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:**

«Использование технологии SQL Server Reporting Services для задачи «Магазин бижутерии»

Выполнил студент Сивак Михаил Николаевич

(Ф.И.О.)

Руководитель проекта пр.-ст. 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)

Консультант: пр.-ст. Бондарчик Е.Н

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

Нормоконтролер: пр.-ст. Бондарчик Е.Н

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

Курсовой проект защищен с оценкой

Минск 2019

**Реферат**

ФИО

Подпись

Дата

Лист

2

КП 00.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

Реферат

Лит.

Листов

1

*БГТУ 71191005, 2019*

Пояснительная записка курсового проекта содержит 36 страниц, 10 таблиц, 23 иллюстрации, 10 источников литературы, 2 приложения.

C#, WPF, MICROSOFT SQL SERVER, DATA BASE, SQL, T-SQL, ADO.NET, SQL SERVER REPORTING SERVICES

Целью курсового проекта является создание базы данных магазина бижутерии, а также ознакомление и применение SQL Server Reporting Services технологии для создания отчётов на основании базы данных.

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

Вторая глава посвящена процессу проектирования модели базы данных.

В третьей главе находится описание процесса разработки, принципы функционирования и назначение созданных компонентов проекта.

В четвёртой главе описаны процедуры для работы с XML.

В пятой главе содержатся сведения об использованной технологии.

Шестая глава посвящена тестированию производительности приложения и самого приложения.

В седьмой главе находится описание руководства пользователя

В заключении приведены итоги курсового проектирования.

ФИО

Подпись

Дата

Лист

3

КП 00.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

Содержание

Лит.

Листов

1

*БГТУ 71191005, 2019*

Содержание

[Введение 4](#_Toc27923858)

[1 Постановка задачи 5](#_Toc27923859)

[1.1 Обзор прототипов 5](#_Toc27923860)

[1.2 Анализ прототипов 7](#_Toc27923861)

[1.3 Описание используемых технологий 7](#_Toc27923862)

[2 Проектирование модели базы данных 9](#_Toc27923863)

[3 Разработка необходимых объектов 10](#_Toc27923864)

[3.1 Таблицы 10](#_Toc27923865)

[3.2 Пользователи 13](#_Toc27923866)

[3.3 Процедуры 14](#_Toc27923867)

[4 Описание процедур импорта и экспорта данных 16](#_Toc27923868)

[4.1 Процедура импорта данных из XML-файла 16](#_Toc27923869)

[4.2 Процедура экспорта данных в XML-файл 16](#_Toc27923870)

[5 Описание технологии 17](#_Toc27923871)

[5.1 Описание применения технологии 17](#_Toc27923872)

[6 Тестирование 20](#_Toc27923873)

[6.1 Тестирование производительности базы данных 20](#_Toc27923874)

[6.2 Тестирование работоспособности приложения 21](#_Toc27923875)

[7 Руководство пользователя 24](#_Toc27923876)

[Заключение 29](#_Toc27923877)

[Список литературных источников 30](#_Toc27923878)

[Приложение А 31](#_Toc27923879)

[Приложение Б 32](#_Toc27923880)

# Введение

ФИО

Подпись

Дата

Лист

4

КП 00.00.ПЗ

Разраб.

*Сивак М.Н.*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

Введание

Лит.

Листов

1

*БГТУ 71191005, 2019*

На сегодняшний день будет достаточно трудно представить информационную систему, которая не имела бы в качестве основы или важной составляющей базу данных. Особенно не могут обойтись без внедрения различных технологий для повышения удобства пользователей интернет-магазины.

Хорошо построенные архитектура базы данных и приложения позволяют сделать просмотр и заказы товаров максимально удобным и значительно сократить затраты времени на это.

Целью моего курсового проекта является создание базы данных интернет-магазина, ознакомление и применение системы отчётов в базе данных.

Основными задачами курсовой работы являются:

* провести аналитический обзор литературы;
* спроектировать базу данных;
* реализовать функциональность базы данных;
* провести тестирование используемой технологии;
* разработать приложение для работы с базой данных;
* написать руководство пользователя.

В соответствии с заданием курсового проекта для проектирования базы данных интернет-магазина используется система управления базами данных Microsoft SQL Server Management Studio.

Также в задачу разработчика входит создание удобного пользовательского интерфейса для комфортного взаимодействия приложения с базой данных.

Для разработки приложения использовалась технология WPF и язык программирования C#. Для взаимодействия приложения с базой данных использовалась технология ADO.NET.

# Постановка задачи

ФИО

Подпись

Дата

Лист

5

КП 01.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

1 Постановка задачи

Лит.

Листов

4

*БГТУ 71191005, 2019*

В соответствии с заданием курсового проекта, помимо создания базы данных, необходимо написать приложение, которое сможет продемонстрировать возможности базы данных. Для формирования окончательных требований к проектируемому программному средству, будет разумным взглянуть на уже существующие прототипы приложений такой же тематики.

## Обзор прототипов

Посмотрим на сайт ювелирных изделий ziko.by. Сразу же хочется сказать, что интерфейс выглядит достаточно удобным для использования, что не может не привлекать большинство пользователей. Также здесь всё разделено на вкладки, при наведении на которые будет отображаться список товаров по выбранному типу товара.

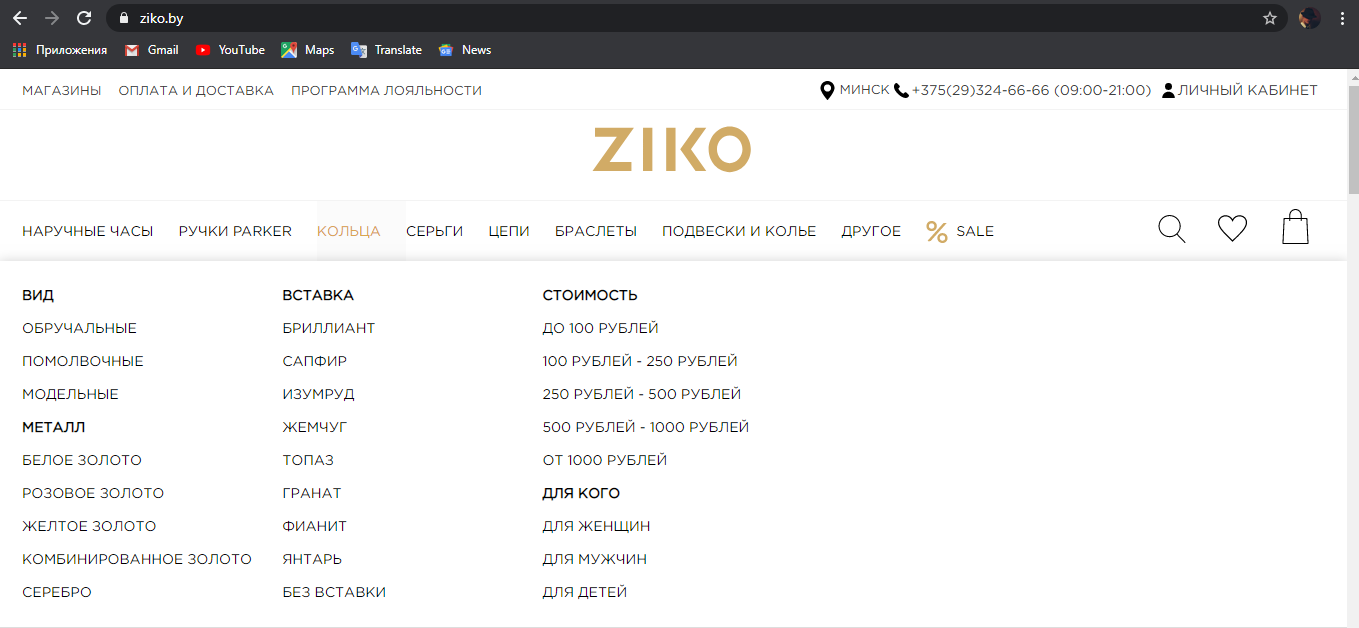


Рисунок 1.1 – Интерфейс сайта ziko.by

Также существуют и другие магазины бижутерии, которые являются достаточно удобными для просмотра информации о товарах, а также для заказа товара, не выходя из дома, что позволяет сэкономить много времени.

Стоит обратить внимание на такой сайт, как monomax.by. Данный сайт достаточно удобен для просмотра ассортимента ювелирных изделий различного типа, но есть существенный минус: на этом сайте нельзя совершить покупку по сети, из-за чего он и уступает в удобстве другим сайтам.

