**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет имени Гагарина Ю.А.»**

**(СГТУ имени Гагарина Ю.А.)**

**ПРОФЕССИОНАЛЬНО-ПЕДАГОГИЧЕСКИЙ КОЛЛЕДЖ**

09.02.07 Информационные системы и программирование

**КУРСОВОЙ ПРОЕКТ**

МДК 02.01 Технология разработки программного обеспечения

Тема:**Разработка информационной системы для магазина бытовой техники**

Студента 4 курса

Группы ИСП-941

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Свириденко Д.Д.

(подпись)

Руководитель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Добрынина И.А.

(подпись)

Работа защищена с оценкой

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_202\_\_\_г.

г. Саратов 2022

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc96917166)

[1 Анализ предметной области и разработка требований к системе 3](#_Toc96917167)

[1.1 Исследование предметной области. Формирование функциональных требований к системе 3](#_Toc96917168)

[1.2 Проектирование программного обеспечения для магазина бытовой техники 3](#_Toc96917169)

[1.3 Проектирование и разработка базы данных 5](#_Toc96917170)

[1.3.1 Проектирование БД на концептуальном уровне 5](#_Toc96917171)

[1.3.2 Логическое проектирование БД 11](#_Toc96917172)

[1.3.3 Физическое проектирование БД 12](#_Toc96917173)

[2 Разработка и тестирование программного обеспечения для аттестации персонала 16](#_Toc96917174)

[2.1 Разработка программных модулей 16](#_Toc96917175)

[2.2 Тестирование и отладка 21](#_Toc96917176)

[2.3 Соответствие кода стандартам кодирования 25](#_Toc96917177)

[Заключение 27](#_Toc96917178)

[Список использованных источников 28](#_Toc96917179)

[Приложение А Техническое задание 29](#_Toc96917180)

Программная часть:

1. Скрипт создания БД и ее объектов

2. Код программы

# ВВЕДЕНИЕ

1. **Анализ предметной области и разработка требований к системе**

Информационные технологии играют важную роль в современном мире. Их применение дает новые возможности для развития и оптимизации бизнеса, способствуют для расширения рынков сбыта, производительности труда, эффективному использованию ресурсов, повышению качества управления бизнесом и предоставления услуг.

Сегодня, острая конкуренция развивается во многих отраслях рынка, одни компании пытаются соответствовать ценам и характеристиками продукции других, покупатели зачастую получают возможность выбирать среди массы конкурирующих товаров, не различающихся по качеству.

В такой ситуации любая компания, ставящая во главу угла вопросы об удовлетворении актуальных потребностей покупателей, предлагающая клиенту высокий уровень обслуживания, имеет несомненное преимущество, позволяющее создавать долговременные отношения. Не имея единой выверенной базы с данными о клиентах, компания теряет существенные средства из-за недостаточно эффективной работы отделов продаж, маркетинга и обслуживания.

Особое значение приобретает деятельность отдела продаж. Во многих компаниях она нередко выглядит как «черный ящик» на входе – некое множество контактов с потенциальными клиентами, а на выходе тот или иной финансовый результат. В сложившейся ситуации контроль за ходом работы с клиентами и оценка действий каждого из менеджеров, как минимум, затруднены, а главное – руководитель теряется в ответе: где скрыты решения для повышения объема продаж.

На данный момент сейчас повсеместно идет внедрение и развитие различных информационных технологий на предприятии. В компаниях, занимающихся коммерческой деятельностью, главной задачей является совершенствование взаимодействия между клиентов и компанией.

Решением этой задачи занимаются в рамках электронной коммерции, которая в свою очередь является одним из направлений электронного бизнеса. Одним из инструментов электронной коммерции является внедрение информационной системы с базой данных, хранящей в себе информацию об ассортименте товаров, различных их характеристиках, менеджерах и персонале компании, а самое главное – клиентах организации.

При такой системе облегчается работа продавцов-консультантов, ведь клиент самостоятельно может при посещении магазина выбрать интересующий товар, изучить его характеристики и сделать заказ, а за сотрудниками магазина остается работа за выдачей товара по подготовленному товарному чеку.

## 1.1 Исследование предметной области. Формирование функциональных требований к системе

Информационная система предназначена для обеспечения доступа клиентам целевой компании к информации о товарах и возможности совершить заказ с использованием рабочей станции, находящегося в торговом зале. Для сотрудников компании информационная система должна обеспечить доступ к ассортименту магазина, возможности его редактировать, добавлять, изменять и удалять позиции товаров.

Авторизованный клиент должен иметь возможность сортировки и удобного поиска необходимого товара в различных категориях, добавлять интересующий товар в пользовательскую корзину и совершать заказ. Программа должна генерировать шаблонный товарный чек, содержащий информацию о приобретенных товарах.

В магазине бытовой техники предусмотрена скидочная система для каждого отдельного клиента компании. Для реализации данной системы должна быть предусмотрена регистрация и авторизация пользователей в программе, а также учет потраченных в магазине средств для каждого из клиентов.

При этом, благодаря удобной для клиентов скидочной системе, компания собирает базу своих активных покупателей, благодаря чему может улучшить показатели продаж, ведя маркетинговую компанию.

## 1.2 Проектирование программного обеспечения для магазина бытовой техники

Функциональная схема или схема данных - схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств. Для изображения функциональных схем используют специальные обозначения, установленные стандартом.

Функциональная схема программного продукта представлена на рисунке 1.

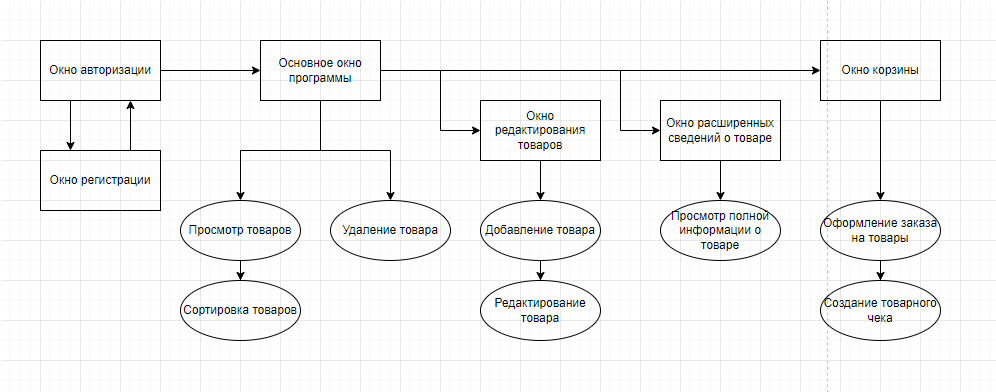


Рисунок 1 – Функциональная схема программного продукта

Диаграмма вариантов использования — это диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей. Данная диаграмма представлена на рисунке 2.

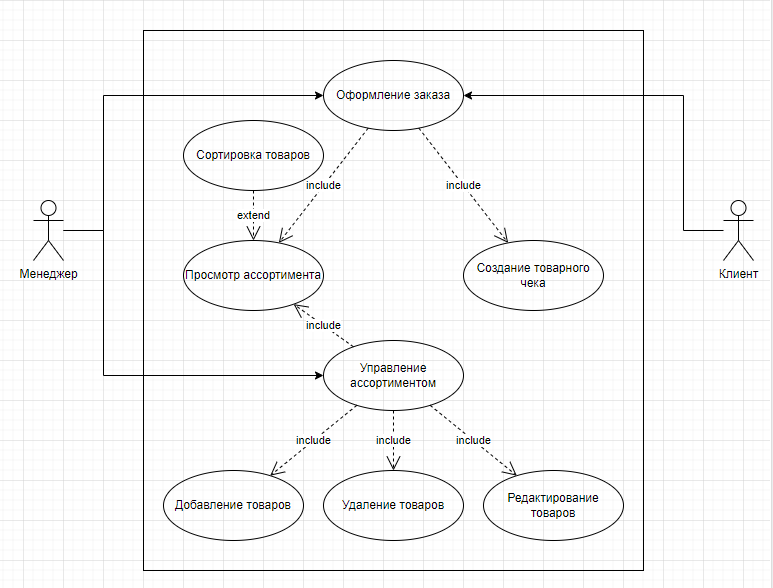


Рисунок 2 – Диаграмма прецедентов

## 1.3 Проектирование и разработка базы данных

### 1.3.1 Проектирование БД на концептуальном уровне

Предметной областью называется фрагмент реальности, который описывается или моделируется с помощью БД и ее приложений. В предметной области выделяются информационные объекты – идентифицируемые объекты реального мира, процессы, системы, понятия и т.д., сведения о которых хранятся в БД.

Магазин — предприятие розничной торговли, оборудованном для продаж товаров и оказания услуг. Для продуктивной работы необходимо контролировать кадровый состав каждого магазина, представленного в сети компании.

Список сущностей базы данных представлен в таблице 1.

Таблица 1 – Список сущностей

|  |  |  |
| --- | --- | --- |
| № | Название | Назначение |
| 1 | Клиенты | Содержит информацию о клиентах организации |
| 2 | Менеджеры | Содержит информацию о менеджерах организации |
| 3 | Товары | Содержит информацию о товарном ассортименте |
| 4 | Категории | Содержит информацию о категориях товаров |
| 5 | Производители | Содержит информацию о производителях товаров |
| 6 | Элементы покупки | Содержит информацию о записях в товарных чеках |
| 7 | Покупки | Содержит информацию о товарных чеках |
| 8 | Уровни скидок | Содержит информацию о уровнях скидок для клиентов |
| 9 | Цвета | Содержит информацию о цветах товара |
| 10 | Расположения морозильной камеры | Содержит информацию о расположениях морозильных камер холодильников |
| 11 | Классы энергоэффективности | Содержит информацию о классах энергоэффективности товаров |
| 12 | Типы подсветки | Содержит информацию о типах подсветки телевизоров |
| 13 | Операционные системы | Содержит информацию об операционных системах телевизоров |
| 14 | Диагонали экранов | Содержит информацию о диагоналях экранов телевизоров |
| 15 | Разрешения экранов | Содержит информацию о разрешениях экранов телевизоров |

Список атрибутов таблицы Клиенты представлен в таблице 2.

Таблица 2 – Список атрибутов таблицы Клиенты

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID клиента | Уникальный идентификатор клиента |
|  | Номер телефона | Номер телефона клиента |
|  | Пароль | Пароль клиента |
|  | Фамилия | Фамилия клиента |
|  | Имя | Имя клиента |
|  | Отчество | Отчество клиента |
|  | Сумма покупок | Сумма всех покупок клиента для расчета скидки |
| ВК | Уровень скидки | ID текущего уровня скидки клиента |
|  | Последняя покупка | Дата последней покупки |
|  | Дата регистрации | Дата регистрации клиента |

Список атрибутов таблицы Менеджеры представлен в таблице 3.

Таблица 3 – Список атрибутов таблицы Менеджеры

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID менеджера | Уникальный идентификатор менеджера |
|  | ФИО менеджера | Фамилия, имя и отчество менеджера |
|  | Логин | Логин менеджера |
|  | Пароль | Пароль менеджера |

Список атрибутов таблицы Категории представлен в таблице 4.

Таблица 4 – Список атрибутов таблицы Категории

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID категории | Уникальный идентификатор категории |
|  | Название категории | Название категории товаров |
|  | Описание категории | Подробное описание категории товаров |

Список атрибутов таблицы Производители представлен в таблице 5.

Таблица 5 – Список атрибутов таблицы Производители

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID производителя | Уникальный идентификатор производителя |
|  | Название компании | Название компании-производителя |
|  | Страна происхождения | Страна происхождения компании |
|  | Номер технической поддержки | Номер технической поддержки компании |

Список атрибутов таблицы Уровни скидок представлен в таблице 6.

Таблица 6 – Список атрибутов таблицы Уровни скидок

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID уровня | Уникальный идентификатор уровня скидки |
|  | Название уровня | Маркетинговое название уровня скидки |
|  | Сумма для уровня | Сумма покупок клиента, необходимая для получения уровня скидки |
|  | Процент скидки | Процент скидки для данного уровня |

Список атрибутов таблицы Цвета представлен в таблице 7.

Таблица 7 – Список атрибутов таблицы Цвета

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID цвета | Уникальный идентификатор цвета |
|  | Название цвета | Название цвета товара |

Список атрибутов таблицы Расположения морозильной камеры представлен в таблице 8.

Таблица 8 – Список атрибутов таблицы Компоновки морозильной камеры

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID компоновки | Уникальный идентификатор компоновки |
|  | Название компоновки | Название компоновки морозильной камеры |

Список атрибутов таблицы Классы энергоэффективности представлен в таблице 9.

Таблица 9 – Список атрибутов таблицы Классы энергоэффективности

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
|  | ID класса | Уникальный идентификатор класса энергоэффективности |
|  | Название класса | Название класса энергоэффективности |

Список атрибутов таблицы Типы подсветки представлен в таблице 10.

Таблица 10 – Список атрибутов таблицы Типы подсветки

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID типа подсветки | Уникальный идентификатор типа подсветки телевизора |
|  | Название типа подсветки | Маркетинговое название типа подсветки телевизора |

Список атрибутов таблицы Операционные системы представлен в таблице 11.

Таблица 11 – Список атрибутов таблицы Операционные системы

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID операционной системы | Уникальный идентификатор операционной системы телевизора |
|  | Название операционной системы | Название операционной системы телевизора |

Список атрибутов таблицы Диагонали экранов представлен в таблице 12.

Таблица 12 – Список атрибутов таблицы Диагонали экранов

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID диагонали | Уникальный идентификатор диагонали телевизора |
|  | Диагональ в дюймах | Численная диагональ телевизора в дюймах |
|  | Диагональ в сантиметрах | Численная диагональ телевизора в сантиметрах |

Список атрибутов таблицы Разрешения экранов представлен в таблице 13.

Таблица 13 – Список атрибутов таблицы Разрешения экранов

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID разрешения | Уникальный идентификатор разрешения экрана телевизора |
|  | Название разрешение | Маркетинговое название разрешения экрана телевизора |
|  | Наименование разрешения | Численное значение разрешения экрана телевизора |

Список атрибутов таблицы Товары представлен в таблице 14.

Таблица 14 – Список атрибутов таблицы Товары

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID товара | Уникальный идентификатор товара |
|  | Модель | Модель товара |
| ВК | ID категории | ID категории товара |
|  | Цена | Цена товара |
|  | Описание | Подробное маркетинговое описание товара |
| ВК | ID производителя | ID производителя товара |
| ВК | ID цвета | ID цвета товара |
|  | Ширина | Ширина товара в сантиметрах |
|  | Высота | Высота товара в сантиметрах |
|  | Глубина | Глубина товара в сантиметрах |
|  | Гарантия | Гарантия на товар в месяцах |
|  | Потребление энергии | Потребление энергии товара в кВт/ч |
| ВК | ID класса энергоэфф. | ID класса энергоэффективности товара |
|  | Изображение | Путь к файлу изображения товара |
| ВК | ID компоновки морозильной камеры | ID компоновки морозильной камеры холодильника |
|  | Зона свежести | Наличие или отсутствие зоны свежести |
|  | Инверторный компрессор | Наличие или отсутствие инверторного компрессора |
|  | Объем холодильной камеры | Объем холодильной камеры в см3 |
|  | Объем морозильной камеры | Объем морозильной камеры в см3 |
|  | Температурный дисплей | Наличие или отсутствие температурного дисплея |
| ВК | ID типа подсветки | ID типа подсветки телевизора |
| ВК | ID диагонали экрана | ID диагонали экрана телевизора |
| ВК | ID разрешения экрана | ID разрешения экрана телевизора |
|  | Поддержка SmartTV | Наличие или отсутствие SmartTV |
| ВК | ID операционной системы | ID операционной системы телевизора |
|  | Поддержка Bluetooth | Наличие или отсутствие Bluetooth модуля |
|  | Поддержка HDR | Наличие или отсутствие поддержки HDR |
|  | Расход воды | Расход воды в л/ч |
|  | Количество программ | Количество программ товара |
|  | Количество комплектов посуды | Количество комплектов посуды, помещающихся в посудомоечную машину |
|  | Максимальная загрузка | Максимальная загрузка в кг |
|  | Диапазон температур | Диапазон температур для стирки белья |
|  | Прямой привод | Наличие или отсутствие прямого привода |
|  | Максимальное число оборотов в минуту | Максимальное число оборотов барабана стиральной машины в минуту |
|  | Внутренний объем | Внутренний объем товара |
|  | Наличие гриля | Наличие или отсутствие гриля |

Список атрибутов таблицы Покупки представлен в таблице 15.

Таблица 15 – Список атрибутов таблицы Покупки

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID покупки | Уникальный идентификатор совершенной покупки |
|  | Дата покупки | Дата совершения покупки |
| ВК | ID клиента | ID клиента, совершившего покупку |

Список атрибутов таблицы Элементы покупки представлен в таблице 16.

Таблица 16 – Список атрибутов таблицы Элементы покупки

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК | ID покупки | Уникальный идентификатор записи в товарном чеке |
| ПК | ID товара | ID товара в данной записи |
|  | Число товаров | Число приобретенных товаров данного типа |

Выявленные связи представлены в таблице 17.

Таблица 17 – Список связей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название связи | Сущности, участвующие в связи | Назначение |
| 1 | 1:М | Manufacturers - Products | У одного производителя может быть множество товаров |
| 2 | 1:М | Categories - Products | К одной категории может относиться множество товаров |
| 3 | 1:М | OperatingSystems - Products | Одна операционная может быть установлена на множество товаров |
| 4 | 1:М | ScreenSizes - Products | Типовые диагонали экранов могут быть у множества товаров |
| 5 | 1:М | Colors - Products | Один цвет может быть у множества товаров |
| 6 | 1:М | ScreenResolutions - Products | Типовые разрешения экранов могут быть у множества товаров |
| 7 | 1:М | BacklightTypes - Products | Один тип подсветки может быть у множества товаров |
| 8 | 1:М | FreezerLocations - Products | Одно расположение морозильной камеры может быть у множества товаров |
| 9 | 1:М | EnergyClasses - Products | Один класс энергоэффективности может быть у множества товаров |
| 10 | 1:М | Products – PurchaseItems | Один товар может быть куплен множество раз |
| 11 | 1:М | Purchases – PurchaseItems | В одном товарном чеке может быть множество |
| 12 | 1:М | Clients – Purchases | Один клиент может совершить множество покупок |
| 13 | 1:М | DiscountLevels - Clients | К одному уровню скидки может относиться множество клиентов |

### 1.3.2 Логическое проектирование БД

Для проектирования БД на логическом уровне используем приложение Draw.io

Draw.io — это бесплатное онлайн-приложение для создания диаграмм для рабочих процессов. Draw.io поможет в создании UML-моделей, графиков и прочих форм, в режиме реального времени.

Построение ER-диаграммы для данной предметной области, представлено на рисунке 3.

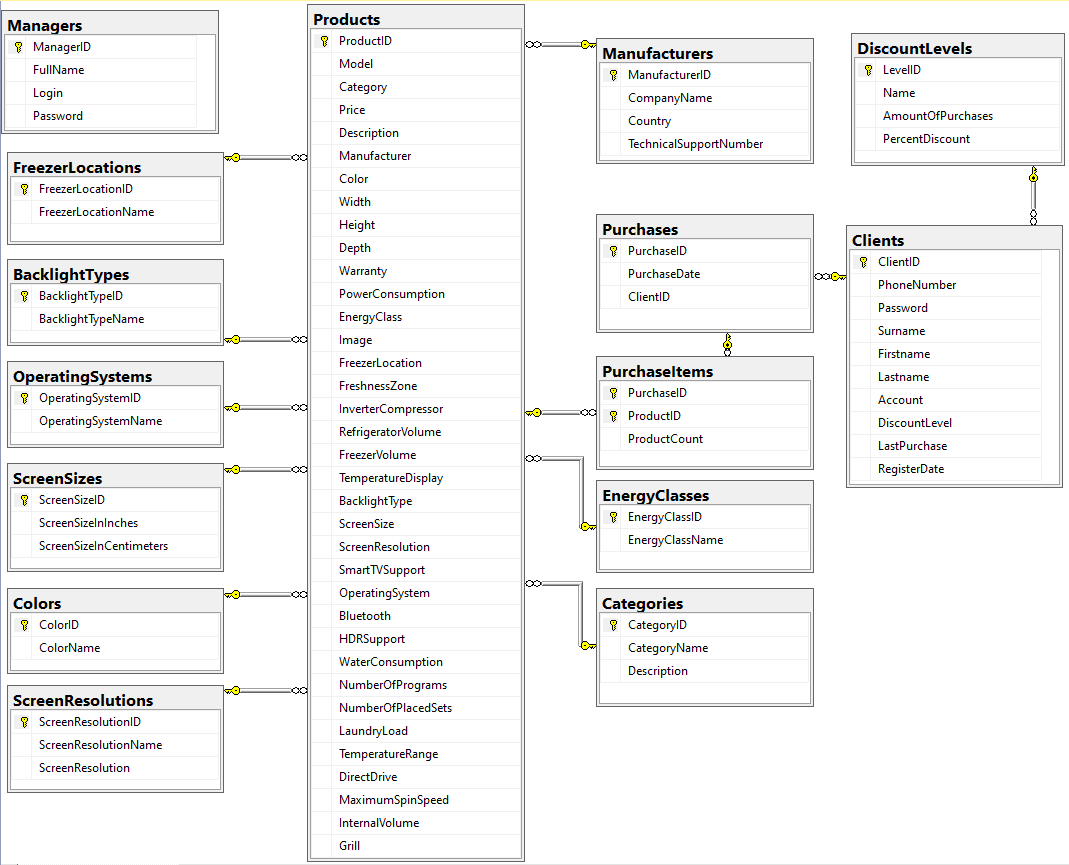


Рисунок 3 – ER-диаграмма

### 1.3.3 Физическое проектирование БД

Расширенная информация о атрибутах представлена в таблицах 18-.

Таблица 18 –Список атрибутов таблицы «Clients»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID Клиента | ClientID | INT | Да | ПК |
| 2 | Номер телефона | PhoneNumber | NVARCHAR(25) | Да |  |
| 3 | Пароль | Password | NVARCHAR(16) | Да |  |
| 4 | Фамилия | Surname | NVARCHAR(25) | Да |  |
| 5 | Имя | Firstname | NVARCHAR(25) | Да |  |
| 6 | Отчество | Lastname | NVARCHAR(25) |  |  |
| 7 | Сумма покупок | Account | Money | Да |  |
| 8 | Уровень скидки | DiscountLevel | TINYINT | Да |  |
| 9 | Последняя покупка | LastPurchase | DATETIME2(0) |  |  |
| 10 | Дата регистрации | RegisterDate | DATETIME2(0) | Да |  |

Таблица 19 –Список атрибутов таблицы «Managers»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID менеджера | ManagerID | INT | Да | ПК |
| 2 | ФИО менеджера | FullName | NVARCHAR(30) | Да |  |
| 3 | Лоин | Login | NVARCHAR(16) | Да |  |
| 4 | Пароль | Password | NVARCHAR(16) | Да |  |

Таблица 20 –Список атрибутов таблицы «Categories»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID категории | CategoryID | SMALLINT | Да | ПК |
| 2 | Название категории | CategoryName | NVARCHAR(50) | Да |  |
| 3 | Описание категории | Description | NVARCHAR(500) |  |  |

Таблица 21 –Список атрибутов таблицы «Manufacturers»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID производителя | Manufacturer ID | SMALLINT | Да | ПК |
| 2 | Название компании | Company Name | NVARCHAR(50) | Да |  |
| 3 | Страна происхождения | Country | NVARCHAR(20) | Да |  |
| 4 | Номер технической поддержки | Technical Support Number | NVARCHAR(25) |  |  |

Таблица 22 –Список атрибутов таблицы «DiscountLevels»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID уровня | LevelID | TINYINT | Да | ПК |
| 2 | Название уровня | Name | NVARCHAR(20) | Да |  |
| 3 | Сумма для уровня | AmountOfPurchases | INT | Да |  |
| 4 | Процент скидки | PercentDiscount | DECIMAL(4,2) | Да |  |

Таблица 23 –Список атрибутов таблицы «Colors»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID цвета | ColorID | TINYINT | Да | ПК |
| 2 | Название цвета | ColorName | NVARCHAR(50) | Да |  |

Таблица 24 –Список атрибутов таблицы «FreezerLocations»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID компоновки | FreezerLocationID | TINYINT | Да | ПК |
| 2 | Название компоновки | FreezerLocationName | NVARCHAR(30) | Да |  |

Таблица 25 –Список атрибутов таблицы «EnergyClasses»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID класса | EnergyClassID | TINYINT | Да | ПК |
| 2 | Название класса | EnergyClassName | NVARCHAR(10) | Да |  |

Таблица 26 –Список атрибутов таблицы «BacklightTypes»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID типа подсветки | BacklightTypeID | TINYINT | Да | ПК |
| 2 | Название типа подсветки | BacklightTypeName | NVARCHAR(30) | Да |  |

Таблица 27 –Список атрибутов таблицы «OperatingSystems»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID операционной системы | Operating SystemID | TINYINT | Да | ПК |
| 2 | Название операционной системы | Operating SystemName | NVARCHAR(20) | Да |  |

Таблица 28 –Список атрибутов таблицы «ScreenSizes»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID диагонали | ScreenSizeID | TINYINT | Да | ПК |
| 2 | Диагональ в дюймах | ScreenSizeInInches | TINYINT | Да |  |
| 3 | Диагональ в сантиметрах | ScreenSize InCentimeters | TINYINT | Да |  |

Таблица 29 –Список атрибутов таблицы «ScreenResolutions»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID разрешения | ScreenResolutionID | TINYINT | Да | ПК |
| 2 | Название разрешение | Screen Resolution Name | NVARCHAR(10) | Да |  |
| 3 | Наименование разрешения | Screen Resolution | NVARCHAR(10) | Да |  |

Таблица 30 –Список атрибутов таблицы «Products»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID товара | ProductID | INT | Да | ПК |
| 2 | Модель | Model | NVARCHAR(100) | Да |  |
| 3 | ID категории | Category | SMALLINT | Да | ВК |
| 4 | Цена | Price | DECIMAL(10,2) | Да |  |
| 5 | Описание | Description | NVARCHAR(900) |  |  |
| 6 | ID производителя | Manufacturer | SMALLINT |  | ВК |
| 7 | ID цвета | Color | TINYINT |  | ВК |
| 8 | Ширина | Width | DECIMAL(6,2) |  |  |
| 9 | Высота | Height | DECIMAL(6,2) |  |  |
| 10 | Глубина | Depth | DECIMAL(6,2) |  |  |
| 11 | Гарантия | Warranty | SMALLINT |  |  |
| 12 | Потребление энергии | Power Consumption | SMALLINT |  |  |
| 13 | ID класса энергоэфф. | EnergyClass | TINYINT |  | ВК |
| 14 | Путь к файлу изображения | Image | NVARCHAR(100) |  |  |
| 15 | ID компоновки морозильной камеры | Freezer Location | TINYINT |  | ВК |
| 16 | Зона свежести | Freshness Zone | BIT |  |  |
| 17 | Инверторный компрессор | Inverter Compressor | BIT |  |  |
| 18 | Объем холодильной камеры | Refrigerator Volume | SMALLINT |  |  |
| 19 | Объем морозильной камеры | Freezer Volume | SMALLINT |  |  |
| 20 | Температурный дисплей | Temperature Display | BIT |  |  |
| 21 | ID типа подсветки | Backlight Type | TINYINT |  | ВК |
| 22 | ID диагонали экрана | ScreenSize | TINYINT |  | ВК |
| 23 | ID разрешения экрана | Screen Resolution | TINYINT |  | ВК |
| 24 | Поддержка SmartTV | SmartTV Support | BIT |  |  |
| 25 | ID операционной системы | Operating System | TINYINT |  | ВК |
| 26 | Поддержка Bluetooth | Bluetooth | BIT |  |  |
| 27 | Поддержка HDR | HDRSupport | BIT |  |  |
| 28 | Расход воды | Water Consumption | TINYINT |  |  |
| 29 | Количество программ | Number OfPrograms | TINYINT |  |  |
| 30 | Количество комплектов посуды | Number OfPlacedSets | TINYINT |  |  |
| 31 | Максимальная загрузка | LaundryLoad | TINYINT |  |  |
| 32 | Диапазон температур | Temperature Range | NVARCHAR(10) |  |  |
| 33 | Прямой привод | DirectDrive | BIT |  |  |
| 34 | Максимальное число оборотов в минуту | Maximum SpinSpeed | SMALLINT |  |  |
| 35 | Внутренний объем | Internal Volume | SMALLINT |  |  |
| 36 | Наличие гриля | Grill | BIT |  |  |

Таблица 31 –Список атрибутов таблицы «Purchases»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение |
| 1 | ID покупки | PurchaseID | INT | ДА | ПК |
| 2 | Дата покупки | PurchaseDate | DATETIME2(0) | ДА |  |
| 3 | ID клиента | ClientID | INT | ДА | ВК |

Таблица 32 –Список атрибутов таблицы «PurchaseItems»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Название | Идентификатор | Тип | Не пусто | Ограничение | |
| 1 | ID покупки | PurchaseID | INT | ДА | ВК | ПК |
| 2 | ID товара | ProductID | INT | ДА | ВК |
| 3 | Число товаров | ProductCount | SMALLINT | ДА |  | |

**Скрипт создания БД и ее объектов**

use master;

GO

    /\*\*\* Создание базы данных \*\*\*/

if DB\_ID('ApplianceStore') is NOT NULL

    drop database ApplianceStore;

GO

create database ApplianceStore;

GO

use ApplianceStore;

    /\*\*\* Создание таблицы Managers \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'Managers')

    DROP TABLE Managers;

GO

CREATE TABLE Managers

(   ManagerID   INT             NOT NULL,

    FullName    NVARCHAR(30)    NOT NULL,

    Login       NVARCHAR(16)    NOT NULL UNIQUE,

    Password    NVARCHAR(16)    NOT NULL,

    CONSTRAINT PK\_Managers

        PRIMARY KEY (ManagerID)

);

GO

    /\*\*\* Создание таблицы DiscountLevels \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'DiscountLevels')

    DROP TABLE DiscountLevels;

GO

CREATE TABLE DiscountLevels

(   LevelID             TINYINT         NOT NULL,

    Name                NVARCHAR(20)    NOT NULL UNIQUE,

    AmountOfPurchases   INT             NOT NULL,

    PercentDiscount     DECIMAL(4,2)    NOT NULL,

    CONSTRAINT PK\_DiscountLevels

        PRIMARY KEY (LevelID)

);

GO

    /\*\*\* Создание таблицы Clients \*\*\*/

GO

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'Clients')

    DROP TABLE Clients;

GO

CREATE TABLE Clients

(   ClientID        INT             NOT NULL,

    PhoneNumber     NVARCHAR(25)    NOT NULL UNIQUE,

    Password        NVARCHAR(16)    NOT NULL,

    Surname         NVARCHAR(25)    NOT NULL,

    Firstname       NVARCHAR(25)    NOT NULL,

    Lastname        NVARCHAR(25),

    Account         Money           NOT NULL DEFAULT 0,

    DiscountLevel   TINYINT         NOT NULL DEFAULT 0,

    LastPurchase    DATETIME2(0),

    RegisterDate    DATETIME2(0),

    CONSTRAINT PK\_Clients

        PRIMARY KEY (ClientID),

    CONSTRAINT FK\_Clients\_DiscountLevel

        FOREIGN KEY (DiscountLevel)

            REFERENCES DiscountLevels(LevelID)

);

GO

    /\*\*\* Создание таблицы Categories \*\*\*/

GO

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'Categories')

    DROP TABLE Categories;

GO

CREATE TABLE Categories

(

    CategoryID      SMALLINT        NOT NULL,

    CategoryName    NVARCHAR(50)    NOT NULL,

    Description     NVARCHAR(500),

    CONSTRAINT PK\_Categories

        PRIMARY KEY (CategoryID),

);

GO

/\*\*\* Создание таблицы Manufacturers \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'Manufacturers')

    DROP TABLE Manufacturers;

GO

CREATE TABLE Manufacturers

(

    ManufacturerID          SMALLINT        NOT NULL,

    CompanyName             NVARCHAR(50)    NOT NULL,

    Country                 NVARCHAR(20)    NOT NULL,

    TechnicalSupportNumber  NVARCHAR(25),

    CONSTRAINT PK\_Manufacturers

        PRIMARY KEY (ManufacturerID)

);

GO

    /\*\*\* Создание таблицы Colors \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'Colors')

    DROP TABLE Colors;

GO

CREATE TABLE Colors

(

    ColorID         TINYINT         NOT NULL,

    ColorName       NVARCHAR(50)    NOT NULL,

    CONSTRAINT PK\_Colors

        PRIMARY KEY (ColorID)

);

GO

    /\*\*\* Создание таблицы EnergyClasses \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'EnergyClasses')

    DROP TABLE EnergyClasses;

GO

CREATE TABLE EnergyClasses

(

    EnergyClassID       TINYINT         NOT NULL,

    EnergyClassName     NVARCHAR(10)    NOT NULL,

    CONSTRAINT PK\_EnergyClasses

        PRIMARY KEY (EnergyClassID)

);

GO

    /\*\*\* Создание таблицы FreezerLocations \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'FreezerLocations')

    DROP TABLE FreezerLocations;

GO

CREATE TABLE FreezerLocations

(

    FreezerLocationID       TINYINT         NOT NULL,

    FreezerLocationName     NVARCHAR(30)    NOT NULL,

    CONSTRAINT PK\_FreezerLocations

        PRIMARY KEY (FreezerLocationID)

);

GO

    /\*\*\* Создание таблицы BacklightTypes \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'BacklightTypes')

    DROP TABLE BacklightTypes;

GO

CREATE TABLE BacklightTypes

(

    BacklightTypeID         TINYINT         NOT NULL,

    BacklightTypeName       NVARCHAR(30)    NOT NULL,

    CONSTRAINT PK\_BacklightTypes

        PRIMARY KEY (BacklightTypeID)

);

GO

    /\*\*\* Создание таблицы ScreenSizes \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'ScreenSizes')

    DROP TABLE ScreenSizes;

GO

CREATE TABLE ScreenSizes

(

    ScreenSizeID            TINYINT     NOT NULL,

    ScreenSizeInInches      TINYINT     NOT NULL,

    ScreenSizeInCentimeters TINYINT     NOT NULL,

    CONSTRAINT PK\_ScreenSizes

        PRIMARY KEY (ScreenSizeID)

);

GO

    /\*\*\* Создание таблицы ScreenResolutions \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'ScreenResolutions')

    DROP TABLE ScreenResolutions;

GO

CREATE TABLE ScreenResolutions

(

    ScreenResolutionID      TINYINT         NOT NULL,

    ScreenResolutionName    NVARCHAR(10)    NOT NULL,

    ScreenResolution        NVARCHAR(10)    NOT NULL,

    CONSTRAINT PK\_ScreenResolutions

        PRIMARY KEY (ScreenResolutionID)

);

GO

    /\*\*\* Создание таблицы OperatingSystems \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'OperatingSystems')

    DROP TABLE OperatingSystems;

GO

CREATE TABLE OperatingSystems

(

    OperatingSystemID       TINYINT         NOT NULL,

    OperatingSystemName     NVARCHAR(20)    NOT NULL,

    CONSTRAINT PK\_OperatingSystems

        PRIMARY KEY (OperatingSystemID)

);

GO

    /\*\*\* Создание таблицы Products \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'Products')

    DROP TABLE Products;

GO

CREATE TABLE Products

(   ProductID           INT             NOT NULL,

    Model               NVARCHAR(100)   NOT NULL,

    Category            SMALLINT        NOT NULL,

    Price               DECIMAL(10,2)   DEFAULT 0,

    Description         NVARCHAR(1000),

    Manufacturer        SMALLINT,

    Color               TINYINT,

    Width               DECIMAL(6,2),

    Height              DECIMAL(6,2),

    Depth               DECIMAL(6,2),

    Warranty            SMALLINT,

    PowerConsumption    SMALLINT,

    EnergyClass         TINYINT,

    Image               NVARCHAR(100)   DEFAULT('pack://application:,,,/Data/Images/default.png'),

    /\* Холодильники\*/

    FreezerLocation     TINYINT,

    FreshnessZone       BIT,

    InverterCompressor  BIT,

    RefrigeratorVolume  SMALLINT,

    FreezerVolume       SMALLINT,

    TemperatureDisplay  BIT,

    /\* Телевизоры \*/

    BacklightType       TINYINT,

    ScreenSize          TINYINT,

    ScreenResolution    TINYINT,

    SmartTVSupport      BIT,

    OperatingSystem     TINYINT,

    Bluetooth           BIT,

    HDRSupport          BIT,

    /\* Посудомойки и стиралки\*/

    WaterConsumption    TINYINT,

    NumberOfPrograms    TINYINT,

    NumberOfPlacedSets  TINYINT,

    LaundryLoad         TINYINT,

    TemperatureRange    NVARCHAR(10),

    DirectDrive         BIT,

    MaximumSpinSpeed    SMALLINT,

    /\* Микроволновки \*/

    InternalVolume      SMALLINT,

    Grill               BIT,

    CONSTRAINT PK\_Products

        PRIMARY KEY (ProductID),

    CONSTRAINT FK\_Products\_Categories

        FOREIGN KEY (Category)

            REFERENCES Categories(CategoryID),

    CONSTRAINT FK\_Products\_Manufacturers

        FOREIGN KEY (Manufacturer)

            REFERENCES Manufacturers(ManufacturerID),

    CONSTRAINT FK\_Products\_Colors

        FOREIGN KEY (Color)

            REFERENCES Colors(ColorID),

    CONSTRAINT FK\_Products\_EnergyClasses

        FOREIGN KEY (EnergyClass)

            REFERENCES EnergyClasses(EnergyClassID),

    CONSTRAINT FK\_Products\_FreezerLocations

        FOREIGN KEY (FreezerLocation)

            REFERENCES FreezerLocations(FreezerLocationID),

    CONSTRAINT FK\_Products\_BacklightTypes

        FOREIGN KEY (BacklightType)

            REFERENCES BacklightTypes(BacklightTypeID),

    CONSTRAINT FK\_Products\_ScreenSizes

        FOREIGN KEY (ScreenSize)

            REFERENCES ScreenSizes(ScreenSizeID),

    CONSTRAINT FK\_Products\_ScreenResolutions

        FOREIGN KEY (ScreenResolution)

            REFERENCES ScreenResolutions(ScreenResolutionID),

    CONSTRAINT FK\_Products\_OperatingSystems

        FOREIGN KEY (OperatingSystem)

            REFERENCES OperatingSystems(OperatingSystemID)

);

GO

    /\*\*\* Создание таблицы Purchases \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'Purchases')

    DROP TABLE Purchases;

GO

CREATE TABLE Purchases

(

    PurchaseID      INT             NOT NULL,

    PurchaseDate    DATETIME2(0)    NOT NULL DEFAULT GETDATE(),

    ClientID        INT             NOT NULL,

    CONSTRAINT PK\_Purchases

        PRIMARY KEY (PurchaseID),

    CONSTRAINT FK\_Purchases\_Clients

        FOREIGN KEY (ClientID)

            REFERENCES Clients(ClientID)

);

GO

    /\*\*\* Создание таблицы PurchaseItems \*\*\*/

IF EXISTS (SELECT \* FROM sys.tables WHERE NAME = 'PurchaseItems')

    DROP TABLE PurchaseItems;

GO

CREATE TABLE PurchaseItems

(

    PurchaseID      INT         NOT NULL,

    ProductID       INT         NOT NULL,

    ProductCount    SMALLINT    NOT NULL DEFAULT 1,

    CONSTRAINT PK\_PurchaseItems

        PRIMARY KEY (PurchaseID, ProductID),

    CONSTRAINT FK\_PurchaseItems\_Purchases

        FOREIGN KEY (PurchaseID)

            REFERENCES Purchases(PurchaseID),

    CONSTRAINT FK\_PurchaseItems\_Products

        FOREIGN KEY (ProductID)

            REFERENCES Products(ProductID)

);

GO

    /\*\*\* Создание триггеров \*\*\*/

IF EXISTS (SELECT \* FROM sys.server\_triggers WHERE NAME = 'T\_Purchases\_Insert')

    DROP TRIGGER T\_Purchases\_Insert ON ALL SERVER;

GO

CREATE TRIGGER T\_Purchases\_Insert

    ON Purchases AFTER INSERT

    AS  UPDATE Clients

            SET LastPurchase = GETDATE()

                WHERE ClientID = (SELECT ClientID FROM inserted)

        UPDATE Purchases

            SET PurchaseDate = GETDATE()

                WHERE PurchaseID = (SELECT PurchaseID FROM inserted)

GO

IF EXISTS (SELECT \* FROM sys.server\_triggers WHERE NAME = 'T\_Clients\_Insert')

    DROP TRIGGER T\_Clients\_Insert ON ALL SERVER;

GO

CREATE TRIGGER T\_Clients\_Insert

    ON Clients AFTER INSERT

    AS  UPDATE Clients

            SET RegisterDate = GETDATE()

                WHERE ClientID = (SELECT ClientID FROM inserted)

GO

# 2 Разработка и тестирование программного обеспечения для аттестации персонала

## 2.1 Разработка программных модулей

Разработка будет произведена в среде Microsoft Visual Studio на платформе .Net Framework. Язык программирования - C#. Программа создана с помощью системы WPF, локального сервера MS SQL Server и технологии работы с базами данных Entity Framework. Программа имеет 6 окон:

Окно авторизации пользователей для входа в систему представлено на рисунке 4. Поле номера телефона поддерживает ввод буквенных символов для авторизации менеджеров магазина. При нажатии на ссылку «Зарегистрируйтесь» откроется окно регистрации.

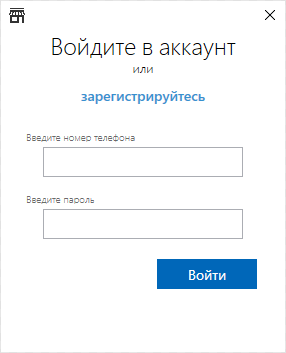


Рисунок 4 – Окно авторизации пользователей

Окно регистрации аккаунтов клиентов представлено на рисунке 5. Поля имеют ограничение типа вводимых символов и проверку целостности данных для регистрации.

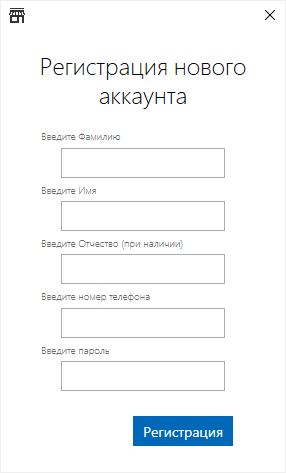


Рисунок 5 – Окно регистрации клиентов

После авторизации пользователя на экран отображается основной каталог программы. В верхней части окна представлен перечень доступных категорий товаров и кнопка «Корзина» со счетчиком товаров, хранящихся в ней. В центральной части окна расположены плитки товаров с их изображением и кратким описанием, а также фильтр по товарам выбранной категории. В нижней части окна указаны фамилия, имя и отчество авторизованного пользователя, а также текущий уровень его скидки.

Основной каталог программы представлен на рисунке 6.

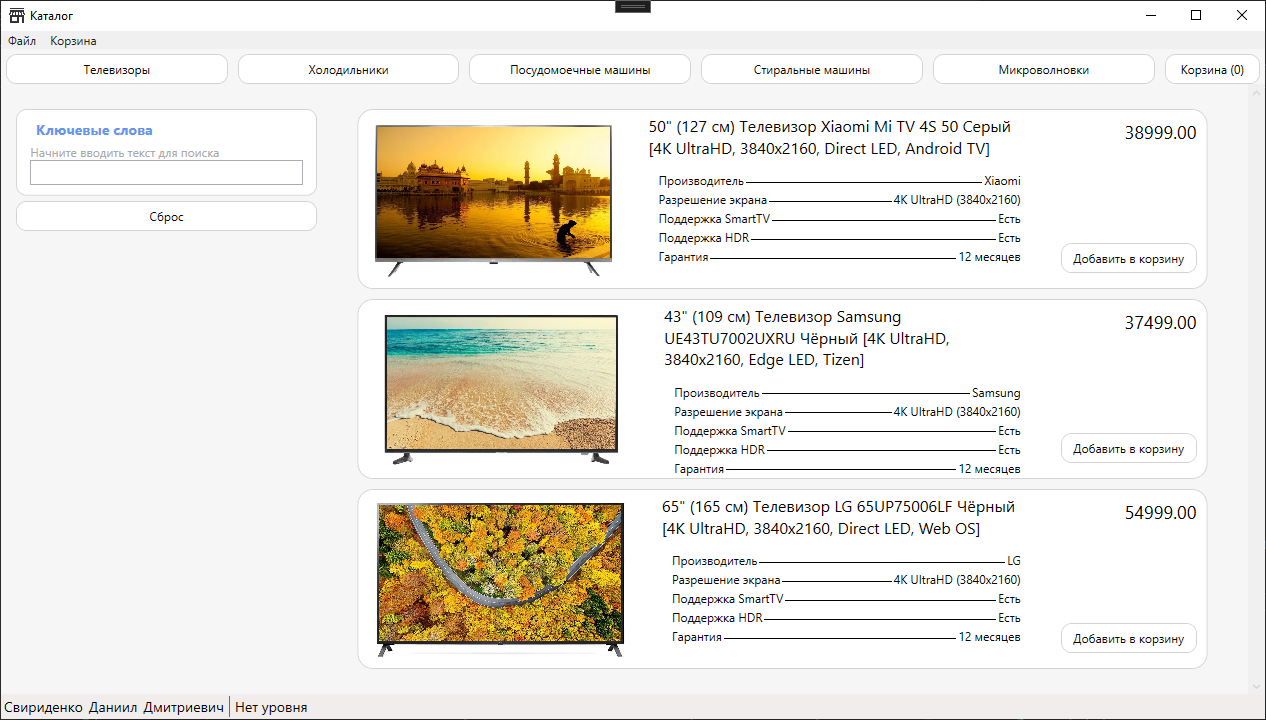


Рисунок 6 – Главное окно программы для пользователя

При нажатии на одну из плиток товаров откроется окно расширенного описания товара, представленное на рисунке 7.

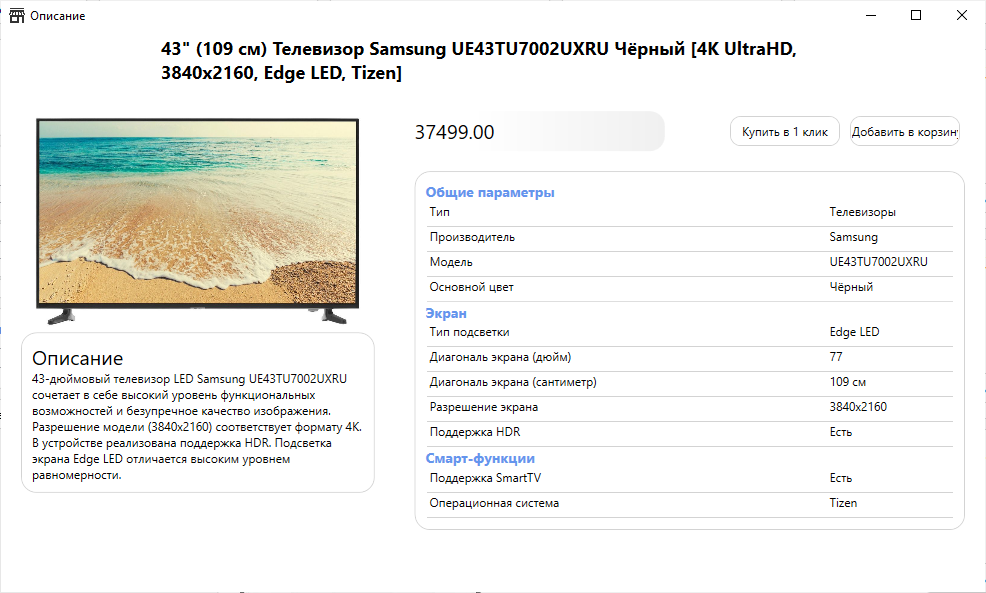


Рисунок 7 – Окно расширенного описания товаров

При щелчке на кнопку «Добавить в корзину» у одного из товаров, он будет записан в список товаров пользовательской корзины. Один товар добавить в корзину можно только один раз. Окно корзины представлено на рисунке 8. Здесь пользователь может изменить количество необходимых товаров, отметить нужные ему товары или снять их выделение, просмотреть сумму и оформить заказ.

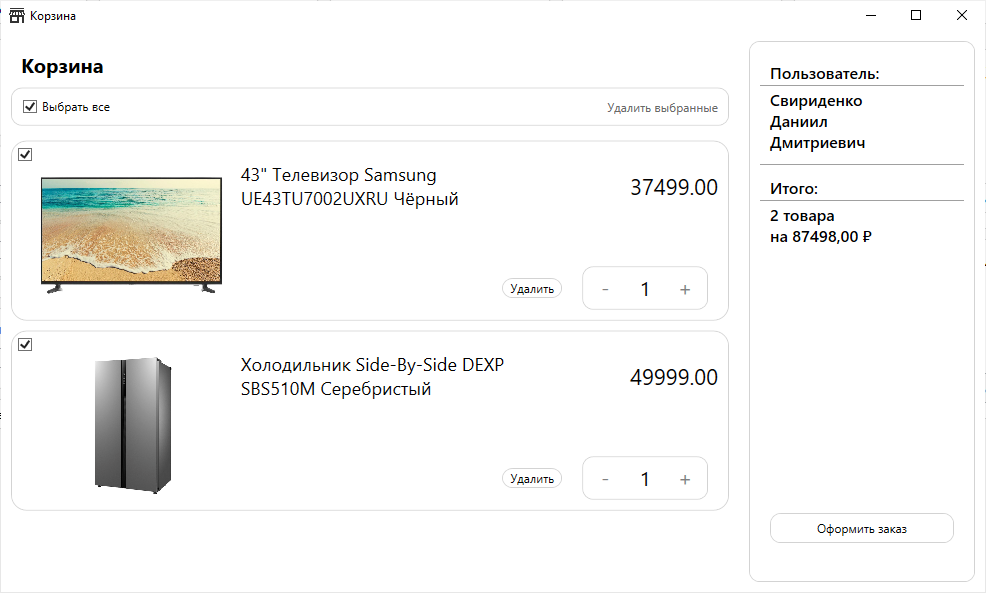


Рисунок 8 – Окно корзины

При нажатии на кнопку оформления заказа будет вызвано сообщение, требующее подтверждения действия. При положительном ответе на сообщение будет создан товарный чек, содержащий заранее заданную информацию о магазине, перечень товаров и контактный телефон. Внешний вид товарного чека представлен на рисунке 9.

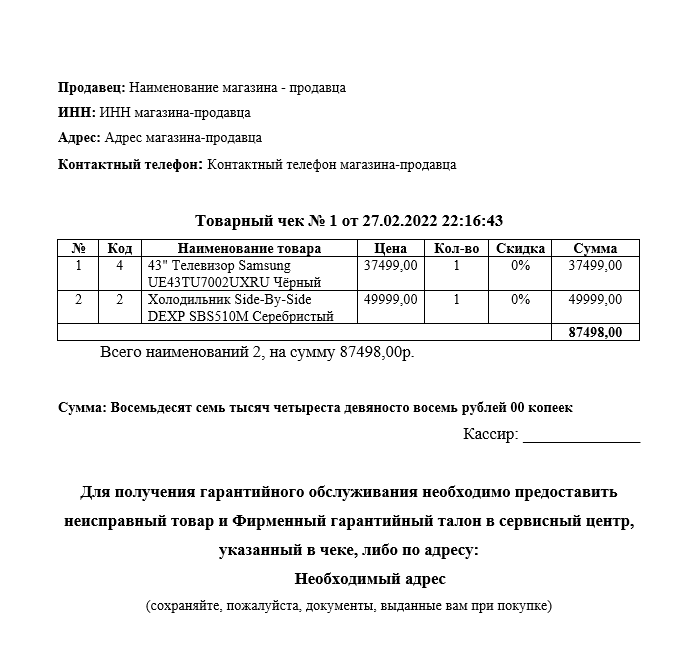


Рисунок 9 – Внешний вид товарного чека.

Если пользователь авторизуется в аккаунт менеджера, то главное окно будет иметь другой внешний вид, представленный на рисунке 10.

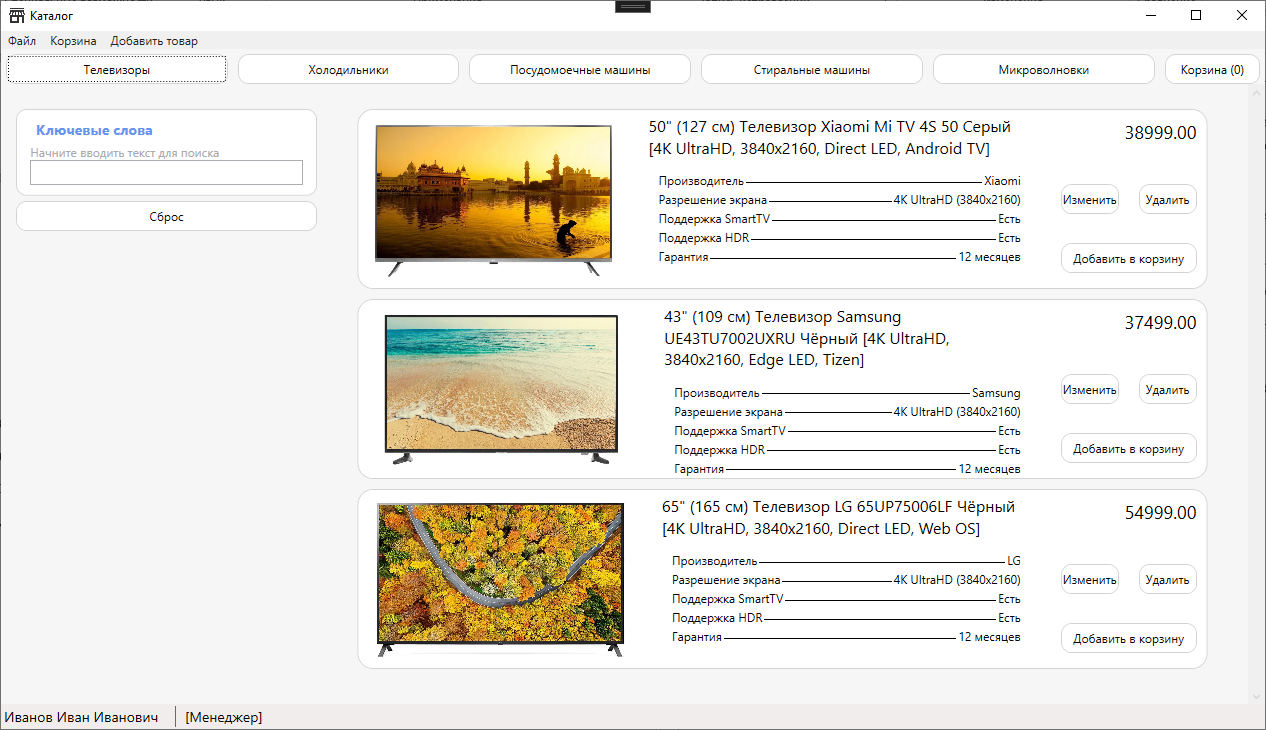


Рисунок 10 – Главное окно программы для менеджера

Менеджеру доступны функции по изменению и удалению существующих товаров и добавлении новых товаров. При нажатии на кнопку изменения или удаления товара, на экране отобразится окно редактора товаров, представленное на рисунке 11.

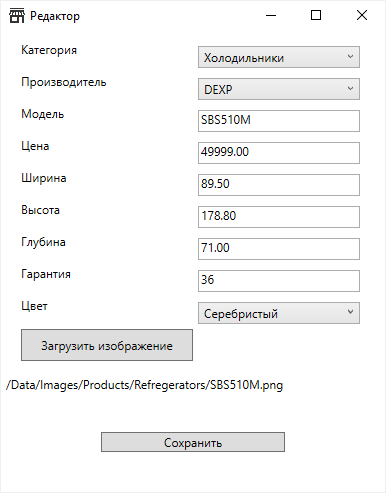


Рисунок 11 – Окно редактирования ассортимента

## 2.2 Тестирование и отладка

Таблица 33 – Тестирование информационной системы

|  |  |  |  |
| --- | --- | --- | --- |
| № | Исходные данные | Ожидаемый результат | Фактический результат |
| 1 | 2 | 3 | 4 |
| Тестирование базовых функций | | | |
| 1 | Запустить программу | Откроется окно авторизации. | Пройден |
| 2 | В поле Логин ввести q,  в поле пароль – q. | Цвет шрифта элементов станет красным. | Пройден |

Продолжение таблицы 33.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 3 | В поле логин ввести 1, в поле пароль – 1. | Появится приветственное сообщение, содержащее имя и отчество клиента. Откроется основное окно. Клиенту не доступны функции изменения, удаления и добавления товаров. Выбрана категория телевизоры. В центральной части окно есть список доступных моделей. | Пройден |
| 4 | Кликнуть на кнопку, соответствующую категории Холодильники. | Список товаров обновился. Теперь отображаются товары, относящиеся к категории холодильники | Пройден |
| 5 | Кликнуть на кнопку «Корзина» | Открылось окно корзины. В корзине отсутствуют товары, значения количество наименований и суммы пусты. | Пройден |
| 6 | Закрыть окно корзины.  Нажать на кнопку «Телевизоры».  Нажать на кнопку «Добавить в корзину», соответствующую товару Xiaomi Mi 4S 50. | Окно корзины закрылось.  В главном окне отображаются товары, соответствующие категории Телевизоры.  Кнопка «Добавить в корзину» стала неактивной, на кнопке «Корзина» отображается число товаров – 1. | Пройден |
| 7 | Нажать на кнопку «Корзина». | В корзине отображается товар Xiaomi Mi 4S 50. Чекбокс «Выбрать все» и чекбокс, соответствующий товару отмечены. В графе количество наименований указано количество – 1. В графе сумма указана сумма – 38999.00. | Пройден |

Продолжение таблицы 33.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 8 | Нажать на кнопку «Оформить заказ». Подтвердить заказ кнопкой Да.  Выбрать Рабочий стол в окне проводника и нажать на кнопку «Сохранить» | