

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет _____ Информационных технологий _____

Кафедра _____ Информационные системы и технологии _____

Специальность: 1–40 01 01 Программное обеспечение информационных технологий

Специализация Программирование интернет приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

« Служба доставки »

Выполнил студент _____ Хорошко Егор Викторович _____
(Ф.И.О.)

Руководитель проекта _____ *Бондарчик Е.Н.* _____
(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой _____ к.т.н., доц. Смелов В.В . _____
(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой _____

Минск 2019

Реферат

Пояснительная записка курсового проекта содержит 30 страница пояснительной записки, 32 иллюстрации, 3 источника литературы, 2 приложения.

C#, ADO.NET, SQL: 2016, БАЗА ДАННЫХ, SSRS.

Целью курсового проекта является разработка базы данных службы доставки и приложения для взаимодействия с базой данных службы доставки.

В первой главе проводится аналитический обзор литературы по тематике курсового проекта и содержит описание технологий, использованных во время выполнения проекта.

Вторая глава посвящена процессу проектирования системы.

В третьей главе описывается процесс разработки, принципы функционирования и назначение созданных компонент проекта.

В четвёртой главе описано руководство пользователя, позволяющее подробно понять интерфейс программного средства.

В заключении приведены результаты проделанной работы.

				<i>КП 00.00.ПЗ</i>						
	ФИО	Подпись	Дата							
Разраб.	<i>Хорошко Е.В.</i>			<i>Реферат</i>			Лит.	Лист	Листов	
Провер.	<i>Бондарчик Е.Н.</i>								1	1
Н. контр.	<i>Бондарчик Е.Н.</i>						<i>БГТУ 71172553, 2019</i>			
Утверд.	<i>Бондарчик Е.Н.</i>									

Содержание

Реферат	4
Введение	6
1 Обзор технических решений и литературных источников	7
1.1 Обзор аналогов	7
1.2 Обзор средств разработки	10
1.2.1 Описание технологии SQL Server Reporting Services	10
1.3 Вывод по разделу	11
2 Проектирование	12
2.1 Модель базы данных (подраздел)	12
2.2 Пользователи	13
2.3 Структура приложения	14
2.4 Вывод по разделу	14
3 Разработка программного средства	15
3.1 Таблицы (подраздел)	15
3.2 Хранимые процедуры (подраздел)	15
3.3 Работа с базой данных	17
3.4 Реализация технологии SQL Server Reporting Services	18
3.5 Вывод по разделу	20
4 Руководство пользователя	21
4.1 Обзор возможностей пользователя	21
4.2 Тестовый пример планирования нагрузки	27
4.3 Вывод по разделу	28
Заключение	29
Список использованной литературы	30
Приложение А	31
Приложение Б	32

				<i>КП 00.00.ПЗ</i>						
	ФИО	Подпись	Дата							
Разраб.	<i>Хорошко Е.В.</i>			<i>Содержание</i>			Лит.	Лист	Листов	
Провер.	<i>Бондарчик Е.Н.</i>								1	1
							<i>БГТУ 71172553, 2019</i>			
Н. контр.	<i>. Бондарчик Е.Н.</i>									
Утверд.	<i>Бондарчик Е.Н.</i>									

Введение

Развитие современных информационных технологий привнесло ощутимые изменения в работу служб доставки, сделав их более доступными и удобными в использовании реализуя возможности заказа онлайн.

В настоящее время существует множество современных компьютерных приложений, которые предоставляют возможность с помощью удобного интерфейса работать с данными, относящимися непосредственно к службе доставки, как со стороны пользователя (добавление заказа, вход в кабинет, отслеживание состояния заказа) так и со стороны администратора (отслеживание статуса всех заказов, модерация, управление товарами).

Спрос на данные программы постоянен, ведь не у каждого человека есть возможность забрать товары самовывозом или из другой страны и потому ежедневно происходят тысячи запросов на доставку товаров в разные точки нашей планеты. Это делает службы доставки востребованными на рынке.

С учетом имеющихся факторов, я выбрал в качестве темы своего курсового проекта разработку программы-службы доставки, функционал которой позволяет пользователям заказывать товары по определённому адресу, варьировать способ оплаты, просматривать заказы за определённый период в виде отчёта, а администраторам контролировать учёт товаров, заказов.

В основе разработки курсового проекта лежит удобство использования и изменения базы данных, наглядность информации, простота и интуитивная понятность интерфейса, которая, является одним из важнейших элементов для любого программного продукта.

В записке представлена информация после аналитического разбора литературы по теме данного курсового проекта с целью формирования требований к программному продукту, определение списка оптимальных используемых технологий. Произведен анализ существующих аналогов продукта, с целью оптимизации проекта, выявления достоинств и недостатков. Составлено руководство пользователя, способствующее более быстрому освоению работников и пользователей в среде нового программного продукта. Все аспекты программного обеспечения будут протестированы на корректность работы выполнения их функций, так же работа программы будет проанализирована при работе с большим объемом данных.

				КП 00.00.ПЗ							
	ФИО	Подпись	Дата								
Разраб.	Хорошко Е.В.			Введение				Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.									1	1
Н. контр.	. Бондарчик Е.Н.							БГТУ 71172553, 2019			
Утверд.	Бондарчик Е.Н.										

1 Обзор технических решений и литературных источников

1.1 Обзор аналогов

На сегодняшний день службы доставки становятся все популярнее, они увеличиваются в объеме, становятся динамичнее. Контроль над процессами в этой области требует использования надежных технологий и программного обеспечения. Это становится одной из главных причин все более активного внедрения компаниями и интернет-магазинами инструментария, позволяющего более наглядно вести учет операций.

На данный момент сфера заполнена различными сайтами и программными продуктами, большая часть которых имеет схожий интерфейс и функционал, хотя некоторые из них и выделяются на фоне других.

Я представляю 2 наиболее популярных решения:

Frontpad (рисунок 1.1)

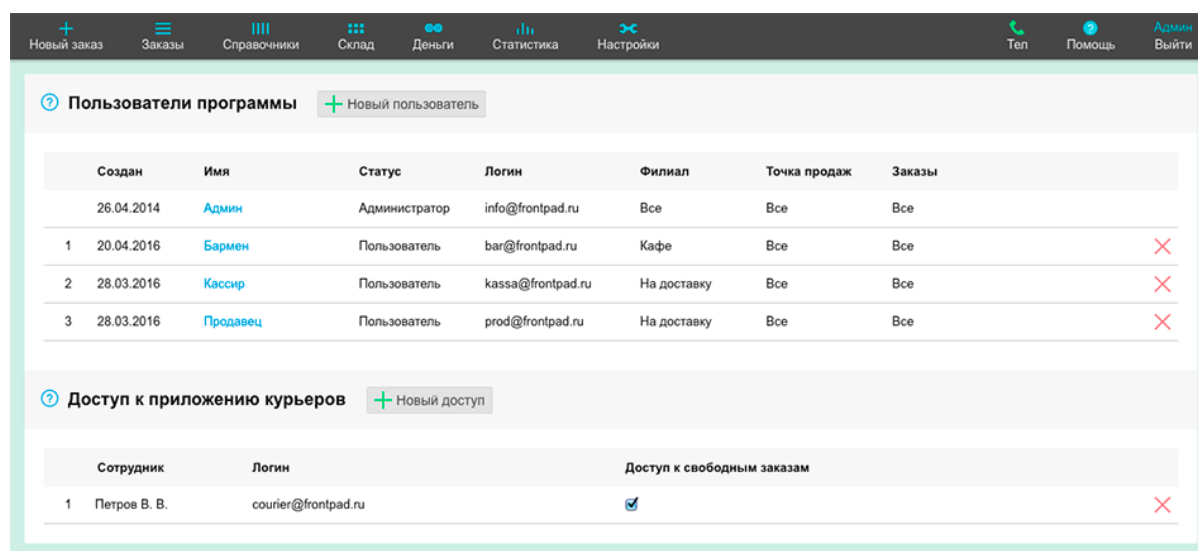


Рисунок 1.1 – Интерфейс Frontpad

Frontpad: программа для автоматизации службы доставки, розницы, кафе или бара. Удобный интерфейс для создания заказов и ведения складского учета. Система автоматического распределения заказов по зонам доставки и курьерам.

К функциям данной программы относится (версия «Профи»):

- Оформление заказов
- Каталог товаров
- Складской учёт

				КП 00.01.ПЗ			
	ФИО	Подпись	Дата				
Разраб.	Хорошко Е.В.			1 Обзор технических решений и литературных источников		Лит.	Лист
Провер.	Бондарчик Е.Н.						Листов
							1
Н. контр.	Бондарчик Е.Н.					БГТУ 71172553, 2019	
Утверд.	Бондарчик Е.Н.						