Но это не всё. Сайт monomax.by имеет другой удобный функционал: в нём можно посмотреть на карте места в городе, где клиент может приобрести тот или иной продукт.

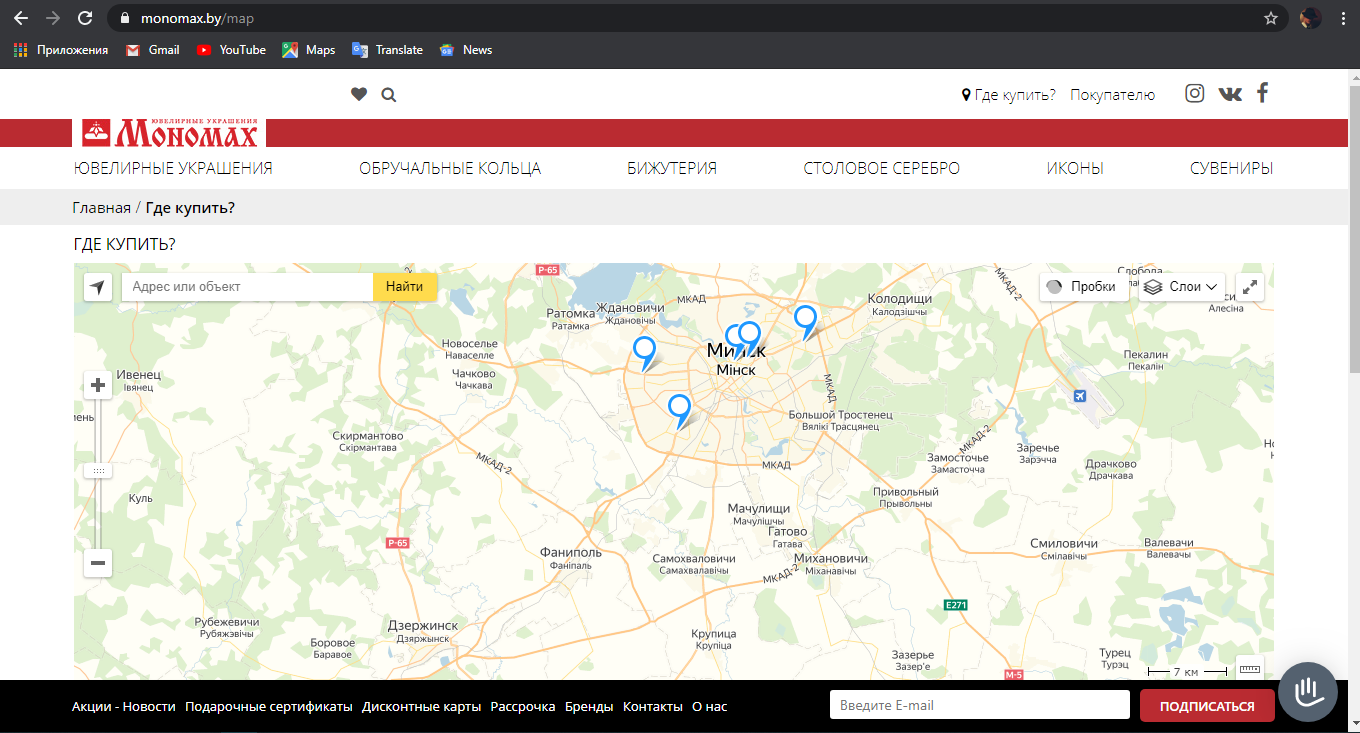


Рисунок 1.2 – Интерфейс сайта monomax.by

Также нельзя пройти стороной мимо сайта уже давно существующего в Беларуси магазина ювелирных изделий slavia.by. Данный сайт также имеет очень удобный интерфейс, а что касается функционала, то в нём есть возможность забронировать заказ, что гарантирует наличие забронированного товара до приезда потенциального покупателя.

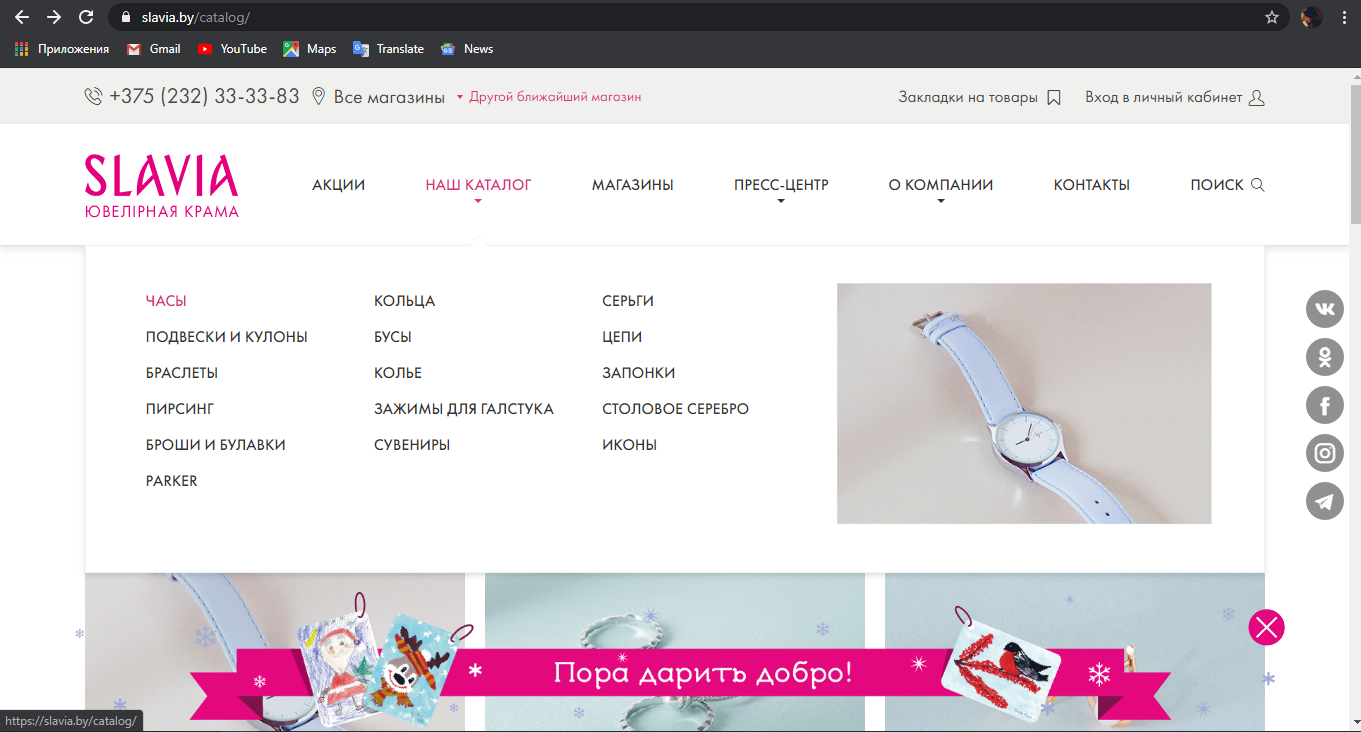


Рисунок 1.3 – Интерфейс сайта slavia.by

**1.2 Анализ прототипов**

Проанализировав прототипы сайтов, можно отметить ожидаемую схожесть в их функциональности и интерфейсе.

Исходя из полученных результатов, можно сформулировать основные функциональные требования курсовой работы:

* регистрация и авторизация клиентов;
* авторизация администратора;
* авторизация кассиров;
* представление списка товаров клиенту;
* оформление клиентом заказа;
* возможности выбора товара по различным критериям;
* просмотр клиентом информации о товарах;
* добавление администратором товаров;
* добавление администратором новых кассиров;
* редактирование профилей кассиров, клиентов и администратора;
* удаление администратором товаров;

Данные требования будут реализованы в базе данных в виде хранимых процедур, которые описаны в главе 3 данной пояснительной записки.

**1.3 Описание используемых технологий**

Существует огромное множество СУБД: Oracle, MS SQL Server, Microsoft Access, MySql и так далее. Для организации работы с базой данных в данной работе использовалась одна из наиболее популярных СУБД «Microsoft SQL Server» [4], так как она предоставляет необходимые возможности оперирования объектами базы данных.

В качестве интерфейса прикладного программирования был выбран API-интерфейс – Windows Presentation Foundation (WPF) [5], платформа уровня представления для построения графических интерфейсов. Основывается на векторной системе визуализации и ориентирована на разработку клиентских Windows приложений, базирующихся на технологии Microsoft.NET.

