Министерство общего и профессионального образования

Ростовской области

государственное бюджетное профессиональное образовательное учреждение

Ростовской области

«Волгодонский техникум информационных технологий,

бизнеса и дизайна имени В. В. Самарского»

**ОТЧЕТ**

о производственной практике

**09.02.07 Информационные системы и программирование**

**Квалификация: Программист**

Выполнил:

студентгруппы ***ИСП-19,***

***Аксёнов А.И.***

Принял:

преподаватель ***Ромашов Р.В.***

Оценка:

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

Волгодонск

2022

СОДЕРЖАНИЕ

[Введение 3](#_Toc120901113)

[1 Анализ предметной области 4](#_Toc120901114)

[2 Постановка задачи 5](#_Toc120901115)

[3 Проектирование программного модуля 6](#_Toc120901116)

[3.1 Инфологическая модель 6](#_Toc120901117)

[3.2 Описание входных и выходных данных 7](#_Toc120901118)

[4 Технологии и инструменты разработки 8](#_Toc120901119)

[4.1 Технологии для разработки 8](#_Toc120901120)

[4.2 Инструменты разработки 9](#_Toc120901121)

[5 Разработка программного модуля 11](#_Toc120901122)

[5.1 Пользовательский интерфейс программного модуля 11](#_Toc120901123)

[5.2 Описание методов разработки 12](#_Toc120901124)

[Заключение 15](#_Toc120901125)

[Список использованных источников 16](#_Toc120901126)

# Введение

Целью данной производственной практики будет получение навыка разработки приложения для предприятия ООО «БИЗНЕС РЕШЕНИЯ», приложение будет создано с помощью программного обеспечения установленного на ПК практиканта. Приложение будет сделано по техническому заданию которое было выдано руководителем практики.

# 1 Анализ предметной области

Основным видом деятельности компании ООО «БИЗНЕС РЕШЕНИЯ» является разработка компьютерного программного обеспечения на платформе 1С, консультационные услуги в области разработки компьютерного программного обеспечения на платформе 1С и другие сопутствующие услуги.

# 2 Постановка задачи

Задача производственной практики – создание программного модуля работы с заказами на базе ОС Windows для ООО «БИЗНЕС РЕШЕНИЯ».

Разрабатываемый модуль должен позволять производить различные действия над базой данных содержащей в себе информацию о заказчиках и их заказах с помощью программного модуля установленного на персональный компьютер.

Программный модуль должен быть создан на языках С# и XAML при помощи системы для построения клиентских приложений Windows с возможностями взаимодействия с пользователем. Графический интерфейс программы должен быть дружественный и понятный, сама программа должна быть проста в использовании и не требовать специального образования или прохождения курсов.

В программе должен быть использован шаблон Model-View-ViewModel, Command и дополнительные пакеты Microsoft.EntityFrameworkCore.

# 3 Проектирование программного модуля

## 3.1 Инфологическая модель

Инфологическая модель - это потоки информации, сущности и связи данной области. В такой модели указываются связи между сущностями данной предметной области.

Для выполнения проекта была выбрана СУБД SQL Server. Инфологическая модель (схема данных) БД представлена на рисунке 3.1.

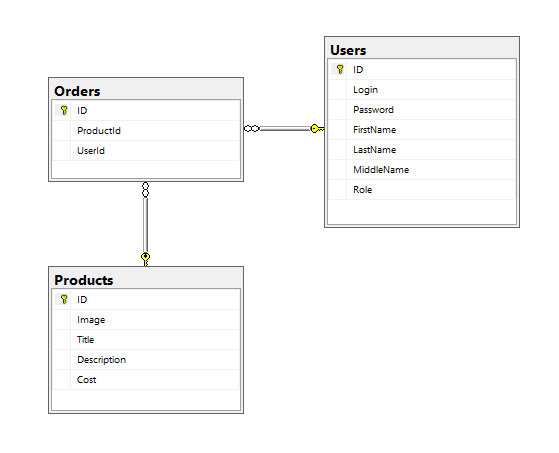


Рисунок 3.1 – Схема данных

Структура таблиц «Продукты», «Пользователи», и «Заказы» базы данных представлена на рисунках 3.2 – 3.4 соответственно.