- Отчёты, статистика
- Клиенты
- Автописание
- API

К достоинствам можно отнести оформление приложения и его многофункциональность. К недостаткам относится необходимость в дублировании данных при ошибках, а некоторые данные, которые должны заполняться автоматически приходится вносить вручную.


Мастер Доставки (рисунок 1.2)


Наименование товара	Количество	Цена	Сумма
Apple iPhone 5S White 16Gb	1	29990,00	29990,00

Отчеты

- Доходы
- Клиент
- Продавец
- Курьер
- Товары

Артикул: GPS-00001
Тариф: До 20 кг
Станция метро: Варшавская




МАСТЕР ДОСТАВКИ

Логин: admin
Пароль:

Размер: 11*7,8*2 Вес: 198 г Район города: Левобережье Курьер: Печкин Павел Стоимость доставки: 500,00 руб. ИТОГО: 18 500,00 руб.

Артикул	Описание	Количество	Цена за единицу	Всего
GPS-00001	GPS навигатор JJ-CONNECT Auto Navigator 2000	2	9000	18000

Адрес	Курьер	Доставка, руб.	Сумма заказа, руб.
ул. Солнечная, 20, кв. 50	Печкин Павел	300	30290

Документы

- Заказ-накладная F8
- Путевой лист F9
- Товарный чек F10

Домашняя страница программного продукта: www.masterdostavki.su

Рисунок 1.2 – Интерфейс Мастер Доставки

Многопользовательская программа доставки позволяет оперативно вести учет доставляемых продуктов или товаров. В программе создается структурированный прейскурант, ведется база клиентов. Заказ на доставку включает в себя: перечень продуктов или товаров, контактные данные заказчика, стоимость услуги доставки и имя курьера. По каждому заказу формируется печатная форма заказа-накладной. По совокупности заказов можно сформировать путевой лист для курьера. Программа по автоматизации доставки снабжена возможностью выгружать различные статистические отчеты за период: по продукции, по клиентам, по менеджерам, по курьерам, по доходам и так далее.

Преимущества программы доставки:

- Высокая скорость обработки данных и выполнения операций;
- Простота настройки системы;
- Оперативная высококвалифицированная техническая поддержка;
- Отсутствие специальных требований к аппаратной части компьютеров;
- Система автоматизирует рутинные процессы и вычисления;

- Отчётность
 - Высокий уровень безопасности хранения конфиденциальных данных;
 - Возможность доработки программы под специфику бизнеса;
 - Многоступенчатый фильтр заказов согласно заданным условиям отбора;
- Недостатки:
- Перегруженность интерфейса;
 - Приложение используется для учёта и статистики, но не для организации самой доставки;
- ArchiDelivery (рисунок 1.3)

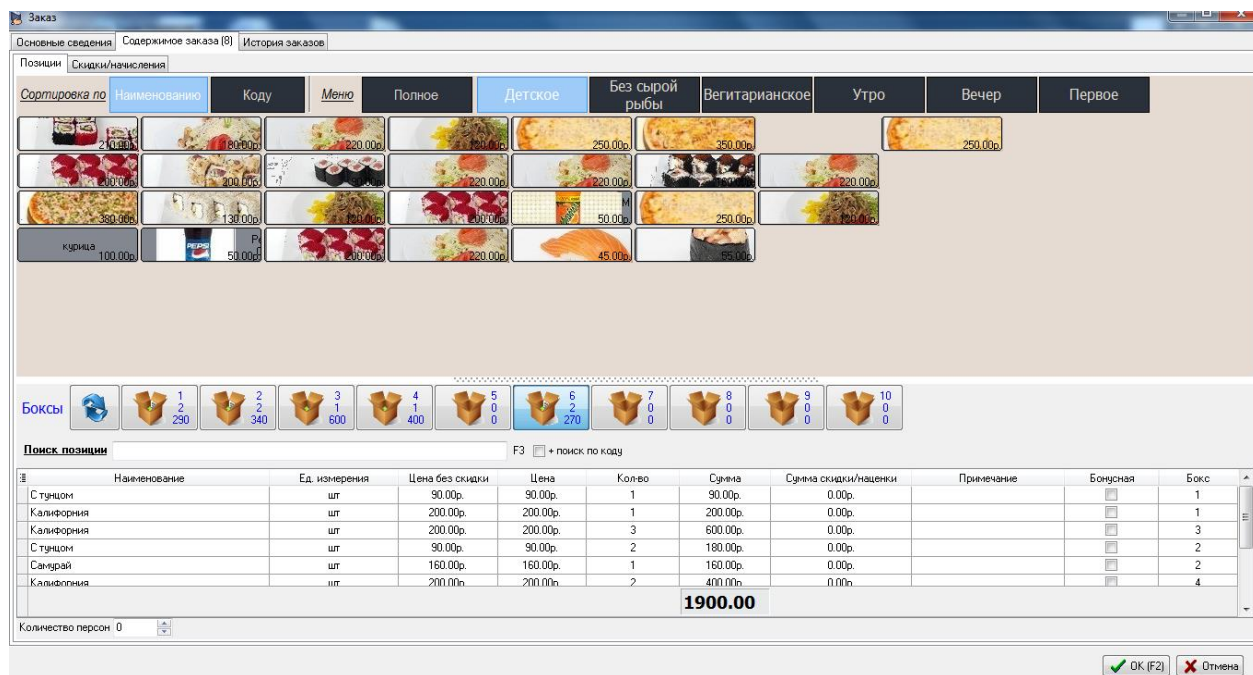


Рисунок 1.3 – Интерфейс ArchiDelivery

Программа для автоматизации предприятий общественного питания и служб доставки. Позволяет автоматизировать ведение складского учета, клиентской базы, принятие и администрирование заказов, ведение логистики, учет заработной платы сотрудников.

Преимущества программы:

- Управление заказами;
- Продуктовый каталог;
- Системы лояльности;
- Мониторинг эффективности персонала;
- Отчёты;
- Интеграция с почтой;
- Email-рассылки;
- Хранилище файлов;
- Экспорт/импорт данных;
- API для интеграции;
- Веб-формы;

Недостатки:

- Отсутствие возможности установки ПО, т.к. программа является облачным решением;

- Недоступна история взаимодействия с клиентами;

Рассмотрев приведённые выше примеры и учитывая их функционал и особенности, можно прийти к выводу о некоторых сходствах программных средств:

- Управление заказами

- Формирование отчётности

- Поиск по заданным критериям

1.2 Обзор средств разработки

1.2.1 Описание технологии SQL Server Reporting Services

SQL Server Службы Reporting Services предлагают несколько программных интерфейсов, которые могут быть использованы в пользовательских приложениях. Можно использовать существующие функции и возможности служб Службы Reporting Services для построения пользовательских средств создания отчетов и средств управления на сайтах и в приложениях Windows или расширять платформу служб Службы Reporting Services .

Расширение платформы служб Службы Reporting Services включает создание новых компонентов и ресурсов, которые могут использоваться для доступа к данным, доставки отчетов и других задач. Эти компоненты и ресурсы можно предлагать компаниям, использующим в своей работе службы Службы Reporting Services .

Службы Службы Reporting Services представляют собой открытую и расширяемую платформу создания отчетов, которая позволяет предоставить разработчикам всеобъемлющий набор API для разработки решений.

SSRS реализован как web-служба и ее администрирование, а также управление отчетами производится через веб интерфейс. Стандартный доступ к отчетам предоставляется пользователям также через веб интерфейс. SSRS предоставляет возможность интегрировать разработанные отчеты в сторонние приложения, т.е. имеется некий API функционал. Например, SSRS можно интегрировать с SharePoint.

Для того чтобы сервер отчетов имел возможность хранить опубликованные отчеты, модели отчетов и иерархию папок требуется база данных SQL Server.

Все отчеты, созданные с помощью служб Reporting Services можно экспортировать в разные форматы (*с сохранением визуального оформления*), например:

- Excel;

- Word;

- PDF;

- CSV;

- XML;

- TIFF;

- MHTML (Web Archive).

Предусмотрены три способа интеграции служб Службы Reporting Services в пользовательские приложения:

- веб-служба сервера отчетов, которую также называют SOAP API Службы Reporting Services,
- элементы управления средства просмотра отчетов для Microsoft Visual Studio
- средства доступа по URL-адресу.

В каждом из этих вариантов реализуется отдельный подход к интеграции служб Службы Reporting Services в приложения.

Веб-служба сервера отчетов является основным интерфейсом разработки приложений служб Службы Reporting Services. Эта веб-служба предоставляет все необходимые методы для интеграции служб Службы Reporting Services в приложения, независимо от того, ведется ли разработка кода для управления каталогом отчетов или для подготовки отчетов в поддерживаемом формате. Примером такого приложения может служить веб-портал, который предоставляется вместе с Службы Reporting Services и который использует веб-службу для управления базой данных сервера отчетов.

