МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-47 01 02 «Дизайн электронных и веб-изданий

Направление специальности 1-47 01 02 «Дизайн электронных и веб-изданий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированное программирование»

Тема Программное средство «Поставка товаров из IKEA»

Исполнитель

студент 2 курса группы 9 Змитревич Диана Андреевна

(Ф.И.О.)

Руководитель работы Гук В.С.

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

Курсовой проект защищен с оценкой

Председатель Белькевич Р.И.

(подпись)

Минск 2022

**Содержание**

[**ВВЕДЕНИЕ** 3](#_Toc73065782)

[**1 Аналитический обзор литературы и постановка задачи** 4](#_Toc73065783)

[1.1 Анализ прототипов 4](#_Toc73065784)

[1.2 Постановка задачи 6](#_Toc73065785)

[**2 Анализ требований к программному средству и разработка функциональных требований** 7](#_Toc73065786)

[2.1 Определение требований к программному средству 7](#_Toc73065787)

[2.2 Описание средств разработки 7](#_Toc73065788)

[2.3 Описание функциональности программного средства 8](#_Toc73065789)

[2.4 Спецификация функциональных требований 8](#_Toc73065790)

[**3 Проектирование и создание программного средства** 10](#_Toc73065791)

[3.1 Архитектура системы 10](#_Toc73065792)

[3.2 Диаграммы UML 10](#_Toc73065793)

[3.3 Проектирование логической структуры базы данных 11](#_Toc73065794)

[3.4 Структура проекта 12](#_Toc73065795)

[**4 Реализация программного средства** 14](#_Toc73065796)

[4.1 Реализация MVVM 14](#_Toc73065797)

[4.2 Реализация авторизации и регистрации пользователей 15](#_Toc73065798)

[4.3 Добавление и удаления товара из корзины 18](#_Toc73065799)

[4.4 Отправка сообщений на почту и генерация чека 20](#_Toc73065801)

[4.5 Формирование заказа 21](#_Toc73065802)

[**5 Тестирование** 24](#_Toc73065803)

[**6 Руководство по использованию** 28](#_Toc73065804)

[6.1 Регистрация и авторизация 28](#_Toc73065805)

[6.2 Использование приложения обычным пользователем 29](#_Toc73065806)

[6.3 Использование приложения администратором 29](#_Toc73065807)

[**ЗАКЛЮЧЕНИЕ** 31](#_Toc73065808)

[**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ** 32](#_Toc73065809)

[Приложение А 33](#_Toc73065810)

[Приложение Б 34](#_Toc73065811)

[Приложение В 35](#_Toc73065812)

[Приложение Г 36](#_Toc73065813)

[Приложение Д 37](#_Toc73065814)

# **ВВЕДЕНИЕ**

У многих крупных и не очень организаций уже существует возможность просмотра и приобретения продукции через веб-сайты и приложения. Поэтому темой курсового проекта является разработка программного средства «Поставка товаров из IKEA». Данное программное средство должно позволить пользователям в лице клиентов заказывать продукцию на единой площадке и иметь возможность отслеживать статус заказа. Пользователи в лице администраторов могут добавлять новые товары, категории, просматривать заказы и отвечать на отзывы. Так же администраторы имеют возможность удалять товары и чистить заказы.

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.

Хранение данных осуществляется в Microsoft SQL Server.

# 

# **1 Аналитический обзор литературы и постановка задачи**

Для того чтобы окончательно определиться с постановкой задачи курсового проекта, необходимо проанализировать прототипы программных средств выбранной темы.

## **1.1 Анализ прототипов**

Ikea.ru ­– платформа, на которой предоставлен широкий выбор товаров для дому и другой продукции (рисунок 1.1).

Достоинства:

* большой выбор товаров и их категоризация
* личный кабинет
* доставка
* приятный глазу дизайн

Недостатки: не выявлены.