WPF предоставляет средства для создания визуального интерфейса, включая язык XAML (eXtensible Application Markup Language), элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление. Что является преимуществом перед более ранней технологией создания пользовательских интерфейсов – Windows Forms.

Для работы с WPF использовался объектно-ориентированный язык программирования – С#, разработанный для создания приложений на платформе Microsoft .NET Framework [6].

Для осуществления связи между базой данных и WPF приложением необходим посредник. И именно таким посредником является технология ADO.NET [7]. ADO.NET – это интерфейс прикладного уровня Microsoft, который является самостоятельной технологией, частью .NET Framework. Для обеспечения защиты паролей клиентов применяется технология хеширования.

Для создания отчётов в базе данных достаточно удобной является программная серверная система создания отчётов SQL Server Reporting Services [8], разработанная корпорацией Microsoft. Она также может быть использована для подготовки множества интерактивных и печатных отчётов. Система администрируется через веб-интерфейс. Reporting Services используют интерфейс веб-служб для поддержки разработки обычных отчётных приложений.

# 

# Проектирование модели базы данных

ФИО

Подпись

Дата

Лист

9

КП 02.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

2 Проектирование модели базы данных

Лит.

Листов

1

*БГТУ 71191005, 2019*

Для реализации поставленной задачи была создана база данных JewerlyStore. Диаграмма структуры полученной базы данных, разработанной в СУБД «Microsoft SQL Server», представлена в приложении А.

Для базы данных было разработано 10 таблиц, 8 из которых связаны друг с другом внешними ключами.

Таблица Users хранит данные о клиентах, кассирах и администраторе.

Таблица Roles хранит роли пользователей.

Таблица Countries содержит названия стран производителей товаров.

Таблица Manufacturers содержит названия производителей товаров и страны производителей.

Таблица Orders хранит данные о текущих заказах клиентов.

Таблица OrdersHistory хранит историю всех заказов клиентов.

Таблица Products содержит информацию о товарах.

Таблица ProductsMaterials хранит названия материалов из которых изготовлены бижутерии.

Таблица ProductsTypes хранит названия (типы) товаров.

Таблица WorkTime хранит время работы кассиров.

Создание перечисленных таблиц описано в главе 3.

# Разработка необходимых объектов

ФИО

Подпись

Дата

Лист

10

КП 03.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

3 Разработка необходимых объектов

Лит.

Листов

7

*БГТУ 71191005, 2019*

## Таблицы

Таблицы являются неотъемлемой частью любой реляционной базы данных. Краткая характеристика каждой из таблиц была предоставлена в разделе 2, а код их создания можно увидеть в Приложении Б. Ниже мы рассмотрим каждую таблицу подробнее.

Таблица Users состоит из семи столбцов:

– idUser – хранит идентификаторы всех пользователей;

– idRole – хранит идентификаторы ролей пользователей;

– email – хранит названия почты каждого пользователя;

– password – хранит пароли всех пользователей;

– first\_name – имена пользователей;

– last\_name – фамилии пользователей;

– phone\_number – номера телефонов всех пользователей.

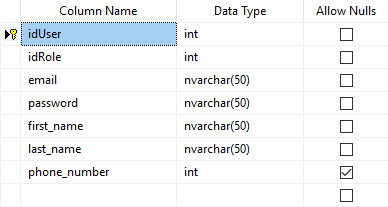


Рисунок 3.1 – Описание таблицы Users

Таблица Roles состоит из двух столбцов:

– idRole – хранит идентификаторы ролей;

– roleName – хранит имена ролей.

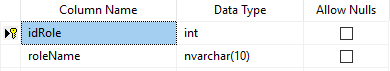


Рисунок 3.2 – Описание таблицы Roles

Таблица Countries состоит из двух столбцов:

– idCountry – хранит идентификаторы стран;

– nameCountry – хранит названия стран.

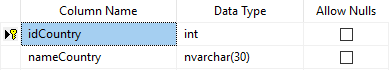


Рисунок 3.3 – Описание таблицы Countries

Таблица Manufacturers состоит из трёх столбцов:

– idManufacturer – хранит идентификаторы производителей;

– nameManufacturer – хранит имена производителей;

– idCountry – идентификаторы стран производителей.

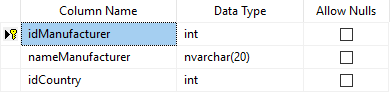


Рисунок 3.4 – Описание таблицы Manufacturers

Таблица Orders состоит из пяти столбцов:

– idOrder – хранит идентификаторы текущих заказов;

– idUser – хранит идентификаторы клиентов, которые совершил заказ;

– idProduct – хранит идентификаторы товаров;

– orderDate – хранит даты текущих заказов;

– totalPrice – хранит общую стоимость товаров.

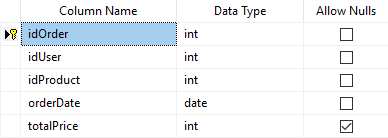


Рисунок 3.5 – Описание таблицы Orders

Таблица OrdersHistory состоит из пяти столбцов:

– idOrder – хранит идентификаторы заказов;

– idUser – хранит идентификаторы клиентов, которые совершил заказ;

– idProduct – хранит идентификаторы товаров;

– orderDate – хранит даты заказов;

– totalPrice – хранит общую стоимость товаров.

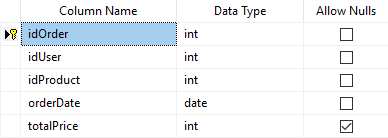


Рисунок 3.6 – Описание таблицы OrdersHistory

Таблица Products состоит из шести столбцов:

– idProduct – хранит идентификаторы товаров;

– productPrice – хранит цены на продукты;

– productQuantity – хранит количество каждого продукта;

– idManufacturer – хранит идентификаторы производителей;

– idType – хранит идентификаторы названий (типов) товаров;

– idMaterial – хранит идентификаторы названий материалов.

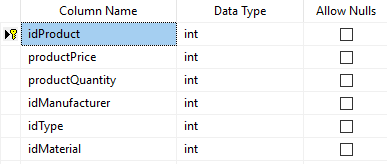


Рисунок 3.7 – Описание таблицы Products

Таблица ProductsMaterials состоит из двух столбцов:

– idMaterial – хранит идентификаторы названий материалов;

– nameMaterial – хранит названия материалов.

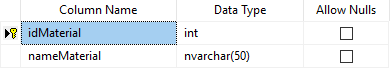


Рисунок 3.8 – Описание таблицы ProductsMaterials

Таблица ProductsTypes состоит из двух столбцов:

– idType – хранит идентификаторы названий (типов) товаров;

– nameType – хранит названия (типы) товаров.

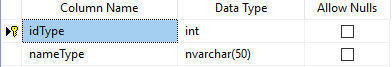


Рисунок 3.9 – Описание таблицы ProductsTypes

Таблица WorkTime состоит из двух столбцов:

– idUser – хранит идентификаторы кассиров;

– time – хранит время работы каждого кассира.

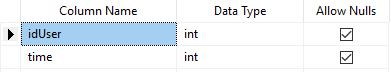


Рисунок 3.10 – Описание таблицы WorkTime

## Пользователи

Пользователь базы данных – это физическое или юридическое лицо, которое имеет доступ к БД и пользуется услугами информационной системы для получения информации. Пользователи базы данных получают права доступа для чтения, вставки, обновления и удаления конкретных объектов.

Администратор интернет-магазина наделён правами на выполнение большинства хранимых процедур, разработанных для базы данных.

Клиенту позволено выполнять процедуры для авторизации, регистрации, процедуры для получения информации из БД, процедуры для заказа товаров, изменения своих личных данных.

Кассир может выполнять процедуры отправки товаров, просмотра списка текущих заказов, изменение своих личных данных.

## Процедуры

Использование хранимых процедур позволяет ограничить либо вообще исключить непосредственный доступ пользователей к таблицам базы данных, оставив пользователям только разрешения на выполнение хранимых процедур, обеспечивающих косвенный и строго регламентированный доступ к данным.

