

Министерство цифрового развития государственного управления,  
Информационных технологий и связи Республики Татарстан  
государственное автономное профессиональное образовательное учреждение  
«Международный центр компетенций –  
Казанский техникум информационных технологий и связи»

ОТЧЕТ  
ПО УЧЕБНОЙ ПРАКТИКЕ

ПМ.02 УП02.01 «Осуществление интеграции программных модулей»

Выполнил обучающийся Крайчик Александр Евгеньевич  
(ФИО)

Группа 422 специальность 09.02.07 «Информационные системы и  
программирование: АБД»

Оценка \_\_\_\_\_(\_\_\_\_\_)

Начало практики: 02.03.2022 г.

\_\_\_\_\_  
(Ф.И.О. подпись руководителя практики)

Окончание практики: 22.03.2022 г.

\_\_\_\_\_  
(Ф.И.О. подпись руководителя практики)

М.П.

Казань, 2022г.

## СОДЕРЖАНИЕ

1	ОБОСНОВАНИЕ СРЕДСТВ РАЗРАБОТКИ .....	3
1.1	Обоснование выбора языка программирования .....	3
1.2	Обоснование выбора СУБД .....	5
2	ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ .....	8
2.1	Описание предметной области «Салон красоты» .....	8
2.2	Реализация .....	9
2.3	Руководство пользователя.....	10
2.4	Тестирование .....	18
	ЗАКЛЮЧЕНИЕ .....	25
	АТТЕСТАЦИОННЫЙ ЛИСТ	

# 1 ОБОСНОВАНИЕ СРЕДСТВ РАЗРАБОТКИ

## 1.1 Обоснование выбора языка программирования

В качестве среды разработки была выбрана Visual Studio, так как данная среда поддерживает язык программирования C# и позволяет соединить программное обеспечение с базой данных через ADO.NET, что необходимо в случае разработки данного ПО.

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.

По умолчанию в ADO.NET имеются следующие встроенные провайдеры:

- провайдер для MS SQL Server;
- провайдер для OLE DB (Предоставляет доступ к некоторым старым версиям MS SQL Server, а также к БД Access, DB2, MySQL и Oracle);
- провайдер для ODBC (Провайдер для тех источников данных, для которых нет своих провайдеров);
- провайдер для Oracle;
- провайдер EntityClient. Провайдер данных для технологии ORM Entity Framework;
- провайдер для сервера SQL Server Compact 4.0.

Основные пространства имен, которые используются в ADO.NET:

- System.Data: определяет классы, интерфейсы, делегаты, которые реализуют архитектуру ADO.NET;
- System.Data.Common: содержит классы, общие для всех провайдеров ADO.NET;
- System.Data.Design: определяет классы, которые используются для создания своих собственных наборов данных;

- `System.Data.Odbc`: определяет функциональность провайдера данных для ODBC;
- `System.Data.OleDb`: определяет функциональность провайдера данных для OLE DB;
- `System.Data.Sql`: хранит классы, которые поддерживают специфичную для SQL Server функциональность;
- `System.Data.OracleClient`: определяет функциональность провайдера для баз данных Oracle;
- `System.Data.SqlClient`: определяет функциональность провайдера для баз данных MS SQL Server;
- `System.Data.SqlServerCe`: определяет функциональность провайдера для SQL Server Compact 4.0;
- `System.Data.SqlTypes`: содержит классы для типов данных MS SQL Servera;
- `Microsoft.SqlServer.Server`: хранит компоненты для взаимодействия SQL Server и среды CLR.

Так же Visual Studio поддерживает язык разметки XAML, с помощью которого был разработан интерфейс приложения. XAML (eXtensible Application Markup Language) - язык разметки, используемый для инициализации объектов в технологиях на платформе .NET.

Применительно к WPF данный язык используется прежде всего для создания пользовательского интерфейса декларативным путём. Хотя функциональность XAML только графическими интерфейсами не ограничивается: данный язык также используется в технологиях WCF и WF, где он никак не связан с графическим интерфейсом. XAML - не является обязательной частью приложения, можно обходиться без него, создавая все элементы в файле связанного с ним кода на языке C#. Однако использование XAML все-таки несёт некоторые преимущества:

- возможность отделить графический интерфейс от логики приложения, благодаря чему над разными частями приложения могут относительно автономно работать разные специалисты: над интерфейсом - дизайнеры, над кодом логики – программисты;

- компактность и понятность.

При разработке программного обеспечения «системы учета парковки» был выбран язык программирования C#, так как данный язык имеет следующим рядом достоинств:

- C# является объектно-ориентированным языком программирования;

- имеется удобная и бесплатная среда разработки;

- прост в освоении и постоянно совершенствуется;

- позволяет создавать мобильные приложения.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — C#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

## 1.2 Обоснование выбора СУБД

В качестве СУБД была выбрана SQL Server Management Studio, так как она является наиболее удобной и функциональной среди других СУБД.

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server, Базы данных SQL Azure и Azure Synapse Analytics, а также управления ими. Среда SSMS предоставляет единую полнофункциональную служебную программу, которая сочетает в себе обширную группу графических инструментов с рядом отличных редакторов сценариев для доступа к службе SQL Server для разработчиков и администраторов баз данных всех профессиональных уровней.

SQL Server Management studio имеет следующие достоинства:

- СУБД масштабируется, поэтому работать с ней можно на портативных ПК или мощной мультипроцессорной технике. Процессор может одновременно обрабатывать большой объем запросов;
- размер страниц – до 8 кб, поэтому данные извлекаются быстро, подробную и сложную информацию хранить удобнее. Система позволяет обрабатывать транзакции в интерактивном режиме, есть динамическая блокировка;
- Microsoft SQL Server Developer Edition бесплатен для разработки и тестирования. В апреле 2016 года Microsoft наконец-то сделала бесплатной версию для разработчиков, которая по своему функционалу полностью совпадает с Enterprise. До этого стоимость одной лицензии разработчика была примерно 2-3 тысяч рублей;
- рутинные административные задачи автоматизированы: это управление блокировками, памятью, редактора размеров файлов. У системы продуманы настройки, можно создать профили пользователей;
- реализован поиск по фразам, тексту, словам, можно создавать ключевые индексы;
- в SQL Server есть репликации через интернет, предусмотрена синхронизация. Есть полноценный веб-ассистент для форматирования страниц;

- в систему интегрирован сервер интерактивного анализа для принятия решений, создания корпоративных отчётов. Есть службы преобразования информации;
- запросы можно формулировать на английском языке, без программирования;
- СУБД поддерживает работу с другими продуктами Microsoft: Access, MS Excel;
- СУБД имеет высокую производительность;
- SQL Server достаточно популярная и имеет хорошую репутацию среди других;
- SQL Server Management Studio кроссплатформенная и хорошо поддерживается на других устройствах.

Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основным используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

## 2 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

### 2.1 Описание предметной области «Салон красоты»

Вашей задачей является разработка системы для компании, которая оказывает услуги клиентам и продаёт определенные товары.

Данную систему можно разделить на несколько подсистем:

- подсистема по работе с клиентами;
- подсистема по работе с сотрудниками;
- подсистема по работе с товарами;
- подсистема по работе с услугами;
- подсистема для управления компанией.

Основное задание на разработку.

В нашем салоне красоты «Бровушка» администраторы продают дополнительные товары для клиентов. Например, это могут быть шампуни, маски для лица, предметы личной гигиены и прочие товары. Клиентам очень удобно, что теперь нет необходимости долго искать рекомендованные мастерами товары в сторонних магазинах – они могут приобрести все сразу после проведения процедур. Однако для администраторов салона красоты необходима система для учета товаров.

Вам предстоит разработать подсистему для работы с товарами нашей компании, которая должна включать в себя следующий функционал:

- Просмотр списка товаров;
- добавление/удаление/редактирование данных о товарах;
- предложения дополнительных товаров к основным;
- просмотр истории покупок товаров салона красоты.

Для достижения поставленной цели были сформулированы и решены следующие задачи:

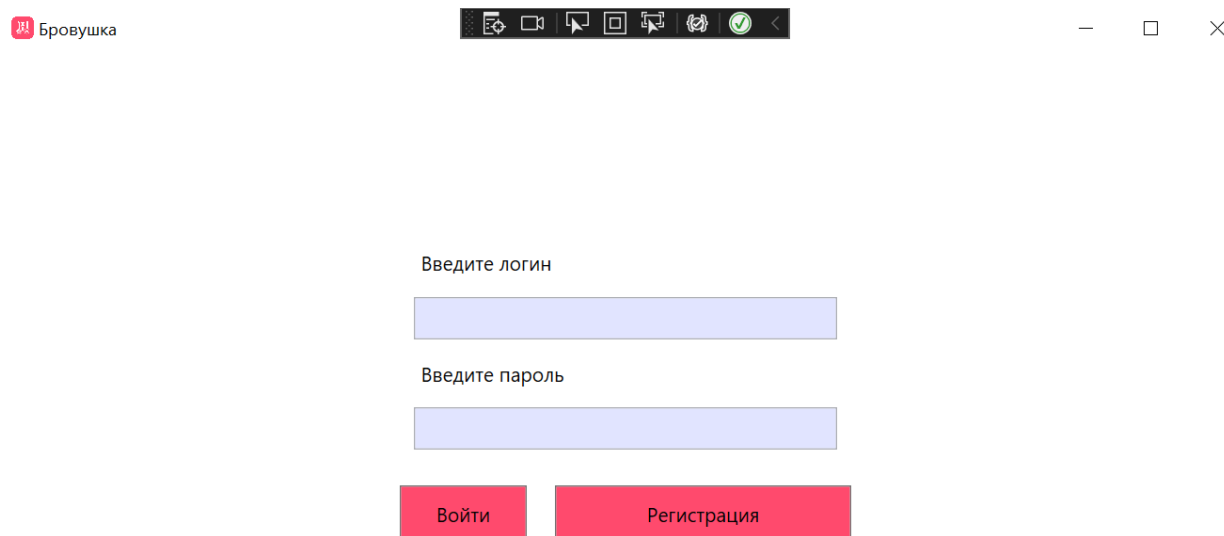
1. Изучение предметной области;
2. разработка базы данных;
3. создание интерфейса;





## 2.3 Руководство пользователя

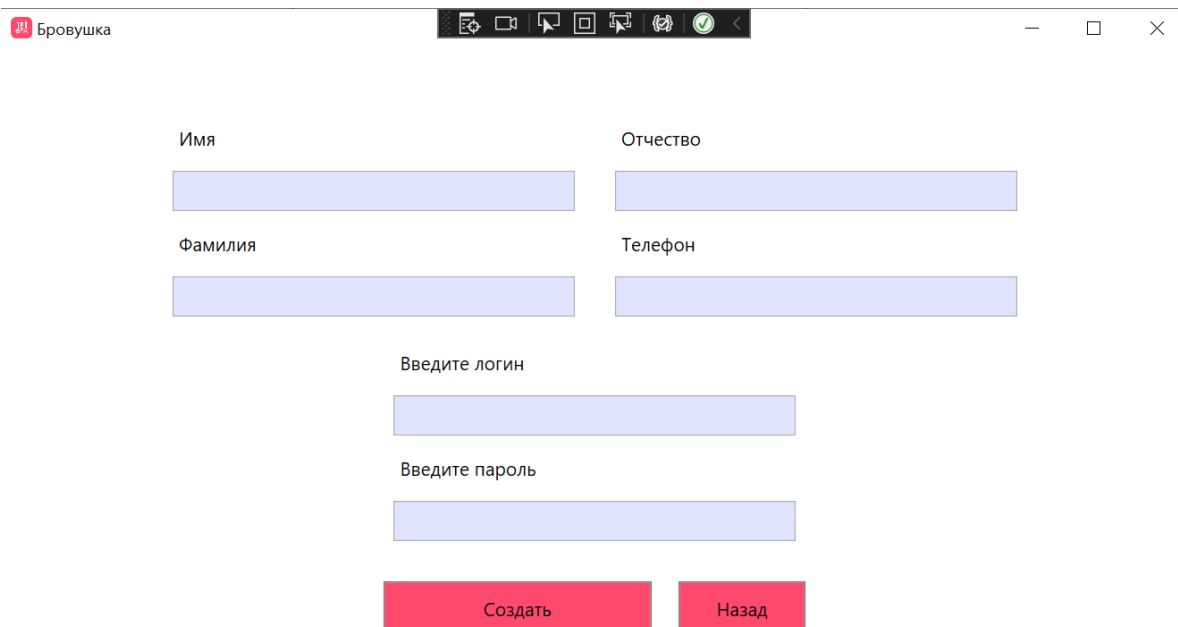
Для начала работы нужно войти в учётную запись, или зарегистрировать нового пользователя (Рисунок 2.2).