Элементы управления средства просмотра отчетов, включенные в состав Visual Studio, используются для интеграции средств просмотра отчетов в приложения. Имеется два элемента управления: один для приложений на основе Windows Forms, а другой – для приложений Web Forms. Каждый элемент управления обеспечивает возможность просмотра отчетов, развернутых на сервере отчетов, а также возможность отображения отчетов, существующих в среде, где сервер отчетов пока еще не установлен.

Доступ по URL-адресу представляет собой еще один метод интеграции средств просмотра отчетов в приложения; он используется в случаях, когда применение элементов управления средства просмотра отчетов не представляется возможным. Метод доступа по URL-адресу также позволяет отправлять пользователям ссылки на отчеты по электронной почте.

Мною был использован метод доступа по URL-адресу. Для его реализации я использовал Report Builder. Построитель отчетов является автономным приложением и устанавливается на компьютере пользователем или администратором.

Обычно администратор устанавливает и настраивает службы Службы Reporting Services, предоставляет разрешение на скачивание построитель отчетов с веб-портала и управляет папками и разрешениями для отчетов, элементов отчетов и общих наборов данных, сохраняемых на сервере отчетов.

Пользователь Report Builder может соединяться с сервером Reporting Services, открывать хранящиеся там источники данных, отчеты и модели, работать с ними и затем снова сохранять их на сервере. Report Builder выполнен как самостоятельное приложение, которое можно бесплатно установить на компьютер клиента.

1.3 Вывод по разделу

В разделе были описаны аналоги программного средства, были рассмотрены их преимущества и недостатки, а также описана технология, используемая в приложении.

2 Проектирование

2.1 Модель базы данных (подраздел)

Для создания, хранения и управления данными, была использованная СУБД MS SQL Server. Диаграмма модели базы данных представлена в Приложении А. База данных в полной мере описывает необходимые таблицы для решения поставленной задачи данного курсового проекта.

Таблица Product содержит информацию о товарах:

- id_product – первичный ключ, идентификатор товара;
- name_product – наименование товара;
- cost – стоимость товара;
- Count – количество товара в наличии;

Таблица Users содержит информацию о пользователях:

- id_users – идентификатор пользователя;
- Username – первичный ключ, логин пользователя;
- Password – пароль пользователя;
- Role – роль пользователя;
- First_Name – Имя пользователя;
- Last_Name – Фамилия пользователя;
- Phone_Number – Номер телефона пользователя;
- Address – адрес доставки ;
- [E-mail] – электронная почта пользователя;

Таблица Card содержит информацию о банковских картах пользователя:

- Username – внешний ключ, логин пользователя;
- card_number – номер карты;
- owner – владелец карты;
- ММYY – Месяц/год до которого действительна карта;
- CVC – код с обратной стороны карты;

Таблица Address содержит информацию об адресах доставки пользователю:

- Username – внешний ключ, логин пользователя;
- City – город доставки;
- Street – улица доставки;
- NumberHouse – номер дома/квартиры доставки;

Таблица Orders содержит информацию о заказах пользователей:

- id_order – идентификатор заказа;
- order_number – номер заказа;
- name_product – наименование товара заказа;
- cost – стоимость товара заказа;
- count – количество заказанного товара;

				КП 00.02.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Хорошко Е.В.			2 Проектирование						
Провер.	Бондарчик Е.Н.									
Н. контр.	Бондарчик Е.Н.									
Утверд.	Бондарчик Е.Н.									
				Лит.		Лист	Листов			
						1	3			
				БГТУ 71172553, 2019						

- TCost – общая стоимость заказанного товара;
- address – адрес доставки заказа;
- name_of_delivery – способ доставки;
- name_of_payments – способ оплаты;
- card_number – номер карты пользователя;
- order_date – дата заказа;
- delivery_date – дата выполнения заказа;
- Username – внешний ключ, логин пользователя;
- Status – статус выполнения заказа;

Таблица Basket содержит информацию, что хочет заказать пользователь:

- id_product – внешний ключ, идентификатор товара;
- name_product – наименование товара добавленного в корзину;
- cost – стоимость одного товара добавленного в корзину;
- count – количество заказанных товаров;
- order_date – дата заказа;
- Username – внешний ключ, логин пользователя;

Таблица Delivery_method содержит информацию о способах доставки:

- id_delivery – идентификатор способа доставки;
- name_of_delivery – способ доставки;

Таблица Payments_method содержит информацию о способах оплаты:

- id_payments – идентификатор способа оплаты;
- name_of_payments – способ оплаты;

2.2 Пользователи

Пользователь базы данных – это лицо, которое имеет доступ к БД и пользуется услугами информационной системы для получения информации. При проектировании базы данных понадобилось 2 пользователя (user, admin). У каждого пользователя есть определенные права, предназначенные для выполнения соответствующих ему требований. Их права представлены на рисунке 2.1.

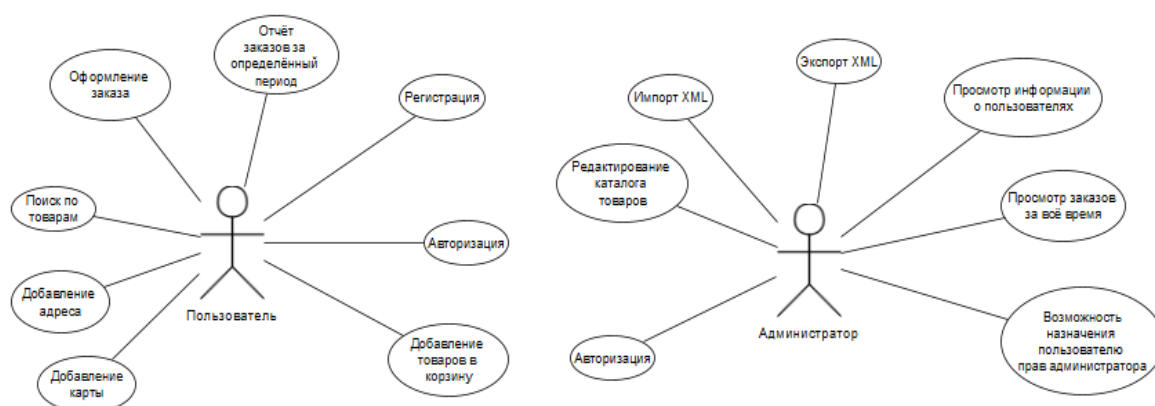


Рисунок 2.1 – Диаграмма использования

Пользователь user наделен правами, имитирующего пользователя, который может оформлять заказ, просматривать отчёт заказов за определённый период, регистрироваться и входить в систему, добавлять товары в корзину, производить поиск по товарам, добавлять адрес и карту.

Пользователь admin наделен правами, имитирующими администратора платформы, который может добавлять и удалять новые товары, производить импорт и экспорт товаров из или в XML, просматривать информацию о пользователях и в случае необходимости назначать или лишать их прав администратора, просматривать заказы за всё время.

2.3 Структура приложения

На диаграмме, представленной в графической части 1, отображена диаграмма классов. Диаграмма классов показывает взаимосвязи между классами, описывает атрибуты и методы каждого класса. Приложение представлено различными типами классов.

Классы, представленные на диаграмме, являются обработчиками запросов и после обработки запроса вызывают нужные команды. Команды вызывают нужные методы из хранимых процедур. Хранимые процедуры позволяют нам взаимодействовать с базой данных. На диаграмме, представленной в графической части 2, отображена структурная схема проекта.

Приложение представляет собой сервер, который состоит из обработчика запросов, подсистемы взаимодействия с клиентами, подсистемы взаимодействия с базой данных, подсистемы внутренних сервисов и базы данных.

Приложение является многостраничным приложением WPF. Контроллер вызывает нужную команду из запроса и возвращает нужную страницу после добавления данных в сессию. Схема приложения представлена на рисунке 2.2.

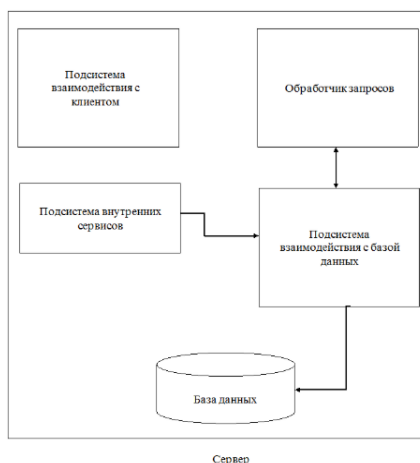


Рисунок 2.2 – Схема приложения

2.4 Вывод по разделу

В разделе была рассмотрена структура приложения, варианты использования приложения пользователями, описание таблиц и их полей, СУБД используемая в курсовом проекте.