Всего было разработано 55 процедур для работы с данными. Краткое описание процедур описано далее.

addAdmin – процедура для добавления администратора магазина;

addCashier – процедура для регистрации кассиров;

addClient – процедура для регистрации клиентов;

addCountry – процедура для добавления стран;

addHistoryOrder – процедура для добавления заказов в таблицу истории заказов;

addManufacturer – процедура для добавления производителей;

addMaterial – процедура для добавления материлов;

addOrder – процедура для добавления текущих заказов в таблицу текущих заказов;

addProduct – процедура для добавления товаров;

addRoles – процедура для добавления ролей;

addType – процедура для добавления названий (типов) товаров;

addWorkTime – процедура для добавления времени работы кассиров;

deleteCashier – процедура для удаления кассиров;

deleteClient – процедура для удаления клиентов;

deleteCountry – процедура для удаления стран;

deleteManufacturer – процедура для удаления производителей;

deleteMaterial – процедура для удаления материалов;

deleteOrder – процедура для отправки товаров и удаления текущего заказа;

deleteProduct – процедура для удаления товаров;

deleteType – процедура для удаления названий (типов) товаров;

deleteUsers – процедура для удаления пользователей;

getAllCountries – процедура для получения списка всех стран;

getAllManufacturers – процедура для получения списка всех производителей;

getAllMaterials – процедура для получения списка всех материалов;

getAllOrders – процедура для получения списка текущих заказов;

getAllTypes – процедура для получения списка всех типов товаров;

getAllUsers – процедура для получения списка всех пользователей;

getCashiers – процедура для получения списка всех кассиров;

getCashiersEmails – процедура для получения почты кассиров;

getClients – процедура для получения списка всех клиентов;

getClientsEmails – процедура для получения почты клиентов;

getCountries – процедура для получения списка всех стран;

getCountryId – процедура для получения идентификатора страны по её имени;

getHistoryOrders – процедура для получении истории всех заказов за всё время;

getIdProduct – процедура для получения идентификаторов продуктов;

getIdRole – процедура для получения идентификаторов ролей;

getIdUser – процедура для получения идентификаторов пользователей;

getManufacturerId – процедура для получения идентификатора производителя;

getManufacturers – процедура для получения списка всех производителей;

getMaterialId – процедура для получения идентификатора материала;

getMaterials – процедура для получения списка всех материалов;

getMnftrByCntryName – процедура для получения списка имён производителей по названию страны.

getMoney – процедура для получения общего дохода;

getProductPrice – процедура для получения стоимости товаров;

getProducts – процедура для получения списка всех товаров;

getQuantityProducts – процедура для получения общего количества товаров;

getTypeById – процедура для получения типа товара по его идентификатору;

getTypeId – процедура для получения идентификатора типа товара;

getWorkTime– процедура для получения времени работы каждого кассира;

InfoUser – процедура для получения информации о текущем пользователе;

insertUsers – процедура для добавления пользователей;

insertUsers – процедура для добавления пользователей;

productsImportXml – процедура для импорта XML в таблицу базы данных;

updateUser – процедура для обновления данных текущего пользователя;

productsExportXml – процедура для Экспорта данных из таблицы базы данных в XML.

Все скрипты хранимых процедур приложены в отдельных файлах в корне директории прилагаемого диска.

# Описание процедур импорта и экспорта данных

ФИО

Подпись

Дата

Лист

16

КП 04.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

4 Описание процедур импорта и экспорта данных

Лит.

Листов

1

*БГТУ 71191005, 2019*

База данных обычно имеет не самостоятельную ценность, является частью информационной системы. Независимо от того, как устроена эта система, на противоположном от БД конце находится интерфейс взаимодействия с пользователем, и задача программиста предоставить простой и понятный способ работы с хранящимися в БД данными и объектами.

При всей своей отлаженности и очевидности, классический способ хранения и представления объектов развитой структуры имеет и вполне определенные недостатки и может вызывать проблемы, с которыми сталкивался любой разработчик, пытавшийся реализовать таким способом достаточно сложную систему. В некоторых ситуациях, решить эти проблемы позволяет хранение объекта в виде XML.

## Процедура импорта данных из XML-файла

С помощью процедуры productsImportXml есть возможность добавить тип продукта в таблицу ProductsTypes из XML-файла. Листинг данной процедуры можно увидеть в приложении Б.

* 1. Процедура экспорта данных в XML-файл

Также в приложении Б представлена процедура productsExportXml, которая записывает типы продуктов по пути, указанному в path.

Процедура productsExportXml возвращает данные, которые затем вставляются в XML-файл. Её реализацию можно увидеть в приложении Б.

# Описание технологии

ФИО

Подпись

Дата

Лист

17

КП 05.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

5 Описание технологии

Лит.

Листов

1

*БГТУ 71191005, 2019*

SQL Server Reporting Services – программная [серверная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)) система создания [отчётов](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D1%87%D1%91%D1%82), разработанная корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft" \o "Microsoft). Она может быть использована для подготовки множества [интерактивных](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) и печатных отчётов. Система администрируется через [веб](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0)-интерфейс. Reporting services используют [интерфейс](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) [веб-служб](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) для поддержки разработки обычных отчётных приложений.

В SSRS отчёты описываются при помощи [Report Definition Language](https://ru.wikipedia.org/wiki/Report_Definition_Language" \o "Report Definition Language) (RDL) на [языке разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) [XML](https://ru.wikipedia.org/wiki/XML). Отчёты могут проектироваться при помощи последних версий [Microsoft Visual Studio](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio" \o "Microsoft Visual Studio) (включая Visual Studio.NET 2003 и Visual Studio 2005) с входящим в них [дополнением](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD) Business Intelligence Projects или при помощи входящего в комплект Report Builder – упрощённого инструмента, не предлагающего полного функционала Visual Studio.

Отчёты, определённые при помощи RDL, могут создаваться во множестве различных форматов, включая [Excel](https://ru.wikipedia.org/wiki/Microsoft_Excel" \o "Microsoft Excel), [PDF](https://ru.wikipedia.org/wiki/PDF), [CSV](https://ru.wikipedia.org/wiki/CSV), [XML](https://ru.wikipedia.org/wiki/XML), [TIFF](https://ru.wikipedia.org/wiki/TIFF) (и других графических форматах), а также HTML Web Archive. SQL Server 2008 SSRS также может подготавливать отчёты в формате [Microsoft Word](https://ru.wikipedia.org/wiki/Microsoft_Word" \o "Microsoft Word) (DOC).

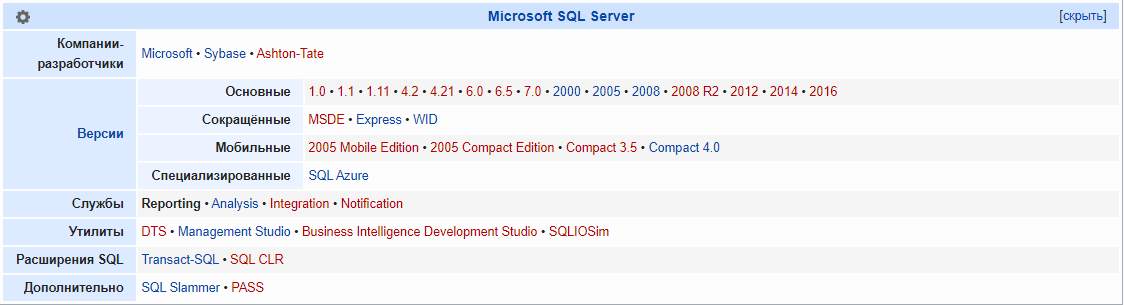


Рисунок 5.1 – Информация о Reporting Services

## 5.1 Описание применения технологии

Для того чтобы создать отчет сначала необходимо установить и настроить сервер отчетов. Для того, чтобы начать установку, нужно скачать файл для установки сервера отчётов. Скачать его можно на официальном сайте Microsoft.com.

После скачивания установочного файла, нужно начать установку сервера, следуя инструкциям по шагам установки.

После того, как сервер будет установлен, нужно открыть приложение с именем Report Server Configuration Manager.

После открытия Report Server Configuration Manager нужно, следуя по шагам, настроить сервер отчётов.

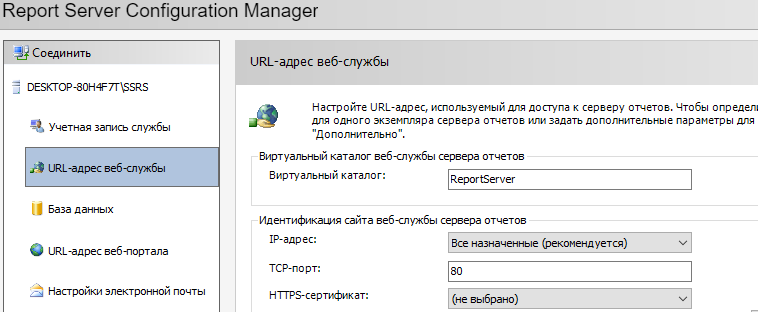


Рисунок 5.2 – Настройка серевера отчётов

Затем можем приступать к созданию проекта Reporting Services, используя Visual Studio.