The screenshot shows the initial screen of the 'Бровушка' application. At the top, there is a browser window title 'Бровушка' and a toolbar with various icons. The main content area contains two input fields for login and password, each with a label 'Введите' (Enter). Below the input fields are two red buttons: 'Войти' (Login) and 'Регистрация' (Registration).

Рисунок 2.2 Начальный экран

Для регистрации необходимо нажать на кнопку регистрации, которая перенаправит пользователя на окно регистрации (Рисунок 2.3).



The screenshot shows the registration window of the 'Бровушка' application. It features four input fields for registration details: 'Имя' (Name), 'Отчество' (Patronymic), 'Фамилия' (Surname), and 'Телефон' (Phone). Below these fields are two input fields for login and password, each with a label 'Введите' (Enter). At the bottom, there are two red buttons: 'Создать' (Create) and 'Назад' (Back).

Рисунок 2.3 Окно регистрации

После успешной регистрации программа уведомит о регистрации, и перенаправит пользователя на начальный экран, где можно авторизоваться в роли администратора или пользователя.

Далее пользователь окажется на главной странице со списком продуктов, где имеется сортировка и переход на доступные страницы (Рисунок 2.4).

При переходе на главную страницу появляется следующий функционал:

1. Поля с сортировкой по возрастанию, убыванию, наименованию и описанию, так же имеется фильтр по производителям;
2. В зависимости от роли появляются следующие кнопки:
  - 2.1 При роли администратора кнопка «Добавить»;
  - 2.2 При роли пользователя кнопка «Моя корзина».
3. Поле с имеющимся количеством продуктов, которое взаимодействует с сортировку;
4. Прокрутка картинок продукта, если таковы имеются.

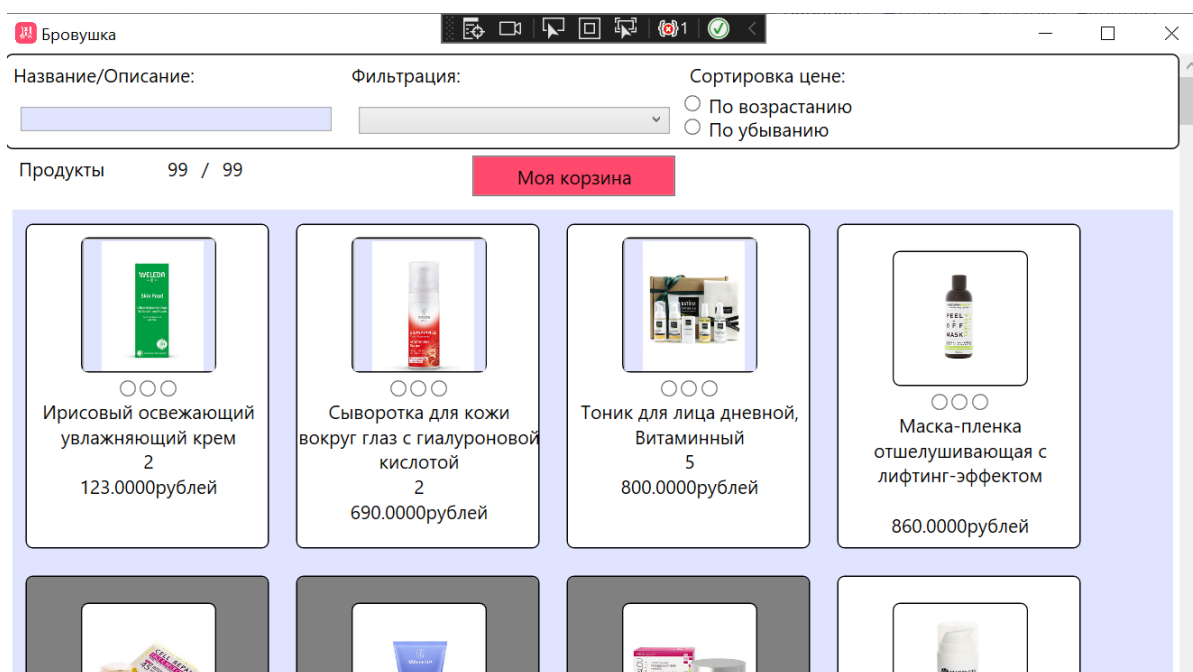


Рисунок 2.4 Главная страница

Нажав на продукт два раза, пользователь перейдет на его страницу (Рисунок 2.5). Для каждой роли имеются определенные ограничения. Пользователи, вошедшие как администраторы могут изменять поля.

При переходе на страницу продукта появляется следующий функционал:

1. Для администратора:
  - Изменять все поля, кроме ID продукта;
  - назначать активность продукту;
  - добавлять дополнительные картинки для продукта или же изменять основную картинку;
  - изменять список предложенных товаров к этому продукту;
  - удалить товар;
  - просмотр истории продаж данного товара (Рисунок 2.6).
2. Для пользователя:
  - Просмотр данных о продукте (Рисунок 2.7).
3. Поля с сортировкой по возрастанию, убыванию, наименованию и описанию, так же имеется фильтр по производителям;
4. В зависимости от роли появляются следующие кнопки:
5. При роли администратора кнопка «Добавить»;
6. при роли пользователя кнопка «Моя корзина».
7. Поле с имеющимся количеством продуктов, которое взаимодействует с сортировку;
8. Прокрутка дополнительных картинок.