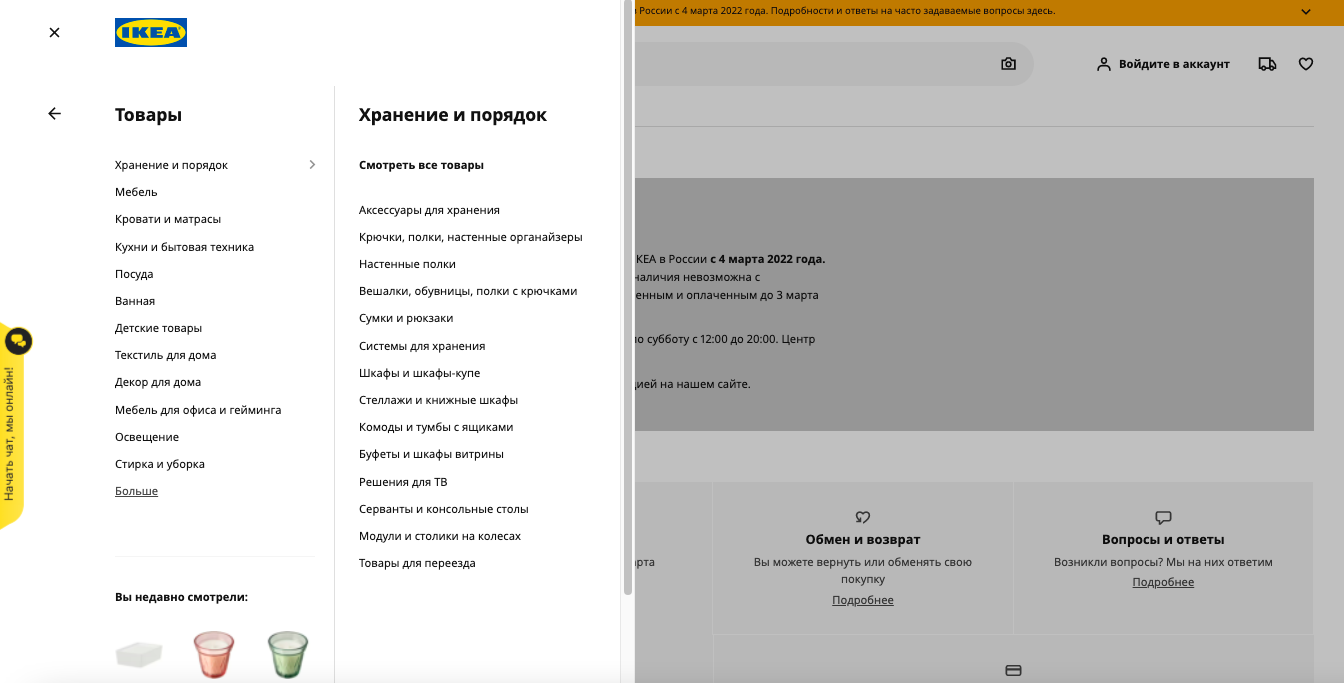


Рисунок 1.1 – Интерфейс веб-приложения «Ikea.ru»

Аikea.by – платформа для заказа товаров из IKEA (рисунок 1.2).

Достоинства:

* аналогичный выбор товаров, как в Ikea.ru
* наличие корзины
* наличие доставки

Недостатки:

* нагруженный интерфейс
* существует только веб-приложение
* отсутствует рассылка по почте

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1.2 – Интерфейс веб-приложения «Аikea.by»

Ikeaby.by – сервис предоставляющий доставку шведской мебели из IKEA (рисунок 1.3).

Достоинства:

* широкий выбор товаров
* наличие доставки

Недостатки:

* устаревший и нагруженный интерфейс
* отсутствие рассылки по почте
* отсутствие корзины



Рисунок 1.3 – Интерфейс веб-приложения «Ikeaby.by»

## **1.2 Постановка задачи**

Проанализировав прототипы, были выделены основная задача и функциональные требования проекта.

Основной задачей курсового проекта является разработка десктопного приложения, позволяющего выполнять авторизацию и регистрацию пользователей, пользователю в лице администратора добавлять продукты с информацией о них с возможностью удаления и редактирования. Пользователю в лице клиента просматривать все продукты, добавлять их в корзину, оставлять комментарии и задавать вопросы под детальными карточками товаров, удалять и оформлять заказ с возможностью прислать детали заказа себе на почту, указанную при регистрации.

Функциональные требования описаны в главе 2.

# **2 Анализ требований к программному средству и разработка функциональных требований**

Для того, чтобы приступить к этапу разработки проекта, необходимо четко сформулировать функциональные требования к программному средству.

## **2.1 Определение требований к программному средству**

Функционально ПС должно выполнять следующие задачи:

* регистрация и авторизация пользователей;
* сохранение рабочей информации в централизованной базе данных;
* для администратора добавление товаров с возможностью их удаления;
* для клиента добавление товаров в корзину с возможностью их удаления;
* для клиента заказ товара из корзины и его подтверждение;
* отправка деталей заказа посредством отправки электронного письма;
* для администратора просмотр всех товаров, комментариев и заказов с возможностью удаления.

## **2.2 Описание средств разработки**

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;
* технология Entity Framework 6.4.4;
* Microsoft SQL Server 2019;

В качестве интерфейса прикладного программирования был выбран обширный API-интерфейс – Windows Presentation Foundation (WPF), предназначенный для создания настольных программ с графически насыщенным пользовательским интерфейсом. В основе графической технологии WPF лежит мощная инфраструктура, основанная на DirectX. Это является одним из основных отличий WPF от более ранней технологии создания пользовательских интерфейсов – Windows Forms.

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

Чтобы осуществлять связь между базой данных и приложением на C# необходим посредник. И именно таким посредником является технология Entity Framework. Она предоставляет собой объектно-ориентированную технологию доступа к данным, является object-relational mapping (ORM) решением для платформы .NET Framework. Entity Framework предоставляет возможность взаимодействия с объектами посредством LINQ to Entities. Центральной концепцией Entity Framework является понятие сущности или entity. Сущность представляет набор данных, ассоциированных с определенным объектом. Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами.

## **2.3 Описание функциональности программного средства**

Описание функциональности программного средства представлено с помощью UML-диаграммы вариантов использования (Приложение А).

Спецификация функциональности программного средства представлена в следующем пункте.

## **2.4 Спецификация функциональных требований**

Для функциональности ПС необходимо создание базы данных для хранения информации приложения. Подробно о базе данных описано в главе 3.

В программном средстве при запуске необходимо реализовать регистрацию и авторизацию пользователей для дальнейшего использования приложения. Для авторизации входными параметрами являются логин и пароль пользователя, которые содержатся в базе данных. Для того, чтобы зарегистрироваться необходимо ввести имя, адрес электронной почты, логин и пароль. Введенные данные, успешно прошедшие валидацию, заносятся в базу данных.

При авторизации пользователя он может в окне «*MainWindow*» просматривать товары, а также осуществлять поиск путем выбора категории товара. В окне «*MyOrdersWindow*» пользователь может просматривать свои заказы. В окне «*CouponsWindow*» пользователь может просматривать существующие купоны и по желанию кликом скопировать из код в буфер обмена. В окне «*MyReviewsWindow*» пользователь может просматривать свои отзывы/вопросы оставленные на карточках товаров и отслеживать на них ответы. В окне «*CartWindow*» пользователь может отслеживать и удалять товары, добавленные в корзину и после заполнения обязательных полей при помощи кнопки оформить заказ. В окне «*ReviewWindow*» пользователь может добавить отзыв/вопрос.