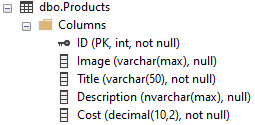


Рисунок 3.2 – Таблица «Продукты»

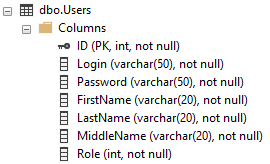


Рисунок 3.3 – Таблица «Пользователи»

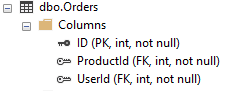


Рисунок 3.4 – Таблица «Заказы»

## 3.2 Описание входных и выходных данных

Входная информация представляет собой данные об продуктах, пользователях и заказах пользователей, которые заносятся в базу данных.

Выходной информацией являются уже обновленные данные об продуктах, пользователях и заказах пользователей. Данные могут быть добавлены, отредактированы и удалены.

# 4 Технологии и инструменты разработки

## 4.1 Технологии для разработки

Для разработки данного приложения был выбран язык C# – объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework и .NET Core. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270 [2].

Windows Presentation Foundation (WPF) — это платформа пользовательского интерфейса для создания клиентских приложений для настольных систем. Платформа разработки WPF поддерживает широкий набор компонентов для разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязки данных, документы и безопасность.

WPF является частью платформы .NET, и если ранее вы создавали приложения в .NET с помощью ASP.NET или Windows Forms, то общий процесс программирования вам уже знаком. WPF использует расширяемый язык разметки для приложений (XAML), чтобы предоставить декларативную модель для программирования приложений [3].

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями [1]. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка [4].

## 4.2 Инструменты разработки

В качестве среды для разработки приложения была выбрана программа Microsoft Visual Studio, линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight. Существует множество сред разработки, но данная была выбрана в ввиду ее удобного графического интерфейса и средств отладки. Microsoft Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня.

Microsoft Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server) [5].

Главные возможности среды разработки Microsoft Visual Studio:

* включены все «интеллектуальные» возможности по редактированию кода;
* есть возможность визуального просмотра будущего приложения;
* сборка проекта работает быстро;
* удобный конструктор интерфейсов;
* удобное и интуитивно понятное логирование проекта;
* указанные цвета и рисунки, использованные в layout'e отображаются на границе в виде небольших превью, которые легко помогают понять какой конкретно ресурс используется;
* среда разработки является технологиями компании Microsoft;
* при выборе ресурса, его содержимое отображается во всплывающих окнах;
* возможность создания одним кликом новых окон и страниц;
* мониторинг используемой памяти;
* возможность подключать сторонние дополнения(плагины);
* возможность добавления новых инструментов.

Рекомендуемые системные требования к компьютеру на которой будет установлена Visual Studio следующие:

* операционная система – Microsoft® Windows® 7/8/10 (32- или 64-bit);
* 2 ГБ ОЗУ. рекомендуется 8 ГБ ОЗУ (минимум 2,5 ГБ при выполнении на виртуальной машине);
* место на жестком диске: до 210 ГБ (минимум 800 МБ) свободного места в зависимости от установленных компонентов; обычно для установки требуется от 20 до 50 ГБ свободного места;
* видеоадаптер с минимальным разрешением 720p (1280 на 720 пикселей); для оптимальной работы Visual Studio рекомендуется разрешение WXGA (1366 на 768 пикселей) или более высокое.

# 5 Разработка программного модуля

## 5.1 Пользовательский интерфейс программного модуля

В окне входа в приложение, представленном на рисунке 5.1, находятся два текстовых поля логин и пароль и кнопка авторизации. Для дальнейшей работы нужно верно указать логин и пароль. Если логин и пароль верны, то откроется основное окно, представленное на рисунке 5.2 с возможностью выбора действий. В главном окне присутствует возможность сортировки, поиска, фильтрации данных, а при нажатии на кнопку заказы откроется окно, со всеми заказами текущего пользователя, показанное на рисунке 5.3.

