МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1-98 01 03 Программное обеспечение информационной

безопасности мобильных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Современные технологии программирования мобильных

систем »

Тема Программное средство «Учет товаров и продаж»

Исполнитель

Студент(ка) 2 курса группы 7 Остапук С.В.

(Ф.И.О.)

Руководитель работы ассистент Панченко О.Л.

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

Курсовой проект защищен с оценкой

Председатель Пацей Н.В.

(подпись)

Минск 2020

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc41512777)

[**1.** **Обзор литературы и аналогов** 4](#_Toc41512778)

[**1.1.** **Постановка задачи** 4](#_Toc41512779)

[**1.1.1.** **Формулировка проблемы** 4](#_Toc41512780)

[**1.1.2.** **Способ решения** 4](#_Toc41512781)

[**1.2**  **Обзор близких аналогов** 5](#_Toc41512783)

[**1.3**  **Обзор литературы** 7](#_Toc41512785)

[**2.** **Анализ требований к программному средству и разработка** 8](#_Toc41512786)

[**2.1.** **Диаграмма вариантов использования** 9](#_Toc41512787)

[**3.** **Разработка архитектуры проекта** 12](#_Toc41512788)

[**3.1.** **Схема базы данных** 12](#_Toc41512788)

[**3.2.** **Проектирование структуры приложения** 13](#_Toc41512789)

[**3.3.** **Диаграмма классов** 16](#_Toc41512790)

[**4 Разработка функциональной модели** 18](#_Toc41512791)

[**4.1 Реализация авторизации и регистрации пользователей** 18](#_Toc41512792)

[**4.2 Реализация функций главной страницы** 19](#_Toc41512793)

[**4.3 Реализация функций страницы склад.** 20](#_Toc41512794)

[**4.4 Реализация отчетных функций.** 22](#_Toc41512795)

[**4.5 Реализация управления списком пользователей.** 23](#_Toc41512796)

[**5. Тестирование** 24](#_Toc41512797)

[**6. Руководство по установке и использованию** 26](#_Toc41512798)

[**6.1 Авторизация** 26](#_Toc41512799)

[**6.2 Использование приложения кассиром** 27](#_Toc41512800)

[**6.3 Использование приложения менеджером** 30](#_Toc41512801)

[**ЗАКЛЮЧЕНИЕ** 32](#_Toc41512802)

[**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ** 33](#_Toc41512803)

# ВВЕДЕНИЕ

На сегодняшний день многое зависит от компьютеров, всевозможные работы и отчеты о их выполнении фиксируются в различных приложениях, поэтому мной и была выбрана данная тема курсового проекта, а именно: “Учет товаров и продаж”.

Учет товаров и продаж какого-либо магазина возможен и без компьютеров, однако это не удобно и, как следствие, тянет за собой огромное количество бумажной работы. Заполнение различных бланков по продажам, ведение учета товаров на складе, все это может вестись и в бумажном виде, но приложение помогло бы сэкономить довольно много времени и денег для небольшого магазина или торгового предприятия.

Учитывая эти факторы приложение может найти свою целевую аудиторию и быть полезным в нынешнем мире, способствуя продвижению молодых индивидуальных предпринимателей.

Целью курсового проекта является: развитие практических навыков вёрстки приложений на базе WPF (Windows Presentation Foundation) с использованием языка разметки — XAML (eXtensible Application Markup Language), и языка программирования — С#; разработка IU и UX дизайна, умение работать в ограниченном промежутке времени и в пределах заданной темы.

Разработка приложения происходит в программной среде Microsoft Visual Studio 2017.

Данные для работы будут храниться в базе данных. Её создание происходит с помощью SQLite.

Для работы с самим SQLite используется Entity Framework, который представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Он позволяет работать напрямую с объектами, данные для которых хранятся в таблицах базы данных.

Курсовой проект включает в себя разделы: введение, содержание, обзор технический методов и программных средств разработки приложения, последовательность вёрстки приложения, составление его структуры, описание хода работы программного средства с копиями экрана пользователя, заключение, список использованной литературы.

Целью данного курсового проекта является разработка приложения учета товаров на складе и их продаж. Для достижения поставленной цели необходимо:

1. Реализовать:

* авторизацию пользователей;
* возможность администратора добавлять/удалять кассиров;
* возможность ведения учета склада;
* возможность выполнять поисковые запросы;
* возможность получения различных отчетов;
* возмжность продажи товара с печатью чека

1. **Обзор литературы и аналогов**
   1. **Постановка задачи**
      1. **Формулировка проблемы**

Бумажный учет не удобен и не пользуется популярностью. Людям часто нужно принимать новые поступления товаров на склад, следить за количеством товаров на складе и так далее.

Все эти проблемы формируют следующие задачи:

* возможность слежки за количеством товаров на складе(учет);
* возможность записи поступивших товаров;
* возможность реализации товаров в магазине.
  + 1. **Способ решения**

Для реализации перечисленных основных задач, в первую очередь, была представлена модель товара. Это подразумевало выделение основных информационных полей для товара и его учета. Также, была разработана модель для хранения информации о зарегистрированных кассирах.

После создания полей в модели товаров и пользователей был продуман абстрактный функционал, который позволял решить минимальные задачи проекта. Основным функционалом, который создавался у каждого пользователя являлись методы взаимодействия базы данных и приложения для отображения верной информации для каждого пользователя приложения.

Следующим шагом, имея в наличии абстрактный объект, метод взаимодействия базы данных и объектов, методы взаимодействия с объектами, являлся интерфейс приложения. Так как приложение подразумевает владение пользователями, которые не знакомы с управлением оконными приложениями, подразумевался интуитивный интерфейс, в котором было бы сложно запутаться. В то же время сохранялись правила дизайна и сетки. Основываясь на двух областях (функционал и свойства объекта) интерфейс был разделен на области(страницы).

Таким образом, был продуман приблизительный план решения задачи при создании приложения на самом раннем этапе.

* + 1. **Требования к программному обеспечению**

Основными требованиями к программному обеспечению (далее ПО) являлись:

* разделение ролей для разных пользователей;
* защита доступа к любым данным;
* функциональный интерфейс;
* многостраничность интерфейса.

Под разделением доступа и его защите подразумевалось создание на базе приложения нескольких пользователей. Так же это означает, что пользователь может добавить аккаунт и работать только с тем функционалом, который ему дозволен программно.

Многостраничность являлась частью функциональности приложения. Основной задачей такого подхода являлось возможное сохранение множества объектов для быстрого доступа к ним по средствам простого клика. Так же все действия над объектами происходят на отдельных страницах.

**1.2 Обзор близких аналогов**

Из открытых источников (Интернет) мною было найдено приложение GS “Учет: Магазин” (Рисунок 1.1)