При авторизации администратора в окне «*AdminMainWIndow*» он может просматривать категории всех товаров, добавлять, удалять и редактировать их. В окне «*AdminCouponsWindow*» администратор может просматривать список всех купонов, редактировать, добавлять и удалять их. В окне «*AdminReviewsWindow*» администратор может просматривать список всех отзывов, удалять их и добавлять на них ответ. В окне «*AdminOrdersWindow*» администратор может просматривать все заказы, удалять их или добавлять к ним соответствующий статус.

Кнопка для выхода из приложения, как и на публичной панели, так и административной, должна возвращать пользователя к окну авторизации.

Все данные о товарах, пользователях и заказах должны заноситься в базу данных в соответствующие таблицы.

# **3 Проектирование и создание программного средства**

Проектирование программного средства — процесс создания проекта программного обеспечения. Целью проектирования является определение внутренних свойств системы и детализации её внешних свойств на основе исходных условий задачи. Исходные условия задачи уже были сформулированы во втором разделе данной пояснительной записки. Этап проектирования подразумевает их анализ.

## **3.1 Архитектура системы**

В данном приложении используется архитектурный паттерн MVVM (Model-View-ViewModel). Он позволяет отделить логику приложения от визуальной части. MVVM состоит из трех частей:

* Модель описывает используемые в приложении данные.
* Представление определяет визуальный интерфейс, через который пользователь взаимодействует с приложением.
* Модель представления связывает модель и представление через механизм привязки данных. ViewModel также содержит логику по получению данных из модели, которые потом передаются в представление.

На рисунке 3.1 представлена диаграмма, которая показывает общую структуру приложения в рамках шаблона MVVM.



Рисунок 3.1 – Структура шаблона MVVM

## **3.2 Диаграммы UML**

UML — унифицированный язык моделирования (Unified Modeling Language), это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования.

Его можно использовать для визуализации, спецификации, конструирова­ния и документирования программных систем.

Диаграмма — это графическое представление набора элементов, чаще всего изображенного в виде связного графа вершин (сущностей) и путей (связей).

Диаграммы последовательностей используются для уточнения диаграмм прецедентов, более детального описания логики сценариев использования. Это отличное средство документирования проекта с точки зрения сценариев использования.

Диаграмма последовательности взаимодействия с окном оформления заказа приведена в приложении Б.

Диаграмма 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. Данная диаграмма приведена в приложении В.

Диаграммы использования показывает актеров и возможные прецеденты, при работе с приложением. Диаграмма использования приведена в приложении А.

Диаграмма классов служит для представления статической структуры мо­дели системы в терминологии классов объектно-ориентированного программирования.

Диаграмма классов ViewModel приведена в приложении Г

## **3.3 Проектирование логической структуры базы данных**

Для реализации поставленной задачи была создана база данных. Для её создания использовались система управления реляционными базами данных Microsoft SQL Server 2019.

База данных – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимо от прикладных программ.

Система управления базами данных – совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами. База данных нашей программной подсистемы состоит из 9 таблиц, ее схема изображена в приложении Д.

Таблица CATEGORIES используется для хранения данных о категориях товаров. Включает 2 столбца: CategoryId – id категории, Name – имя категории.

Таблица COUPONS используется для хранения данных о купонах. Включает 6 столбцов: CouponId – id купона, CouponCode – обязательный код купона в шесть символов, IsPercent – является ли скидка купона в процентах, PercentOff – сумма скидки в процентах, MoneyOff – сумма скидки в денежных единицах, CouponText – текст купона.

Таблица ORDERED-PRODUCTS используется для хранения заказанных товаров. Включает 5 столбцов: OrderedProductId – id заказанного продукта, OrderedQuantity – заказанное количество, LinkToOrderId – id товара, к которому идет привязка, OrderedProdcutPrice – цена заказанного продукта, OperationId – уникальный идентификатор операции.

Таблица ORDERS используется для хранения информации о заказах. Включает 8 столбцов: OrderId – id заказа, LinkAccountId – id аккаунта, к которому осуществляется привязка заказа, OrderPrice – цена заказа без купона, Coupon – код купона, FinalOrderPrice – цена заказа с купоном, PaymеntMethod – способ оплаты, Adress – адрес заказа, Status – статус заказа.

Таблица PERSONAL-ACCOUNTS используется хранения данных об аккаунтах. Состоит из 3 столбцов: AccountId – id аккаунта, Name – имя пользователя, которому принадлежит аккаунт, Email – почта пользователя, которому принадлежит аккаунт.

Таблица PRODUCTS используется для хранения данных о товарах. Включает 6 столбцов: ProductId – id товара, LinkToCategoryId – id категории для привязки, Name – название товара, Description – описание товара, Price – цена товара, Quantity – доступное количество для заказа.

Таблица PRODUCTS-DETAIL используется для хранения информации о детальных страница товаров. Состоит из 7 столбцов: DetailId – id детальной страницы, LinkToProductId – id товара для привязки, ProductCode – шестизначный код товара, Material – описание материала товара, Package – описание упаковки товара, Size – описание размера товара, BigDescription – детальное описание товара.

Таблица REVIEWS хранит информацию об отзывах. Состоит из 6 столбцов: ReviewId – id отзыва, LinkToProductId – id товара для привязки, LinkUserLogin – логин пользователя для отображения на отзыве, ReviewText – текст отзыва, AdminText – текст ответа администратора приложения, AdminName – имя администратора, которое будет отображаться в ответе.

Таблица USERS используется для хранения данных о привязанной карте для оплаты. Состоит из 5 столбцов: UserId – id пользователя, Login – логин пользователя, Password – хешированный пароль пользователя необходимый для входа в аккаунт, LinkAccounId – id аккаунта для привязки, IsAdmin – поле для проверки является ли пользователь администратором.

