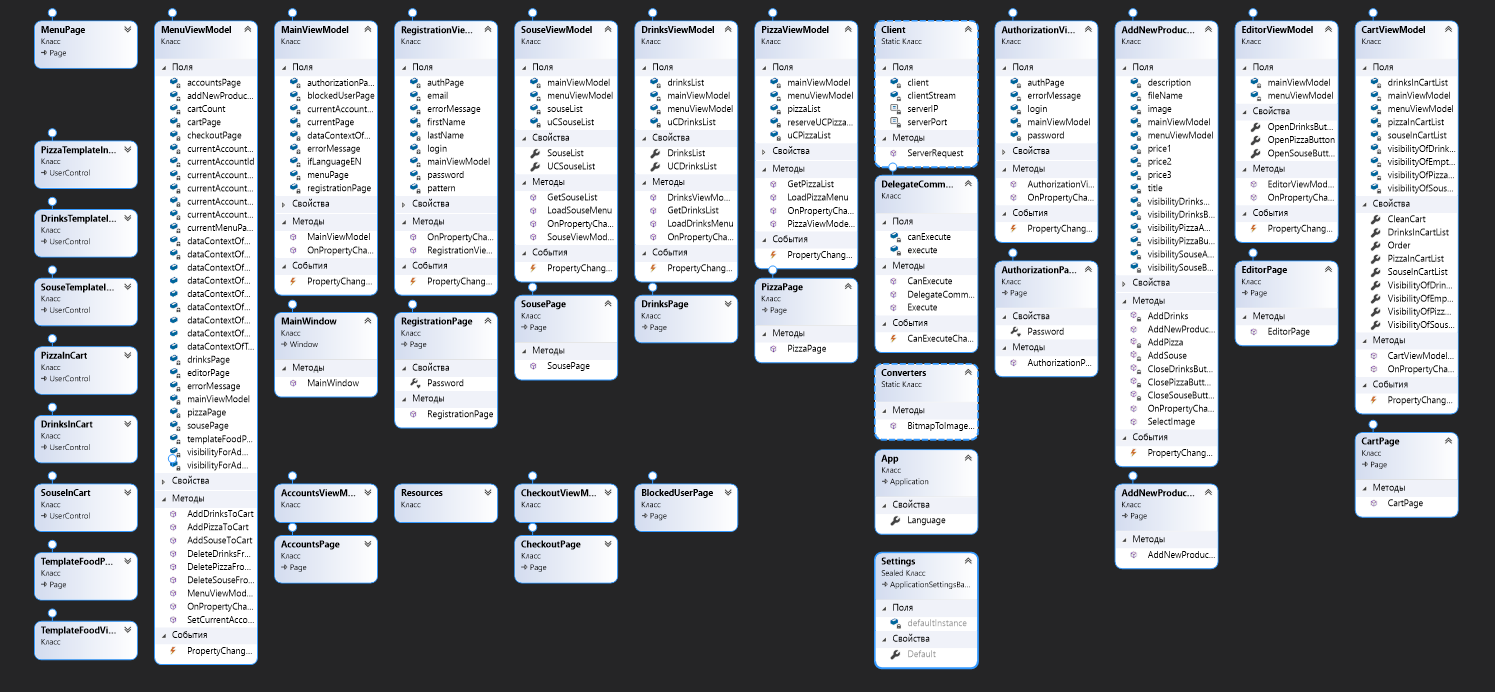


Рисунок – Диаграмма классов клиента

**Приложение Б**

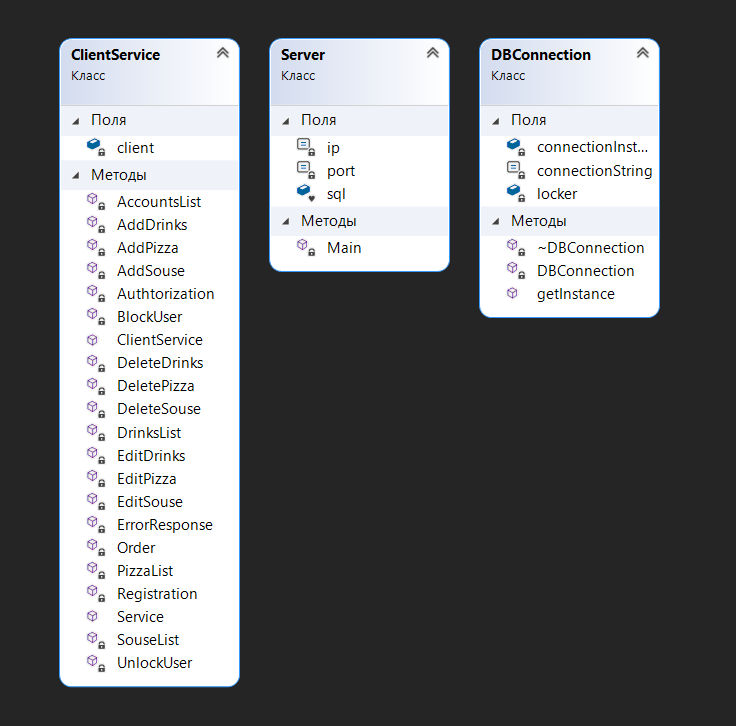


Рисунок – Диаграмма классов сервера