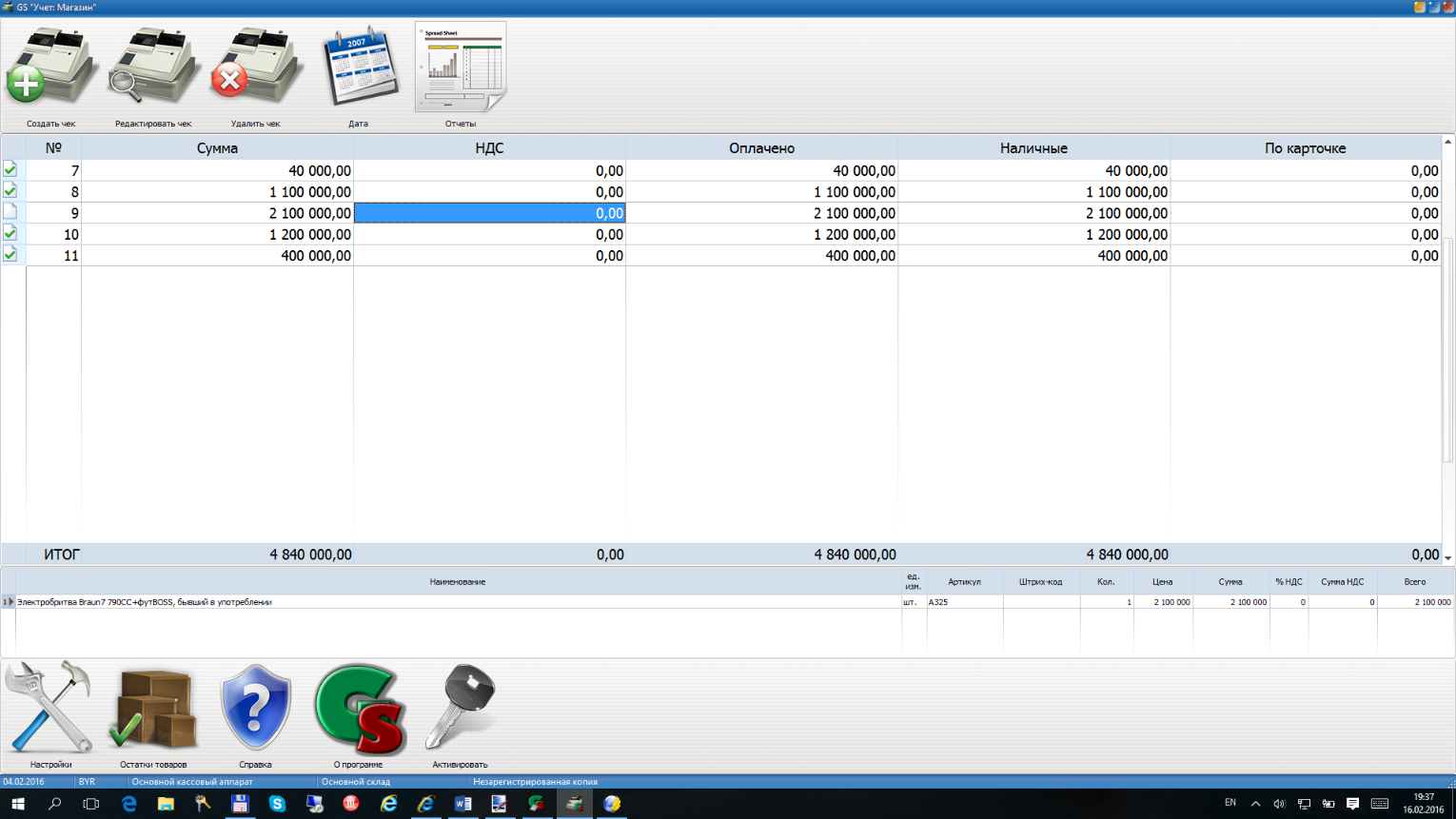


Рисунок 1.1 – приложение GS “Учет: Магазин”

Данное приложение является своего рода приложением учета товаров и продаж. Его главная задача состоит в сборе информации о продажах и наличии/отсутствии товара на складе.

Основной функционал:

* создание чеков и регистрация;
* удаление чеков;
* поиск по дате;
* просмотр остатков товара на складе;

Главный недостаток – отсутствие связи товаров и чеков, т.е. чек хранит в себе лишь сумму продажи, но не количество проданного товара и его единичную стоимость. Отсутствие связи между товарами и складом, делает вкладку «Остаток товаров» бесполезной, ибо возможно добавление различных чеков без ссылки на проданный товар, а просто введя в поле «Сумма» придуманного числа.

Аналогичным образом (Интернет) мною был найден вебсайт moysklad.ru (Рисунок 1.2)

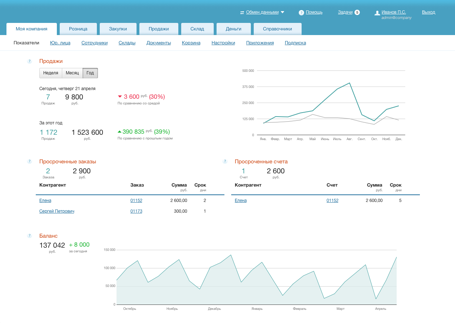


Рисунок 1.2 – вебсайт moysklad.ru

Данный сайт является веб-приложением учета товаров. Его главная задача состоит в сборе информации о наличии/отсутствии товара на складе.

Основной функционал:

* учет товаров;
* статистика по приему/реализации товаров;
* различного рода поиски;
* просмотр остатков товара на складе;
* печать документов;

Главный недостаток – отсутствие полной информации о продажах. Хотя такая информация и имеется, но она является не полной. Отсутствует дата выполнения продажи, цена и еще несколько пунктов. Так же данное приложения является полностью платным, имеется лишь пробный период длиной в несколько суток, в котором ограниченны возможности складского учета.

**1.3 Обзор литературы**

При создании курсового проекта использовалась литература в виде веб-порталов по тематике.

Metanit. Основным источником знаний при работе с C#, Entity Framework 6 (далее EF) были стати данного ресурса. Отличается от большинства простатой подачи информации, краткого теоретического сопровождения и наличием примеров.

Youtube – ресурс для просмотра видео, где есть многочисленное количество разнообразных уроков

SQL.ru . Данный проект предоставляет помощь при работе с базами данных SQL. Большинство запросов производились на тему конструирования запросов с участием EF.

Professor Web. Проект схож с ресурсом Metanit, однако подача материала отличается в сторону более полных определение, примеров и углубленного рассмотрения заданий на примерах.

WPF Tutorial. Данный сайт предоставляет полной информацию по возможностям работы с WPF.

1. **Анализ требований к программному средству и разработка**

**функциональных требований**

Имеются 3 группы пользователей:

* менеджер-админ;
* менеджер;
* кассир

У каждой из этих групп существуют свои права доступа к программе. Кассиру после входа в программу будет предложено меню, где он может просмотреть свою статистику, начать реализовывать. Менеджеру будет доступен список кассиров, генерирование разного рода отчётов, возможность изменять штат кассиров и возможность реализации товаров. И последняя роль – это менеджер-админ (администратор). Ему предоставляется возможности такого же рода как и менеджеру, но с возможностью редактирования полного штата сотрудников (менеджеры и кассиры). Полный функционал показан на рисунке 2.2.

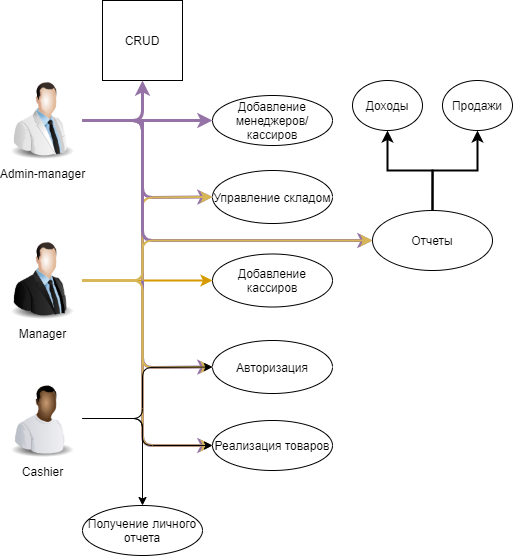


Рисунок 2.1 – возможности ролей.

* 1. **Диаграмма вариантов использования**

В диаграмме использования проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых прецедентов. При этом актером или действующим лицом называется любая сущность, взаимодействующая с системой извне. Другими словами, каждый вариант использования определяет некоторый набор действий, совершаемый системой при диалоге с актером. При этом ничего не говорится о том, каким образом будет реализовано взаимодействие актеров с системой. Диаграмма вариантов использования для пользователя менеджера-админа приведена на рисунке 2.2.