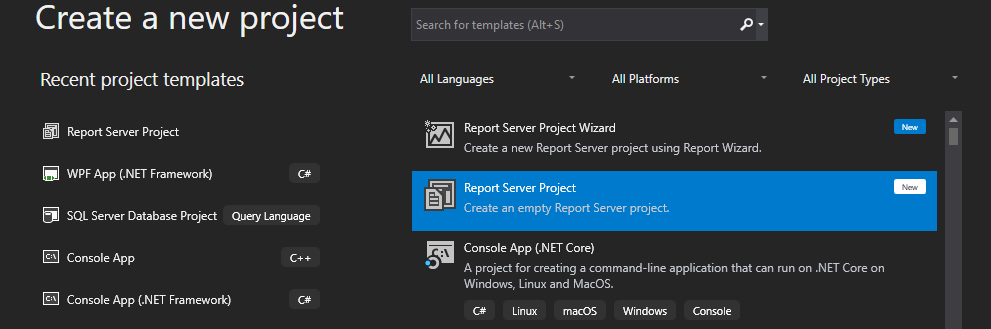


Рисунок 5.3 – Окно создания проекта Reporting Services

Далее необходимо добавить новый отчёт в директорий Reports в обозревателе решений, а затем перейти к данным отчёта, после чего добавить источник данных и набор данных.

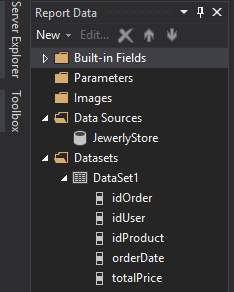
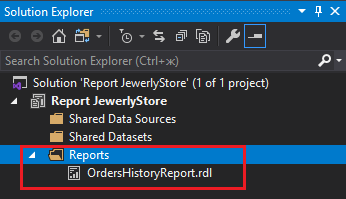


Рисунок 5.4 – Создание отчёта

Далее на панели элементов выбираем нужный нам элемент, в нашем случае это таблица, и заполняем его данными. После успешного заполнения запускаем проект. После успешного развёртывания отчёта, он будет загружен в сервер отчетов Reporting Services.

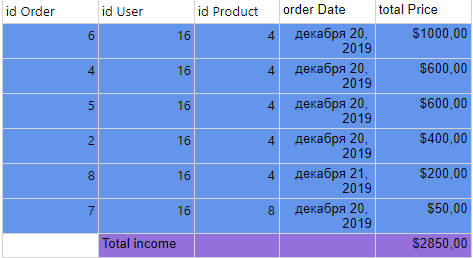


Рисунок 5.5 – Результат выгрузки проекта

Как мы видим, система создания отчётов достаточно удобна для визуализации данных, что является важным, так как человеку проще и комфортнее воспринимать информацию визуально, чем в виде обычного текста.

С помощью технологии Reporting Services также можно создавать отчёты в виде диаграмм и различных графиков, что очень удобно для просмотра статистики полученных данных.

# Тестирование

ФИО

Подпись

Дата

Лист

20

КП 06.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

6 Тестирование

Лит.

Листов

4

*БГТУ 71191005, 2019*

* 1. Тестирование производительности базы данных

Для тестирования производительности была взята за основу таблица Users, которую мы заполнили 100000 строками. После этого был применён SELECT-запрос к данной таблице на столбец email и при помощи стандартных средств IDE MS SQL Server Management Studio оценена цена выборки к таблице. Результат данной оценки запроса приведён на рисунке 6.1.

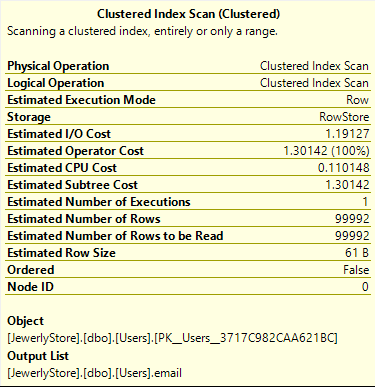


Рисунок 6.1 – Оценка запроса к таблице Users с кластеризованным индексом

Как мы видим, операция запроса для такого большого количества строк является слишком затратным, и при выполнении её в приложении, оно может зависнуть некоторое время, что ухудшает производительность приложения, а также доставляет неудобство пользователю.

После проведения первоначальной оценки был построен некластеризованный индекс к таблице Users, создание которого можно увидеть в приложении Б, и проведена оценка такого же SELECT-запроса к таблице Users. Результаты, полученные во время оценки, представлены на рисунке 6.2.

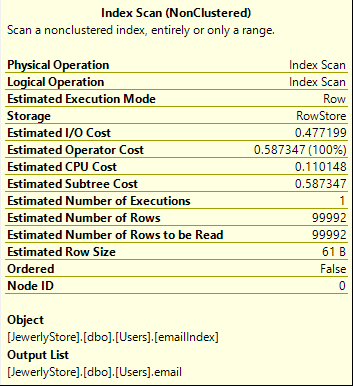


Рисунок 6.2 – Оценка запроса к таблице с построенным некластеризованным индексом

По результатам проведённых оценок можно сказать, что постройка индекса к таблице была более чем оправдана, так как мы получили прирост производительности почти в 2 раза.

## Тестирование работоспособности приложения

Тестирование приложения предназначено для выявления возможных ошибок при разработке и для того, чтобы исключить действия пользователя, которые могут повлечь за собой аварийное завершение программы.

Написание валидации для выявления ошибок и обработки исключений является очень хорошей практикой и профилактикой как для итогового программного продукта, так и в процессе его разработки.

Тестирование работоспособности приложений является очень важным на этапе завершения разработки программного средства, он позволяет выявить все существующие и часто не очевидные ошибки, чтобы улучшить качество своего продукта. Ни одна кампания не обходится без данного этапа развития своего программного продукта, в каждой кампании есть тестировщики, которые проверяют работоспособность программы всеми возможными способами, применяя как ручное, так и автоматизированное тестирование компонентов. Благодаря деятельности тестировщиков, можно создать хороший и качественный продукт, обеспечивая безопасность и защиту личных данных пользователей.

Рассмотрим регистрацию и авторизацию пользователя. Если пользователь не заполнит какие-нибудь поля при регистрации или авторизации, то он будет уведомлён об этом всплывающим окном с текстом.

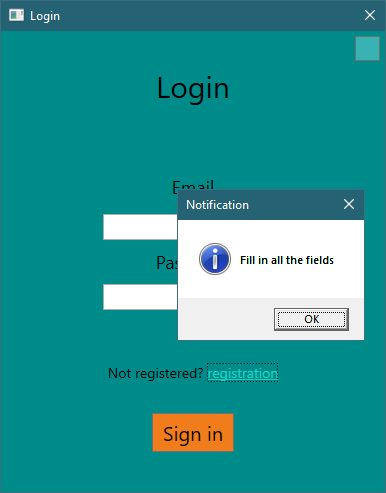
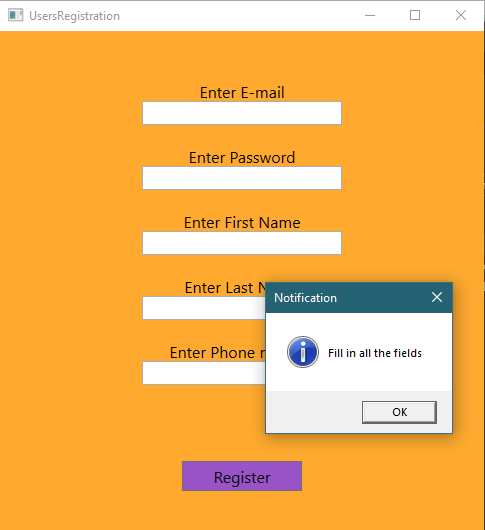


Рисунок 6.3 – Проверка заполнения полей при регистрации и авторизации

В окне авторизации пользователя, если пользователь введёт данные, которых нет в базе данных, то пользователь получит уведомление о некорректно введённых логине или пароле.