3 Разработка программного средства

База данных данного курсового проекта содержит следующие объекты: таблицы и хранимые процедуры. Их подробное описание представлено ниже.

3.1 Таблицы (подраздел)

База данных площадки электронных услуг содержит 8 таблиц, которые описаны в главе 2. SQL-скрипты для создания таблиц находятся в приложении А.

3.2 Хранимые процедуры (подраздел)

Вся работа приложения с базой данных осуществляется через хранимые процедуры и представление, что позволяет инкапсулировать структуру таблиц от пользователя. Полный список процедур и их реализацию можно найти в приложении А. Пример хранимой процедуры, которая осуществляет вставку представлена на рисунке 3.1.

```
CREATE PROCEDURE [dbo].[sp_RegUsers]
@Username nvarchar (50),
@Password int,
@Role nvarchar(50),
@First_Name nvarchar(50),
@Last_Name nvarchar(50),
@Phone_Number nvarchar(50),
@Address nvarchar(50),
@email nvarchar(50)
AS
Insert INTO [dbo].[Users](Username, Password, Role,
First_Name, Last_Name, Phone_Number,
Address, [E-mail])
Values(@Username, @Password, @Role,
@First_Name, @Last_Name,
@Phone_Number, @Address, @Email)
GO
```

Рисунок 3.1 – Хранимая процедура для вставки в таблицу Users

Для экспорта и импорта использовалась технология XML (eXtensible Markup Language) – набор средств и возможностей СУБД и БД в MS SQL Server, ориентированный на работу с форматом XML как по части преобразования, так и хранения данных. Импорт и экспорт производится в формат .xml согласно заданию курсового проекта. Реализация функции для импорта таблицы Products в xml представлена на рисунке 3.2

				КП 00.03.ПЗ							
	ФИО	Подпись	Дата								
Разраб.	Хорошко Е.В.			3 Разработка программного средства				Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.									1	6
								БГТУ 71172553, 2019			
Н. контр.	Бондарчик Е.Н.										
Утверд.	Бондарчик Е.Н.										

```

public void Serealization_clk(object sender, RoutedEventArgs e)
{
    XmlSerializer xml = new XmlSerializer(typeof(DataTable));
    if (File.Exists("E:\\Xmlfile.xml"))
    {
        File.Delete("E:\\Xmlfile.xml");
    }
    using (FileStream fs = new FileStream("E:\\Xmlfile.xml", FileMode.OpenOrCreate))
    {
        SqlConnection sqlCon = new SqlConnection(ConnectionString);
        try
        {
            if (sqlCon.State == ConnectionState.Closed)
                sqlCon.Open();
            string sqlExpression = "sp_Books";
            SqlCommand cmd = new SqlCommand(sqlExpression, sqlCon);
            adapter = new SqlDataAdapter(cmd);

            adapter.Fill(dt);
            xml.Serialize(fs, dt);
            MessageBox.Show("Elements are serialized!");
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        finally
        {
            adapter.Dispose();
            sqlCon.Close();
            sqlCon.Dispose();
        }
    }
}

```

Рисунок 3.2 – Функция импорта в XML

Для импорта используется функция xml.Serialize.

Реализация хранимой процедуры и функции для экспорта таблицы Placement из xml представлена на рисунках 3.3 – 3.4.

```

CREATE PROCEDURE [dbo].[sp_XmlDes]
    @id_product int,
    @name_product nvarchar(50),
    @cost int,
    @count int
AS
if not exists(select * from Products where id_product=@id_product)
BEGIN
    INSERT INTO Products(id_product, name_product, cost, Count) values (@id_product, @name_product,
        @cost, @count)
END
else
BEGIN
    if exists(select * from Products where id_product=@id_product
        and name_product!=@name_product)
        update Products set name_product=@name_product, cost=@cost, Count=@count where id_product=@id_product
    END;
GO

```

Рисунок 3.3 – Хранимая процедура для экспорта из XML

```
//Десериализация
private void Deser(object sender, RoutedEventArgs e)
{
    string path = @"E:\XmlFile.xml";
    using (StreamReader fs = new StreamReader(path))
    {
        DataTable Des = new DataTable();
        Des.Clear();
        XmlSerializer xml = new XmlSerializer(typeof(DataTable));
        Des = (DataTable)xml.Deserialize(fs);
        dataGrid2.ItemsSource = Des.DefaultView;

        IList rows = dataGrid2.Items;
        SqlConnection conn = new SqlConnection(ConnectionString);
        conn.Open();
        foreach (DataRow dr in Des.Rows)
        {
            var cells = dr.ItemArray;
            int ID = Convert.ToInt32(cells[0].ToString());
            string Name = cells[1].ToString();
            int Page = Convert.ToInt32(cells[2].ToString());
            int Cost = Convert.ToInt32(cells[3].ToString());
            int Count = Convert.ToInt32(cells[4].ToString());
            string sqlExp = "sp_XmlDes";
            SqlCommand comm = new SqlCommand(sqlExp, conn);
            comm.CommandType = CommandType.StoredProcedure;
            comm.Parameters.AddWithValue("@id_product", ID);
            comm.Parameters.AddWithValue("@name_product", Name);
            comm.Parameters.AddWithValue("@pages", Page);
            comm.Parameters.AddWithValue("@cost", Cost);
            comm.Parameters.AddWithValue("@count", Count);
            SqlDataAdapter adapter1 = new SqlDataAdapter(comm);
            adapter1.Fill(dt);
        }
    }
}
```

Рисунок 3.4 – Функция для экспорта из XML

3.3 Работа с базой данных

Вызов хранимых процедур осуществляется, с помощью SQL Command, способом, показанным на рисунке 3.5.

```
private void Check_Status_Click(object sender, RoutedEventArgs e)
{
    SqlConnection sqlCon = new SqlConnection(ConnectionString);
    try
    {
        if (sqlCon.State == ConnectionState.Closed)
            sqlCon.Open();
        string sqlExpression = "sp_ConfirmStatus";
        SqlCommand cmd = new SqlCommand(sqlExpression, sqlCon);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Username", user);
        cmd.ExecuteNonQuery();
        FillListOrder();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Рисунок 3.5 – Вызов хранимой процедуры в приложении

Результат хранимой процедуры обрабатывается и может быть помещен в таблицу. Обработка показана на рисунке 3.6.

```

private void FillListOrder()
{
    SqlConnection sqlCon = new SqlConnection(ConnectionString);
    try
    {
        if (sqlCon.State == ConnectionState.Closed)
            sqlCon.Open();
        string sqlExpression = "sp_ViewOrders";
        SqlCommand cmd = new SqlCommand(sqlExpression, sqlCon);
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Parameters.AddWithValue("@Username", user);
        adapter = new SqlDataAdapter(cmd);