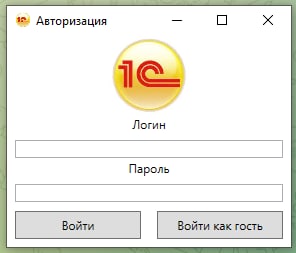


Рисунок 5.1 – Окно входа

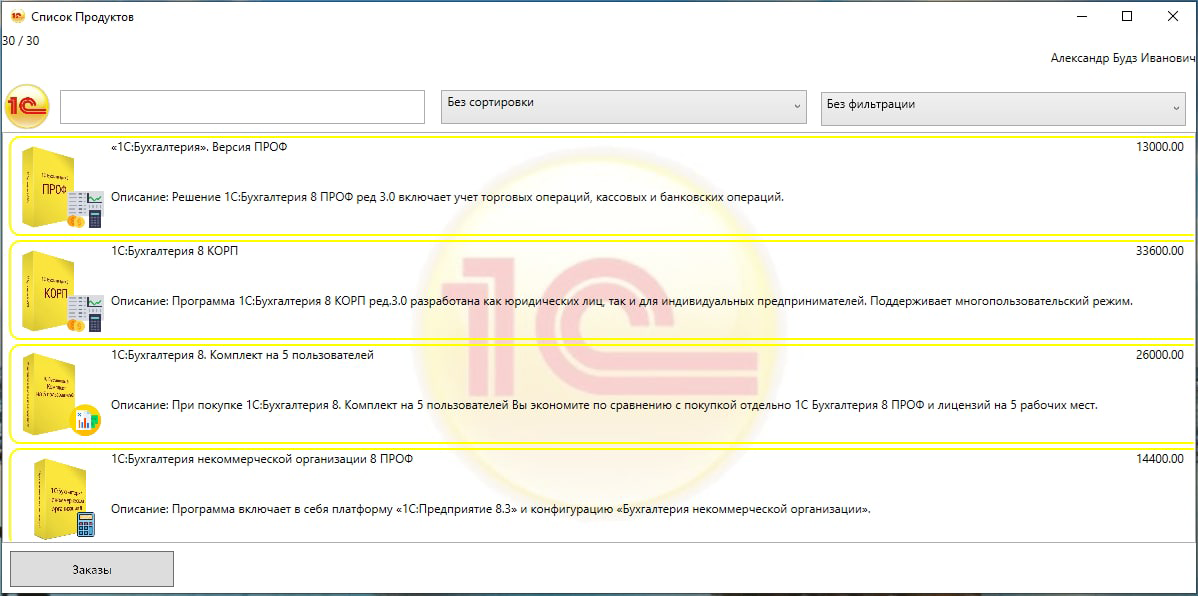


Рисунок 5.2 – Основное окно

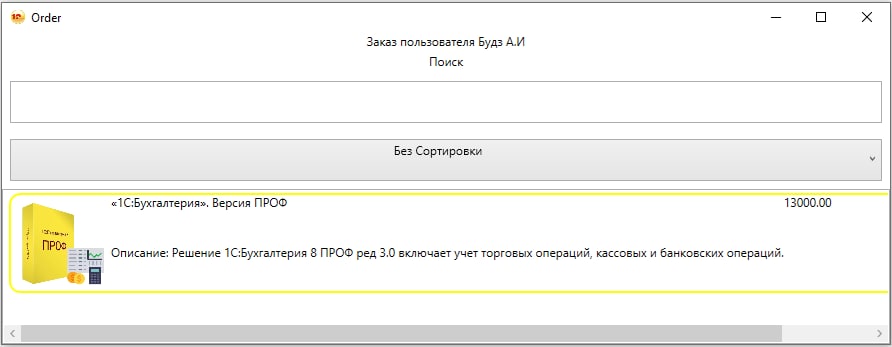


Рисунок 5.3 – Окно просмотра заказов

## 5.2 Описание методов разработки

В шаблоне MVVM есть три основных компонента: модель, представление и модель представления. Каждый из них обслуживает отдельную цель. Кроме понимания обязанностей каждого компонента, важно понимать, как они взаимодействуют друг с другом. На высоком уровне в представлении «известно о модели представления и модели представления» известно о модели, но модель не знает модель представления, а модель представления не знает об этом представлении. Таким образом, модель представления изолирует представление от модели и позволяет модели развиваться независимо от представления.

Ниже приведены преимущества использования шаблона MVVM:

* eсли реализована существующая реализация модели, которая инкапсулирует существующую бизнес-логику, она может быть сложной или рискованной для ее изменения. В этом сценарии модель представления выступает в качестве адаптера для классов модели и позволяет избежать внесения значительных изменений в код модели;
* разработчики могут создавать модульные тесты для модели представления и модели без использования представления. Модульные тесты для модели представления могут выполнять точно те же функциональные возможности, которые используются в представлении;
* пользовательский интерфейс приложения можно переконструировать, не затрагивая код, при условии, что представление полностью реализовано в XAML. Поэтому Новая версия представления должна работать с существующей моделью представления;
* разработчики могут одновременно работать с компонентами в процессе разработки. Дизайнеры могут сосредоточиться на представлении, тогда как разработчики могут работать над моделью представления и компонентами модели [6].

Команды являются привязанными объектами, что позволяет разделить логику и пользовательский интерфейс друг от друга.

Если рассматривать команды более подробно, то они представляют из себя следующее:

* команды представляют собой объекты, реализующие интерфейс ICommand;
* обычно команды связанны с какой либо функцией;
* элементы пользовательского интерфейса привязываются к командам — кода интерфейс активируется пользователем, то выполняется команда — вызывается соответствующая функция;
* команды знают, включены ли они или нет;
* функции могут отключать команды – автоматическое отключение всех пользовательских элементов ассоциированных с ней.

На самом деле существует множество различных применений команд.

Например использование команд для создания асинхронных функций, обеспечивающих логику, которая может быть проверена с/без помощи использования пользовательского интерфейса и др.

Entity Framework Core - представляет собой объектно-ориентированную, легковесную и расширяемую технологию от компании Microsoft для доступа к данным. EF Core является ORM-инструментом (object-relational mapping - отображения данных на реальные объекты). То есть EF Core позволяет работать базами данных, но представляет собой более высокий уровень абстракции: EF Core позволяет абстрагироваться от самой базы данных и ее таблиц и работать с данными независимо от типа хранилища. Если на физическом уровне мы оперируем таблицами, индексами, первичными и внешними ключами, но на концептуальном уровне, который нам предлагает Entity Framework, мы уже работаем с объектами [7].

# Заключение

Разработанный программный модуль работы с заказами для ООО «БИЗНЕС РЕШЕНИЯ» представляет собой простое приложение.

Программа разработана на языке С# в интегрированной среде разработки Microsoft Visual Studio с использованием СУБД SQL Server и шаблона Model-View-ViewModel. Приложение интуитивно понятно даже неопытному пользователю и не затрудняет его работу с оценками, группами, дисциплинами и студентами.

В дальнейшем в приложение будет обновляться и будут добавляться новые функции необходимые для работы с базой данных.

В процессе написания кода для данной программы были приобретены практические навыки в работе в СУБД SQL Server, шаблоном Model-View-ViewModel и языком программирования С#.

# Список использованных источников

Список использованной литературы

1 Моргунов, Е.П. PostgreSQL. Основы языка SQL / Е.П. Моргунов. – Санкт-Петербург : БХВ-Петербург, 2019.

Интернет ресурсы

2 Википедия. Язык программирования C# [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/C\_Sharp

3 Windows Presentation Foundation (WPF) [Электронный ресурс] – Режим доступа: – https://docs.microsoft.com/ru-ru/visualstudio/designers/getting-started-with-wpf?view=vs-2022

4 SQL Server 2019 [Электронный ресурс] – Режим доступа: – https://www.microsoft.com/ru-ru/sql-server/sql-server-2019

5 Microsoft Visual Studio. [Электронный ресурс] – Режим доступа: – https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio

6 Шаблон Model-View-ViewModel. [Электронный ресурс] – Режим доступа: – https://docs.microsoft.com/ru-ru/xamarin/xamarin-forms/enterprise-application-patterns/mvvm

7 Документация по Entity Framework. [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/ru-ru/ef/