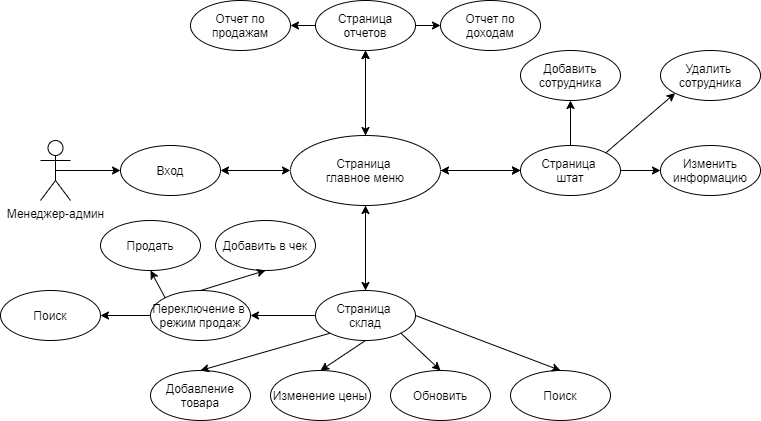


Рисунок 2.2. – Диаграмма вариантов использования для пользователя администратора

На данной диаграмме вариантов использования можно увидеть действия, которые может совершать пользователь-администратор. Открыв приложение, пользователь может авторизоваться и попасть на главную страницу, где он может искать информацию, добавлять новую, редактировать и удалять старую, с помощью различных пунктов меню. Пользователь администратор, так же может переключиться в режим продаж и реализовывать товар, может изменять настройки других пользователей и вести складской учет.

Диаграмма вариантов использования для пользователя с правами кассира на рисунке 2.3.

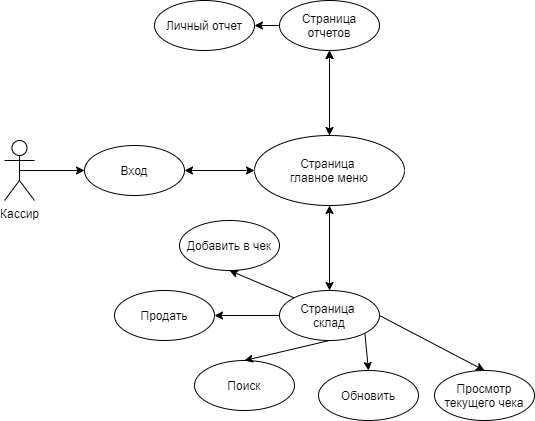


Рисунок 2.3. – Диаграмма вариантов использования для пользователя с правами кассира

На данной диаграмме вариантов использования можно увидеть действия, которые может совершать пользователь с правами кассира. Пользователь может авторизоваться и попасть на главную страницу, где он может получить личный отчет по продажам. Кассир так же может перейти на склад где ему доступны функции продажи, просмотра текущего чека, добавления товара в чек и поиск товаров.

1. **Разработка архитектуры проекта**
   1. **Схема базы данных**

При разработке приложения в первую очередь понадобилось создать базу данных так, чтобы она удовлетворяла потребностям приложения и не была избыточной.

Для реализации задуманной цели была придумана база данных, состоящая из 4 таблиц, ее структура представлена на рисунке 3.1.

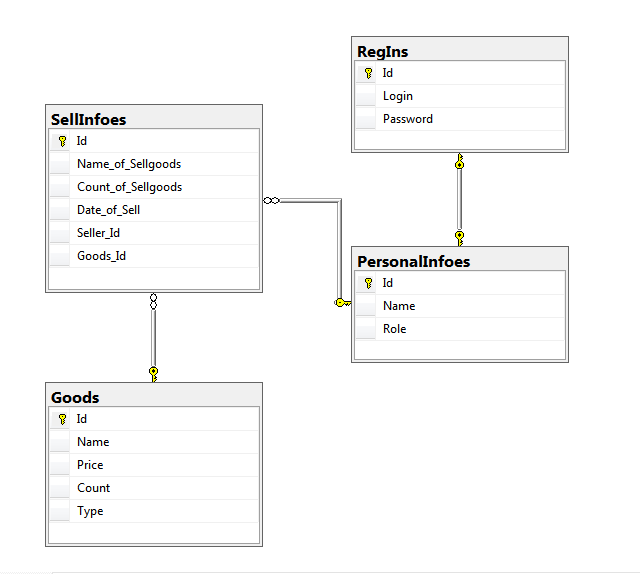


Рисунок 3.1. – Схема базы данных

Таблица Regins представляет собой таблицу, которая хранит информацию о зарегистрированных пользователях и состоит из 3 столбцов:

* Id – ключевое поле таблицы, которое содержит уникальный индекс для каждого пользователя.
* Login – логин пользователя
* Password – зашифрованный пароль пользователя

Таблица PersonalInfo представляет собой таблицу, которая хранит информацию о зарегистрированных пользователях, мною было решено использовать 2 отдельные таблицы для хранения информации о пользователя:

* Id – ключевое поле таблицы, которое содержит уникальный индекс для каждого пользователя, формируется на основе Regins.Id;
* Name – поле которое содержит ФИО пользователя;
* Role – поле которое содержит роль пользователя в использовании приложения.

Таблица Goods представляет собой таблицу, которая хранит информацию о складских товарах и состоит из 6 полей:

* Id – ключевое поле таблицы, которое содержит уникальный индекс для каждого товара;
* Name – поле которое содержит название товара;
* Count – поле содержащее количество товара на складе;
* Price – цена товара;
* Type – тип товара, т.е. указывает является товар штучным или продается по килограмму.

Таблица SellInfo представляет собой таблицу, которая хранит информацию о реализации товаров и содержит в себе 6 полей:

* Id – ключевое поле таблицы, которое содержит уникальный индекс.
* Name\_of\_Sellgoods – поле которое содержит название товара;
* Count\_of\_Sellgoods – поле содержит количество реализованного товара;
* Date\_of\_Sell – дата реализации товара;
* Seller\_Id – Id пользователя, реализовавшего товар;
* Goods\_Id – Id реализованного товара
  1. **Проектирование структуры приложения**

Приложение представлено одним проектом «Ugoods» и представленно на рисунке 3.2.

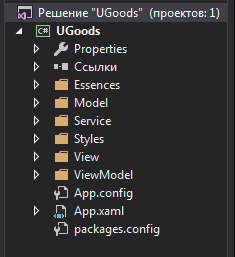


Рисунок 3.2. – Структура приложения

В папке «Essences» содержаться сущности для вывода работы с Entity Framework.

Содержание папки «Essences» представлено на рисунке 3.3

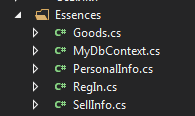


Рисунок 3.3 – Содержимое папки «Essences»

В папке «Styles» содержаться различные картинки для визуального отображения изделия при выборе изделия, а также иконки для добавления, удаления, обновления информации и подтверждения изменения, и стиль для основного окна.

Содержание папки «Styles» представлено на рисунке 3.4

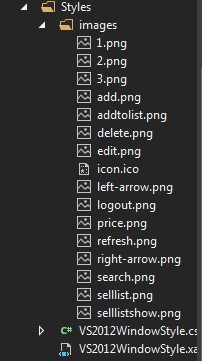


Рисунок 3.4 - Содержимое папки «Styles»

В папке Model содержаться непосредственно классы, которые представляют собой элементы для вывода ListView.

Содержимое папки Model представлено на рисунке 3.5.

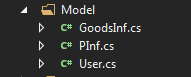


Рисунок 3.5. – Содержимое папки Model

В папке Service представлен класс-сервис для получения различного рода отчетов в асинхронном режиме.

Содержимое папки Service представлено на рисунке 3.6.



Рисунок 3.6. – Содержимое папки Service

В папке View представлены различного рода страницы и окна, которые используются в приложении, для отображения информации (реализован паттерн Model-View-ViewModel).