        DataTable dt = new DataTable("Orders");
        adapter.Fill(dt);
        DataOrder.ItemsSource = dt.DefaultView;
    }
}

```

Рисунок 3.6 – Обработка результатов хранимой процедуры в приложении

Примеры кода приложения приведены в приложении 2.

3.4 Реализация технологии SQL Server Reporting Services

Для реализации технологии SQL Server Reporting Services необходима установка и настройка SSRS, а также программа Report Builder для построения формы отчёта.

Мною был выбран метод реализации работы SSRS с помощью доступа по URL-адресу (рисунок 3.7).

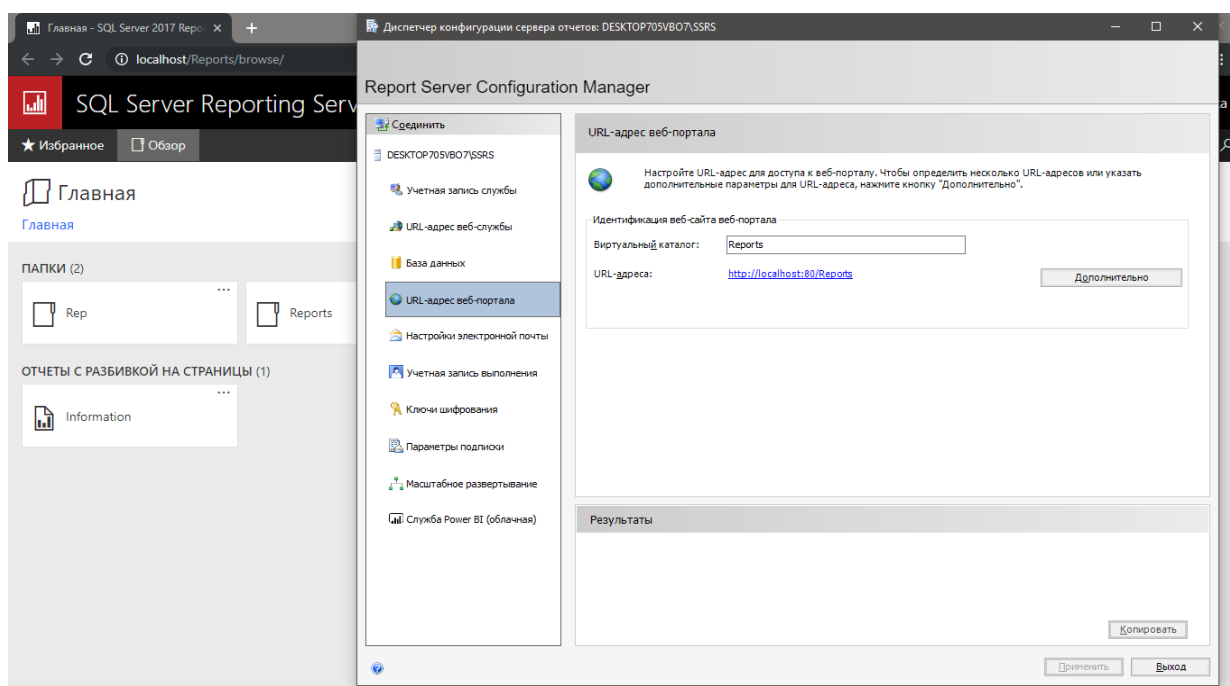


Рисунок 3.7 – Метод доступа по URL-адресу

Доступ по URL-адресу позволяет получать доступ к отчетам с помощью URL-адреса сервера отчетов. Запрос по URL-адресу позволяет получить доступ к выбранному серверу отчетов, а также к отчетам, ресурсам и другим элементам в базе данных сервера отчетов.

Чтобы создать отчёт мною был использован Мастер таблицы или матрицы в Report Builder, где необходимо было указать строку соединения с нашей базой данных и необходимую хранимую процедуру для формирования запроса.

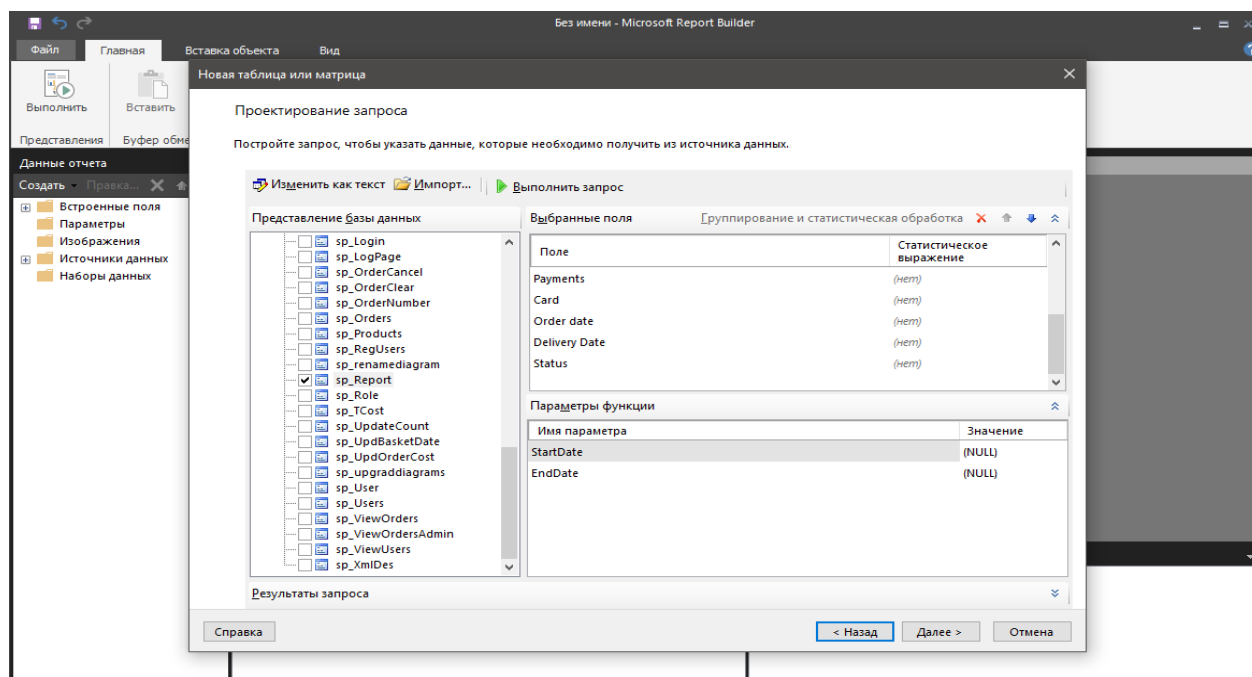


Рисунок 3.8 – Выбор хранимой процедуры при создании отчёта

Хранимая процедура для данного отчёта описывает выборку заказов в определённом временном диапазоне относительно даты заказа. Данная хранимая процедура представлена на рисунке 3.9.