Бровушка

Товар

ID

Название

Цена

Описание

Производитель  ☒ Активный

Фото

Вы так же можете предложить

Добавить доп. картинк

Изменить ориг. картин

Сохранить

Удалить

История продаж

Назад

Рисунок 2.5 Выбранный продукт для администратора

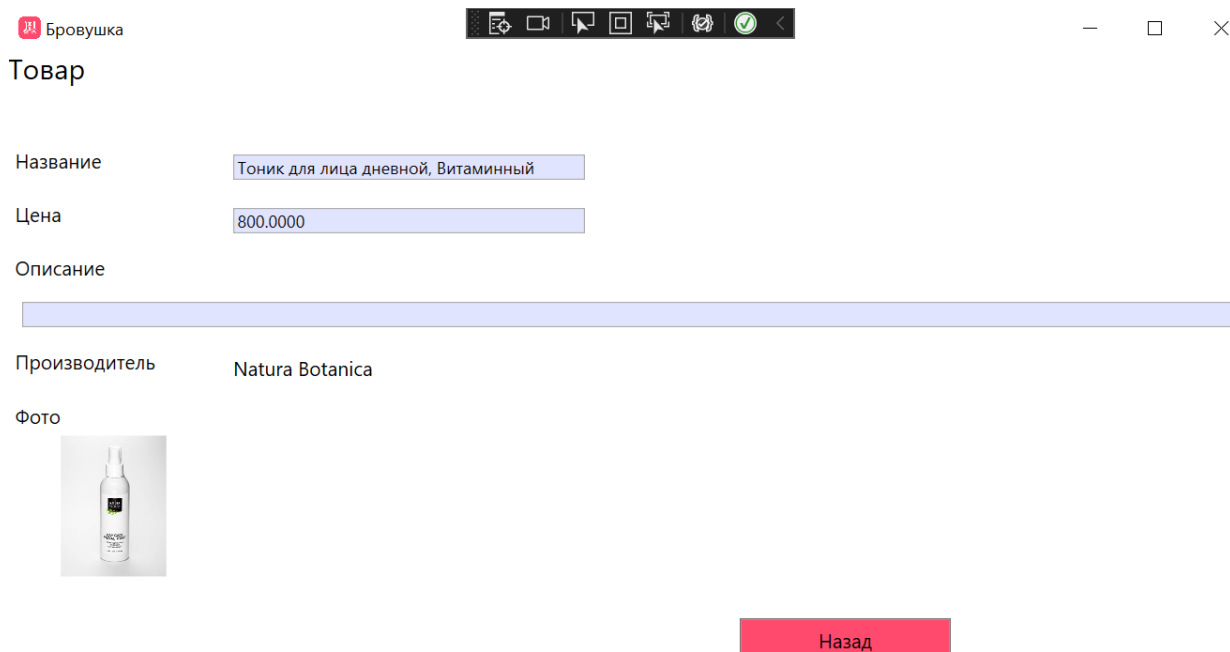
Бровушка

Ирисовый освежающий уе

Товар	Дата	Кол-во
Ирисовый освежающий увлажняющий крем	3/7/2022 12:00:00 AM	11
Ирисовый освежающий увлажняющий крем	3/6/2022 12:00:00 AM	12

Назад

Рисунок 2.6 История продаж выбранного продукта



Бровушка

Товар

Название: Тоник для лица дневной, Витаминный

Цена: 800.0000

Описание:

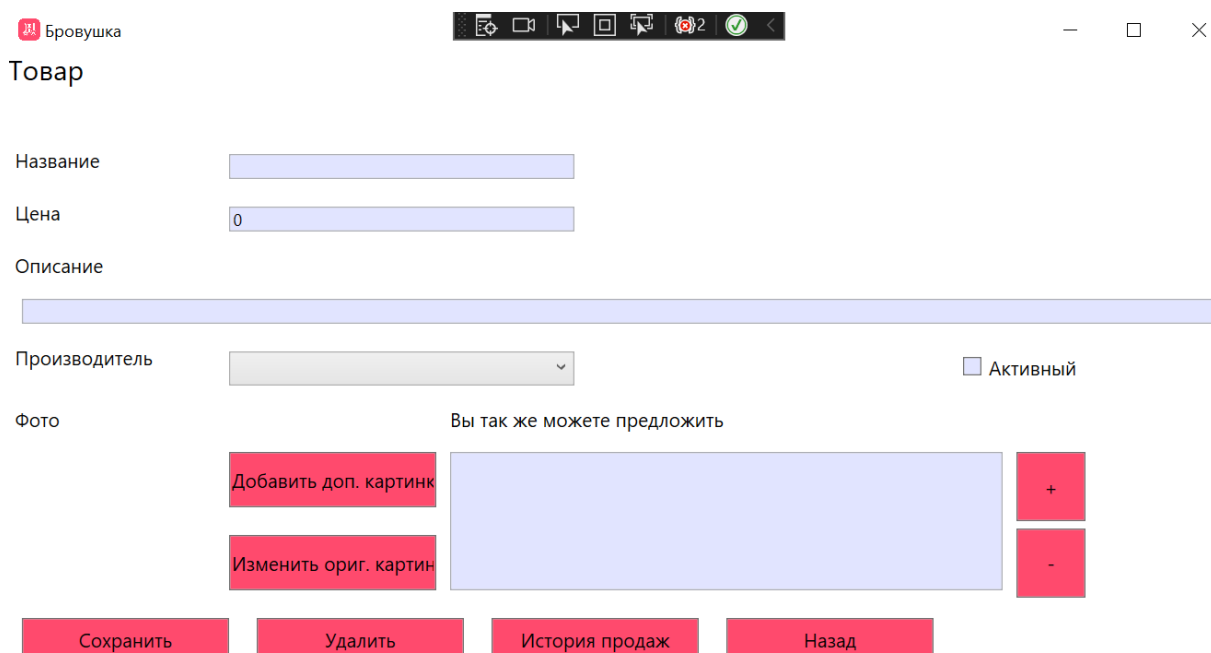
Производитель: Natura Botanica

Фото:

Назад

Рисунок 2.7 Выбранный продукт для пользователя

Для добавления, администратору необходимо нажать на кнопку «Добавить», после чего появится окно добавления (Рисунок 2.8).