## **3.4 Структура проекта**

Структура проекта представлена на рисунке 3.2.

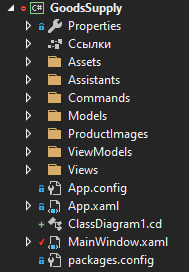


Рисунок 3.2 – структрура проекта

Папка «Assets» содержит файлы шрифтов ПС и различные графическими элементами.

Папка «Assistants» содержит классы для поддержки некоторых функций приложения.

Папка «Commands» содержит классы команд нужных для работы приложения.

Папка «Models» содержит классы моделей.

Папка «ProductImages» содержит изображения добавляемые в коде при добавлении товара.

Папка «ViewModels» содержит классы моделей представлений.

Папка «Views» содержит все представления. Приложение включает в себя 10 окон и 2 пользовательских элемента управления.

# **4 Реализация программного средства**

## **4.1 Реализация MVVM**

Паттерн MVVM реализуется через базовый класс BaseViewModel, который реализует интерфейс INotifyPropertyChanged и от которого наследуются все страницы. Для реализации паттерна файлы программы были распределены по соответствующим пространствам имен (см. 3.2) и реализованы следующие функции. На рисунке 4.1 видим, что класс MainWindowViewModel наследует класс BaseViewModel (рис. 4.2) .

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4.1 – пример использования ViewModelBase



Рисунок 4.2 – структура класса BaseViewModel

В приложении используется паттерн Command который позволяет инкапсулировать запрос на выполнение определенного действия в виде отдельного объекта. В WPF команды представлены интерфейсом ICommand. В приложении он представлен в виде собственной команды Command код которой приведен в приложении. Класс реализует два метода:

* CanExecute: определяет, может ли команда выполняться
* Execute: собственно, выполняет логику команды

## **4.2 Реализация авторизации и регистрации пользователей**

Для того чтобы пользоваться приложением каждый пользователь должен войти в систему, предварительно зарегистрировавшись.

На рисунке 4.3 находится код, реализующий регистрацию.

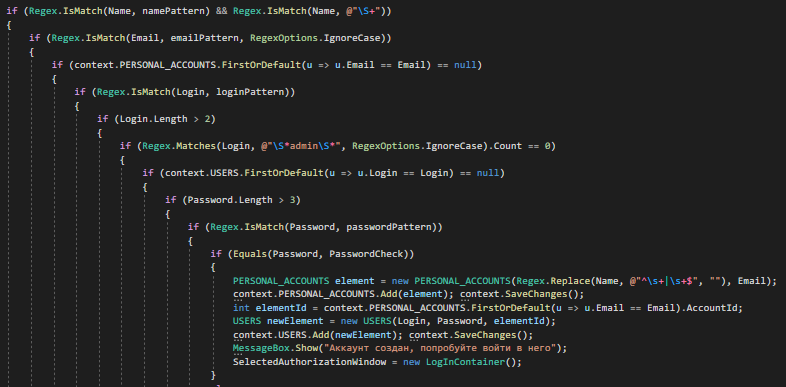


Рисунок 4.3 – Команда, реализующая регистрацию

Для безопас­ности хранения информации используется хэширование, в данном случае применяется встроенный класс MD5. Код метода хэширования приведен на рисунке 4.4.

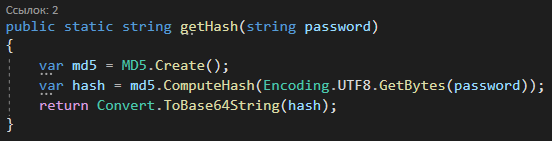


Рисунок 4.4 – Пример метода обеспечивающего хэширование пароля

Данные при регистрации проходят валидацию и поиск на соответствие с уже существующими данными в базе данных. Валидация реализована с использованием регулярных выражений.

После успешного создания аккаунта пользователь автоматически переходит на страницу авторизации. При авторизации проверяется существует ли пользователь с такими данными, а затем проверяется значения хэш-функций паролей. В случае неудовлетворения требований появляется надпись с ошибкой. В ином случае создается окно обычного пользователя или администратора в зависимости от значения в поле IsAdmin в базе данных. Пример команды, реализующей авторизацию приведен на рисунке 4.5.

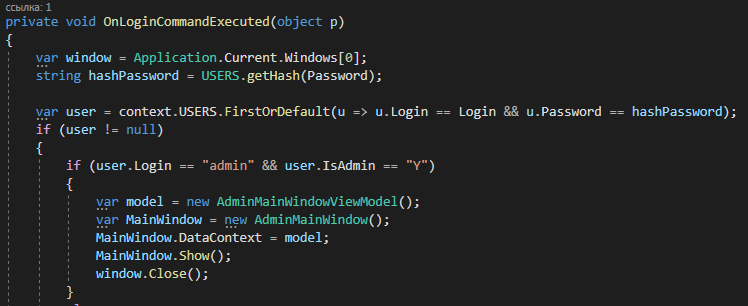


Рисунок 4.5 – Команда, реализующая авторизацию

## **4.3 Добавление и удаления товара из корзины**

Пример команды, реализующей добавление товара в корзину пользователя приведен на рисунке 4.6.

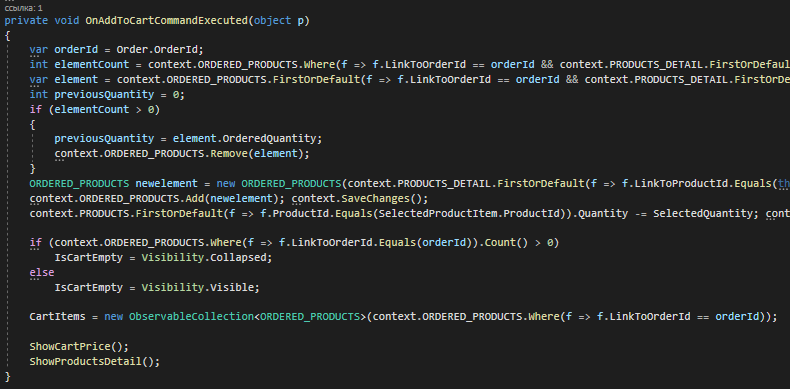


Рисунок 4.6 – Команда, реализующая добавление товара к корзину

Удаление товара из корзины пользователя производится с помощью команды привязанной на странице корзины пользователя. Пример команды, реализующей удаление приведен на рисунке 4.7.

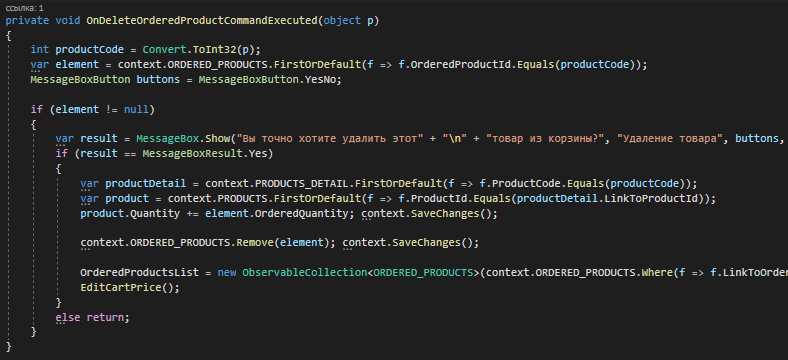


Рисунок 4.7 – Команда, реализующая удаление товара из корзины

## **4.4 Отправка сообщений на почту и генерация чека**

Для отправки сообщений на электронную почту пользователей используются классы, предоставленные пространством имен System.Net.Mail. Данное пространство имен предоставляет классы для работы с SMTP протоколом.

В качестве параметров в первом случае передается адрес электронной почты пользователя, код, тема и тело сообщения, а во втором – адрес электронной почты, тема и тело сообщения.

Пример отправки чека заказа приведен на рисунке 4.8.

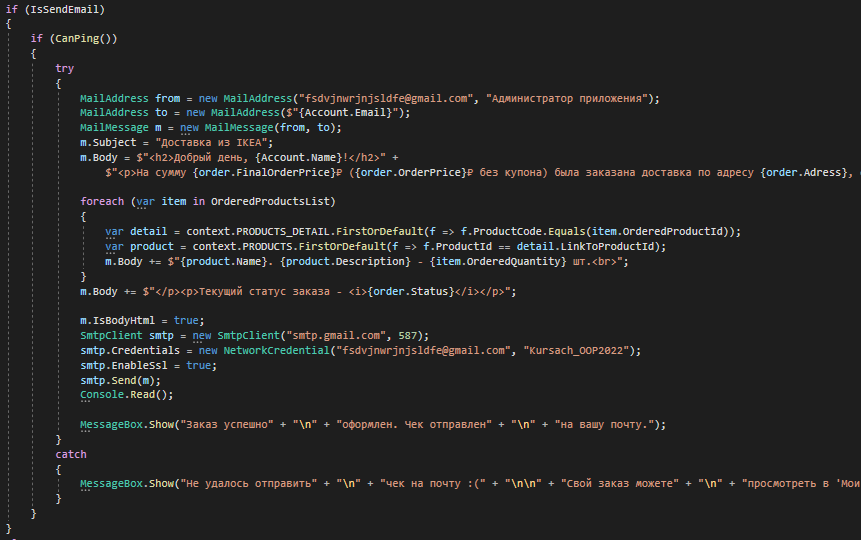


Рисунок 4.8 – Метод отправки чека заказа

## **4.6 Формирование заказа**

Заказ формируется с помощью метода AddOrderCommand. Наличие достаточного количества товаров на складе для оформления заказа проверяется другими методами в различных окнах программ, поэтому наличие дополнительной валидации в данном листинге может быть избыточным.

Пример метода, реализующего формирование заказа приведен на рисунке 4.9.

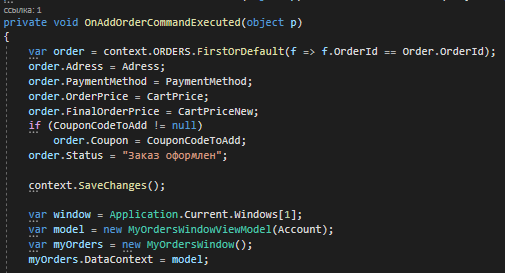


Рисунок 4.9 – Метод, реализующий формирование заказа

# **5 Тестирование**

В курсовом проекте задействуется обработка ошибок, таким образом, что пользователь будет уведомлен о неудачном выполнении запроса к базе данных, , или недоступности данных в формах.