```
CREATE PROCEDURE [dbo].[sp_Report]
@StartDate date,
@EndDate date
AS
Select username AS [User], order_number AS [Order Number], name_product AS [Product],
count AS [Count],
TCost AS [Total Cost], address AS [Address], name_of_delivery AS [Delivery],
name_of_payments AS [Payments], card_number AS [Card],
convert(varchar(10),order_date,101) as [Order date],
convert(varchar(10), delivery_date,101) as [Delivery Date], Status AS [Status]
From Orders where order_date between @StartDate and @EndDate
GO
```

Рисунок 3.9 – Хранимая процедура для формирования отчёта

Чтобы настроить SSRS необходимо использовать Диспетчер конфигурации сервера отчётов.

Основные этапы настройки:

- Указать учётную запись для запуска службы сервера отчётов ;
- Настроить URL-адрес. Для этого нужно указать виртуальный каталог и один или несколько URL-адресов;
- Создать или подключиться к существующей базе данных сервера отчётов;
- Настроить URL-адрес веб-портала для доступа в веб-порталу.

Результатом работы системы и приложения является отчёт, представленный на рисунке 3.11

User	Product	Address	Delivery	Payments	Card	Order date	Delivery Date	Status	Order Number	Count	Total Cost
Shuka	dsf	Klimovich, Pushkinskaya, 38	Pickup	Card	45315345	11/30/2019	01/12/2019	Inactive	1770	30	438678
		Minsk, Belorusskaya, 19	Pickup	Card	45315345	12/02/2019	07/12/2019	Inactive	41	10	4320
Shu	Klimovich, Pushkinskaya, 38	Minsk, Belorusskaya, 19	Pickup	Card	45315345	11/30/2019	01/12/2019	Inactive	1770	30	438678
		Minsk, Belorusskaya, 19	Post	Card	45315345	11/26/2019	11/27/2019	Inactive	7	200	223200
shuk	Klimovich, Pushkinskaya, 38	Minsk, Belorusskaya, 19	Pickup	Card	45315345	11/30/2019	01/12/2019	Inactive	1770	0	438678
		Minsk, Belorusskaya, 19	Post	Card	45315345	11/26/2019	11/27/2019	Inactive	7	100	223200
SL	Klimovich	Pickuo	Card	45315345	11/30/2019	01/12/2019	Inactive	1770	6	438678	

Рисунок 3.11 – Отчёт о заказах за определённый период

3.5 Вывод по разделу

В разделе была рассмотрена реализация технологии SQL Server Reporting Services, работа с базой данных из приложения, реализация функций импорта таблицы из XML и экспорта таблицы в XML, описание объектов базы данных.

4 Руководство пользователя

4.1 Обзор возможностей пользователя

Приложение взаимодействует с базой данных. Оно было реализовано с помощью языка С#.

Начальной страницей приложения является страница авторизации, представленная на рисунке 4.1.

Рисунок 4.1 – Страница авторизации пользователя

Если пользователь не имеет аккаунта, то ему необходимо зарегистрироваться в системе. Для этого необходимо нажать на кнопку «Registration». После нажатия происходит переход на страницу, представленную на рисунке 4.2.

Рисунок 4.2 – Страница регистрации поля

				КП 00.04.ПЗ						
	ФИО	Подпись	Дата							
Разраб.	Хорошко Е.В.			4 Руководство пользователя			Лит.	Лист	Листов	
Провер.	Бондарчик Е.Н.							1	8	
							БГТУ 71172553, 2019			
Н. контр.	. Бондарчик Е.Н.									
Утверд.	Бондарчик Е.Н.									

После ввода данных нового пользователя нажимаем кнопку «Registration». Затем произойдет переход на страницу авторизации и вводим данные для входа в систему. После входа появится страница, представленная на рисунке 4.3. Тут мы можем выбрать на какую страницу нам перейти.

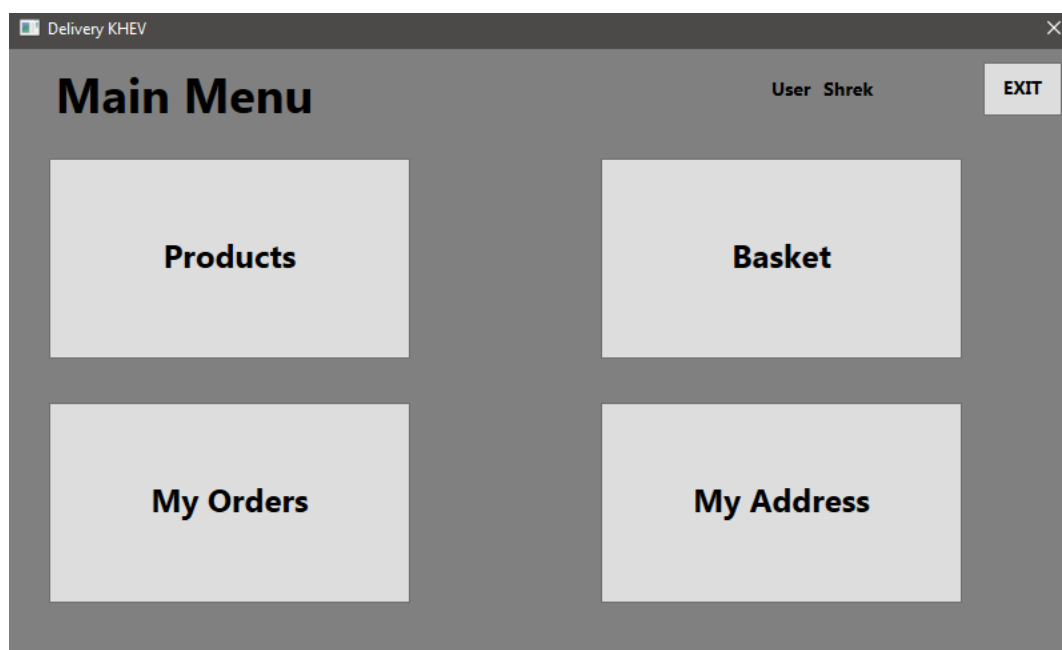


Рисунок 4.3 – Страница главного меню

При нажатии на кнопку «Products» произойдет переход на форму, представленную на рисунке 4.4. Тут мы можем найти нужный товар по стоимости или названию, добавить его в корзину.

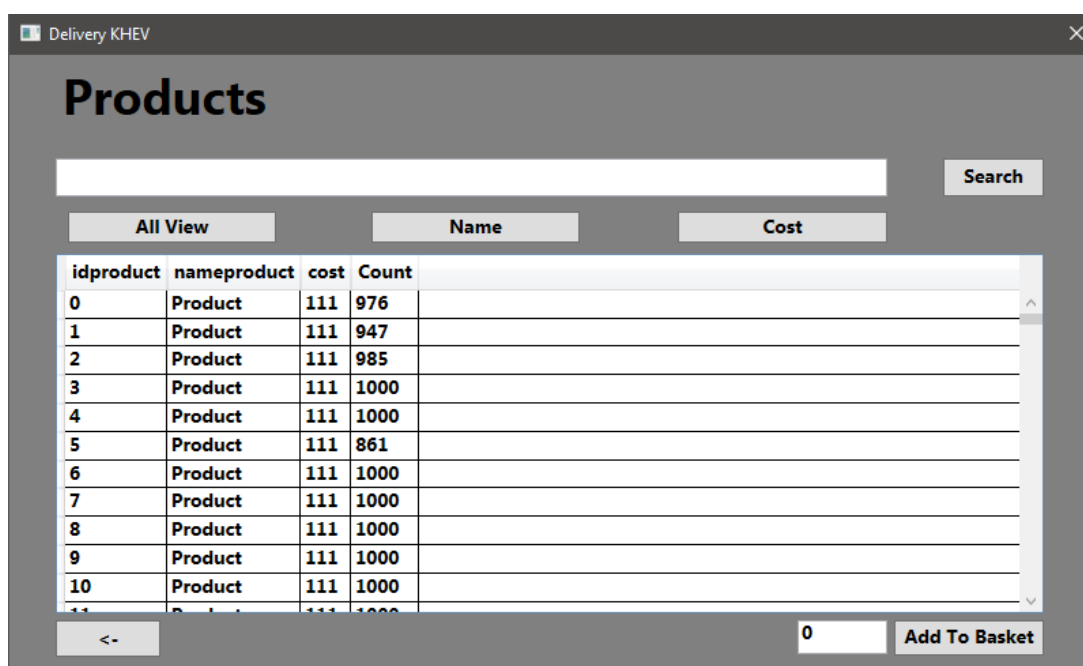
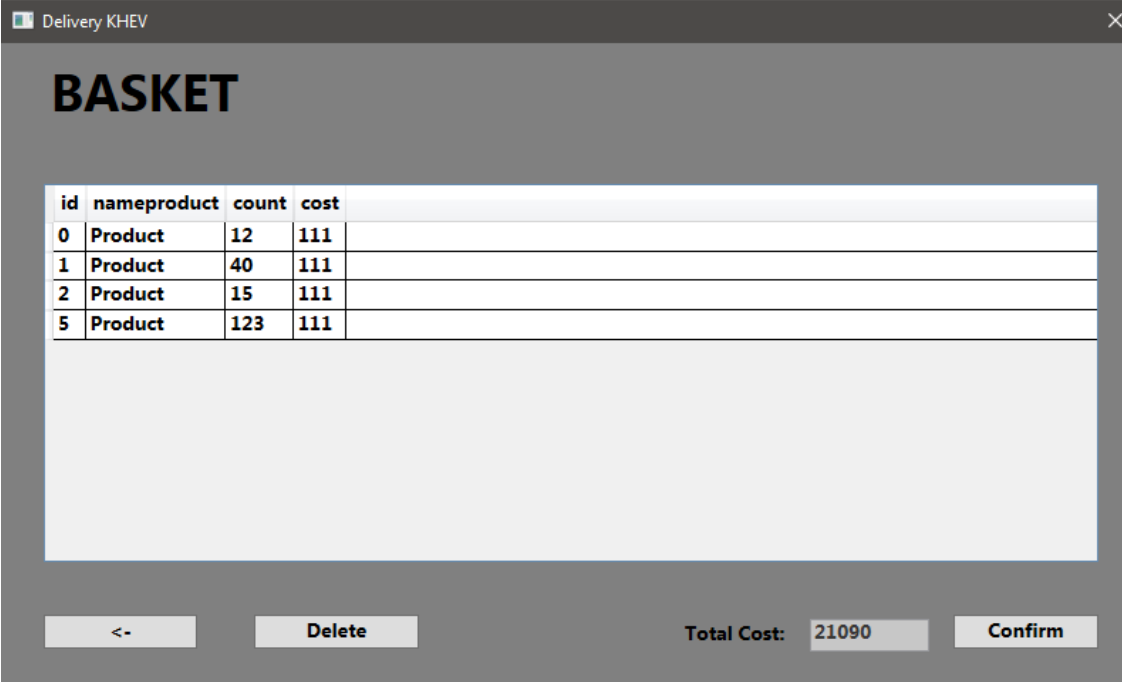


Рисунок 4.4 – Страница каталога продуктов

При нажатии на кнопку «Basket» произойдет переход на форму, где мы можем заказывать или удалять товары, добавленные в корзину (рисунок 4.5).

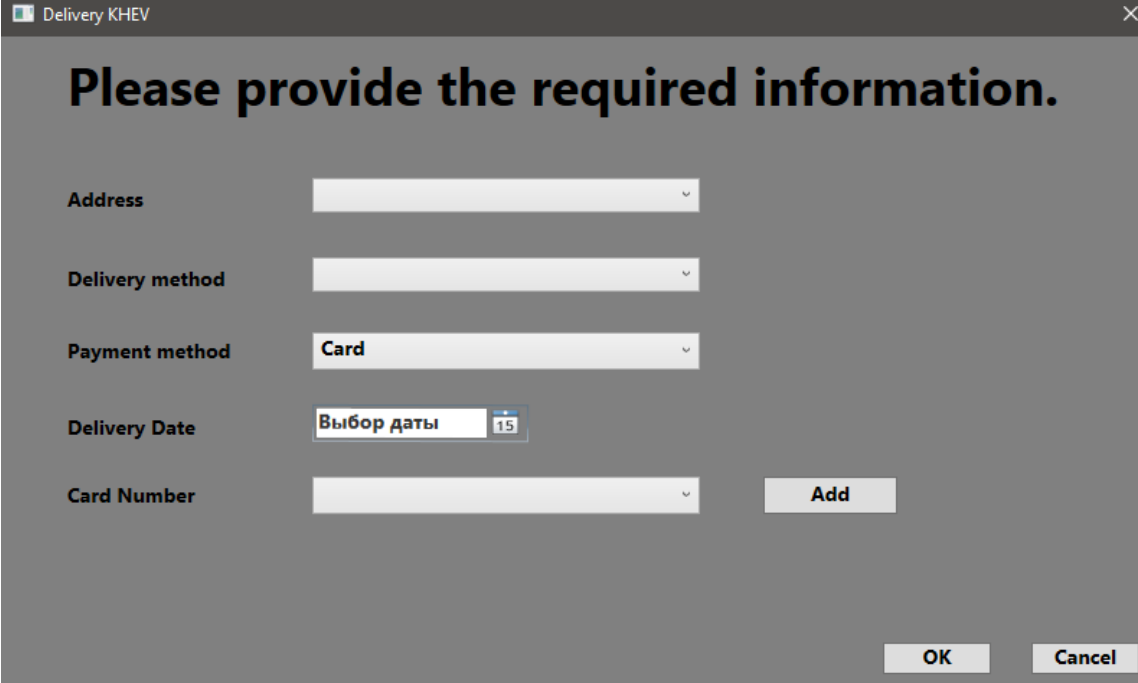


id	nameproduct	count	cost
0	Product	12	111
1	Product	40	111
2	Product	15	111
5	Product	123	111

<- Delete Total Cost: 21090 Confirm

Рисунок 4.5 – Страница корзины заказов

При нажатии кнопки «Confirm» на странице «Basket» происходит переход на страницу для ввода необходимой для формирования заказа и его доставки. Она представлена на странице 4.6. Также с этой страницы возможен переход на страницу «My cards» и при подтверждении заказа на страницу «Orders».