Содержимое папки View представлено на рисунке 3.7.

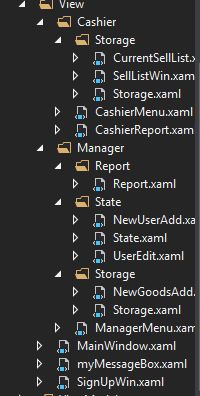


Рисунок 3.7. – Содержимое папки View

В папке ViewModel классы, реализовывающие бизнес-логику моего приложения.

Содержимое папки ViewModel представлено на рисунке 3.8.

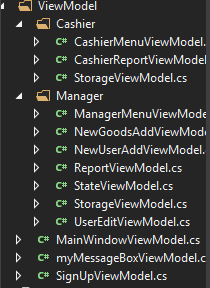


Рисунок 3.8. – Содержимое папки ViewModel

* 1. **Диаграмма классов**

Диаграмма классов служит для представления статической структуры мо­дели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру (поля, методы) и типы отношений (наследование, реализация интерфейсов). На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

Классы, которые находятся в папке Essences созданы при помощи технологии Entity Framework и представляют собой классы, которые являются сущностями для работы с базой данных, а именно SQLite.

Класс User, находящийся в папке Model, является Singleton`ом, который помогает сохранят информацию о авторизованном пользователе и предотвращает ее потерю.

Класс MainWindowViewModel хранит в себе функцию хеширования паролей, для предотвращения хранения открытых паролей в базе данных и приложении.

Остальные классы с припиской ViewModel содержат в себе бизнес логику приложения, т.е. функции, использующиеся в приложении, поля, свойства и т.д.

Схема диаграммы классов, которая показывает нам взаимосвязь между классами нашего приложения представлена на рисунке 3.10.



Рисунок 3.10 – Схема диаграммы классов

1. **Разработка функциональной модели**
   1. **Реализация авторизации и регистрации пользователей**

Войдя в приложение, пользователь если у него есть аккаунт авторизоваться и войти в приложение. При установке приложения, будет создана новая база данных и логин и пароль для владельца будут стандартными – Admin Admin. Реализация создания базы данных при первом входе показана на рисунке 4.1.

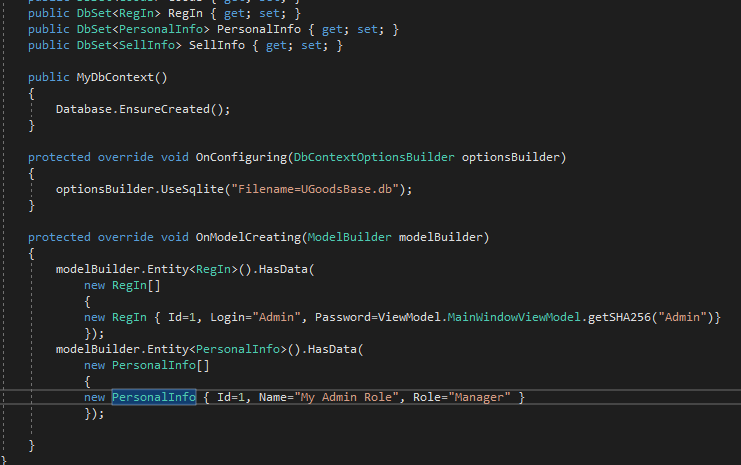


Рисунок 4.1 – реализация создания бд.

В приведенном фрагменте кода мы использовали Entity Framework Core и его обработку сущностей. При первом входе в приложение будет вызван конструктор MyDbContext, который в свою очередь не увидит локальной базы данных на компьютере и создаст новую по заданному шаблону. OnModelCreating создает записи в уже сформированной таблице/таблицах базы данных.

После этого пользователь может пройти авторизацию. Код авторизации представлен на рисунке 4.2

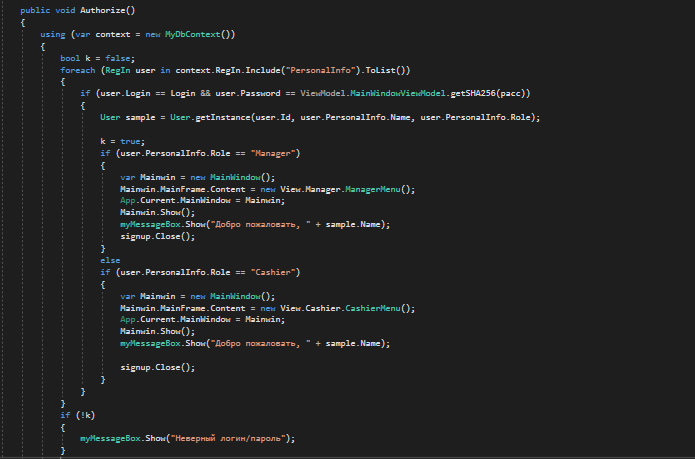


Рисунок 4.2. – Авторизация пользователя

Здесь пользователь вводи свой логин и пароль, данные сверяются с данными в базе данных и если все успешно, то мы присваиваем полям Singletone id пользователя, его имя и логин.

* 1. **Реализация функций главной страницы**

Главная страница имеет один и тот же шаблон для различных пользователей на этой странице имеются кнопки, по нажатию которых пользователь перейдет в нужный ему раздел. Рисунок 4.3. Также имеется кнопка выхода из аккаунта, которая закроет текущее окно и вернет нас на этап авторизации.

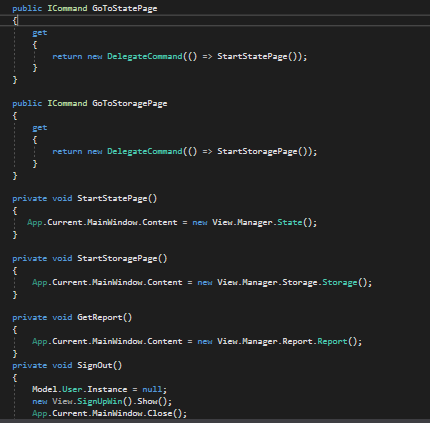


Рисунок 4.3 – реализация главного меню.

* 1. **Реализация функций страницы склад.**

Страница склад представляет из себя ListView с содержанием имеющихся товаров на складе. Для полной работоспособности склада должны быть реализованы некоторые функции, а именно: добавление товара (рис. 4.4), различного рода поиски товара в базе данных (рис 4.5), обновление списка товаров (рис 4.6).

Добавление товара происходит в отдельном окне, используется валидация для корректного ввода всех полей и предотвращения ошибок. Кнопка сохранения результата и добавления нового товара в базу данных скрыта до заполнения всех полей и отсутствия ошибок.

Поиск реализован написанием LINQ to Entities запросов и регулярных выражений, которые способствуют успешному поиску по имени товара. Сортировки реализованы через LINQ to Entities.

Обновление списка товаров происходит с помощью обычного обращения к базе данных и полного заполнения списка товаров на складе по новой.

Так же на этой странице имеется функция изменения цены и перехода в режим продаж. Последний в свою очередь становится доступным просто с помощью открытия окна склада кассира для менеджера. При нажатии кнопки продажи печатается чек.