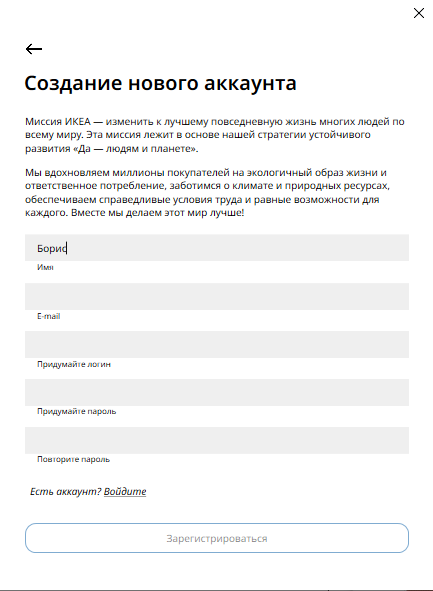


Рисунок 5.1 – Проверка на заполнения всех полей для регистрации

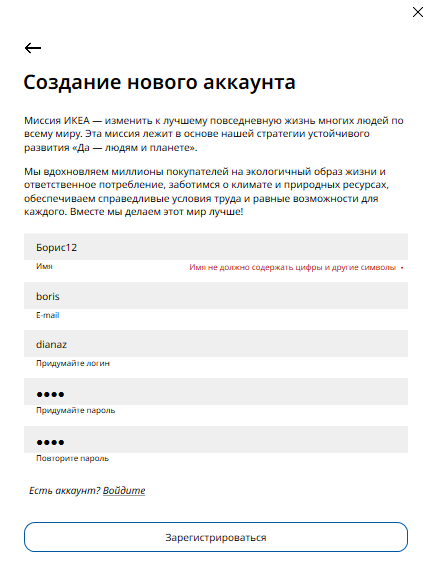


Рисунок 5.2 – Наличие лишних знаков при вводе имени

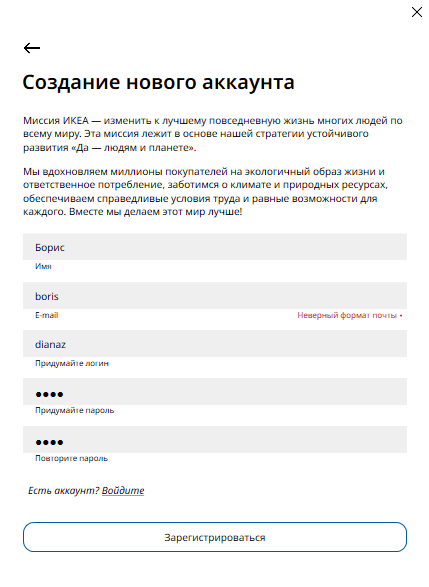


Рисунок 5.3 – Некорректно веденный адрес электронной почты

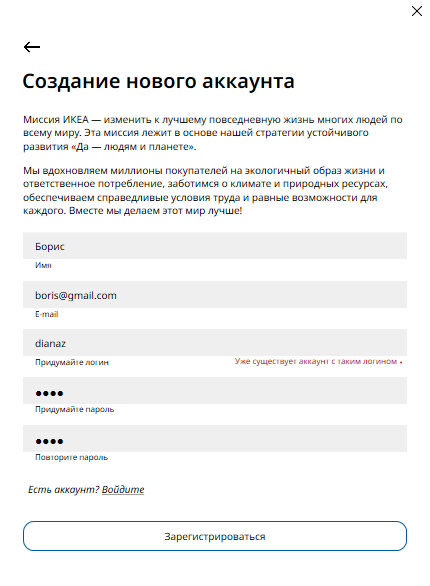


Рисунок 5.4 – Ошибка создания повторяющегося логина пользователя

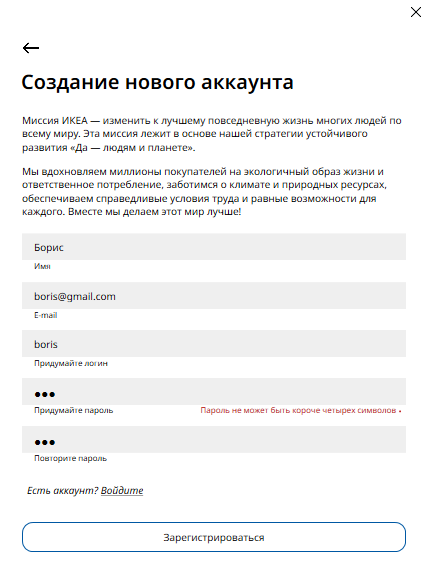


Рисунок 5.5 – Неверный формат пароля

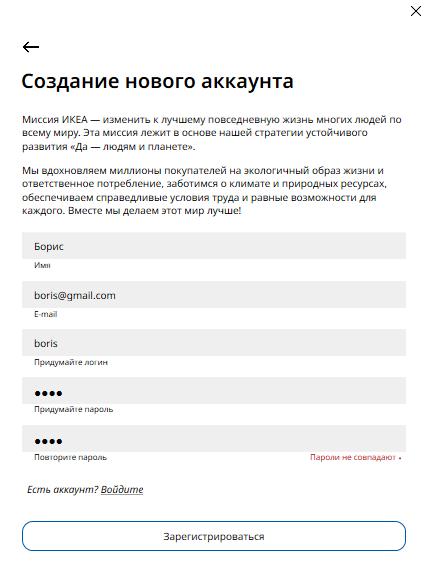


Рисунок 5.6 – Ошибка проверки соответствия паролей

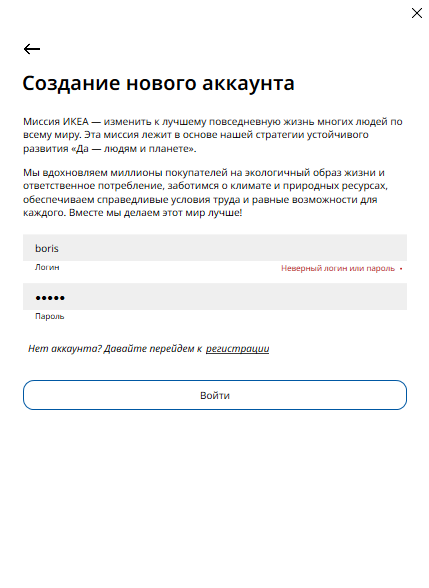


Рисунок 5.7 – Ошибка авторизации

# **6 Руководство по использованию**

## **6.1 Регистрация и авторизация**

После запуска приложения пользователь видит главное окно приложения, на котором он может авторизоваться и получить возможность приобретать товары. Окно представлено на рисунке 6.1.