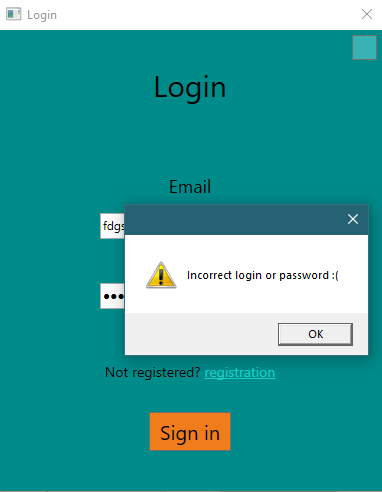


Рисунок 6.4 – Проверка ошибки авторизации

При добавлении администратором каких-либо данных, например, добавление кассира, также проводится контроль заполнения всех полей.

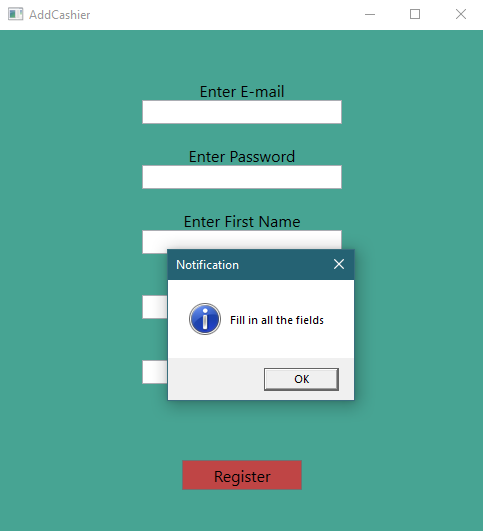


Рисунок 6.5 – Проверка заполнения полей при добавлении нового объекта

Также проверка срабатывает, когда клиент не заполняет все поля формы во время оформления заказа.

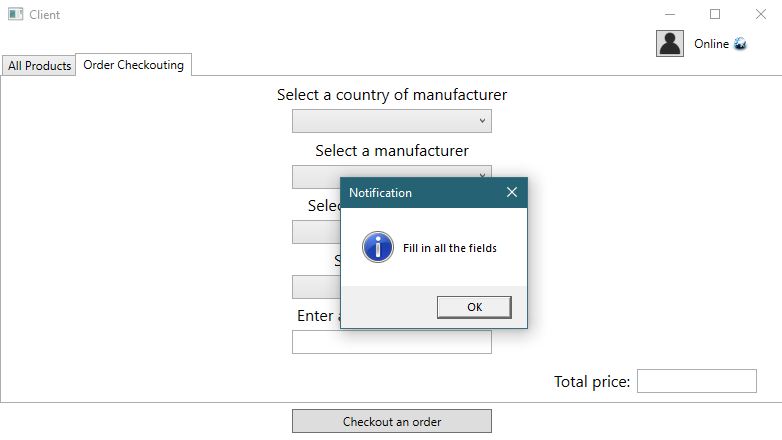


Рисунок 6.6 – Проверка заполнения полей при заказе товара

# Руководство пользователя

ФИО

Подпись

Дата

Лист

24

КП 07.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

7 Руководство пользователя

Лит.

Листов

6

*БГТУ 71191005, 2019*

Первое, что должен сделать пользователь после открытия приложения – это войти, а если он не зарегистрирован, то он может перейти соответственно в окно регистрации, нажав на ссылку «registration», и сначала зарегистрироваться.

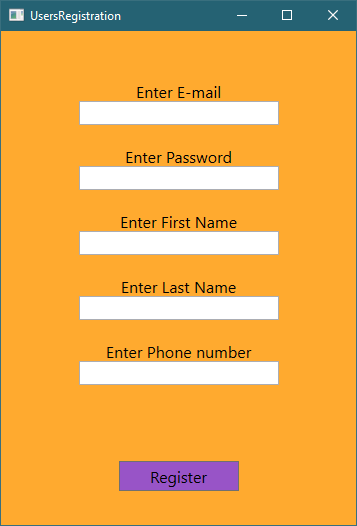
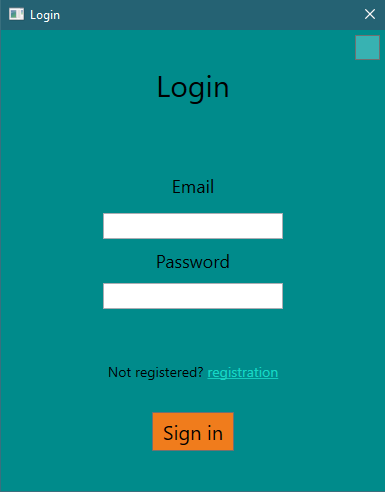


Рисунок 7.1 – Окна авторизации и регистрации

После авторизации пользователь попадает в свой профиль, где может совершать позволенные ему операции по работе с базой данных. Авторизация пользователей предназначена, как для клиентов, так для кассиров и администратора магазина.

После того, как кассир зайдёт в свой аккаунт, ему сразу будет отображена таблица с информацией о текущих товарах. Он может перейти к просмотру либо редактированию профиля либо отправить заказ клиенту.

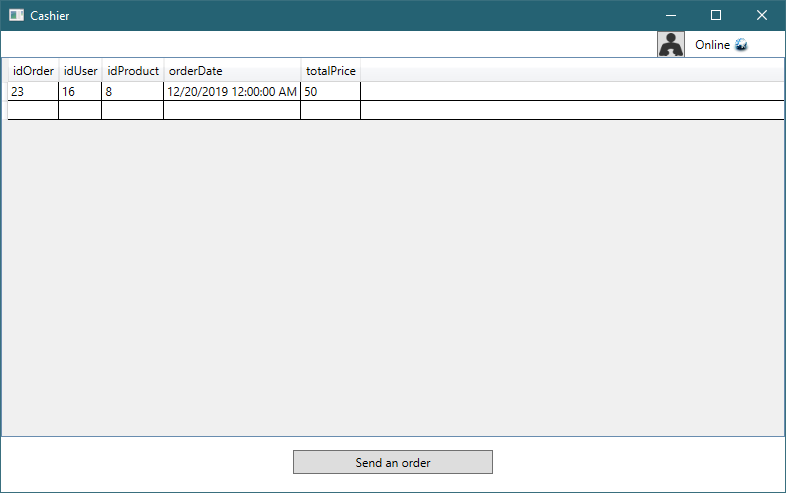


Рисунок 7.2 – Окно кассира

При оформлении клиентом заказа нужно нажать на вкладку Order Checkouting и заполнить все поля, расположенные на форме, после чего нажать на кнопку «Checkout an order». После перечисленных действий появится уведомление об успешном добавлении заказа.

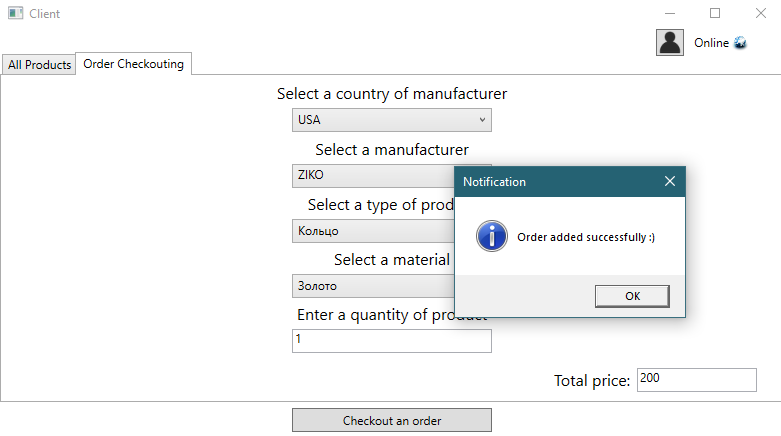


Рисунок 7.3 – Оформление заказа клиентом

Теперь, после оформление пользователем заказа, в окне кассира в списке текущих заказов отображается новый заказ.