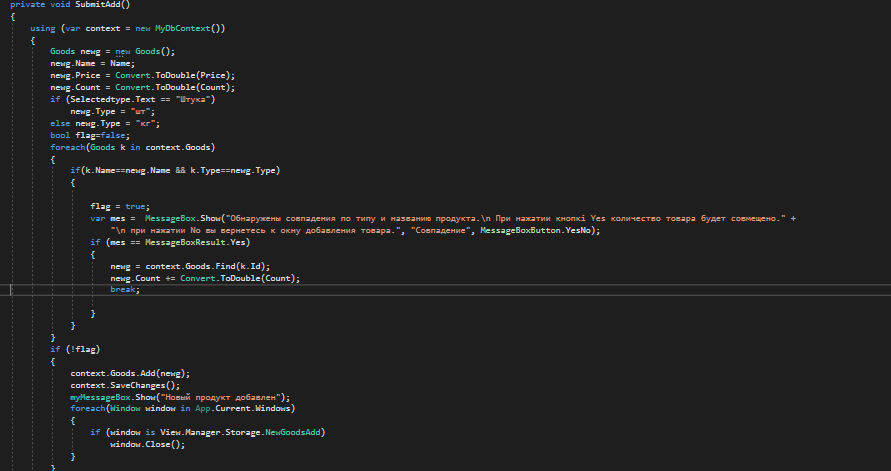


Рисунок 4.4 – реализация добавления товара.

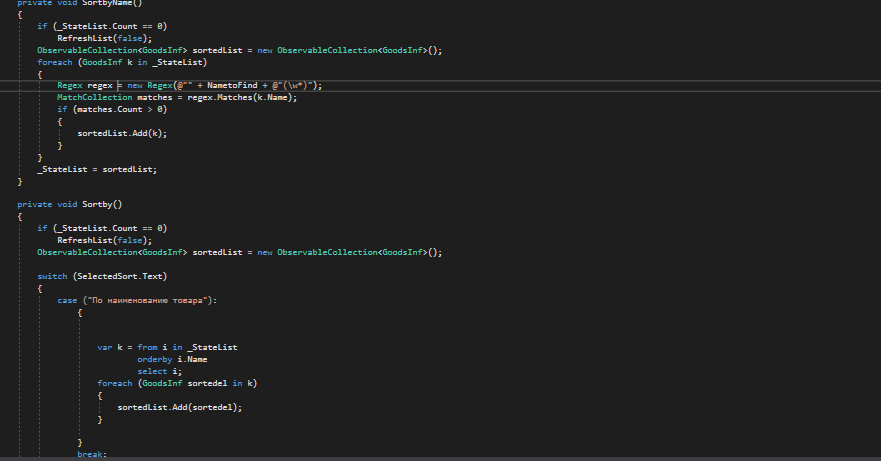


Рисунок 4.5 – поиск товара.

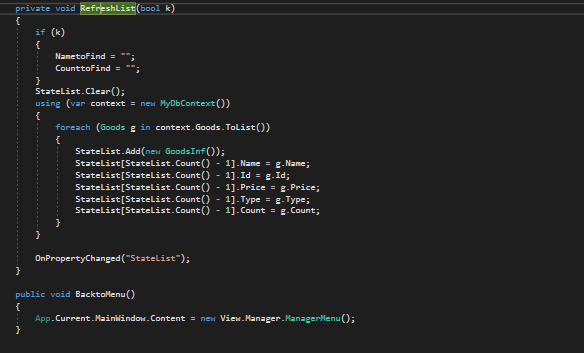


Рисунок 4.5 – обновление списка.

* 1. **Реализация отчетных функций.**

На странице отчеты доступны 3 вида отчета, а именно: столбчатые диаграммы для кассира (просмотр личных продаж за последнее время) и два другие отчета генерируются с помощью заполнения документов Word. Реализация столбчатых диаграмм показана на рисунке 4.6, а вывод в документы на рисунке 4.7

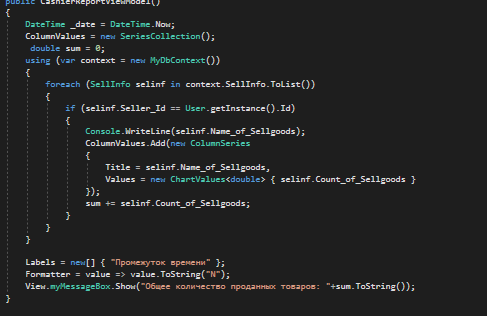


Рисунок 4.6 – реализация столбчатых диаграмм.

Стоит отметить, что для реализации диаграмм был скачан пакет из Visual Studio NuGet, под названием lvc. Данный пакет позволяет строить множество различных графиков и диаграмм.

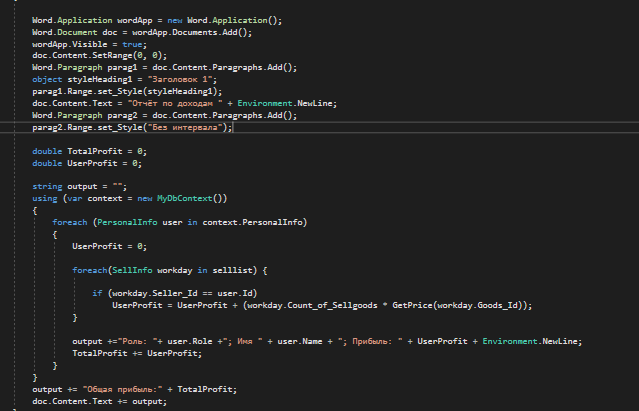


Рисунок 4.7 – генерация отчетов в Microsoft Word.

* 1. **Реализация управления списком пользователей.**

Как и страница товары, данная страница имеет в себе ListView со списком сотрудников, которые пользуются данным приложением. Менеджер-администратор может управлять полным списком сотрудников, в свою очередь обычный менеджер может просматривать и изменять/удалять/добавлять информацию исключительно о кассирах. Функции удаления и редактирования информации аналогичны в написании, поэтому включу в отчет только удаление – рисунок 4.8.

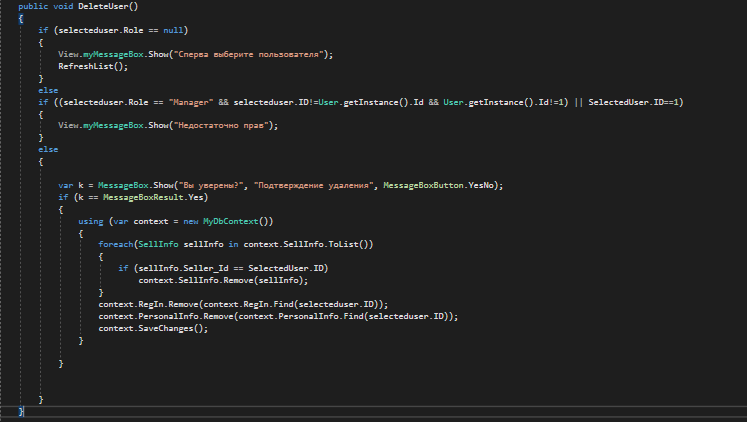


Рисунок 4.8 – Удаление пользователя.

1. **Тестирование**

В курсовом проекте использована обработка исключений и валидация данных, что не позволяет пользователю ввести некорректные данные. Когда происходит исключение, оно обрабатывается и сообщение выводится в отдельное окошко, что дает пользователю понять об некорректности вводимых данных, либо же при ошибке текст будет подсвечиваться красной рамкой и при наведении будет видна информации об ошибке.

При неправильном вводе пароля пользователь получает сообщение, показанное на рисунке 5.1

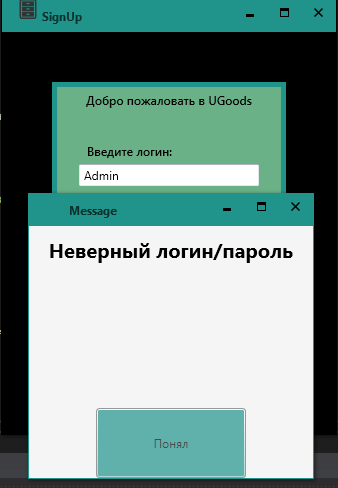


Рисунок 5.1 – Сообщение вводе пароля/логина.

При регистрации нового пользователя менеджер может ввести различные данные неправильно или ввести логин, который уже есть в базе данных и ему выдаст исключение. Пример сообщения о том что такие данные уже хранятся в базе данных приведены на рисунке 5.2.