Please provide the required information.

Address

Delivery method

Payment method Card

Delivery Date 15

Card Number Add

OK Cancel

Рисунок 4.6 – Страница для ввода информации необходимой для заказа

При нажатии кнопки «Add» происходит переход на страницу «My cards», представленной на рисунке 4.7. Тут пользователь может добавить банковскую карту для использования её при оплате заказов.

My Cards

Card Number

Owner

MMYY

CVC

Add **Reset** **Cancel**

Рисунок 4.7 – Страница для добавления банковских карт пользователя

При нажатии кнопки «My Address» происходит переход на страницу, представленную на рисунке 4.8. Здесь пользователь может добавлять и удалять адреса доставки.

My Address

City	Street	NumberHouse
Minsk	Sverdlova	13
Minsk	Belorusskaya	19/808A

City

Street

House / apartment number

Add **Delete**

Go To Menu **Reset**

Рисунок 4.8 – Страница адресов пользователя

При нажатии на кнопку «My Orders» происходит переход на форму, где мы можем просмотреть активные заказы. При нажатии на кнопку «Report» произойдёт переход по ссылке на сервер отчётов, для отображения отчёта заказов юзера за определённый период. При нажатии на кнопку «Confirm Order» мы можем подтвердить заказ. Страница представлена на рисунке 4.9.

MY ORDERS

Search

Confirm order Report

Order Number	Product	Count	Total Cost	Address	Delivery	Payments	Card	Ord
4182	Product	12	21090	Minsk, Sverdlova, 13	Courier	Card	134534651	12/1
4182	Product	40	21090	Minsk, Sverdlova, 13	Courier	Card	134534651	12/1
4182	Product	15	21090	Minsk, Sverdlova, 13	Courier	Card	134534651	12/1
4182	Product	123	21090	Minsk, Sverdlova, 13	Courier	Card	134534651	12/1

Рисунок 4.9 – Страница заказов пользователей

В случае авторизации под профилем администратора, мы получаем форму, представленную на рисунке 4.10. С возможностью перехода на 4 различные формы.

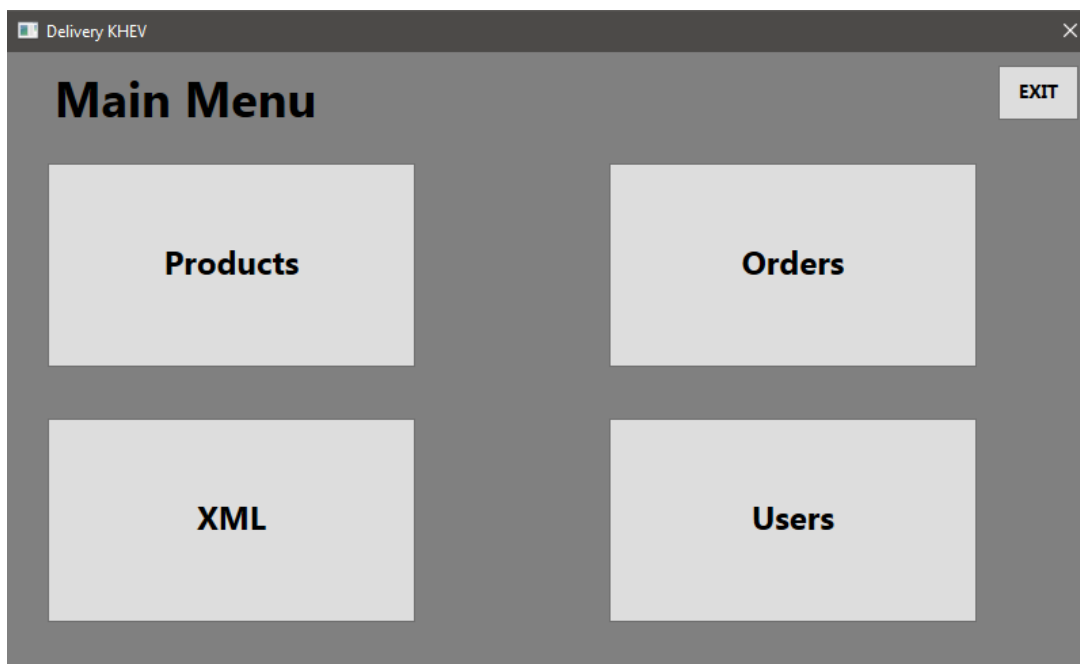


Рисунок 4.10 – Страница главного меню администратора

При нажатии на кнопку «Products» происходит переход на форму управления каталогом товаров. Она представлена на рисунке 4.11.

Products

Search

Products

ID:

Books Name

Cost:

Count

Add Delete

Update Reset

<-

idproduct	nameproduct	cost	Count
0	Product	111	976
1	Product	111	947
2	Product	111	985
3	Product	111	1000
4	Product	111	1000
5	Product	111	861
6	Product	111	1000
7	Product	111	1000
8	Product	111	1000
9	Product	111	1000
10	Product	111	1000
11	Product	111	1000
12	Product	111	1000

Рисунок 4.11 – Страница управления каталогом товаров

При нажатии на кнопку «Orders» происходит переход на форму содержащую все заказы. Она представлена на рисунке 4.12. На данной форме, есть функции просмотра активных или неактивных заказов, а также проверка на выполнение заказа.

ALL ORDERS

Search

All orders Inactive orders Active orders Check order status

Order Number	Product	Count	Total Cost	Address	Delivery	Payments	Card
7	Shu	200	223200	Minsk, Belorusskaya, 19	Post	Card	45315
7	shuk	100	223200	Minsk, Belorusskaya, 19	Post	Card	45315
590	dsf	10	146226	Klimovichi, Pushkinskaya, 38	Pickup	Card	45315
590	Shu	10	146226	Klimovichi, Pushkinskaya, 38	Pickup	Card	45315
590	shuk	0	146226	Klimovichi, Pushkinskaya, 38	Pickup	Card	45315
590	SL	2	146226	Klimovichi, Pushkinskaya, 38	Pickup	Card	45315
590	dsf	10	146226	Klimovichi, Pushkinskaya, 38	Pickup	Card	45315
590	Shu	10	146226	Klimovichi, Pushkinskaya, 38	Pickup	Card	45315
590	shuk	0	146226	Klimovichi, Pushkinskaya, 38	Pickup	Card	45315
590	SL	2	146226	Klimovichi, Pushkinskaya, 38	Pickup	Card	45315

Рисунок 4.12 – Страница просмотра заказов

При нажатии на кнопку XML происходит переход на форму импорта и экспорта XML. Страница представлена на рисунке 4.13.

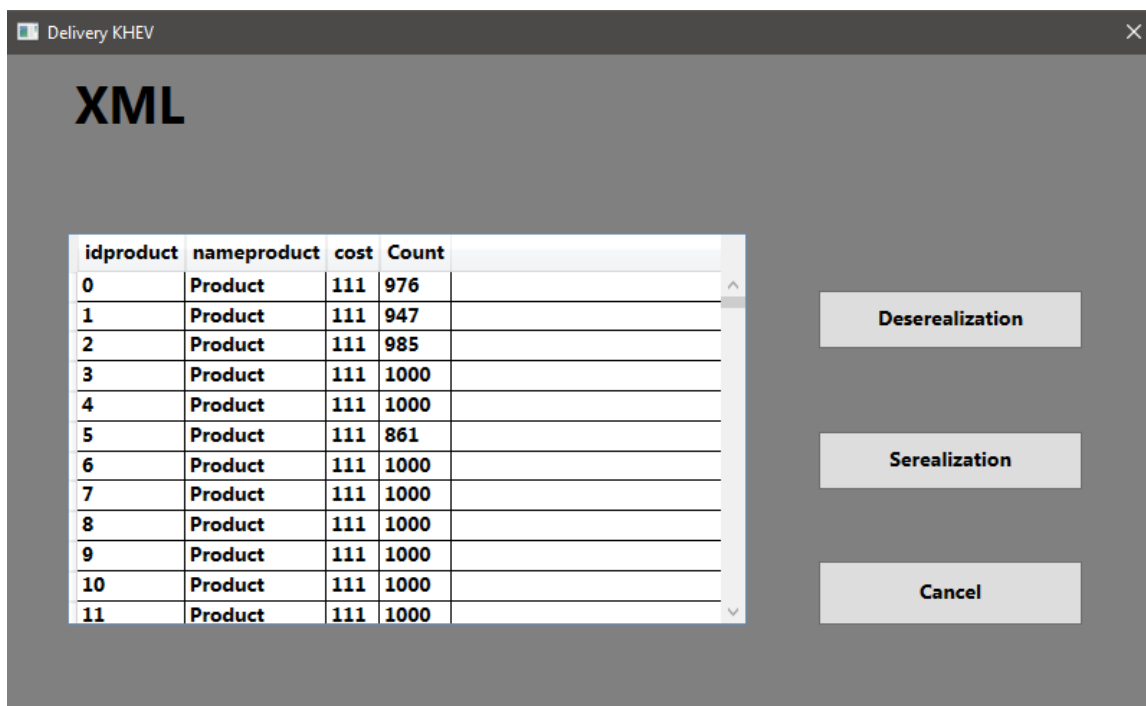


Рисунок 4.13 – Страница импорта и экспорта XML

При нажатии на кнопку «Users» происходит переход на форму, представленную на рисунке 4.14, где есть возможность просмотра всех пользователей и администраторов, а так назначение ролей.

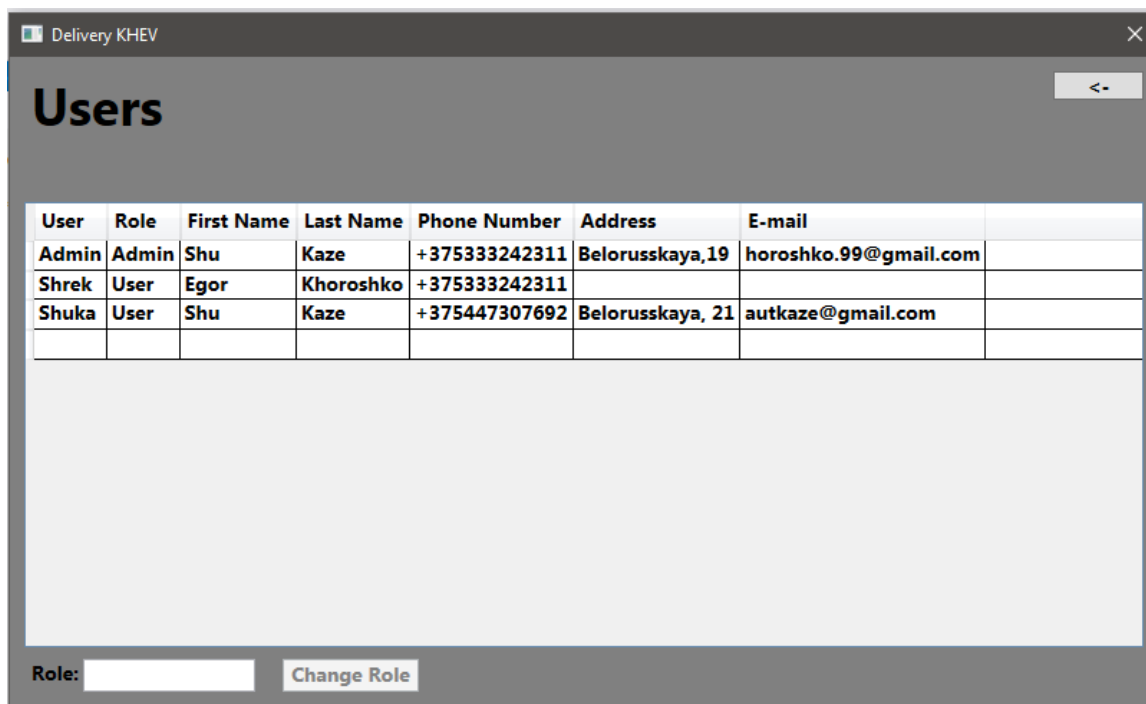


Рисунок 4.14 – Страница просмотра информации о всех пользователей

4.2 Тестовый пример планирования нагрузки

Заполнение таблицы для тестового планирования нагрузки происходит с помощью T-SQL скрипта. Скрипт представлен на рисунке 4.8.