Бровушка

Товар

Название:

Цена: 0

Описание:

Производитель:

Активный ☐

Фото: Вы так же можете предложить

Добавить доп. картинк

Изменить ориг. картин

Сохранить

Удалить

История продаж

Назад

Рисунок 2.8 Страница добавления

После заполнения всех или нескольких полей, необходимо нажать на кнопку «Сохранить». После всех необходимых действий администратор окажется на главной странице. Такая же функция работает и с редактированием выбранного товара.

При нажатии кнопки «Удалить» на странице выбранного продукта откроется окно с подтверждением удаления существующего продукта (Рисунок 2.9).

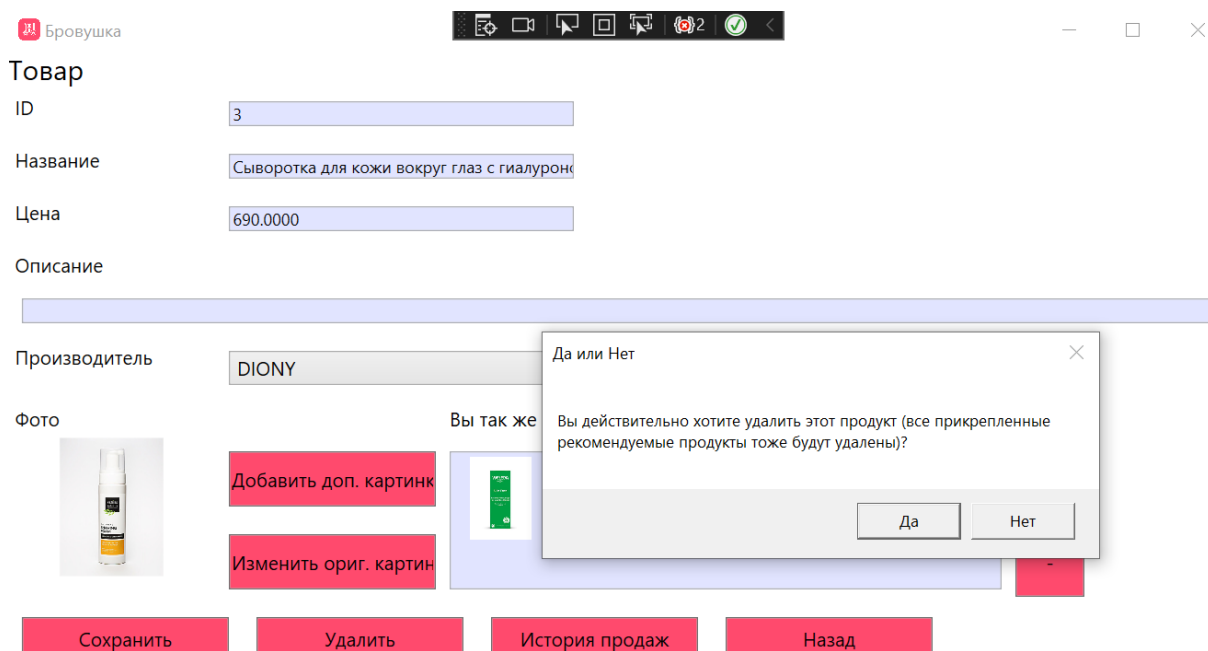


Рисунок 2.9 Удаление продукта

Если товар учитывается в истории продаж, то удаление невозможно (Рисунок 2.10).

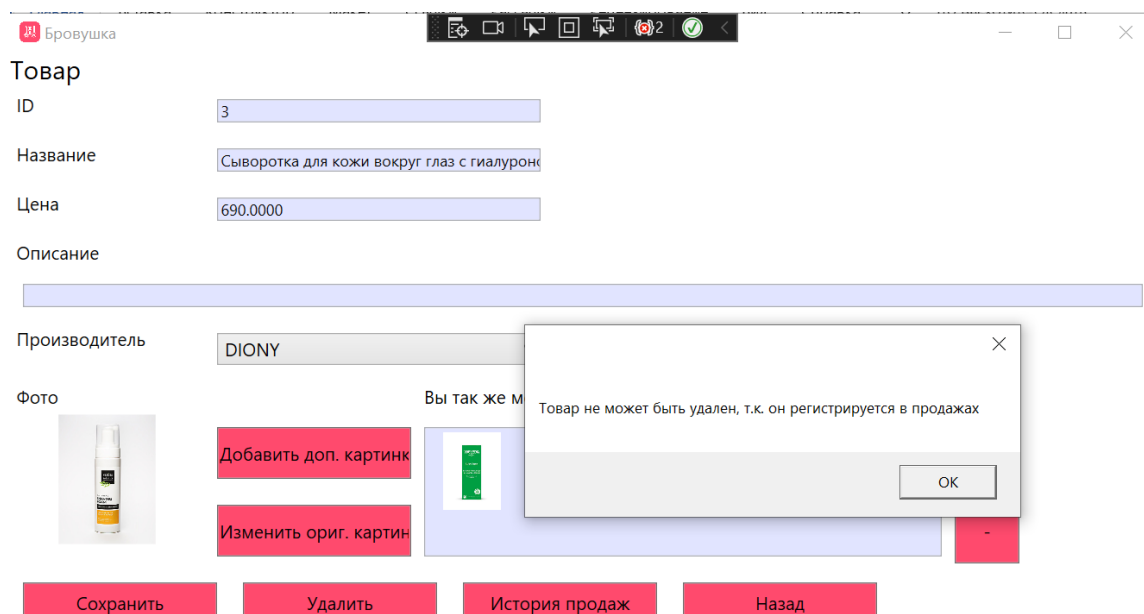


Рисунок 2.10 Окно с сообщением о том, что удаление невозможно

Поиск по названию, комментарию и сортировки доступны как администраторам, так и пользователям.