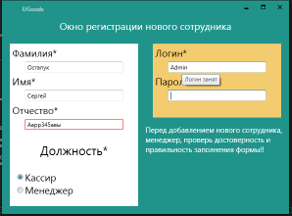


Рисунок 5.2 – Сообщение валидации

При добавлении нового товара, нам нужно проверить, чтобы данные были правильные, и в случае некорректности вводимых данных вывести ошибку. Пример приведен на рисунке 5.3.

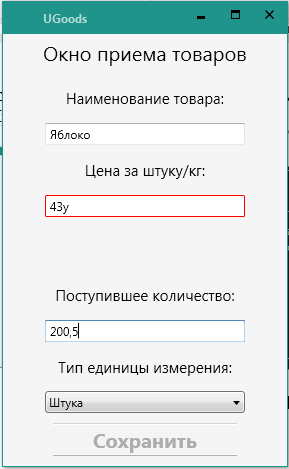
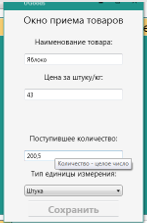


Рисунок 5.3. – Оповещение пользователя об некорректности вводимых данных

.

1. **Руководство по установке и использованию**

Запустить установщик приложения и установить на диск. На рисунке 6.1 представлен установщик. По умолчанию на рабочем столе будет создан ярлык для быстрого запуска приложения.



Рисунок 6.1. – приложение для установки программы

* 1. **Авторизация**

После запуска приложения пользователь видит главное окно приложения, на котором он может авторизоваться и выполнять свою работу. Окно представлено на рисунке 6.2.

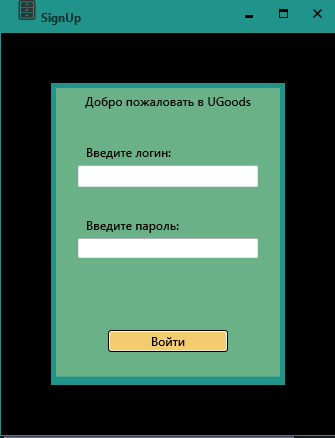


Рисунок 6.2. – Окно входа в аккаунт

Если пользователь не имеет аккаунта, он не получит доступа к приложению.

* 1. **Использование приложения кассиром**

При подтверждении пароля пользователь переходит на главную страницу приложения. Главная страница представлена на рисунке 6.3.

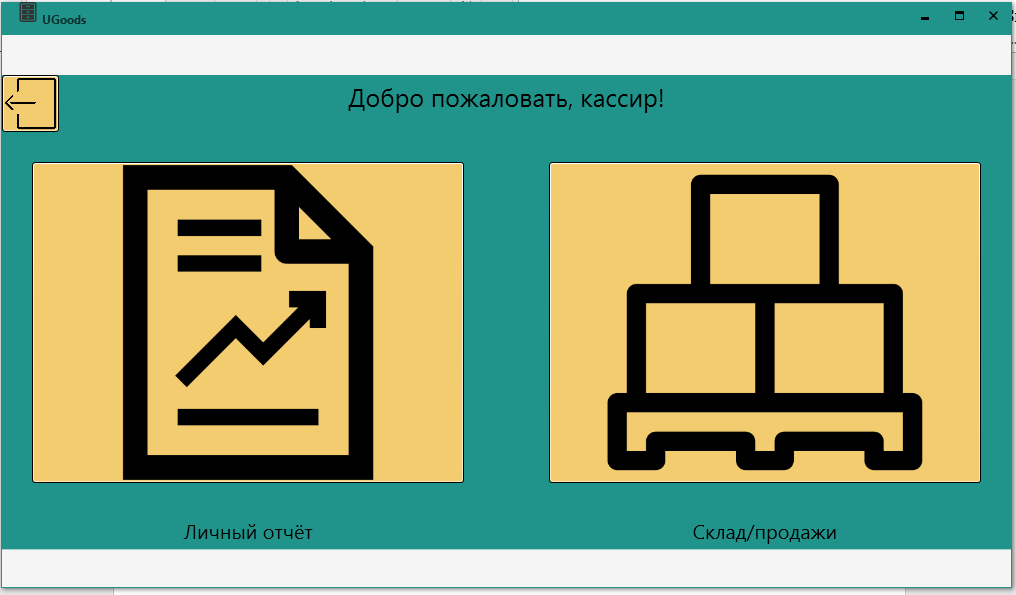


Рисунок 6.3 – Главная страница

При нажатии на кнопку склад/продажи пользователь переходит в окно, которое показано на рисунке 6.5

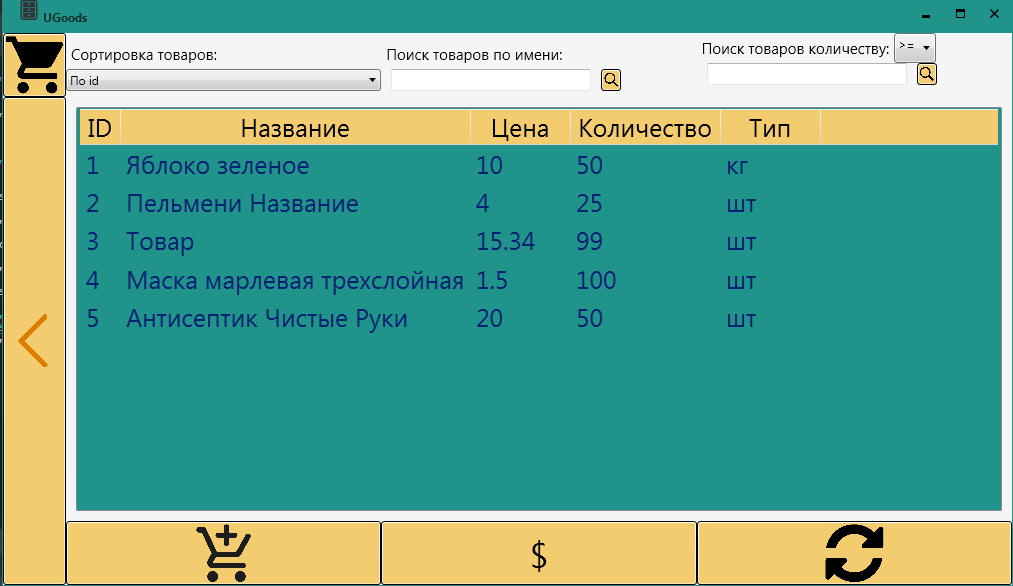


Рисунок 6.4 – Страница склад/продажи

На данной странице кассир может продавать начать продавать товар.

Процесс продажи товара рисунки 6.5-6.7.

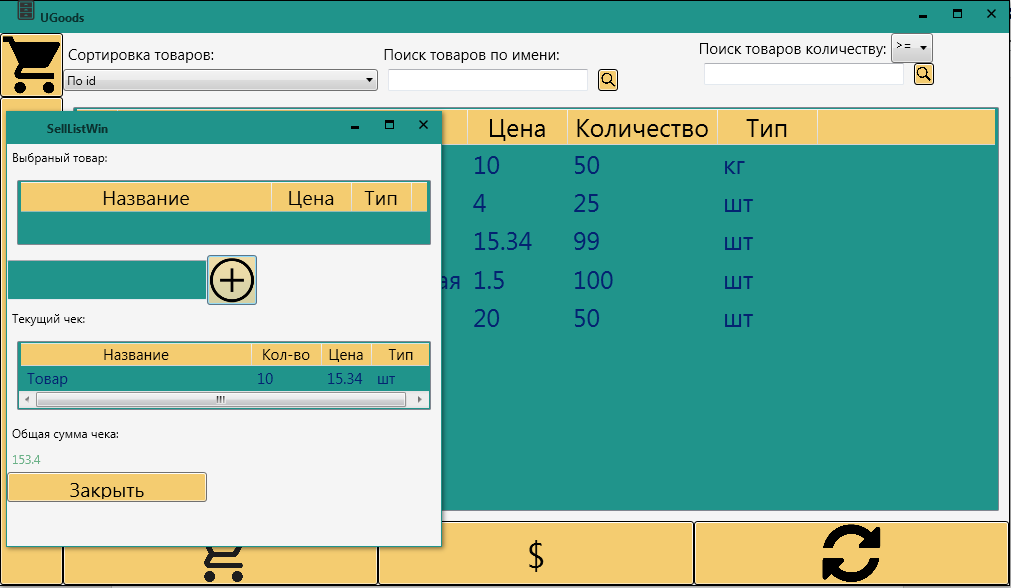


Рисунок 6.5 - Добавление товара в текущий чек.