```

CREATE PROCEDURE [dbo].[sp_100000]
AS
    DECLARE @i int
    Set @i = 0
    WHILE @i < 100000
    BEGIN
        INSERT INTO Products(id_product, name_product, cost, Count)
        Values (@i, 'Product', 123, 111, 1000)
        SET @i = @i + 1
    END;

```

Рисунок 4.15. – SQL скрипт для заполнения таблицы Arendation

Вид страницы после заполнения таблицы представлен на рисунке 4.9.

ID	Name Product	Cost	Count	
0	Product	111	976	
1	Product	111	947	
2	Product	111	985	
3	Product	111	1000	
4	Product	111	1000	
5	Product	111	861	
6	Product	111	1000	
7	Product	111	1000	
8	Product	111	1000	
9	Product	111	1000	
10	Product	111	1000	
11	Product	111	1000	

Рисунок 4.16. – Страница после добавления 100000 строк.

Добавление 10000 строк влияет на быстродействие программы. В будущем, при дальнейшем увеличении числа значений, потребуется добавление новых средств оптимизации программы, к примеру, индексация.

4.3 Вывод по разделу

В разделе были рассмотрены возможности пользователя и администратора в разработанном программном средстве, а также был рассмотрен пример нагрузки для вставки в таблицу Orders 10000 строк и поведение программного средства после вставки строк.

Заключение

В ходе выполнения курсовой работы было разработано приложение, позволяющее арендовать автомобили. Во время выполнения данной курсовой работы было изучено немало теоретического материала, а также просмотрено и разобрано большое количество уже готовых решений тех или иных задач.

Разработаны следующие процедуры, некоторые из них:

- оформление заказов;
- просмотр истории заказов;
- поиск по различным критериям;
- импорт и экспорт данных формата xml;
- добавление новых товаров;
- добавление и удаление адресов доставки;

Также была рассмотрена производительность приложения при вставке 100000 строк в таблицу и рассмотрены проблемы при увеличении числа вставленных строк.

Применение технологии SSRS позволило организовать данные заказов и просматривать их при необходимости.

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

				КП 00.00.ПЗ			
	ФИО	Подпись	Дата				
Разраб.	Хорошко Е.В.			Заключение		Лит.	Лист
Провер.	Бондарчик Е.Н.						1
							1
Н. контр.	Бондарчик Е.Н.					БГТУ 71172553, 2019	
Утверд.	Бондарчик Е.Н.						

Список использованной литературы

1. Мэтью Мак-Дональд «Pro WPF 4.5 in C#: Windows Presentation» –Apress, 2012 –1095 с.
2. Статья «WPF – система авторизации и регистрации» [Электронный ресурс]. – режим доступа. – <https://you-hands.ru/2018/08/31/wpf-sistema-avtorizacii-i-registracii/>
3. Документация для разработчиков служб Reporting Services [Электронный ресурс] – режим доступа – <https://docs.microsoft.com/ru-ru/sql/reporting-services/reporting-services-developer-documentation?view=sql-server-ver15>

				<i>КП 00.00.ПЗ</i>			
	ФИО	Подпись	Дата	<i>Список литературы</i>			
Разраб.	<i>Хорошко Е.В.</i>						
Провер.	<i>Бондарчик Е.Н.</i>						
Н. контр.	<i>Бондарчик Е.Н.</i>						
Утверд.	<i>Бондарчик Е.Н.</i>			<i>БГТУ 71172553, 2019</i>			