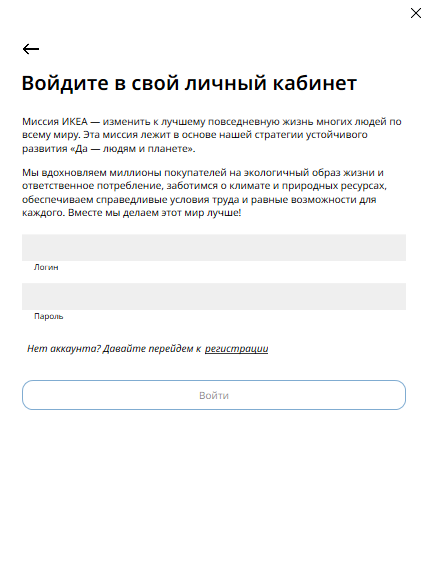


Рисунок 6.1 – Окно входа в аккаунт

Если пользователь ещё не имеет своего аккаунта, то по нажатию на соответствующую кнопку можно перейти к окну регистрации. Пользователь должен придумать логин и пароль, которые будет использовать для авторизации в приложении. Окно регистрации показано на рисунке 6.2.

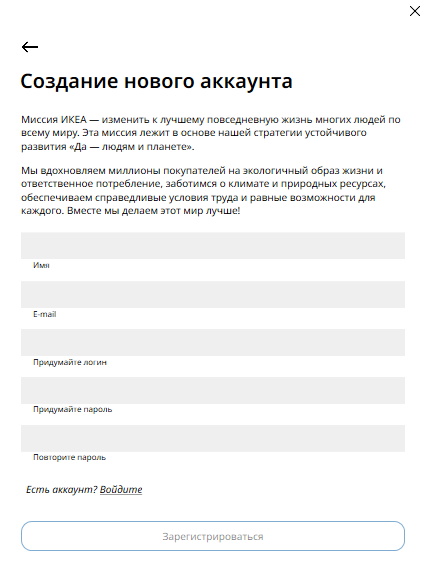


Рисунок 6.2 – Окно регистрации

## **6.2 Использование приложения обычным пользователем**

После успешной авторизации пользователь попадает на главный экран, где находятся все товары магазина, а также предоставлен поиск по выбранной категории. Так же на главном окне имеется 5 кнопок для навигации по страницам и выхода из приложения.

Стартовое окно пользователя приведено на рисунке 6.3.

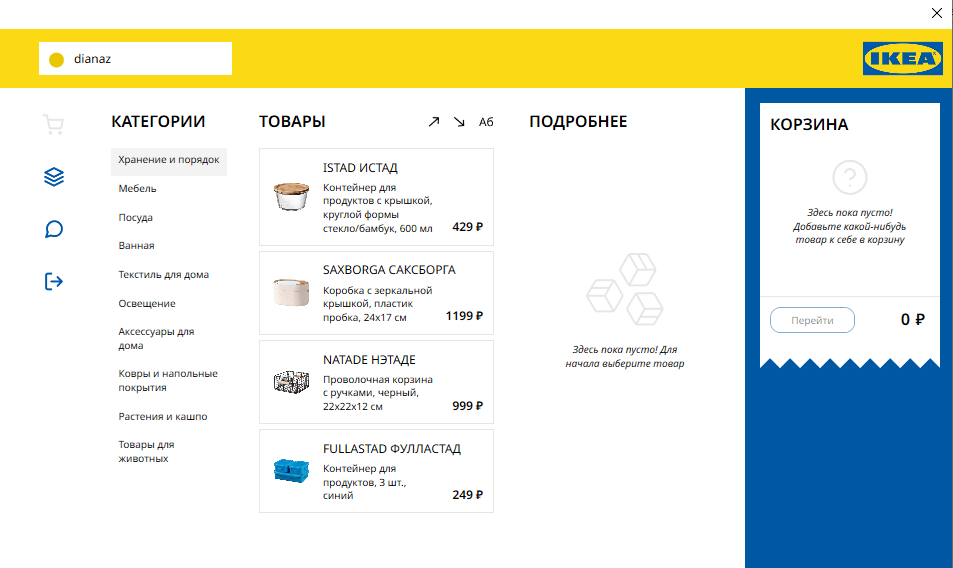


Рисунок 6.3 – Главное окно пользователя

Далее пользователь имеет возможность просматривать и добавлять товары в корзину, а также оформлять заказ.

## **6.3 Использование приложения администратором**

После успешной авторизации администратор попадает в главное окно администратора, где видит 9 кнопок для навигации по страницам и выхода их приложения. Администратор имеет возможность добавлять такие объекты как:

* категории
* товары
* купоны
* ответы на отзывы
* статус у заказа

Так же администратор имеет возможность вносить в изменения любых объектов находящихся в базе данных.

Пример главного окна администратора приведен на рисунке 6.4.