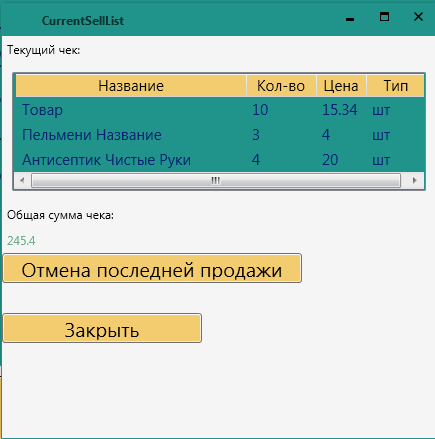


Рисунок 6.6 - Просмотр текущего чека.

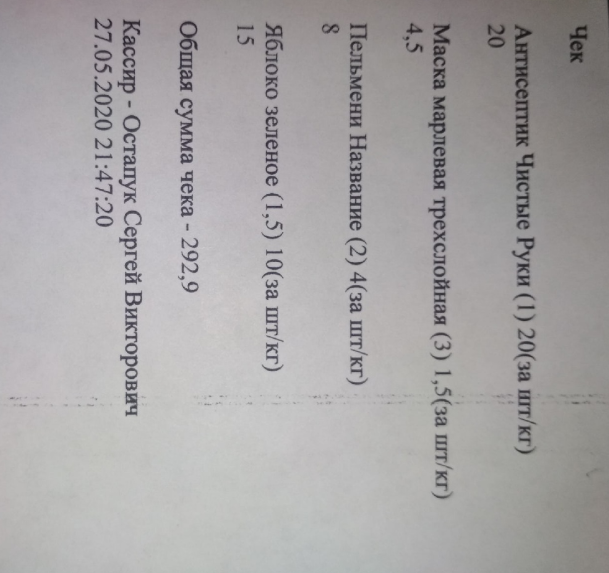


Рисунок 6.7 - Результат печати чека.

Последняя функция доступная кассиру – это просмотр личной статистики о продажах за последнее время – Рисунок 6.8.

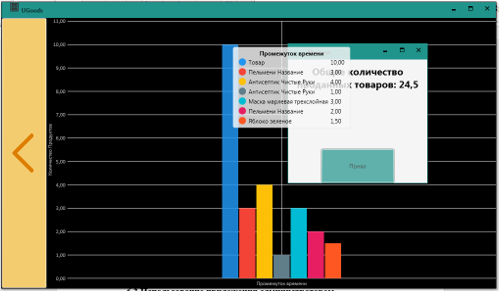


Рисунок 6.8 - Отчет кассира.

* 1. **Использование приложения менеджером**

Если при авторизации пользователя, данные введены верно и статус акка­унта соответствует администратору то пользователь попадает в меню менеджера рисунок 6.9.

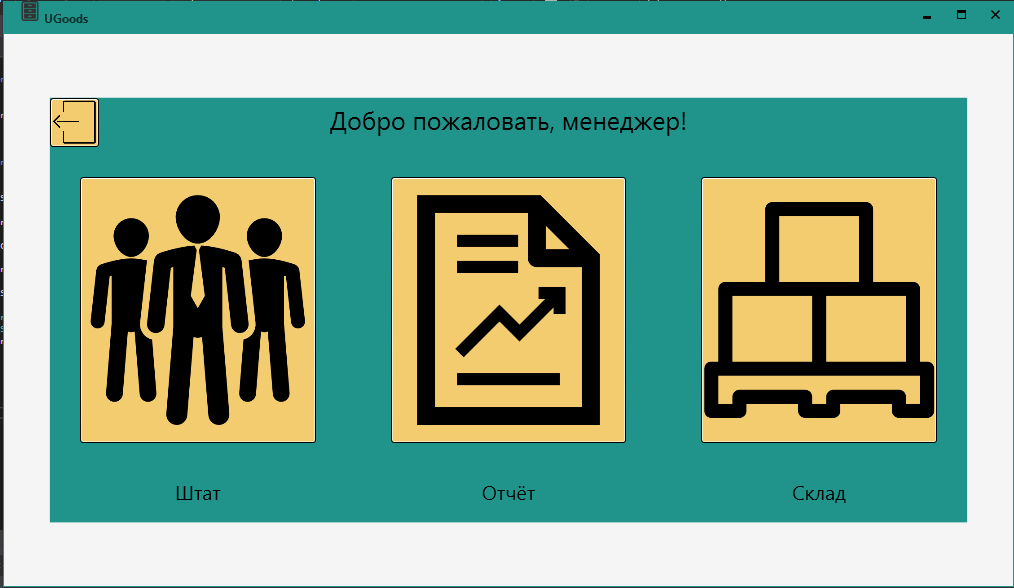


Рисунок 6.9 - Меню менеджера.

Менеджеру доступны другие виды отчетов, а именно отчет по продажам и отчет по доходам по всему штату сотрудников, демонстрация такого отчета будет приведена ниже на рисунке 6.10-6.11.

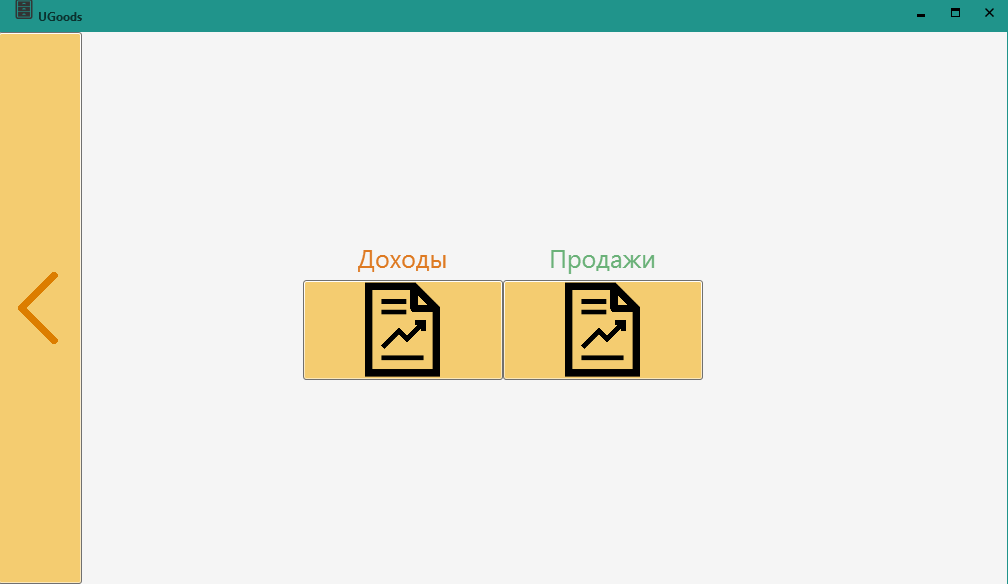


Рисунок 6.10 - Меню Отчет.