Для того, чтобы выполнить поиск по названию продукта, необходимо ввести его вручную (Рисунок 2.9).

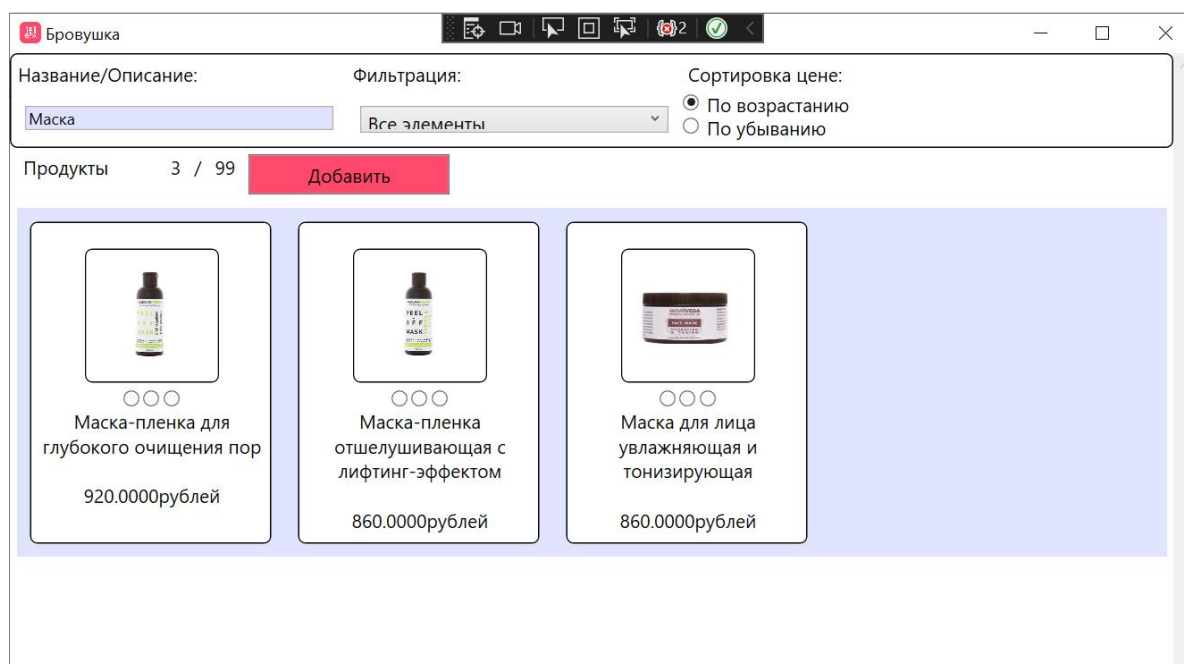


Рисунок 2.11 Поиск по критериям

Для просмотра другой корзины пользователя необходимо нажать на кнопку «Моя корзина», находящаяся на главной странице (Рисунок 2.10).

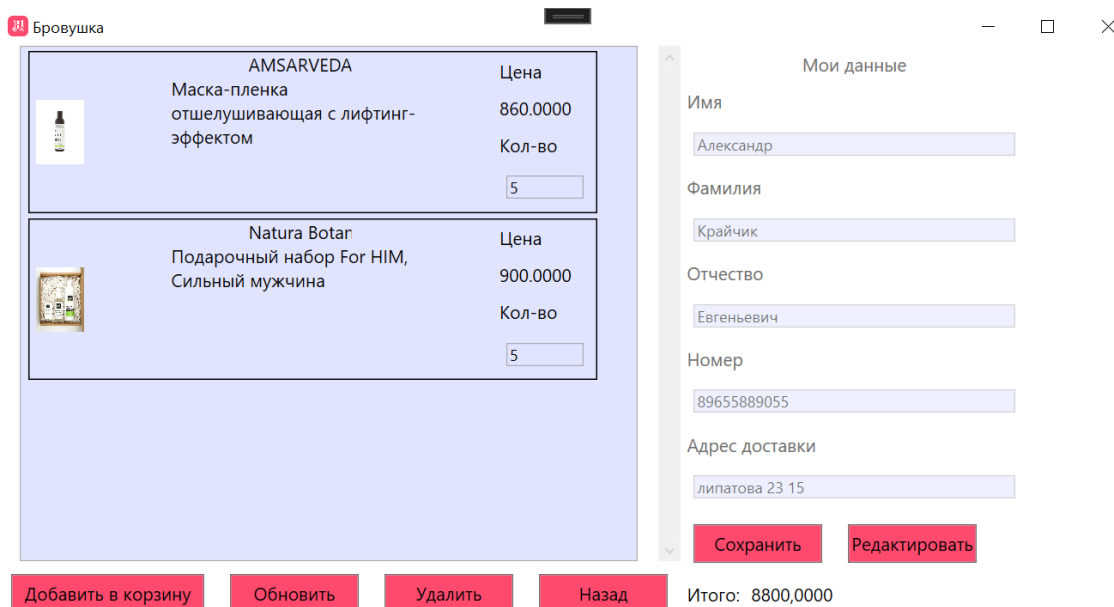


Рисунок 2.12 Страница корзины пользователя

Под данными пользователя выводится строка о общей сумме корзины пользователя.



Пользователь может изменять свои данные, нажав на кнопку «Редактировать» и после на кнопку «Сохранить». Пользователь может добавлять (Рисунок 2.13) и изменять количество (Рисунок 2.14) продуктов, добавленных в корзину.