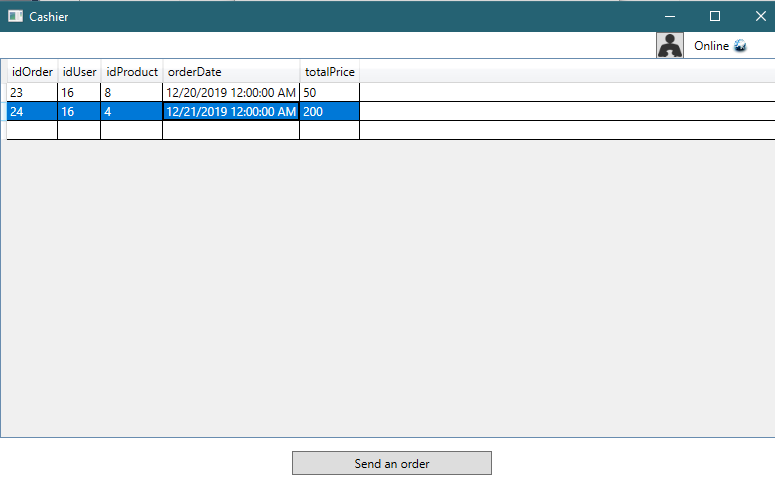


Рисунок 7.4 – Окно кассира со списком текущих заказов

Чтобы кассир отправил заказ клиенту, нужно нажать на кнопку «Send an order». После этого появится окно с полем для ввода идентификатора заказа.

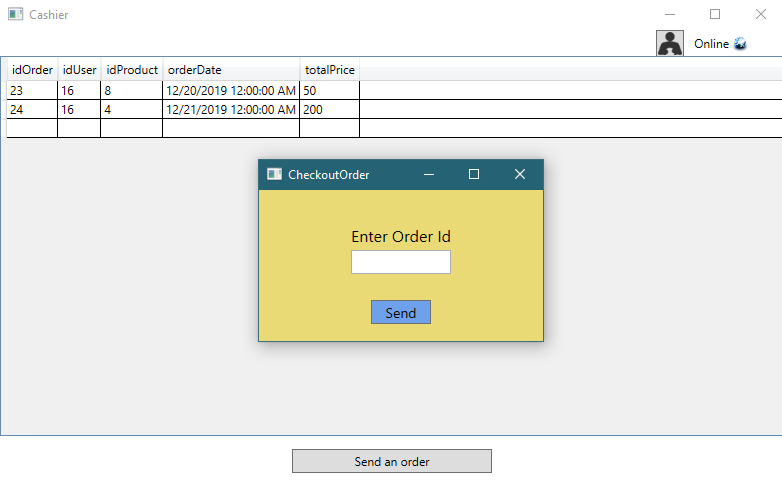


Рисунок 7.5 – Отравление товара

После введения идентификатора заказа в поле формы «CheckoutOrder» и нажатия на кнопку «Send», появится уведомление о успешно отправленном товаре.

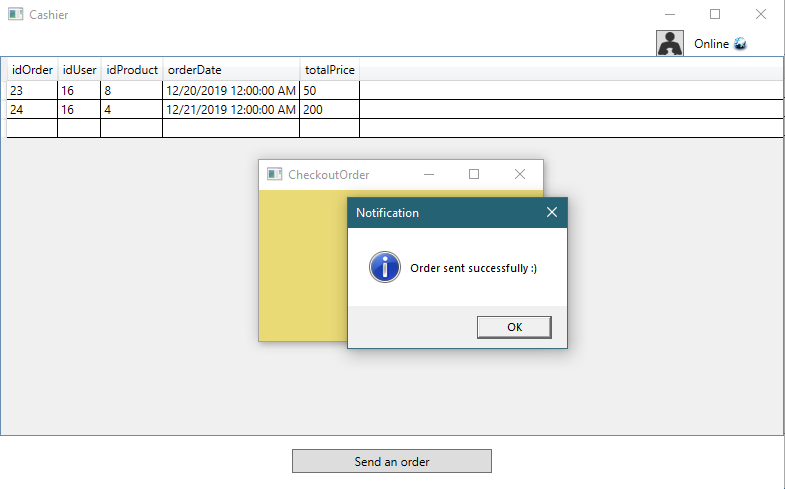


Рисунок 7.6 – Уведомление о успешно отправленном товаре

Каждый пользователь, войдя в свой профиль, может посмотреть, а также изменить свои данные. Чтобы посмотреть информацию о профиле, нужно нажать на иконку в правом верхнем углу формы.

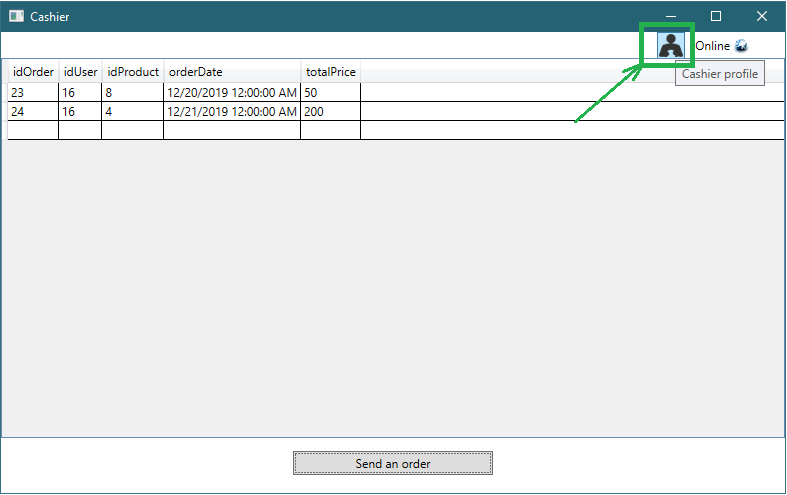


Рисунок 7.7 – Окно пользователя

После нажатия на иконку с картинкой, в окне пользователя будет отображаться личная информация.

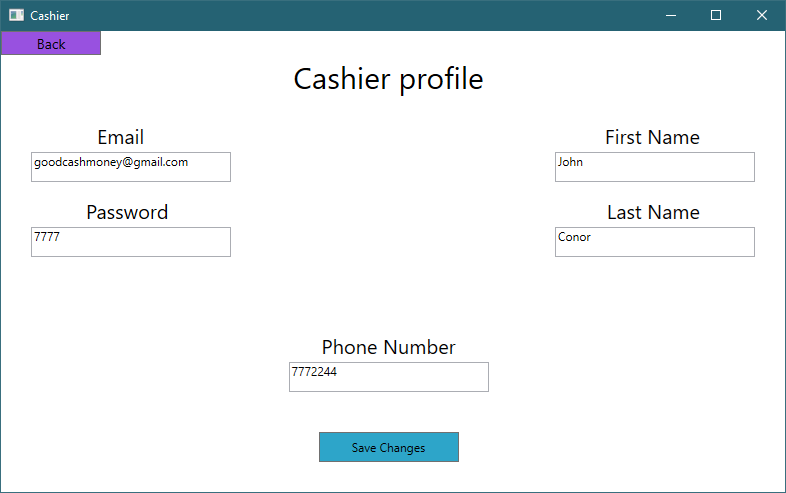


Рисунок 7.8 – Страница с данными профиля пользователя

Теперь пользователь может редактировать свой профиль, меняя содержание полей. После того, как пользователь введёт новые данные в поля, для их сохранения нужно нажать на кнопку «Save Changes». В результате выполнения этих действий появится уведомление о успешно изменённых данных, а также эти данные сразу отобразятся в полях профиля.

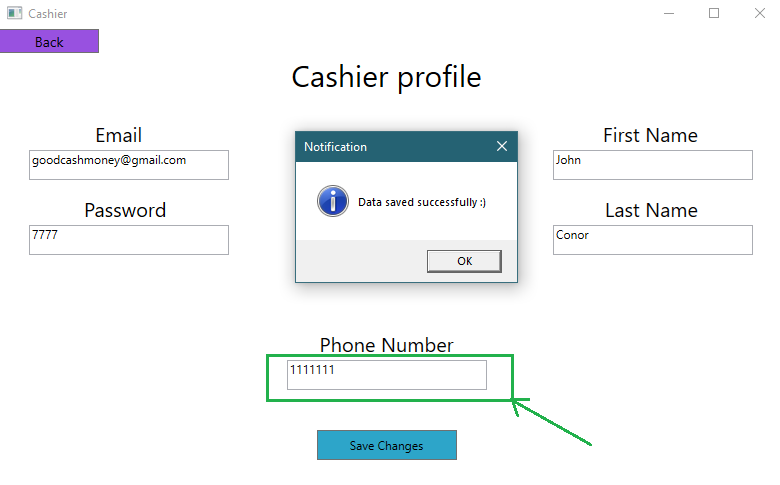


Рисунок 7.9 – Изменение личных данных пользователя

# Заключение

ФИО

Подпись

Дата

Лист

29

КП 00.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

Заключение

Лит.

Листов

1

*БГТУ 71191005, 2019*

Перед началом разработки базы данных был произведен аналитический обзор прототипов приложений подобной тематики и определение функциональных возможностей моей базы данных.