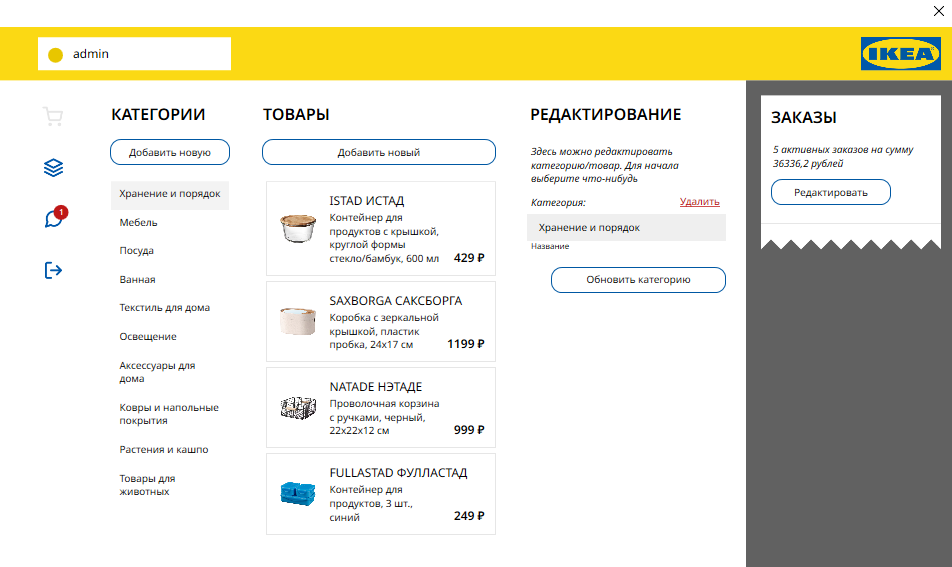


Рисунок 6.4 – Главное окно администратора

# **ЗАКЛЮЧЕНИЕ**

В итоге выполнения курсового проекта было разработано программное средство «Поставка товаров из IKEA», предоставляющее возможность добавлять товары в корзину, а так же косвенно покупать товары, находящиеся в данной корзине. Так же приложение поддерживает отправку чека на электронную почту, которая привязана к аккаунту пользователя. Основные достоинства – простота использования, оригинальный дизайн и гибкий функционал.

В процессе реализации программного средства были достигнуты вышеуказанные требования и задачи.

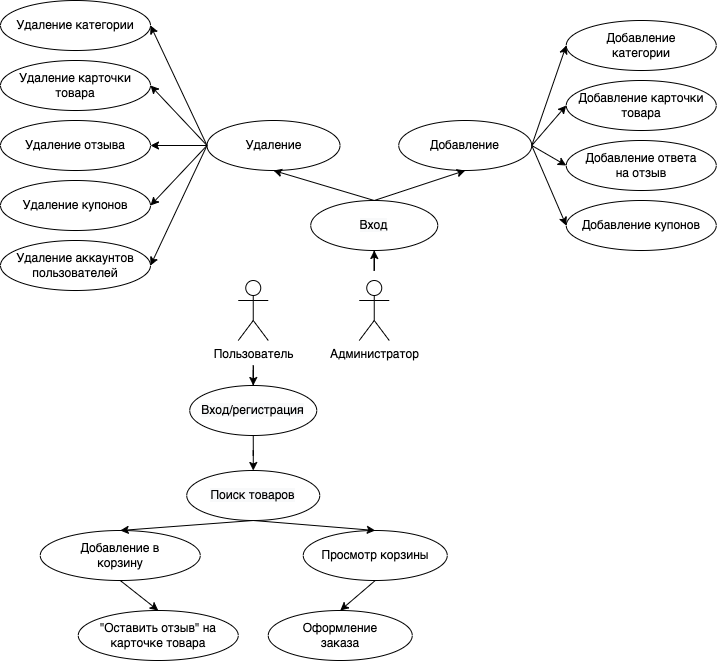
Также в процессе выполнения данного курсового проекта были закреплены навыки в программировании на языке C#, создании приложений на WPF, использование Entity Framework 6, работа с современным паттерном MVVM, проектирование базы данных и реализация их в СУБД MS SQL Server 2019.

Был получен опыт взаимодействия со сторонними библиотеками, изучения их и внедрения в свой проект.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. MSDN сеть разработчиков в Microsoft [Электронный ресурс] – <http://msdn.microsoft.com/library/rus/>. Дата доступа: 22.05.2022
2. Определение паттерна MVVM [Электронный ресурс] – <https://metanit.com/sharp/wpf/22.1.php>. Дата доступа: 22.05.2022
3. Подход Code First [Электронный ресурс] –https://metanit.com/sharp/entityframework/1.2.php. Дата доступа: 23.05.2022

## **Приложение А**

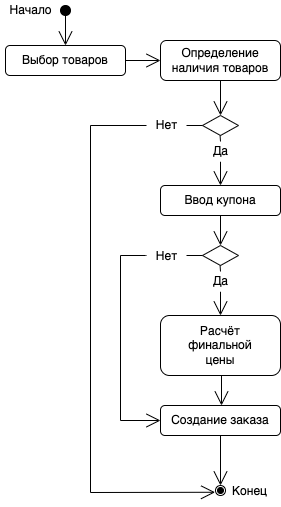


## 

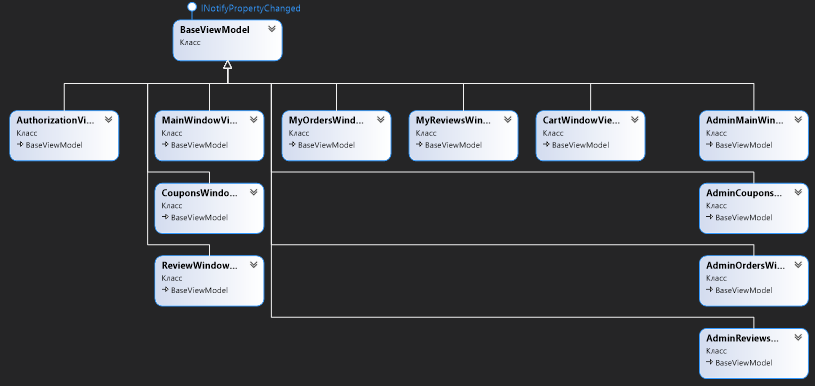
## **Приложение Б**

## 

## **Приложение В**



## **Приложение Г**



## **Приложение Д**