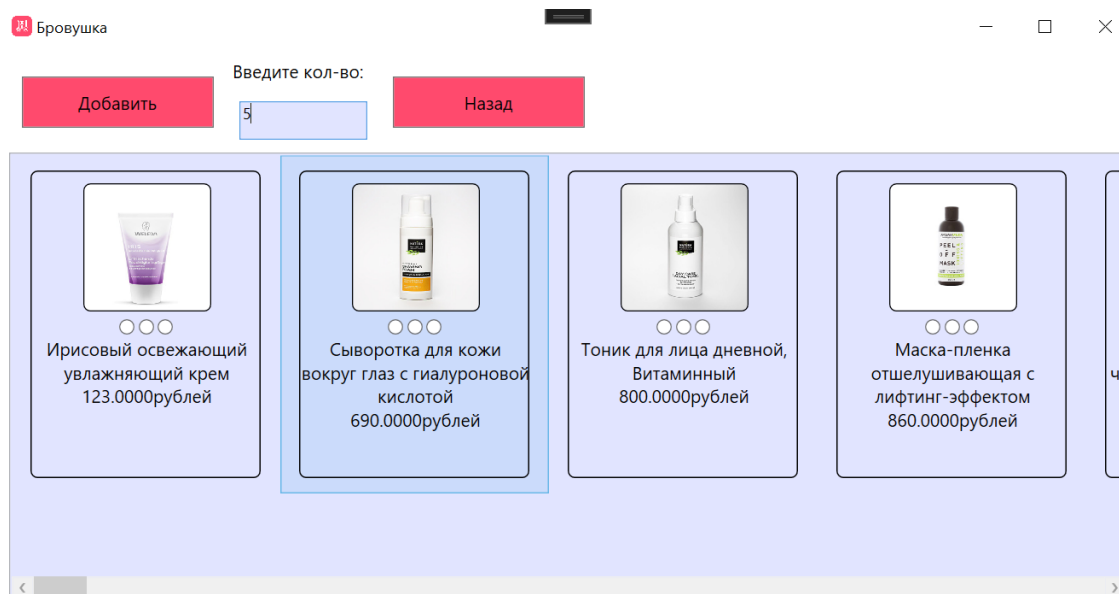


Рисунок 2.13 Страница добавления продукта в корзину

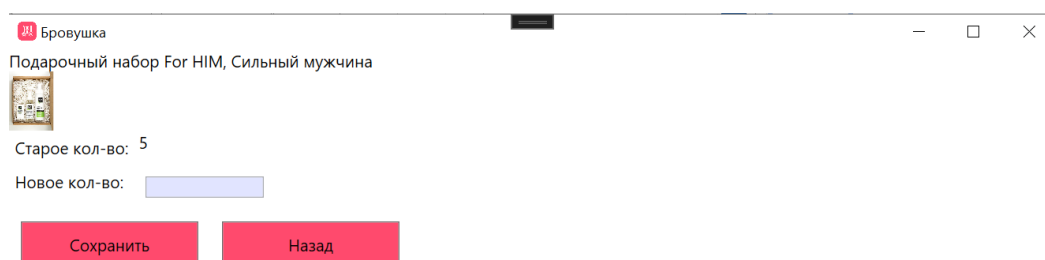


Рисунок 2.14 Страница изменения продукта из корзины

## 2.4 Тестирование

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Цели тестирования:

- повысить вероятность того, что приложение, предназначенное для тестирования, будет работать правильно при любых обстоятельствах;
- повысить вероятность того, что приложение, предназначенное для тестирования, будет соответствовать всем описанным требованиям;
- предоставление актуальной информации о состоянии продукта на данный момент.

Типы тестирования ПО:

- тестирование типа «белый ящик» проверяет внутренние структуры и модули, игнорирует ожидаемую функциональность для конечных пользователей. Это может быть тестирование API, внесение неисправностей (fault injection), модульное тестирование, интеграционное тестирование;
- тестирование типа «чёрный ящик» больше интересуется тем, что делает ПО, а не как делает. Это означает, что тестировщики не обязаны ни разбираться в объекте тестирования, ни понимать, как он работает под капотом. Такой тип тестирования нацелен на конечных пользователей, их опыт взаимодействия с видимым интерфейсом. К «чёрным ящикам» относится тестирование на основе моделей, тестирование способов использования, таблицы переходов состояний, спецификационное тестирование и т. Д.;
- тестирование типа «серый ящик» проектируется со знанием программных алгоритмов и структур данных (белый ящик), но выполняется на пользовательском уровне (чёрный ящик). Сюда относится регрессионное тестирование и шаблонное тестирование (pattern testing).

Был проведен ряд тестов программного продукта, в которых показан результат, указанный в таблицах.

Test case #1:

Test Case #	TC_DB_1
Приоритет теста	высокий
Название тестирования/Имя	Проверка авторизации с проверкой правильным/неправильным логином и паролем
Резюме испытания	<ol style="list-style-type: none"> <li>1. Проверка авторизации с правильным логином и паролем и переходы по формам;</li> <li>2. Проверка невозможности авторизации с неверным логином и паролем</li> </ol>
Шаги тестирования	<p>Для правильной пары логин-пароль:</p> <ol style="list-style-type: none"> <li>1. Ввод пары логин-пароль для учетной записи (УЗ) Администратор;</li> <li>2. Ввод пары логин-пароль для УЗ Пользователь;</li> </ol> <p>Для неправильной пары логин-пароль:</p> <ol style="list-style-type: none"> <li>1. Проверка возможности входа с некорректными данными;</li> <li>2. Проверка возможности входа с незаполненными полями;</li> </ol>
Данные тестирования	<p>Для правильной пары логин-пароль:</p> <ol style="list-style-type: none"> <li>1. Логин: 123 Пароль: 123</li> <li>2. Логин: 1234 Пароль: 1234</li> </ol> <p>Для неправильной пары логин-пароль:</p> <p>Логин: 1111 Пароль: uvb76</p>
Ожидаемый результат	<p>Для правильной пары логин-пароль:</p> <ol style="list-style-type: none"> <li>1. Вход под УЗ Администратора, открытие страницы выбора таблиц;</li> </ol>

	<p>2. Вход под УЗ Пользователя, открытие страницы выбора таблиц;</p> <p>Для неправильной пары логин-пароль:</p> <p>1. Невозможность входа под неправильной парой логин-пароль, вывод сообщение об ошибке;</p> <p>2. Невозможность входа с незаполненными полями, вывод сообщения об ошибке.</p>
Фактический результат	<p>Для правильной пары логин-пароль:</p> <p>1. Вход под УЗ Администратора, открытие страницы выбора таблиц;</p> <p>2. Вход под УЗ Пользователя, открытие страницы выбора таблиц;</p> <p>Для неправильной пары логин-пароль:</p> <p>1. Невозможность входа под неправильной парой логин-пароль, вывод сообщение об ошибке;</p> <p>2. Невозможность входа с незаполненными полями, вывод сообщения об ошибке.</p>
Предпосылки	Запуск приложения и автоматический переход на страницу авторизации.
Постусловия	Реализована связь с базой данных.
Статус (Pass/Fail)	Pass.
Комментарии	При попытке авторизации с пустыми полями, то все исключения обработаны.