В процессе выполнения курсовой работы была спроектирована база данных для интернет-магазина. База данных была разработана с помощью системы управления базами данных «Microsoft SQL Server 2018». Помимо этого, для визуализации данных при создании отчёта я познакомился с использованием системы отчётов SQL Server Reporting Services.

При разработке базы данных было созданы следующие объекты:

– 8 таблиц с внешними связями;

– 2 таблицы без связей;

– 55 хранимых процедур;

– 1 индекс;

– 3 пользователя базы данных.

Также было разработано приложение, демонстрирующее работу с базой данных. Приложение написано с помощью технологии WPF и языка программирования C#. Связь между базой данных и приложением осуществляется с помощью технологии ADO.NET, которая предоставляет набор классов, через которые мы можем установить подключение к базе данных и производить ряд других операций. Также было написано руководство пользователя для созданного приложения.

В соответствии с полученным результатом, можно сказать, что разработанная база данных функционирует верно, требования технического задания реализованы в полном объеме, в созданном приложении предоставлены возможности базы данных, поэтому цель курсового проекта можно считать выполненной.

**Список литературных источников**

ФИО

Подпись

Дата

Лист

30

КП 00.00.ПЗ

Разраб.

*Сивак М. Н*

Провер.

Бондарчик Е.Н.

Н. контр.

.

Утверд.

Список литературных источников

Лит.

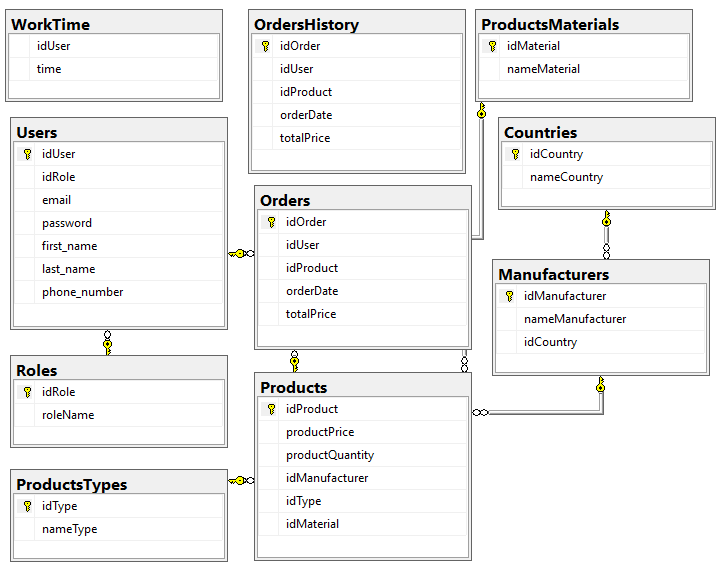
Листов

1

*БГТУ 71191005, 2019*

1. MSDN сеть разработчиков в Microsoft [Электронный ресурс] / Режим доступа: <http://msdn.microsoft.com/library/rus/>.
2. Примеры массового импорта и экспорта XML-документов (SQL Server) [Электронный ресурс] – режим доступа - [https://docs.microsoft.com/ru-ru/sql/relational-databases/import-export/examples-of-bulk-import-and-export-of-xml-documents-sql-server.](https://docs.microsoft.com/ru-ru/sql/relational-databases/import-export/examples-of-bulk-import-and-export-of-xml-documents-sql-server.%20)
3. METANIT.COM Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com>.
4. Документация СУБД «Microsoft SQL Server» [Электронный ресурс] – Режим доступа: <https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15>
5. Мэтью Макдональд — WPF: Windows Presentation Foundation в .NET 4.0 с примерами на C#
6. Язык C# и .NET Framework [Электронный ресурс] — Режим доступа: <http://professorweb.ru/my/csharp/charp_theory/level1/>
7. Руководство по ADO.NET и работе с базами данных [Электронный ресурс] – Режим доступа: <https://metanit.com/sharp/adonet/>
8. Службы SQL Server Reporting Services (SSRS) [Электронный ресурс] / Режим доступа: <https://docs.microsoft.com/ru-ru/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated-reports?view=sql-server-2017>
9. Хранение больших объектов в базе данных или файловой системе в SQL Server [Электронный ресурс] – Режим доступа: <https://www.microsoft.com/en-us/research/publication/to-blob-or-not-to-blob-large-object-storage-in-a-database-or-a-filesystem/>
   * + - 1. 10. Блинова Е.А. [Курс лекций по базам данных] / Е.А. Блинова.

# Приложение А

Диаграмма базы данных JewerlyStore.

# Приложение Б

Листинг создания таблиц

create table Roles --Роли

(

idRole int not null IDENTITY(1,1) primary key,

roleName nvarchar(10) not null

);

create table Users --Все пользователи (Включая администратора и кассиров)

(

idUser int not null IDENTITY(1,1) primary key,

idRole int not null,

email nvarchar(50) not null,

password nvarchar(50) not null,

first\_name nvarchar(50) not null,

last\_name nvarchar(50) not null,

phone\_number int

);

create table Orders --Заказы

(

idOrder int not null IDENTITY(1,1) primary key,

idUser int not null,

idProduct int not null,

orderDate date not null,

totalPrice int

);

create table OrdersHistory --История заказов

(

idOrder int not null IDENTITY(1,1) primary key,

idUser int not null,

idProduct int not null,

orderDate date not null,

totalPrice int

);

create table Products --Товары

(

idProduct int not null IDENTITY(1,1) primary key,

productPrice int not null,

productQuantity int not null,

idManufacturer int not null,

idType int not null,

idMaterial int not null

);

create table Manufacturers --Производители

(

idManufacturer int not null IDENTITY(1,1) primary key,

nameManufacturer nvarchar(20) not null,

idCountry int not null

);

create table Countries --Страны производителей

(

idCountry int not null IDENTITY(1,1) primary key,

nameCountry nvarchar(30) not null

);

create table ProductsTypes --Типы продуктов

(

idType int not null IDENTITY(1,1) primary key,

nameType nvarchar(50) not null

);

create table ProductsMaterials --Материал продуктов

(

idMaterial int not null IDENTITY(1,1) primary key,

nameMaterial nvarchar(50) not null

);

create table WorkTime --Время работы кассира

(

idUser int,

time int

);

Процедура импорта данных из XML-файла

create procedure productsImportXml

@file\_path nvarchar(max)

as

set nocount on

set xact\_abort on

begin transaction

declare @sql nvarchar(MAX) = 'INSERT INTO ProductsTypes (nameType)

SELECT

MY\_XML.productType.query(''nameType'').value(''.'', ''VARCHAR(50)'')

FROM (SELECT CAST(MY\_XML AS xml)

FROM OPENROWSET(BULK ''' + @file\_path + ''' , SINGLE\_BLOB) AS T(MY\_XML)) AS T(MY\_XML)

CROSS APPLY MY\_XML.nodes(''productTypes/productType'') AS MY\_XML (productType)';

execute sp\_executesql @sql;

-- Begin Return Select <- do not remove

select idType, nameType from ProductsTypes

where idType = scope\_identity()

-- End Return Select <- do not remove

commit

go

declare @return\_value int

execute @return\_value = productsImportXml

@file\_path = 'E:\products.xml'

GO

Процедура экспорта данных в XML-файл

create procedure productsExportXml

as

set nocount on

set xact\_abort on

begin transaction

select \* FROM ProductsTypes

for XML path('productType'), ROOT('productTypes');

commit

GO

--drop procedure productsExportXml

execute productsExportXml;

Создание индекса

go

Create procedure insertUsers

as

declare @num int,

@role int,

@email nvarchar(50),

@password int,

@first\_name nvarchar(50),

@last\_name nvarchar(50),

@phone\_number int;

set @num =13;

set @password = RAND(1);

while @num<=100000

begin

if (@num % 100)=0

set @role = 1

else if (@num % 100)=1

set @role = 2

else set @role = 3

set @email = 'user' + cast(@num as nvarchar(8))+'@gmail.com';

set @password = CEILING(rand()\*3000);

set @first\_name = 'firstname' + cast(@num as nvarchar(8));

set @last\_name = 'lastname' + cast(@num as nvarchar(8));

set @phone\_number = 1000000+@num;

insert into Users(idRole,email,password,first\_name,last\_name, phone\_number)

values(@role,@email,@password,@first\_name,@last\_name,@phone\_number)

set @num=@num+1;

end;

exec insertUsers;

create index emailIndex on Users (email);

select email from Users