МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-40 05 01 Информационные системы и технологии

Направление специальности 1-40 01 05 10 Информационные системы и технологии

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Тема Программное средство «Магазин автозапчастей»

Исполнитель

студент 3 курса группы 1 Палазник Арсений Викторович

(Ф.И.О.)

Руководитель работы ассистент Панченко О.Л.

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

Курсовой проект защищен с оценкой

Председатель Смелов В.В.

(подпись)

Минск 2023

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc73065782)

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

[1.1 Анализ прототипов 4](#_Toc73065784)

[1.2 Постановка задачи 6](#_Toc73065785)

[2Анализ 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)

[3Проектирование и создание программного средства 9](#_Toc73065791)

[3.3 Проектирование логической структуры базы данных 9](#_Toc73065794)

[3.2 Проектирование окон приложения 11](#_Toc73065795)

# **ВВЕДЕНИЕ**

Рынок автомобилей с каждым годом растет, количество автомобилей увеличивается, однако каждый автомобиль имеет высокий шанс получить повреждения. В настоящее время автовладельцы для экономии времени предпочитают использовать приложения и онлайн площадки для поиска и покупки автомобильных запчастей. Это так же экономит время, так как многие площадки имеют возможность доставить товар бесплатно.

У многих крупных и не очень организаций уже существует возможность просмотра и приобретения автомобильных запчастей через веб-сайты и приложения. Поэтому темой курсового проекта является разработка программного средства «Магазин автозапчастей». Данное программное средство должно позволить пользователям в лице клиентов заказывать автомобильные запчасти на единой площадке, отслеживать статус заказа. Пользователи в лице администраторов могут добавлять новые товары, просматривать заказы. Так же администраторы имеют возможность удалять товары и поставщиков.

Для успешной реализации курсового проекта необходимо:

* провести анализ соответствующей литературы;
* ознакомиться с прототипами программных средств выбранной мной темы;
* определить функциональные требования;
* продумать структуру базы данных;
* продумать структуру проекта;
* реализовать программное средство;
* протестировать программное средство;
* написать руководство пользователя.

Содержание данной пояснительной записки отражает этапы выполнения курсового проекта.

# **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

Exist.by ­– платформа, на которой предоставлен широкий выбор автомобильных запчастей и аксессуаров для автомобиля (рисунок 1.1). [1]

Достоинства:

* большой выбор товаров
* доставка курьерской службой

Недостатки:

* слишком нагруженный интерфейс
* отсутствие собственных пунктов выдачи товара
* для Windows существует только веб-версия

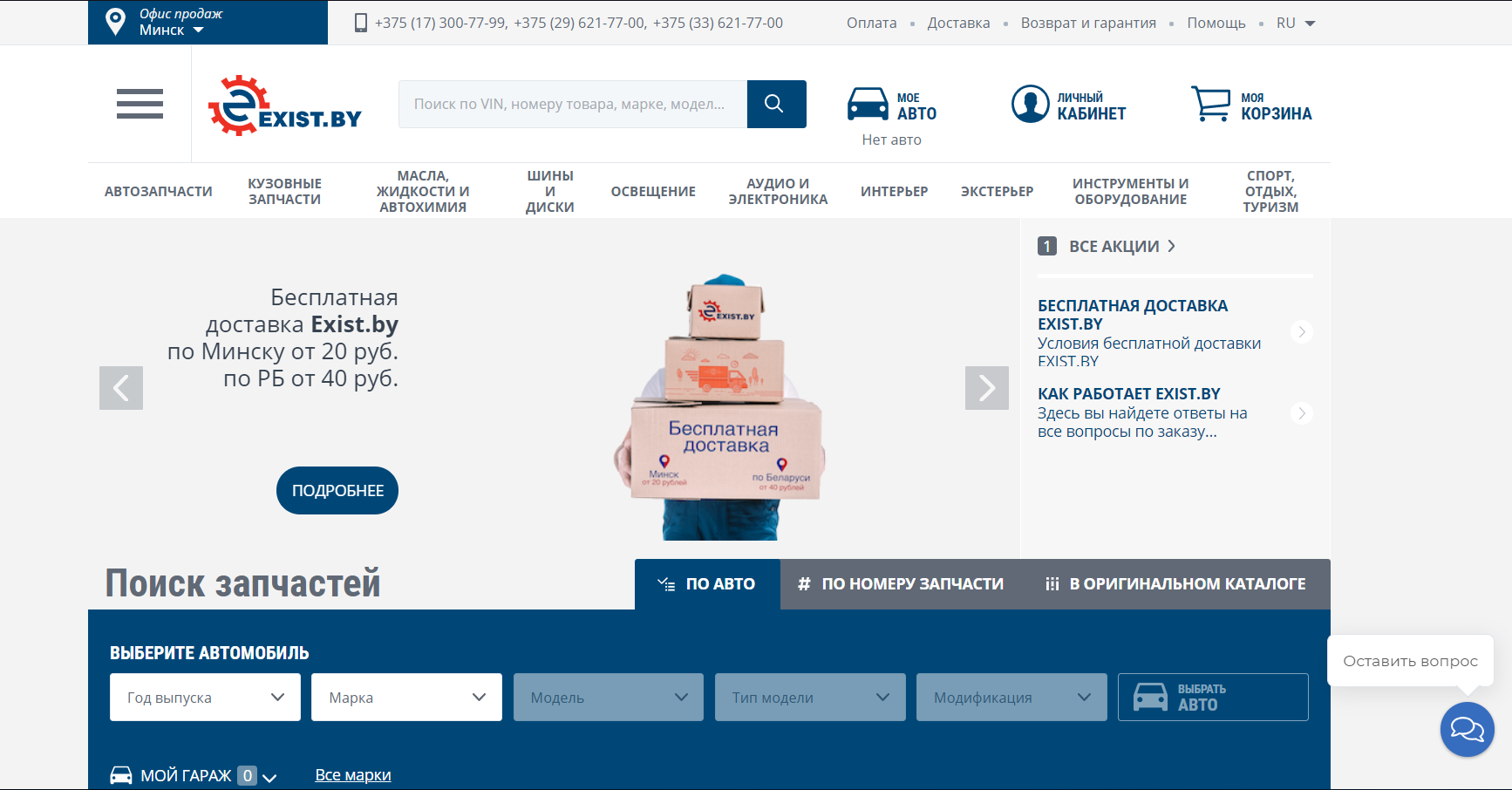


Рисунок 1.1 – Интерфейс веб-приложения «Exist.by»

Zap.by – платформа для заказа автозапчастей (рисунок 1.2). [2]

Достоинства:

* широкий выбор товаров
* бесплатная доставка

Недостатки:

* устаревший интерфейс
* для Windows существует только веб-приложение

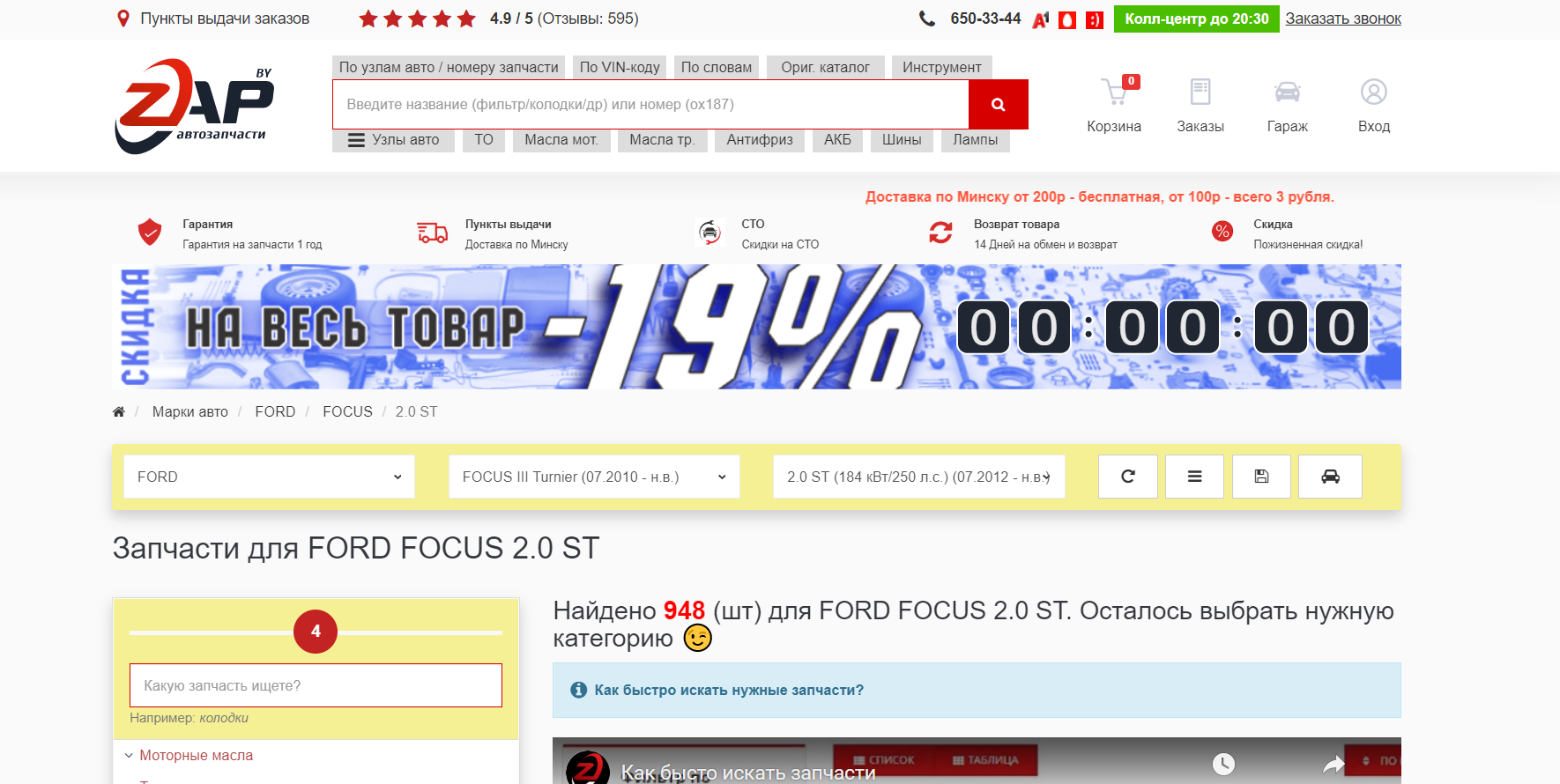


Рисунок 1.2 – Интерфейс веб-приложения «Zap.by»

Auto1.by – сервис предоставляющий широкий выбор автозапчастей и услуги по обслуживанию автомобилей (рисунок 1.3). [3]

Достоинства:

* приятный дизайн
* широкий выбор товаров
* большое количество точек продаж
* наличие рекомендаций по подбору запчастей

Недостатки: не выявлены

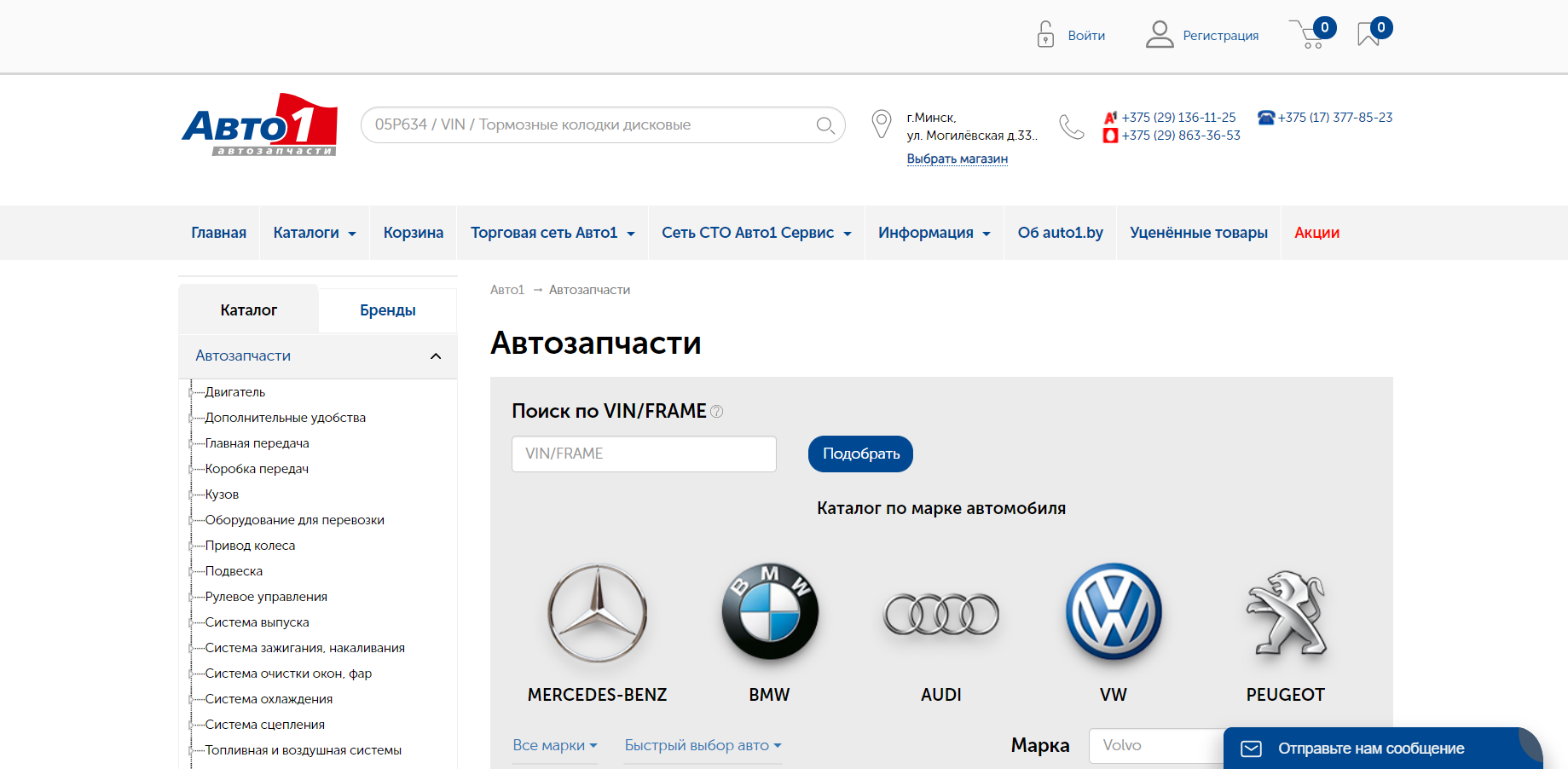


Рисунок 1.3 – Интерфейс веб-приложения «auto1.by»

## 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. Пользователю в лице клиента - просматривать все продукты, осуществлять их поиск и фильтрацию, добавлять их в корзину, удалять и оформлять заказ с последующей возможностью отмены.