Test case #2:

Test Case #	TC_DB_2
Приоритет теста	высокий
Название тестирования/Имя	Регистрация пользователя
Резюме испытания	<ol style="list-style-type: none"> <li>1. Проверка регистрации пользователя.</li> <li>2. Проверка невозможности регистрации.</li> </ol>
Шаги тестирования	<p>Для удачной регистрации пользователя:</p> <ol style="list-style-type: none"> <li>1. Ввод логин-пароль в поля</li> </ol> <p>Для неудачной регистрации пользователя:</p> <ol style="list-style-type: none"> <li>1. Проверка возможности регистрации с неполными данными;</li> <li>2. Проверка возможности регистрации с незаполненными полями;</li> </ol>
Данные тестирования	<p>Для удачной регистрации:</p> <ol style="list-style-type: none"> <li>1. Логин: albedo Пароль: Cine@123;</li> <li>2. Заполнены все поля регистрации;</li> </ol> <p>Для неудачной регистрации:</p> <ol style="list-style-type: none"> <li>1. Не все поля заполнены;</li> </ol>
Ожидаемый результат	<p>Для удачной регистрации:</p> <ol style="list-style-type: none"> <li>1. Регистрация пользователя, переход к странице регистрации</li> </ol> <p>Для неудачной пары логин-пароль:</p> <ol style="list-style-type: none"> <li>1. Невозможность регистрации при неполном заполнении полей, вывод сообщение об ошибке;</li> <li>2. Невозможность входа с незаполненными полями, вывод сообщения об ошибке.</li> </ol>
Фактический результат	Для удачной регистрации:

	<p>1. Регистрация пользователя, переход к странице регистрации</p> <p>Для неудачной пары логин-пароль:</p> <p>1. Невозможность регистрации при неполном заполнении полей, вывод сообщение об ошибке;</p> <p>2. Невозможность входа с незаполненными полями, вывод сообщения об ошибке.</p>
Предпосылки	Запуск приложения и возможность перехода с главной страницы по кнопкам: регистрация, войти
Постусловия	Реализована связь с базой данных
Статус (Pass/Fail)	pass
Комментарии	При попытке регистрации с пустыми и неполными полями, неправильной парой логин-пароль все исключения обработаны

Test case #3:

Test Case #	TC_DB_3
Приоритет теста	высокий
Название тестирования/Имя	Проверка добавления продукта
Резюме испытания	<p>1. Проверка добавления продукта.</p> <p>2. Проверка невозможности добавления продукта.</p>
Шаги тестирования	<p>Для добавления:</p> <p>1. Ввод пары логин-пароль для учетной записи (УЗ) Администратор;</p> <p>2. Добавление данных о продукте</p>

	<p>Для неудачного добавления:</p> <ol style="list-style-type: none"> <li>1. Проверка возможности входа с незаполненными полями;</li> </ol>
Данные тестирования	<p>Для правильных данных:</p> <ol style="list-style-type: none"> <li>1. Крем, Любой производитель, 1800</li> </ol> <p>Для неправильных данных:</p> <ol style="list-style-type: none"> <li>2. Пустые поля</li> <li>3. Palialan,Palialan@gmail.com, sdfsdffs,sixty seven, Dungeon Masters</li> </ol>
Ожидаемый результат	<p>Для добавления продукта:</p> <ol style="list-style-type: none"> <li>1. Вход под УЗ Администратора,</li> <li>2. Открытие страницы</li> <li>3. Добавление нового продукта и сохранение;</li> </ol> <p>Для не добавления продукта:</p> <ol style="list-style-type: none"> <li>1. Вход под УЗ Администратора.</li> <li>2. Открытие страницы.</li> <li>3. Ввод пустых полей или некорректных данных.</li> <li>4. Сообщение об ошибке.</li> </ol>
Фактический результат	<p>Для добавления продукта:</p> <ol style="list-style-type: none"> <li>1. Вход под УЗ Администратора.</li> <li>2. Открытие страницы.</li> <li>3. Добавление нового продукта и сохранение;</li> </ol> <p>Для не добавления продукта:</p> <ol style="list-style-type: none"> <li>1. Вход под УЗ Администратора.</li> <li>2. Открытие страницы.</li> </ol>

	<p>3. Ввод пустых полей или некорректных данных.</p> <p>4. Сообщение об ошибке.</p>
Предпосылки	Запуск приложения и возможность перехода с главной страницы по кнопкам: регистрация, войти
Постусловия	Реализована связь с базой данных
Статус (Pass/Fail)	pass
Комментарии	При попытке добавления продукта с пустыми полями и некорректными данными, все исключения обработаны



## ЗАКЛЮЧЕНИЕ

В процессе прохождения учебной практики была разработана автоматизированная информационная система «Салон красоты». Программный продукт соответствует всем нормам и требованиям, полученным из индивидуального задания. Также программный продукт соответствует стандартам дизайнерских решений и макетов. Программный продукт может использовать весь необходимый функционал и имеет все заложенные в него возможности для выполнения необходимых требований и задач, которые были поставлены в процессе прохождения учебной практики.

В результате достигнута цель и следующие задачи:

1. изучена предметная область;
2. разработана база данных;
3. создан интерфейс;
4. создан функционал приложения с использованием базы данных