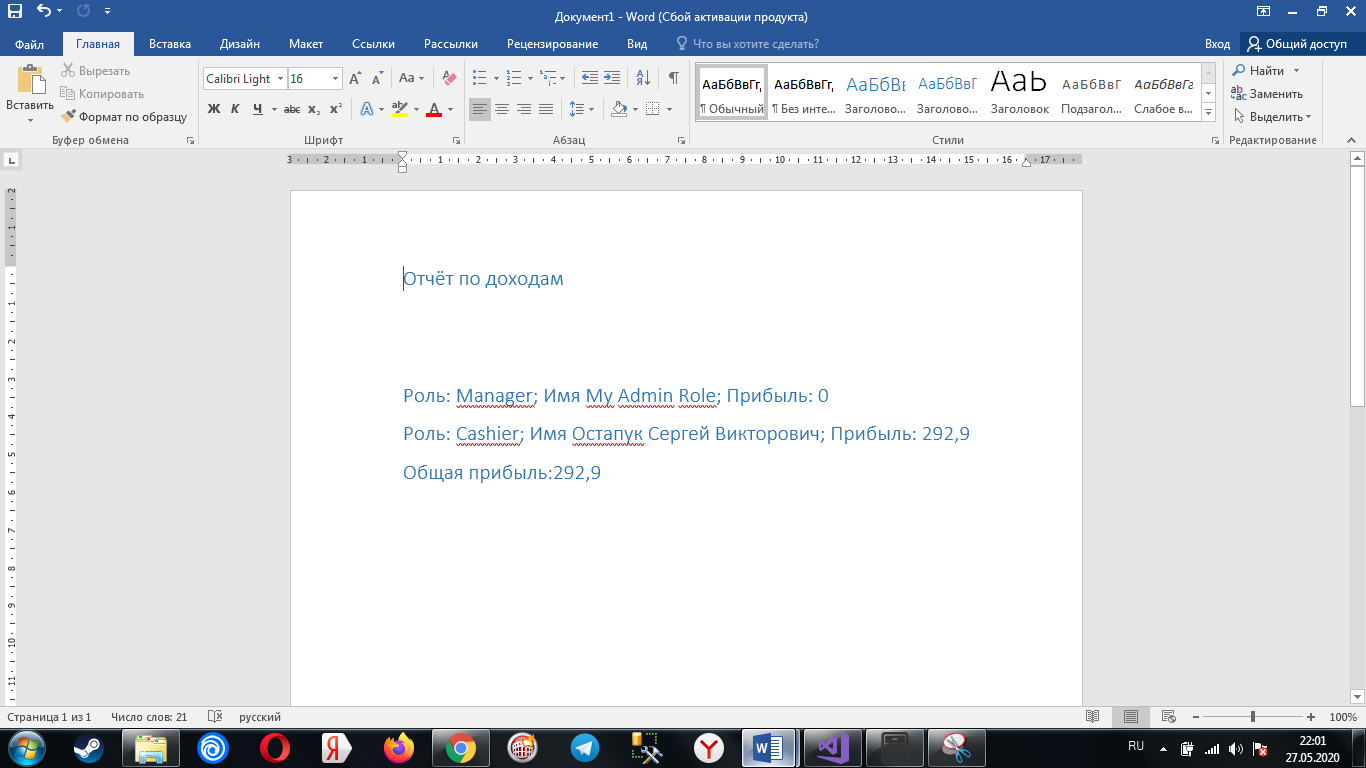


Рисунок 6.11 – Результат генерации отчета.

Также для менеджера доступна страница просмотра штата сотрудиников и информации о них с возможностью редактирования, добавления и удаления новых сотрудников и информации о них – Рисунок 6.12.

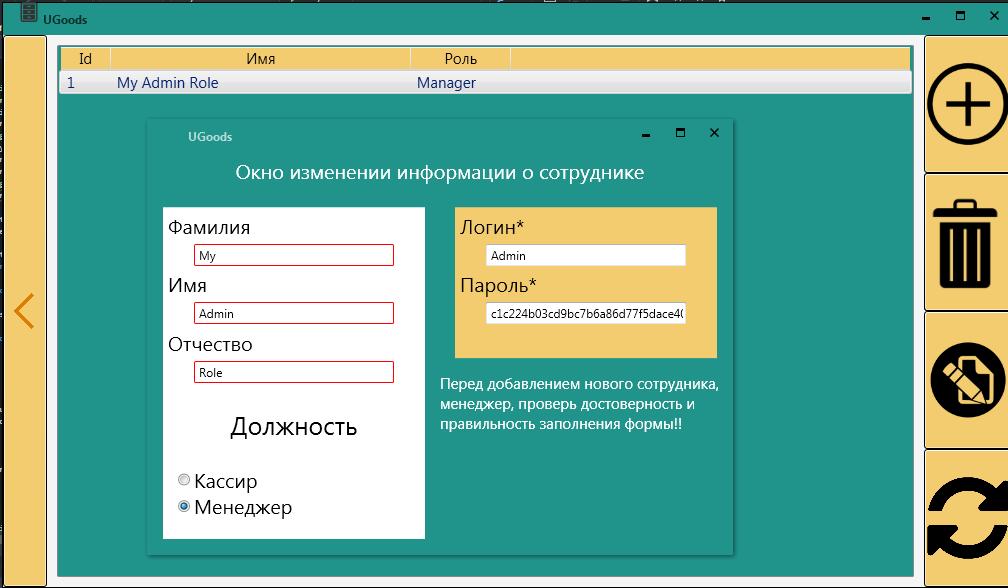


Рисунок 6.12 – Штат.

**ЗАКЛЮЧЕНИЕ**

В итоге выполнения данного курсового проекта было создано приложение для ПК, предназначенное для учета товаров и продаж. Были реализованы основные задачи, которые требовалось реализовать.

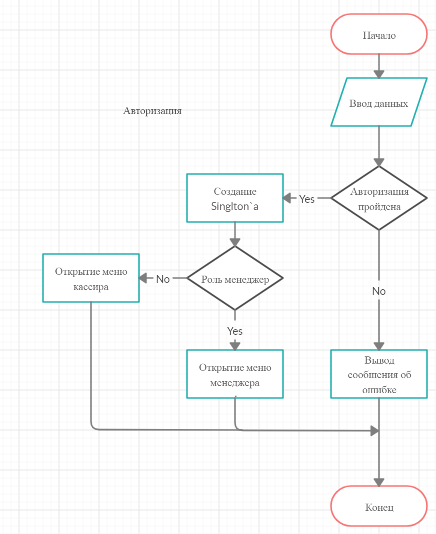
За период выполнения курсового проекта были изучены принципы проектирования десктопных приложений с использованием WPF, Entity Framework , был изучен новый язык разметки XAML. Так же была спроектирована база данных с помощью «SQLite» для хранения информации.

Данная работа мотивировала изучать различные новые инструменты для разработки продукции в сфере информационных технологий, что повысило навыки программирования.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. MSDN сеть разработчиков в Microsoft [Электронный ресурс] – <http://msdn.microsoft.com/library/rus/>. Дата доступа: 10.05.2018
2. Пацей, Н.В. Курс лекций по языку программирования С# / Н.В. Пацей. – Минск: БГТУ, 2016. – 175 с.
3. Generic Repository | Entity Framework 6 [Электронный ресурс] – <https://metanit.com/sharp/entityframework/3.13.php>. Дата доступа: 13.10.2016
4. ProfessorWeb .NET & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru> Дата доступа: 13.04.2018
5. Professor WEB [Электронный ресурс]. – Подстановка шрифтов. – Режим доступа: <https://professorweb.ru/my/WPF/UI_WPF/level6/6_4.php>.
6. Metanit [Электронный ресурс]. – Руководство по WPF. – Режим доступа: <https://metanit.com/sharp/wpf/>

**ПРИЛОЖЕНИЕ А**



**ПРИЛОЖЕНИЕ Б**