Функциональные требования описаны в главе 2.

## 1.3 Вывод по разделу

После оценки всех веб-сервисов можно заметить, что многие веб-сайты имеют схожий интерфейс и концепцию, однако каждый из них реализован по-своему. Проанализировав аналоги веб-сайтов с тематикой «Магазин автозапчастей» были выделены основные задачи проекта: разработка приложения, позволяющего пользователю осуществлять весь базовый функционал (регистрация, поиск товаров, фильтрация, покупка, добавление товара в корзину и т.д.). Интерфейс должен быть удобным и понятным.

# **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 для неавторизованного пользователя должно выполнять следующие задачи:

* регистрация и авторизация пользователей;
* поиск и фильтрация данных.

Функционально ПС для авторизованного пользователя должно выполнять следующие задачи:

* регистрация и авторизация пользователей;
* сохранение рабочей информации в централизованной базе данных;
* добавление товаров в корзину с возможностью их удаления;
* заказ товара из корзины и его подтверждение;
* поиск и фильтрация данных.

Функционально ПС для администратора должно выполнять следующие задачи:

* сохранение рабочей информации в централизованной базе данных;
* просмотр всех товаров, добавление товаров с возможностью их удаления;
* отправка кода подтверждения заказа посредством отправки электронного письма.

## 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

На основе описанного основного функционала программы были определены основные возможности пользователя. Для наглядного отображения возможностей пользователя была создана UML-диаграмма (Рисунок 2.1).

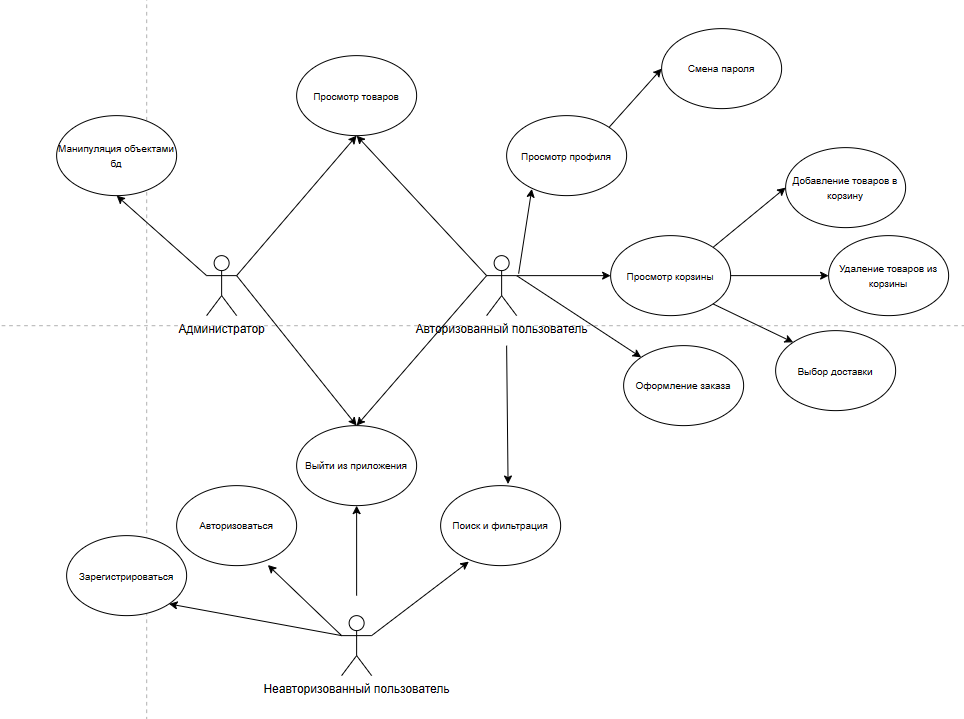


Рисунок 2.1 – UML-диаграмма

# **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 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.

Таблица Users используется для хранения данных о пользователях. Включает 7 столбцов: Id– id пользователя, Login – логин пользователя, Password – пароль пользователя, при записи в базу данных хэшируется, Is\_admin – 0 – пользователь, 1 - администратор, FirstName –имя пользователя, LastName – фамилия пользователя, Email – адрес электронной почты.

Таблица Providers используется для хранения данных о поставщиках товара. Включает 3 столбца: ProviderId – id поставщика, Name – имя поставщика, Email – адрес электронной почты поставщика.

Таблица Parts используется для хранения продуктов. Включает 10 столбцов: PartId – id продукта, Name – название продукта, Quantity – количество товара, доступного для покупки, ProviderId – id поставщика продукт, Price – цена покупки товара, Description – краткое описание товара, FullDescription – полное описание товара, ImageLink – ссылка на изображение товара, CategoryId – id категории товара, MarkId – id марки автомобиля, к которому принадлежит товар.

Таблица Orders используется для хранения информации о заказах. Включает 5 столбцов: OrderId – id заказа, OrderDate – дата открытия заказа, UserId – id пользователя, который сделал заказ, DeliveryId – id службы доставки, OrderState – статус заказа.

Таблица OrderedParts используется для связи между таблицами Orders и Parts. Используется для определения того, какие товары и какое их количество входит в заказ. Состоит из 3 столбцов: PartId – id товара, OrderId – id заказа, Amount – количество единиц определенного товара в заказе.

Таблица Marks используется для хранения данных о марках автомобилей. Включает 2 столбца: MarkId – id марки автомобиля, MarkName – наименование марки автомобиля.

Таблица Deliveries используется для хранения информации о службах доставки. Состоит из 4 столбцов: DeliveryId – id службы доставки, Description – описание службы доставки, Price – цена услуги доставки, Name – наименование службы доставки.

Таблица Categories хранит информацию о категориях товаров. Состоит из 3 столбцов: CategoryId – id категории, Name – имя категории, Description – описание категории.

Таблица Cards используется для хранения данных о привязанной карте для оплаты. Состоит из 5 столбцов: CardId – id карты, CardNumber – номер карты, UserId – id пользователя, CvvCode – код cvv карты, Balance – баланс карты.

На основе этих таблиц и их столбцов была разработана логическая схема БД (Рисунок 3.1).



Рисунок 3.1 – логическая схема БД

## 3.2 Проектирование окон приложения

Приложение включает в себя 5 окон и 19 пользовательских элементов управления, выступающих в качестве страниц. Связи между ними представлены на рисунке 3.2.

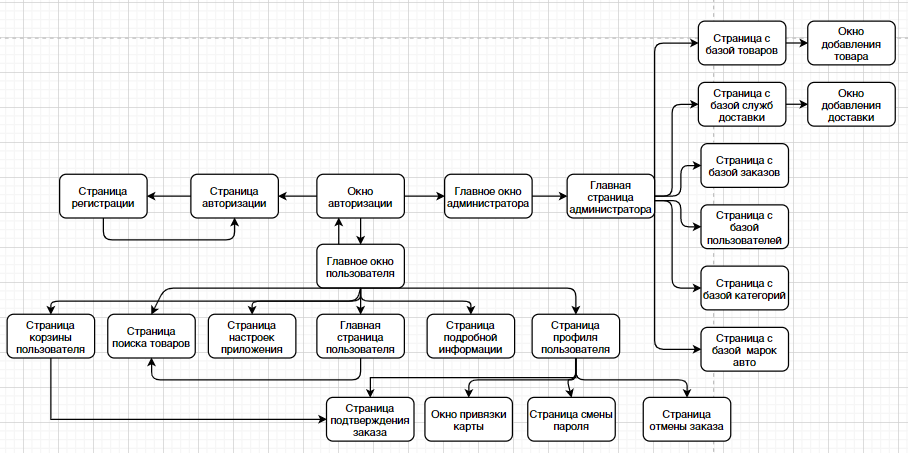


Рисунок 3.2 – связи между окнами

## 3.3 Проектирование окон приложения

Для реализации паттерна MVVM файлы программы были распределены по соответствующим директориям и реализовали соответствующие функции. Разделение проекта на логические модули представлено на рисунке 3.3.

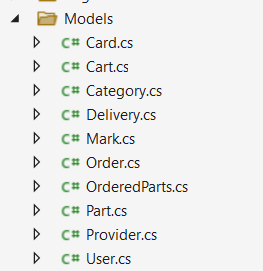
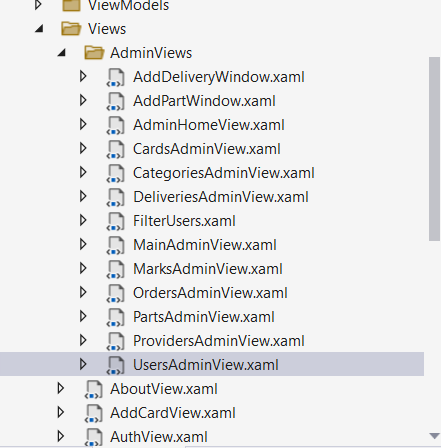




Рисунок 3.3 – логические модули проекта;

Архитектура MVVM (Model-View-ViewModel) является одним из популярных паттернов архитектуры для проектов, связанных с разработкой пользовательских интерфейсов. Модель представляет данные в приложении, Представление отображает данные пользователю, а ViewModel является связующим звеном между Моделью и Представлением, обрабатывая логику, связанную с UI и взаимодействуя с Моделью для запроса и обновления данных. [5]

3.4 Реализация классов Models и ViewModels

В папке Models расположены сущностные классы, которые используются для создания БД. Этими классами являются Card, Cart, Category, Delivery, Mark, Order, OrderedParts, Part, Provider, User.

Классы с суффиксом ViewModel содержат: закрытые поля, открытые свойства и команды, привязываемые к элементам управления на страницах View. ViewModel также содержит логику по получению данных из модели, которые потом передаются в представление. И также VewModel определяет логику по обновлению данных в модели.

**3.5 Реализация Views**

Для разработки графической части приложения была выбрана технология WPF.

Windows Presentation Foundation (WPF) — это библиотека для создания пользовательских интерфейсов для интеллектуальных клиентских приложений

Одной из важных особенностей WPF является использование языка декларативной разметки интерфейса XAML, основанного на XML. Разработка с использованием XAML позволяет отделить графический интерфейс от логики приложения, а также создавать насыщенный интерфейс, используя или декларативное объявление интерфейса, или код на управляемых языках C#.

**3.9 Вывод по разделу**

В данном разделе представлен детальный обзор процесса проектирования базы данных в среде Microsoft SQL Server, а также описаны компоненты, составляющие приложение, и подробная структура файлов проекта. Был Осуществлен анализ и определение структуры базы данных с использованием Microsoft SQL Server. Разработанная база данных включает в себя несколько ключевых таблиц. Каждое поле таблицы имеет свое уникальное предназначение, обеспечивая надежное и эффективное хранение данных.

Относительно структуры приложения, основной платформой для клиентской части был выбран Windows Presentation Foundation (WPF). Для взаимодействия с базой данных применен Entity Framework. Было произведено подробное описание структуры файлов проекта и их назначения.

# **4.** **Тестирование и проверка работоспособности**

# **4.1 Тестирование программных компонентов**

В момент авторизации и регистрации возможна ситуация, когда пользователь вводит некорректные данные, например, неверный пароль, незарегистрированный логин, некорректный адрес электронной почты. Такие исключения обрабатываются программным средством с помощью показа пользователю всплывающих сообщений с текстом ошибки. Примеры обработки разных видов исключительных ситуаций представлены на рисунках 4.1 – 4.3.

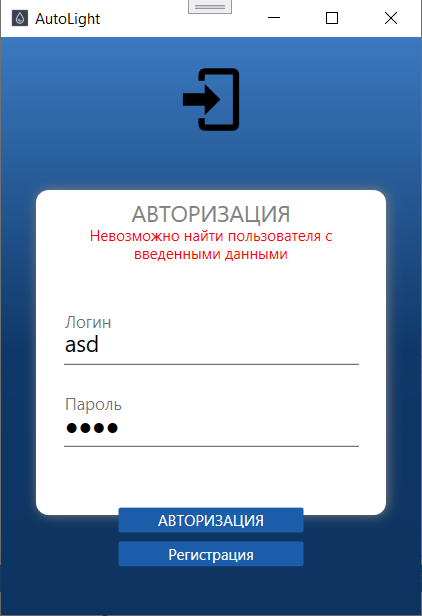


Рис 4.1 – обработка неверного/несуществующего пользователя;

Важной частью валидации стартового окна является и регистрация, для каждого поля которой обработаны случаи неправильного ввода информации и подсветки соответствующего поля ввода.

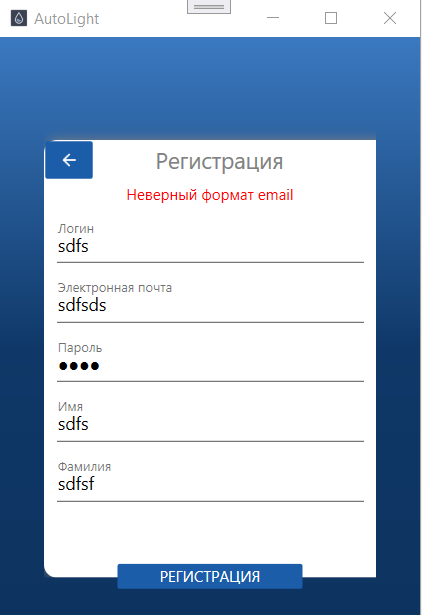


Рис 4.2 – обработка невалидного email;

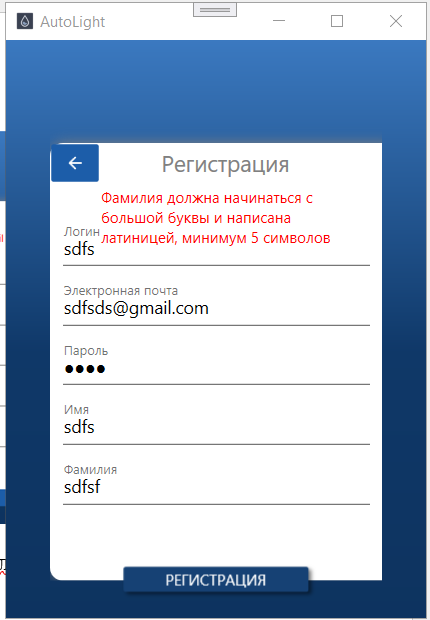


Рис 4.3 – обработка невалидной фамилии;

При заполнении корректных данных окно регистрации закрывается и открывается окно с авторизацией

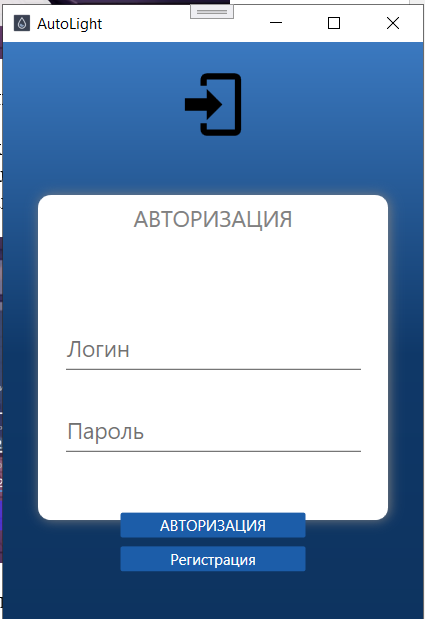


Рис 4.4 – переадресация на окно авторизации;

После окна авторизации пользователь попадает на главную страницу с каталогом товаров (рисунок 4.5).

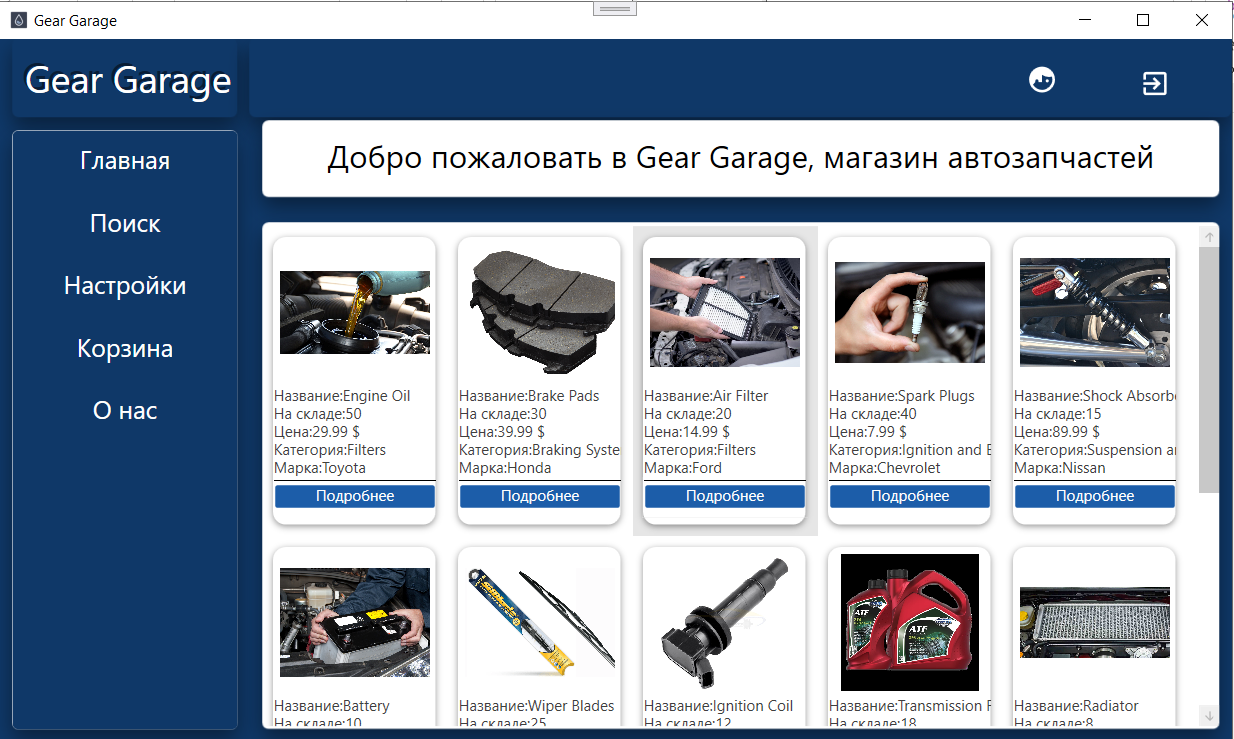


Рис 4.4 – главная страница приложения;

На вкладке Поиск представлен интерфейс позволяющий выполнять поиск и фильтрацию по товарам (Рисунки 4.5 и 4.6).

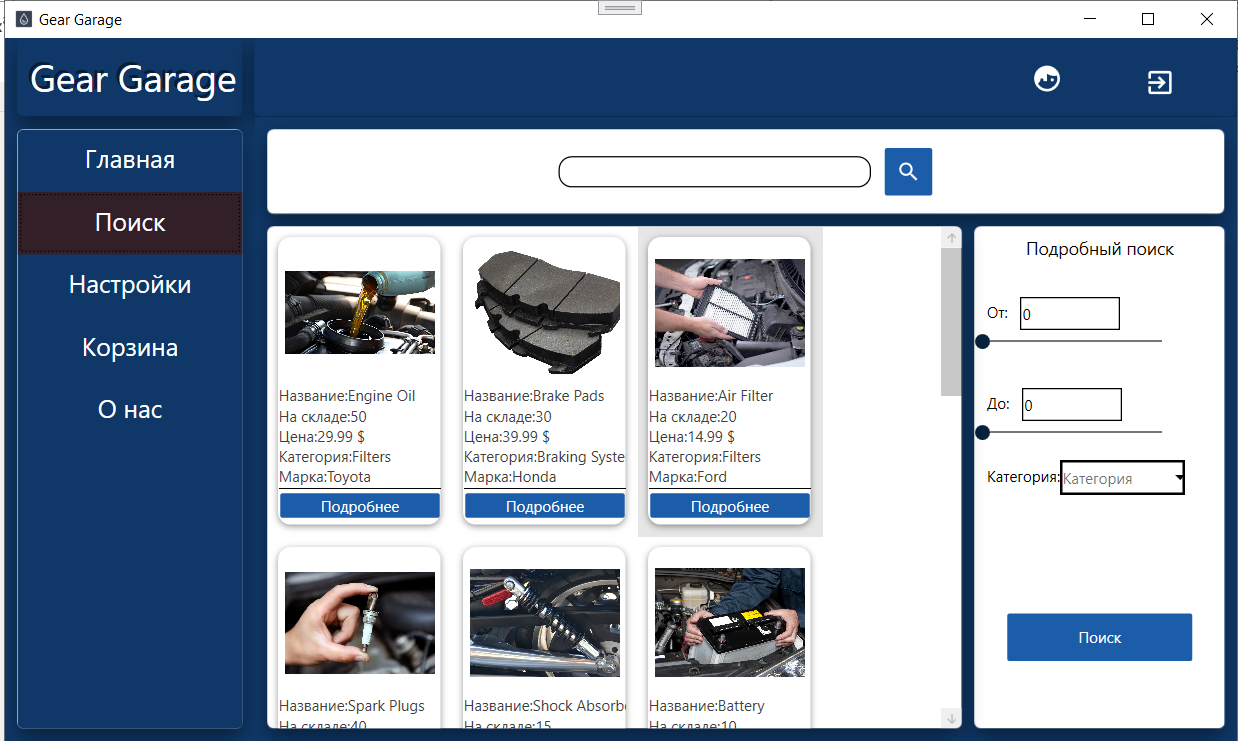


Рис 4.5 – страница «Поиск» до фильтрации;

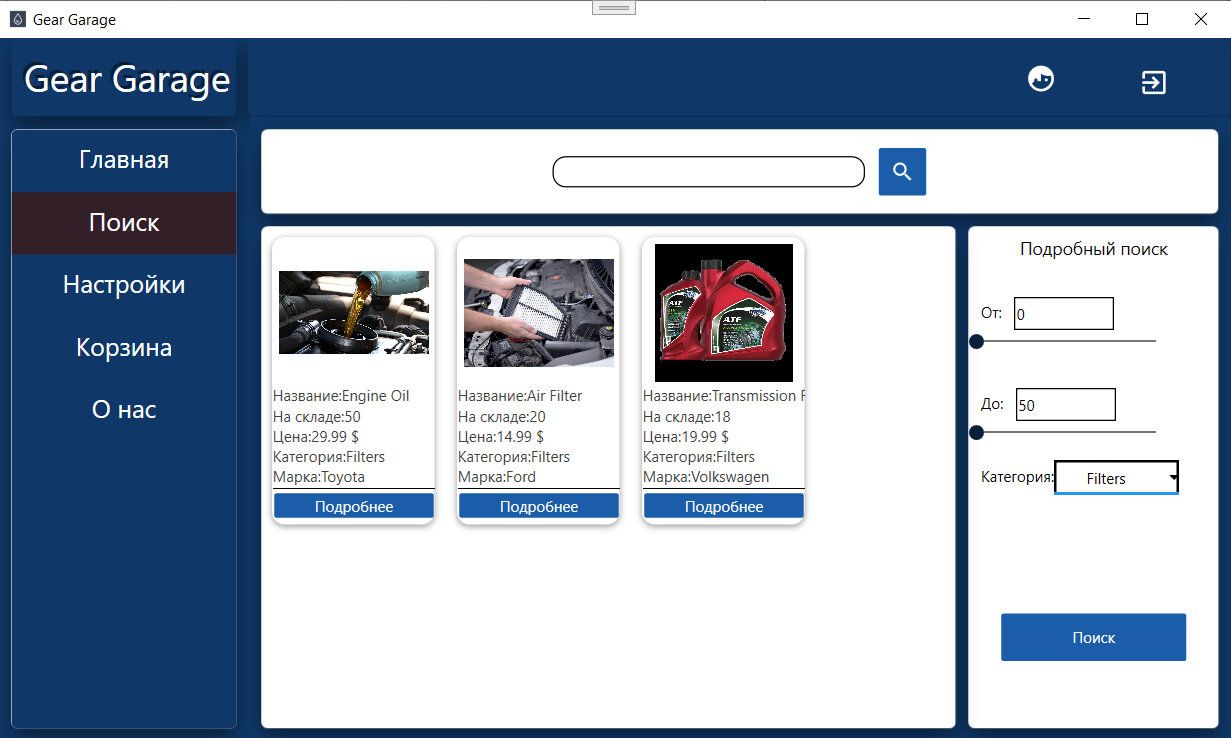


Рис 4.6 – страница «Поиск» после фильтрации;

При нажатии на кнопку «Подробнее» пользователь переходит на страницу товара, с помощью которой он может добавить товар в корзину (Рисунок 4.7).

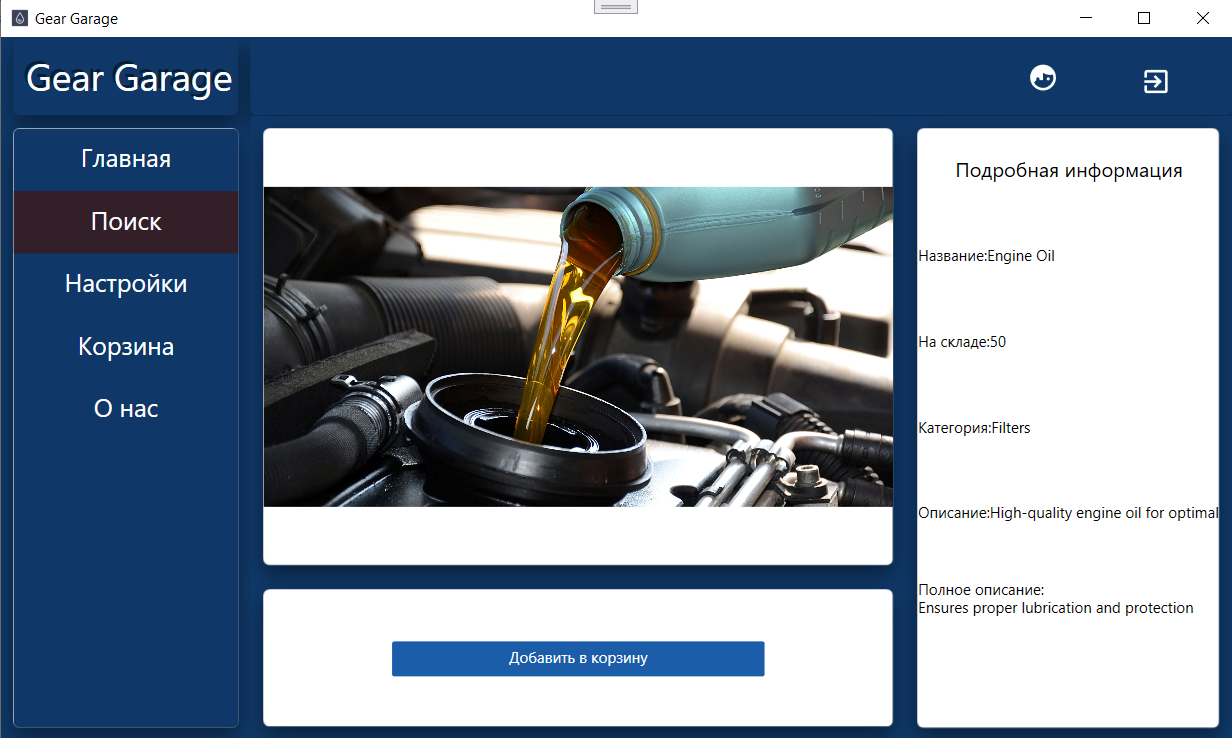


Рис 4.7 – страница товара;

При нажатии на кнопку «Добавить в корзину» товар добавляется в корзину, о чем нас оповещает система (Рисунок 4.8).

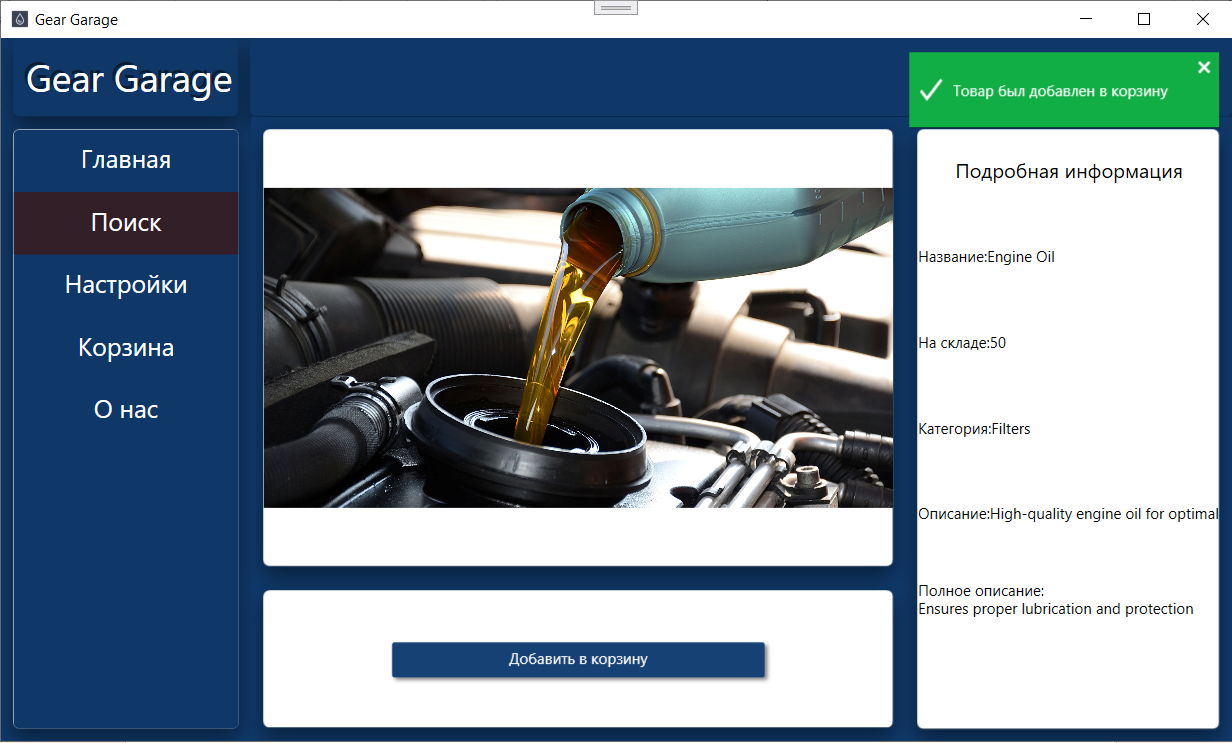


Рис 4.8 – уведомление о добавлении товара в корзину;

При нажатии на вкладку «Настройки» у нас предоставляется возможность изменить язык приложения (Рисунок 4.9).

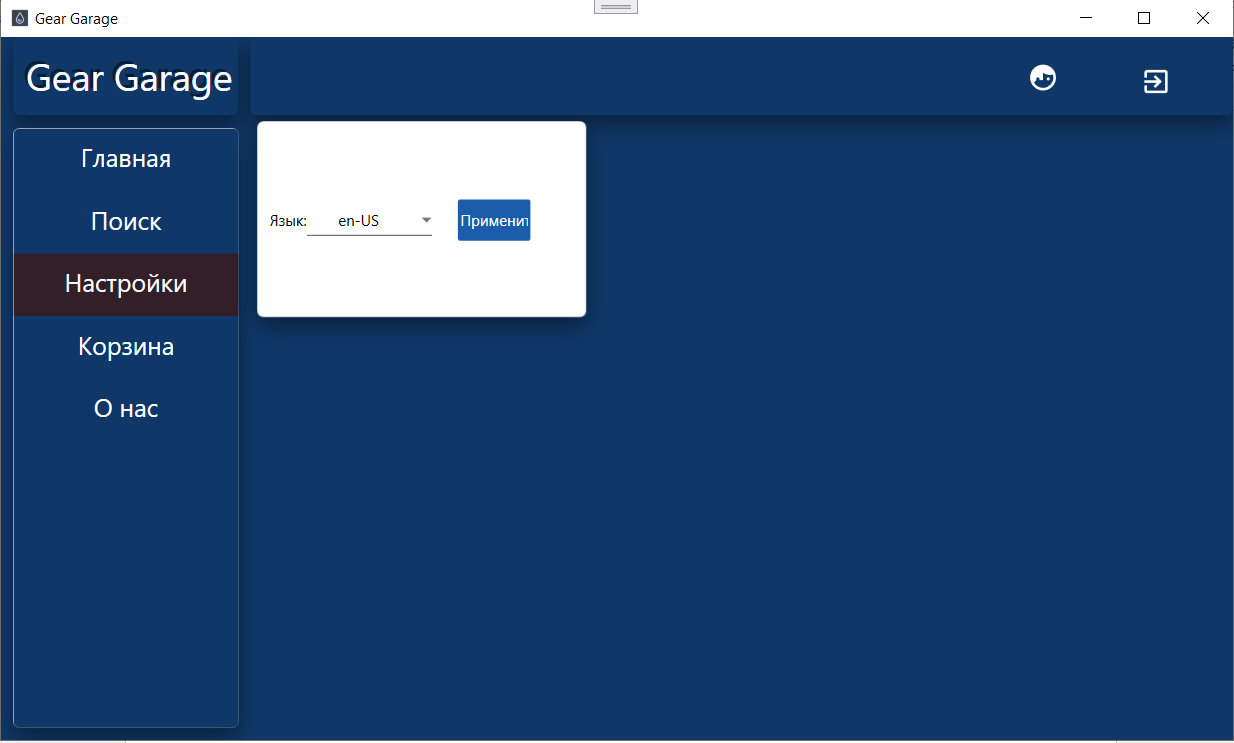


Рис 4.9 – вкладка «Настройки»;

При нажатии на вкладку «Корзина» у нас отображается страница с товарами, которые находятся в корзине (Рисунок 4.10).

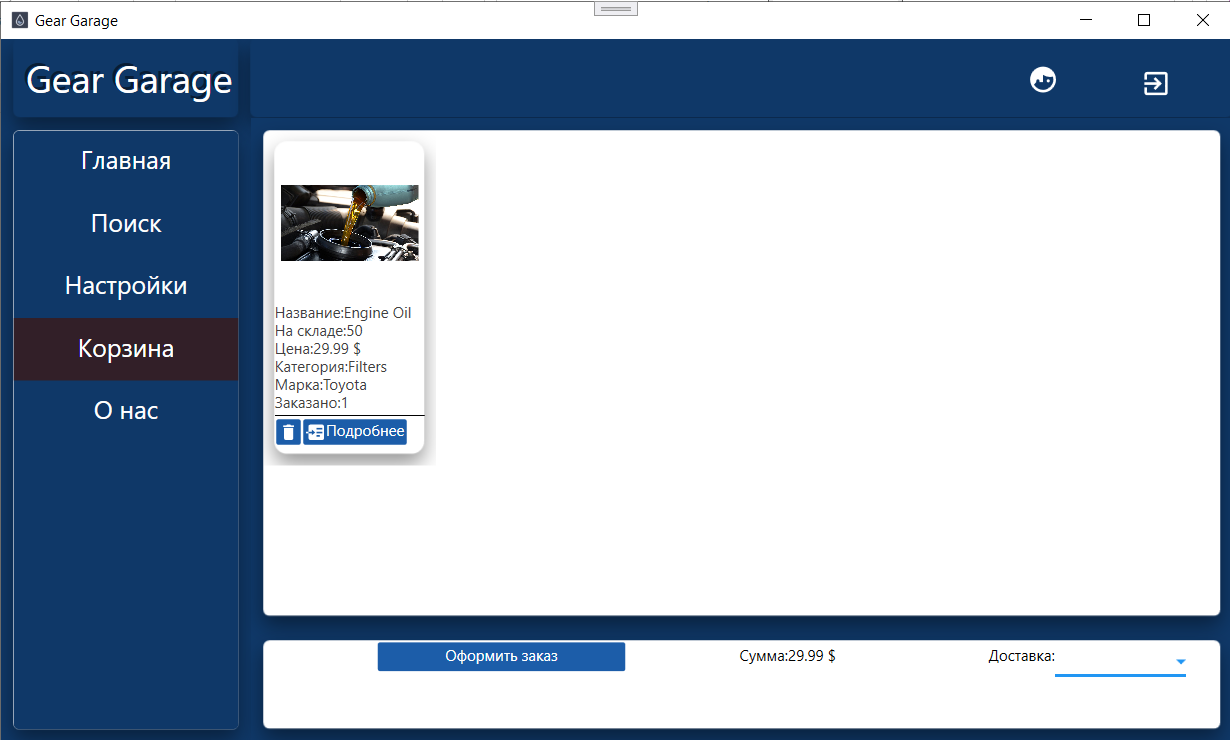


Рис 4.10 – вкладка «Корзина»;

При нажатии на вкладку «О нас» появляется FAQ (Рисунок 4.11)

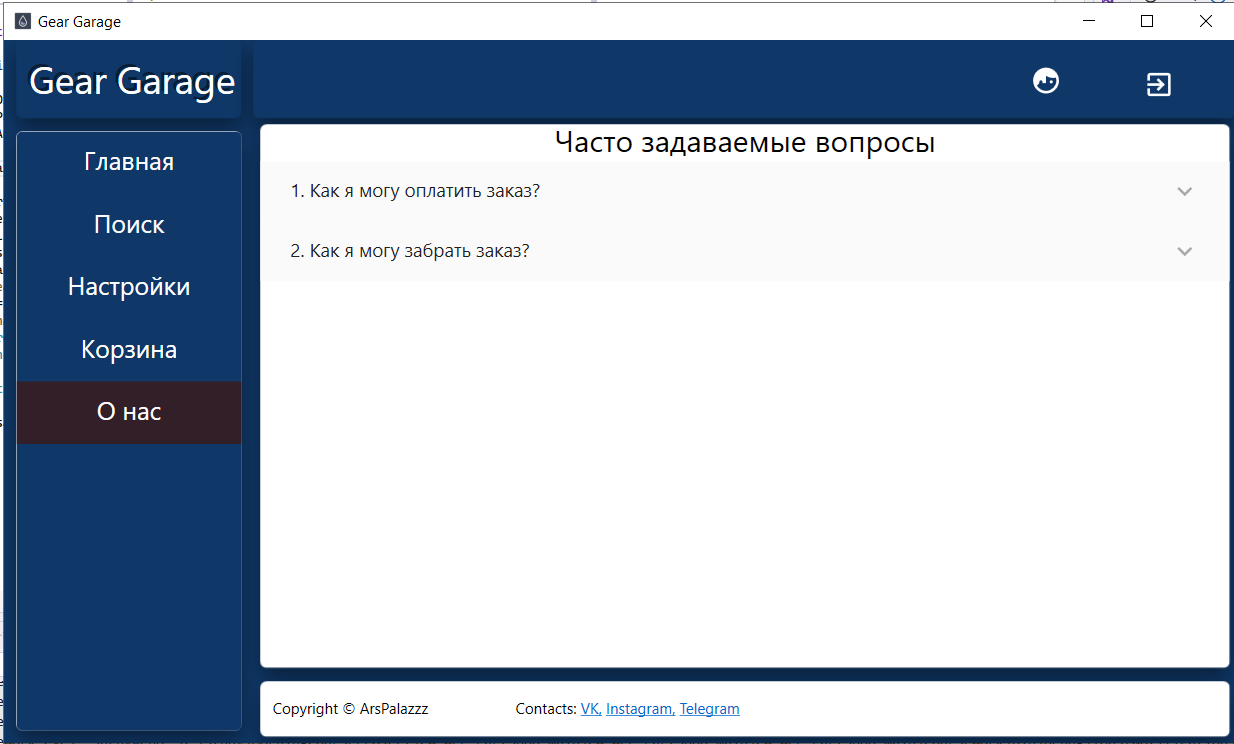


Рис 4.11 – вкладка «О нас»;

При нажатии на иконку профиля открывается страница с подробной информацией о пользователе (Рисунок 4.12)

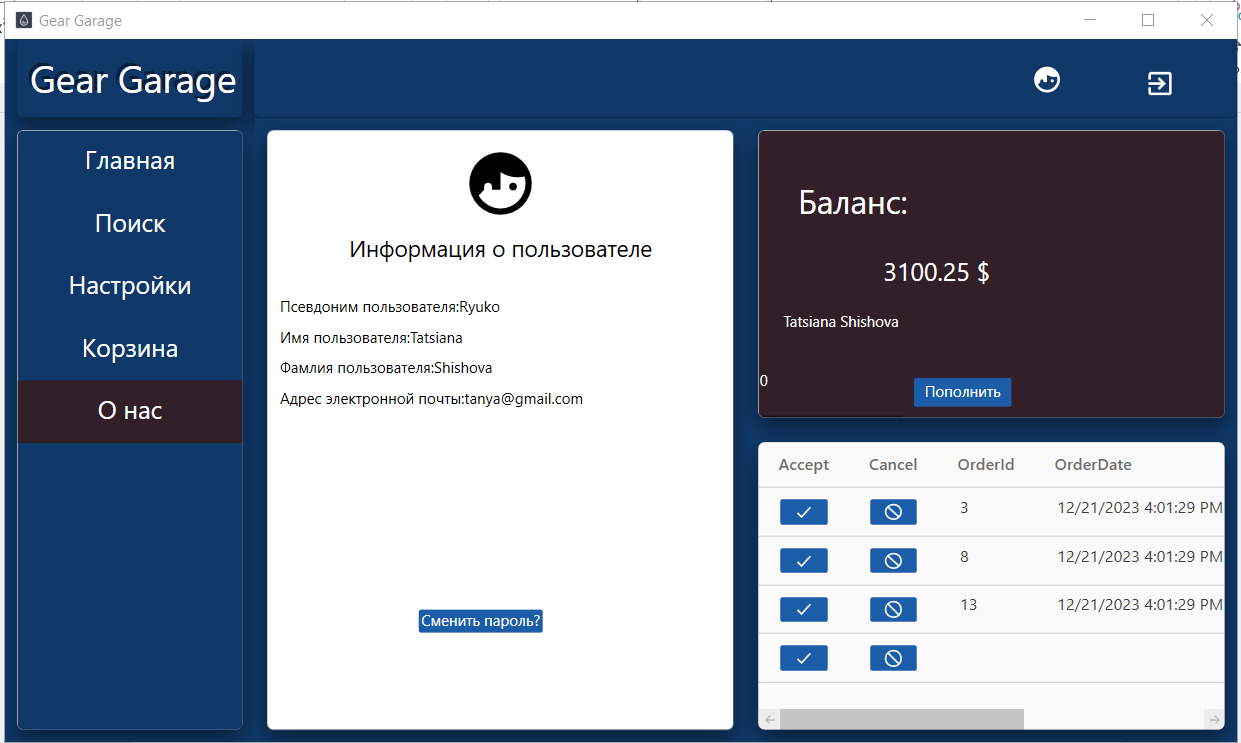


Рис 4.12 – вкладка «Профиль»;

При авторизации пользователя в качестве админа открывается окно для администратора (Рисунок 4.13).

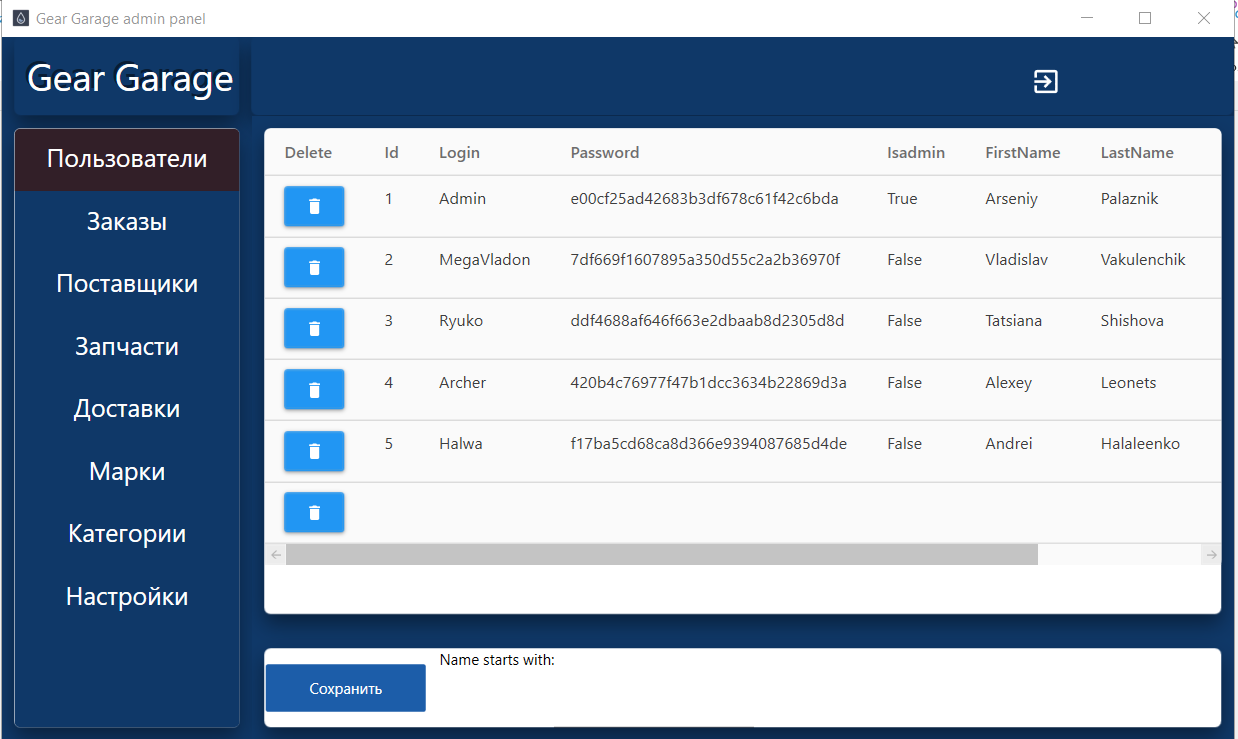


Рис 4.13 – вкладка «Пользователи»;

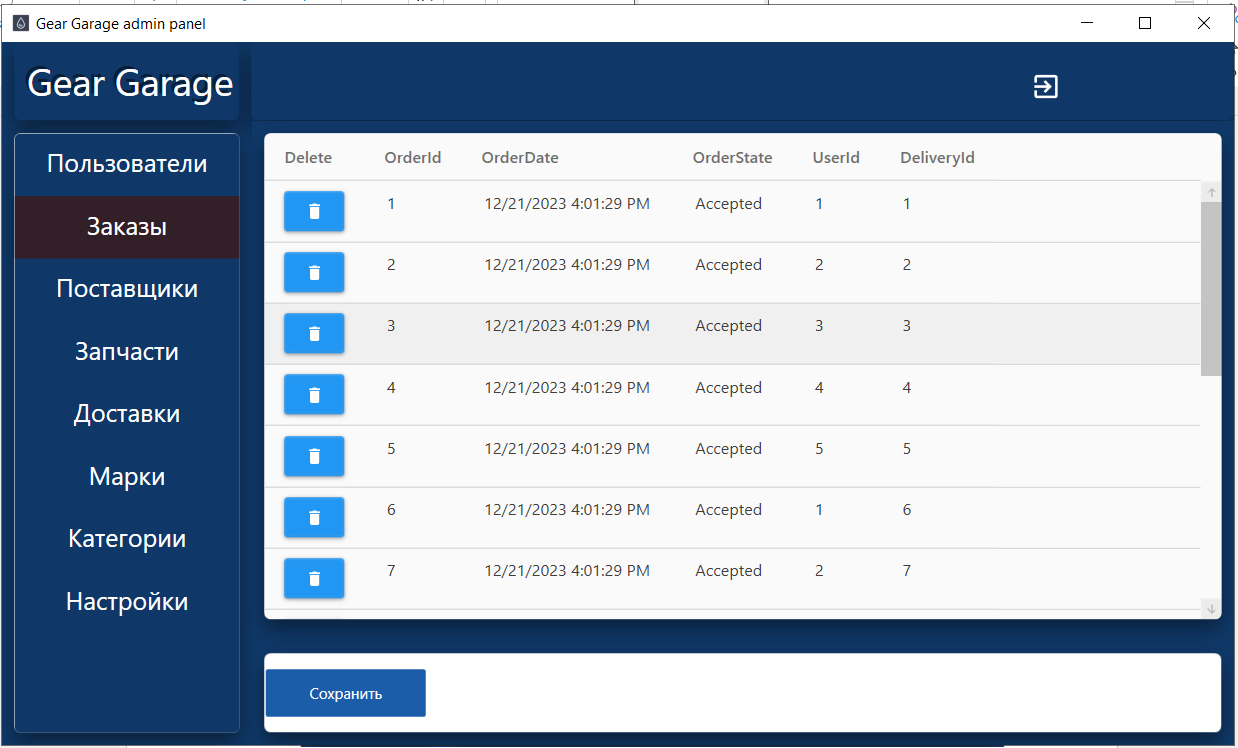


Рис 4.14 – вкладка «Заказы»;

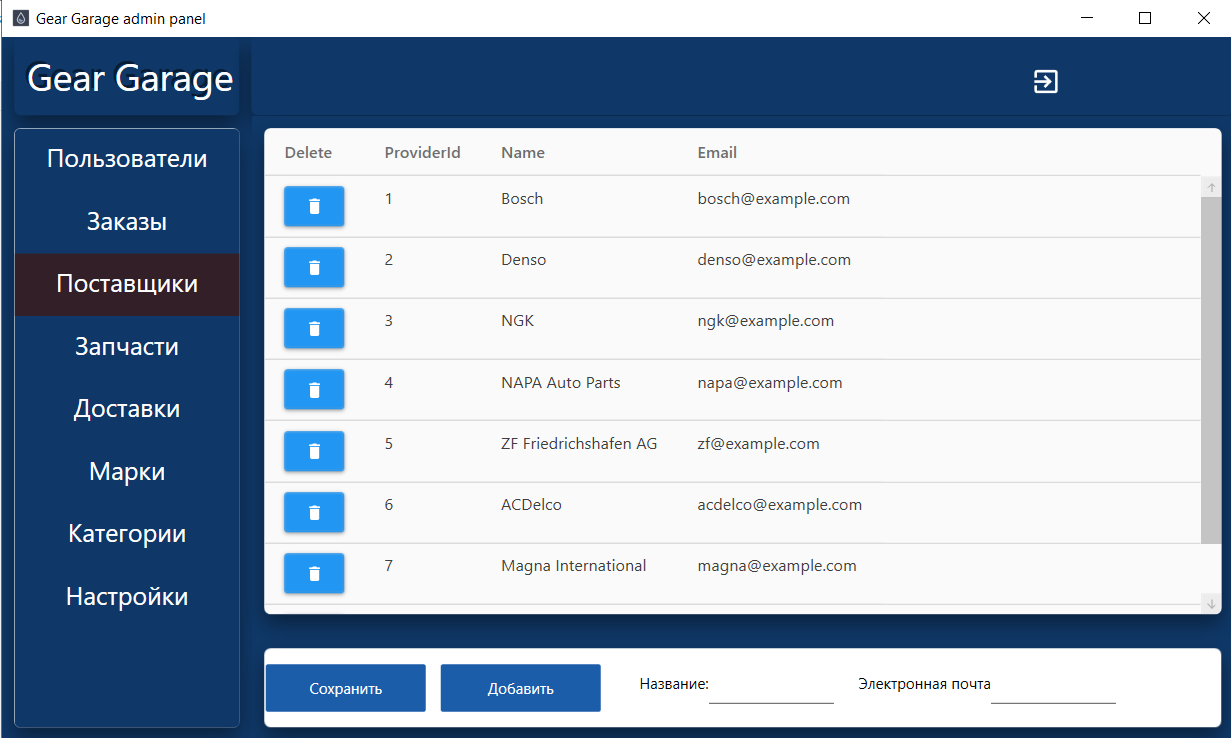


Рис 4.15 – вкладка «Поставщики»;

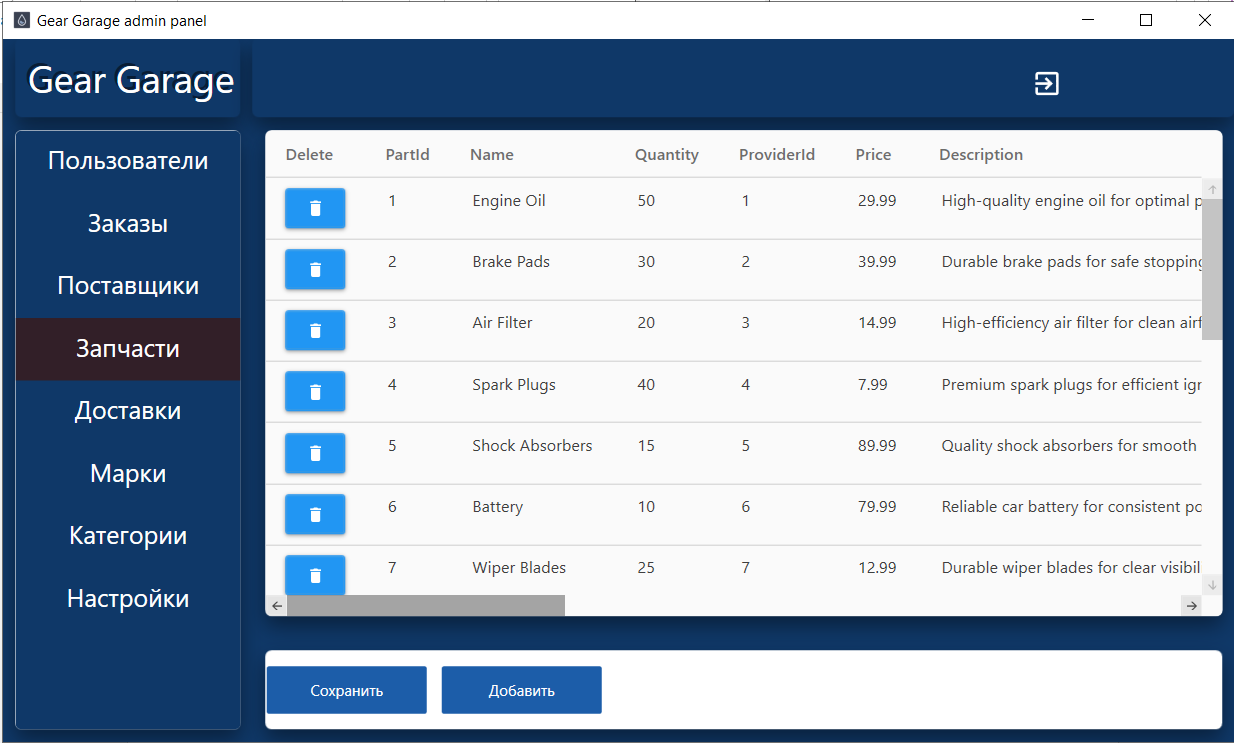


Рис 4.16 – вкладка «Запчасти»;



Рис 4.17 – вкладка «Доставки»;

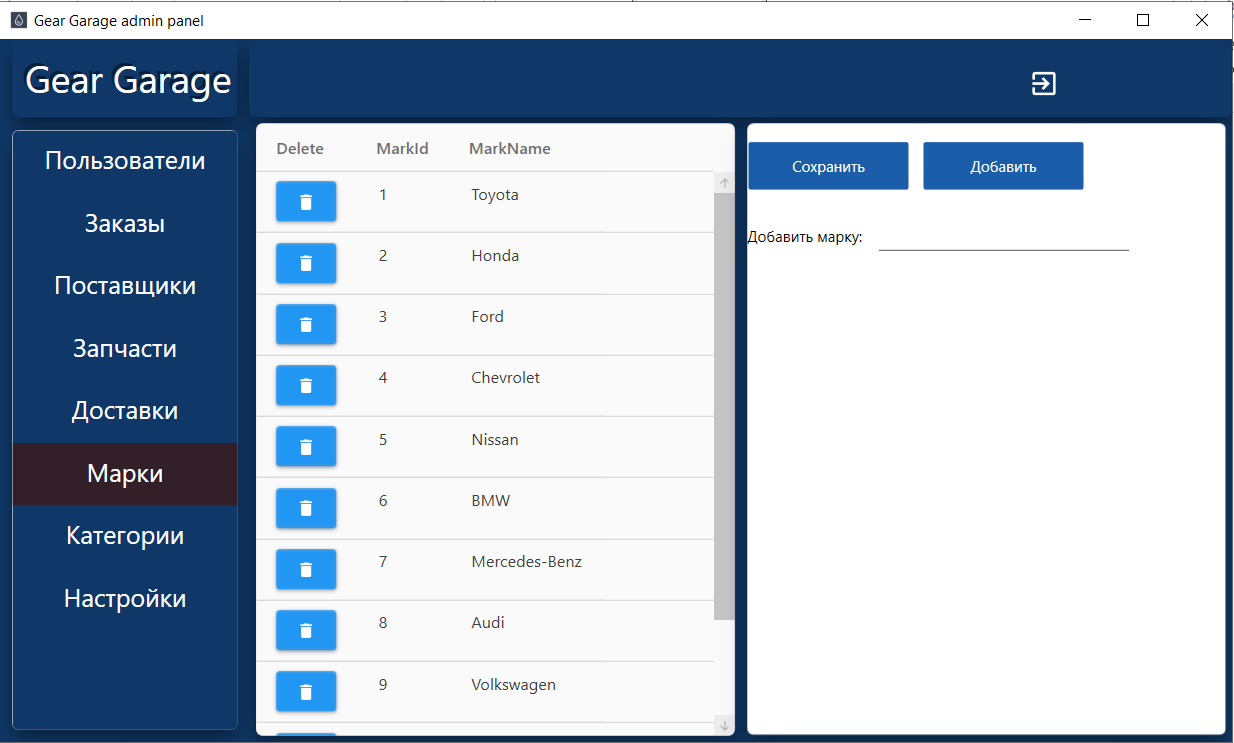


Рис 4.18 – вкладка «Марки»;

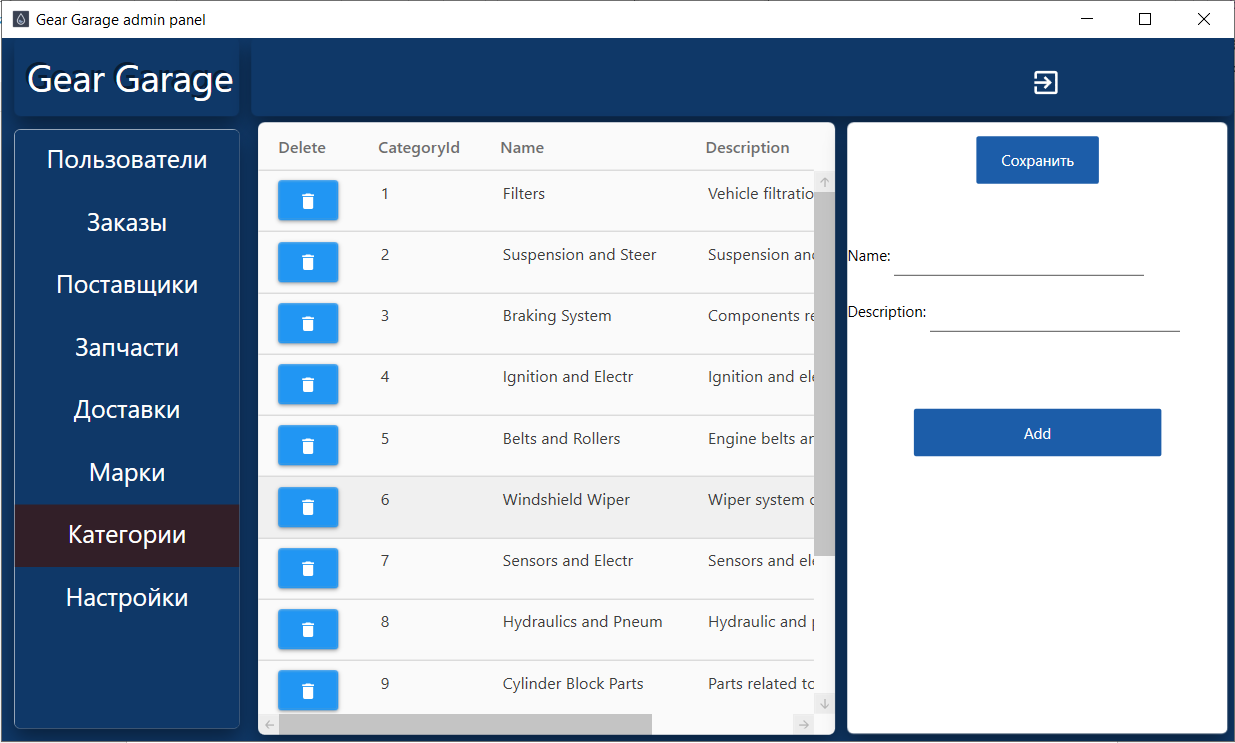


Рис 4.19 – вкладка «Категории»;

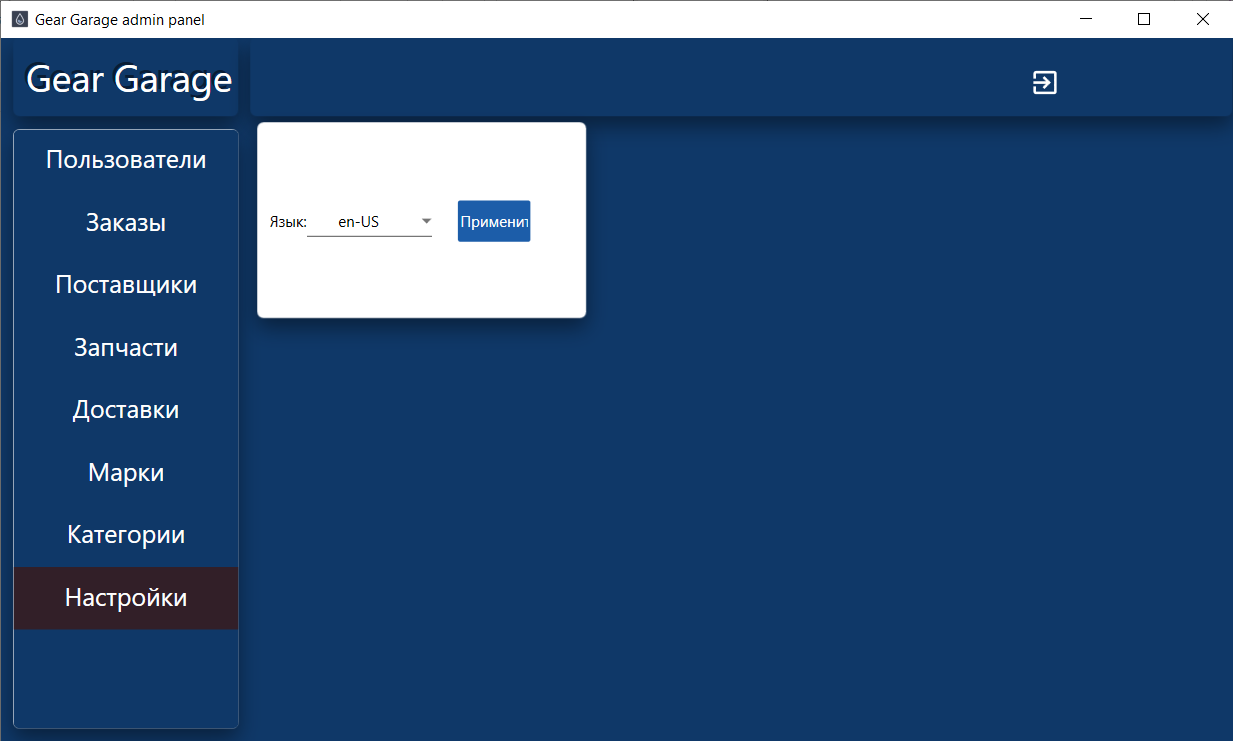


Рис 4.20 – вкладка «Настройки»;

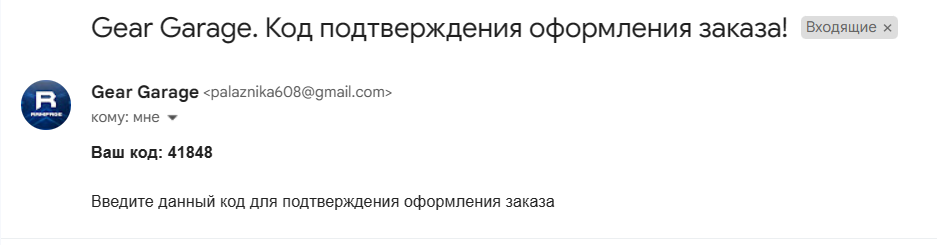


Рис 4.21 – Отправка сообщения на почту

**4.3 Вывод по разделу**

В данном разделе был представлен обзор процесса тестирования и оценки работоспособности разработанного приложения. Прежде всего, был проведен анализ требований к приложению, определены основные функциональные и нефункциональные характеристики, которые подлежит тестированию. Это включало в себя проверку соответствия приложения заявленным требованиям, а также обеспечение его стабильной и безопасной работы. Проведение тестов позволило выявить и устранить различные ошибки и недочеты в функционировании приложения. Таким образом, процесс тестирования и оценки работоспособности приложения позволил не только выявить и устранить возможные проблемы, но и удостовериться в его соответствии заявленным требованиям, обеспечивая стабильное и эффективное функционирование.

# **5. Руководство по установке и использованию**

# **5.1 Системные требования**

Для успешного использования данного приложения пользовательского сбора, необходимо, чтобы компьютер пользователя соответствовал определенным системным требованиям. Соблюдение этих требований крайне важно, так как в противном случае пользователь может столкнуться с трудностями при взаимодействии с приложением. Для удобного использования приложения пользователь должен обладать компьютером с операционной системой не ниже Windows 10 x64, объемом оперативной памяти не менее 3 ГБ для работы системы и достаточным свободным пространством для функционирования приложения. Также необходим процессор Intel Pentium с частотой 2.3 ГГц и выше или AMD Ryzen 3 с частотой 2.2 ГГц и выше.

Желательно наличие дискретной видеокарты начального уровня для плавного воспроизведения анимаций в приложении. Минимальное пространство на диске должно составлять приблизительно 20 ГБ, учитывая, что большая часть этого пространства занимается операционной системой. Также необходим установленный пакет библиотек Visual Studio 2008–2023, однако они обычно уже предустановлены в системе. Этот набор включает разнообразные компоненты и DLL-файлы для совместной работы операционной системы Windows, аппаратных компонентов ПК и приложений. Также следует иметь установленный набор библиотек .NET Framework и должен быть установлен SQL Server Management Studio (желательно 19 версии).

# **5.2 Установка приложения**

Из предоставленных файлов выбираем setup.exe (рисунок 5.1) и проходим стандартный процесс установки приложения (рисунок 5.2 – 5.4).



Рис 5.1 – файлы установщика приложения;

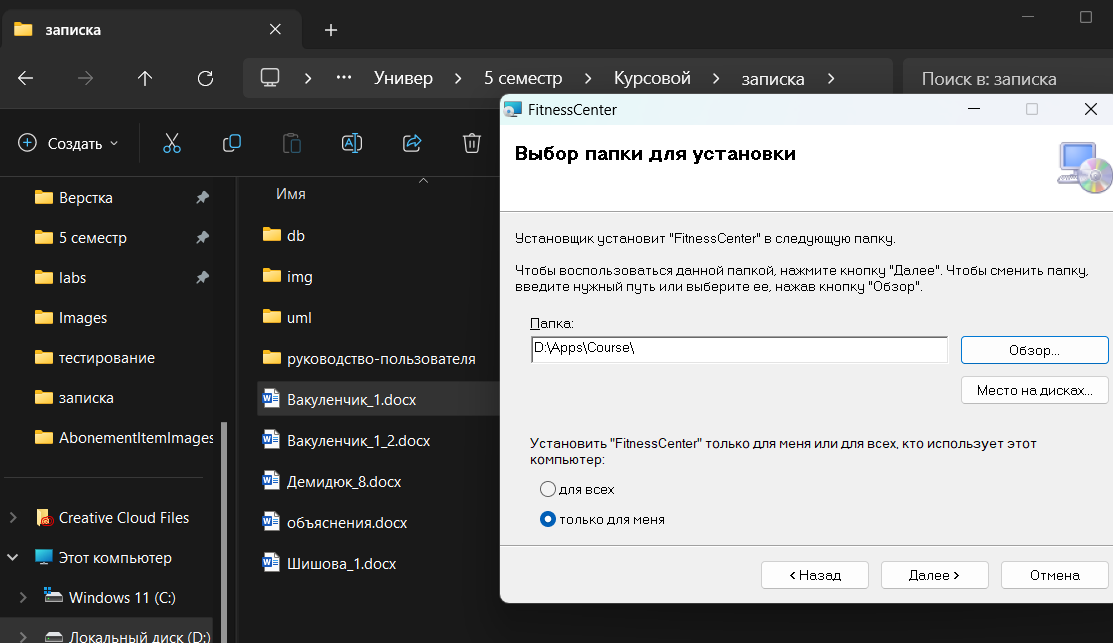


Рис 5.2 – выбор папки сохранения;

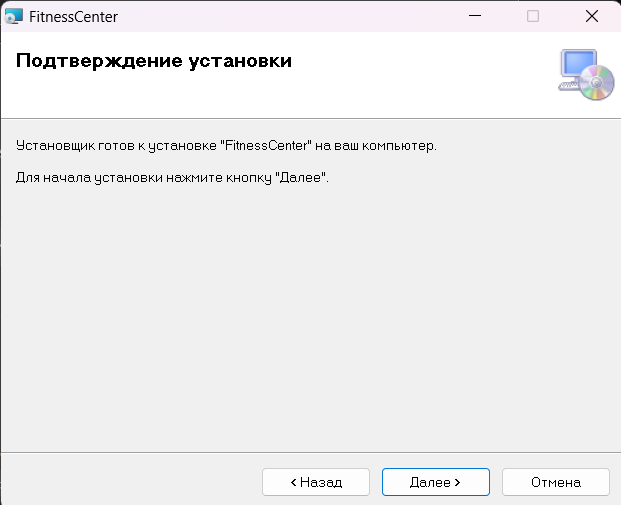


Рис 5.3 – подтверждение установки приложения;

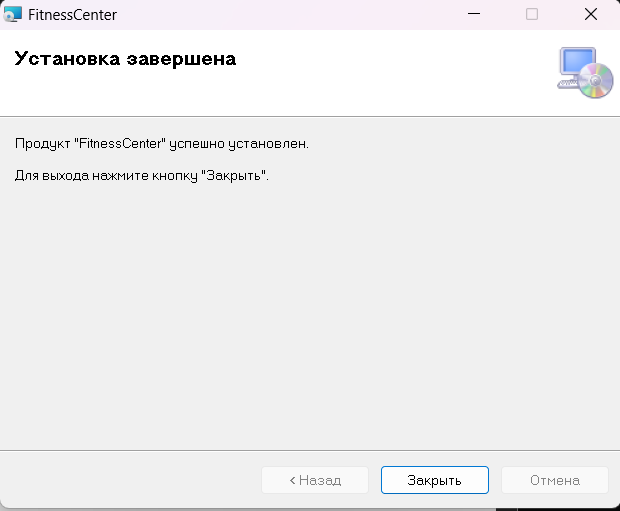


Рис 5.4 – оповещение об успешной установке;

# **5.3 Описание процесса использования приложения**

Теперь при запуске D:\Study\MorningCourseProject\CW-WPF-PartShop-main\CW-WPF-PartShop-main\CW-WPF-PartShop-main. Поскольку, в приложении используются абсолютные пути к различным картинкам, то для корректного отображения, стоит заменить все абсолютные пути в проекте (D:\Study\MorningCourseProject\CW-WPF-PartShop-main\CW-WPF-PartShop-main\CW-WPF-PartShop-main) при необходимости.

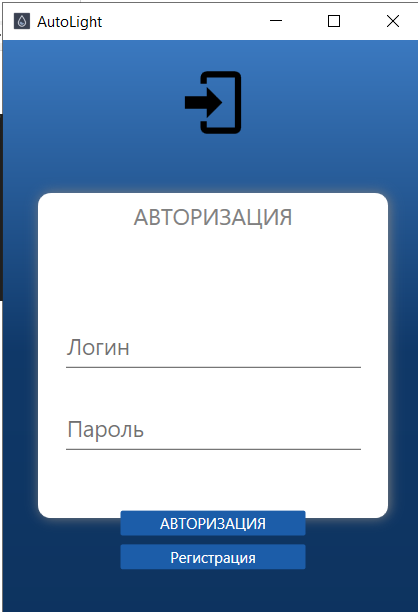


Рис 5.3.1 – Стартовое окно приложения

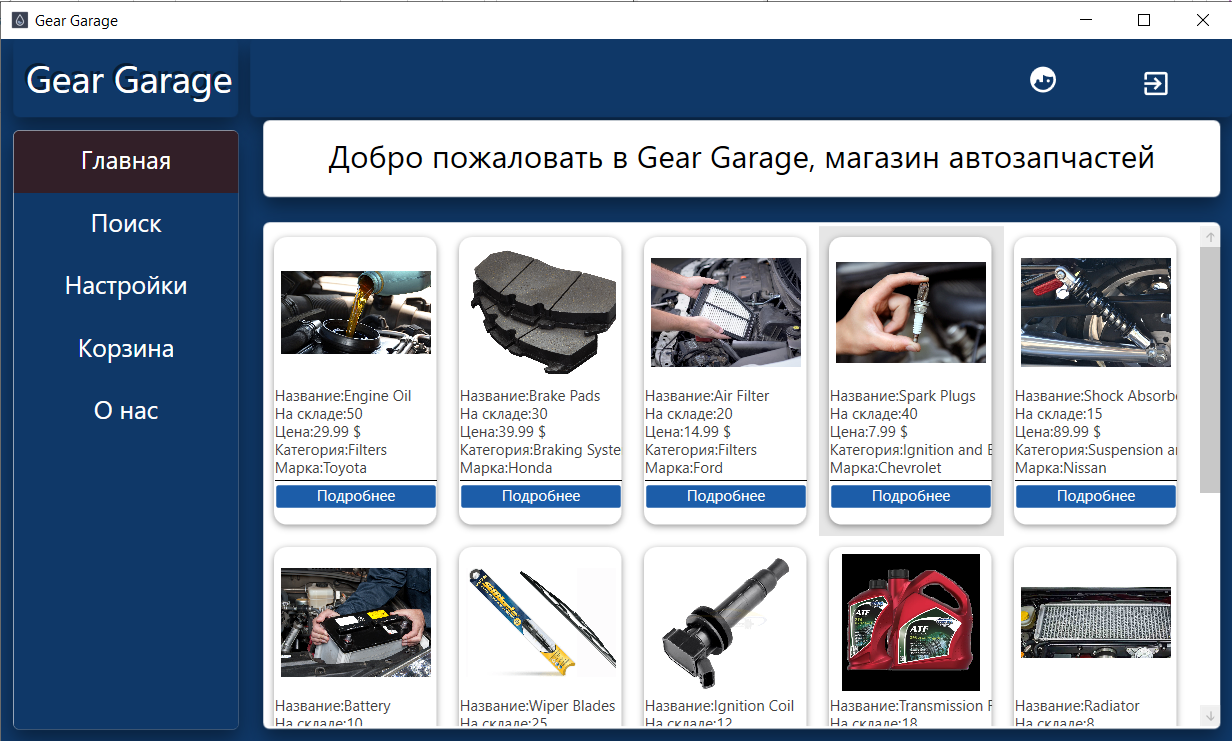


Рис 5.3.2 – Главная страница пользователя

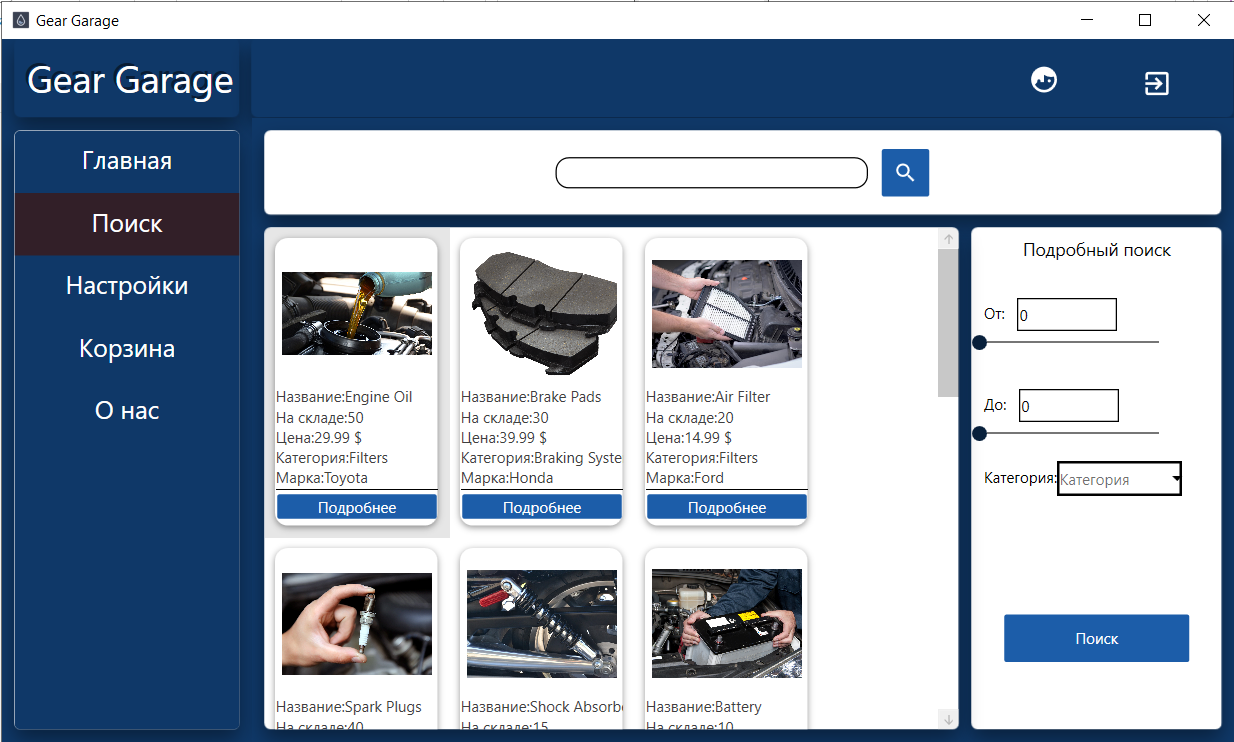


Рис 5.3.3 –страница «поиск»

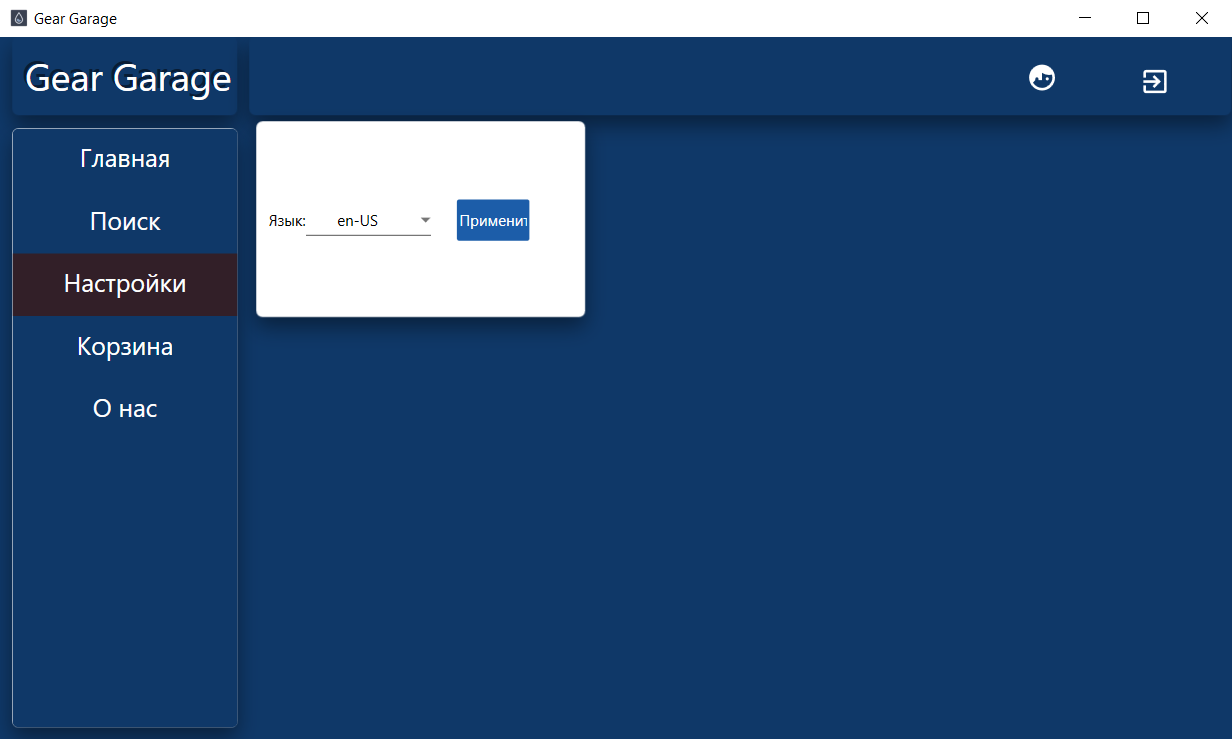


Рис 5.3.4 –страница «настройки»

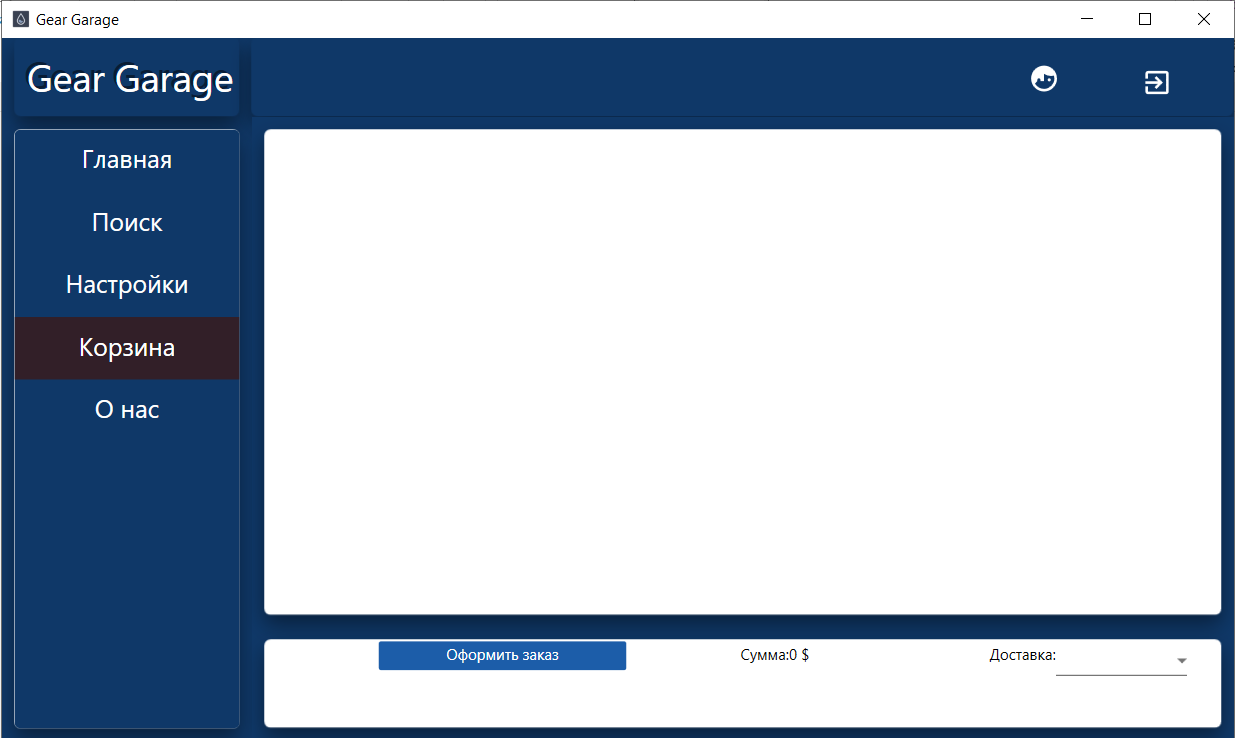


Рис 5.3.5 –страница «корзина»

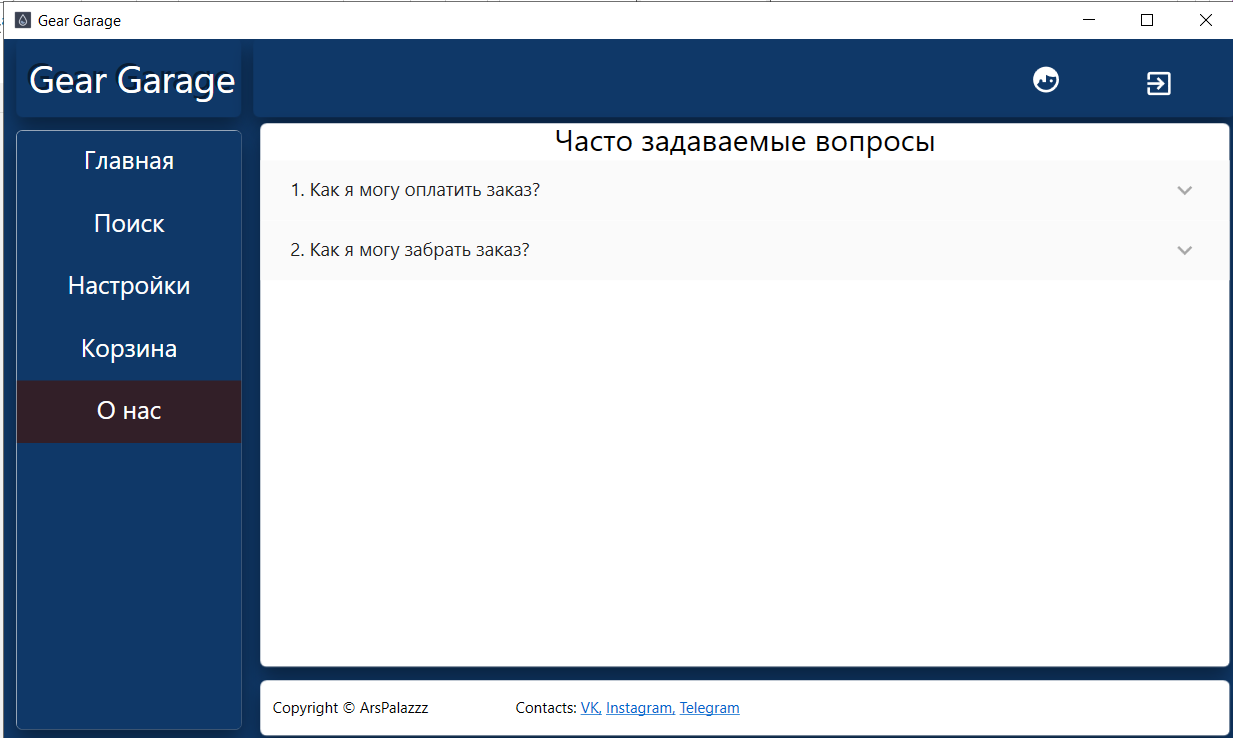


Рис 5.3.6 –страница «о нас»

**5.4 Вывод по разделу**

В данном разделе был представлен обзор процесса установки, настройки и использования разработанного приложения . Системные требования были подробно описаны, обеспечивая пользователей информацией о необходимых характеристиках и ресурсах компьютера для эффективной работы приложения.

Раздел также включает в себя шаги установки приложения, начиная с запуска установочного файла и заканчивая оповещением об успешной установке. Инструкции и скриншоты облегчают процесс установки, обеспечивая пользователей простым и понятным путем. Описание процесса использования приложения предоставляет пользователю взгляд на ключевые функции и возможности.

В целом, раздел предоставляет полное понимание того, как использовать и взаимодействовать с приложением, что делает его удобным и доступным для пользователей с различным опытом использования программного обеспечения.

# **Заключение**

В ходе разработки данного программного приложения были проведены тщательные исследования, анализ аналогов и анализ потребностей пользователей, что позволило сформировать ключевые направления проекта. Проектирование архитектуры, выбор технологий и инструментов, а также разработка функциональной модели и модели данных были выполнены с акцентом на максимальное соответствие требованиям современных фитнес-клубов и ожиданиям пользователей. Процесс разработки приложения затронул следующие основные этапы: анализ концепции и требований, проектирование архитектуры и базы данных, тестирование, установка и использование.

В целом, разработанное приложение для фитнес-центра успешно соответствует поставленным целям и ожиданиям пользователей. Оно предоставляет удобные инструменты для взаимодействия с клиентами, обработки заказов и управления информацией о тренерах, что делает его важным инструментом в эффективном функционировании фитнес-бизнеса.

**Список используемых источников**

1. Cosmosfit [Электронный ресурс] – Режим доступа: https://cosmofit.by – Дата

доступа: 16.10.2023

1. Evo-club [Электронный ресурс] – Режим доступа: https://evo-club.by – Дата

доступа: 17.10.2023

1. MF [Электронный ресурс] – Режим доступа: https://mf.by – Дата

доступа: 20.10.2023

1. Uml [Электронный ресурс] – Режим доступа: https://uml.org/ – Дата

доступа: 21.11.2023

1. Metanit [Электронный ресурс] – Режим доступа: https://metanit.com/ – Дата

доступа: 27.11.2023

# **ПРИЛОЖЕНИЕ А**



# **ПРИЛОЖЕНИЕ Б**

**Список используемых источников**

1. Exist.by [Электронный ресурс] – Режим доступа: https://exist.by/ – Дата доступа: 17.10.2023
2. Zap.by [Электронный ресурс] – Режим доступа: https://zap.by/ – Дата доступа: 17.10.2023
3. auto1.by [Электронный ресурс] – Режим доступа: https://auto1.by/ – Дата доступа: 17.10.2023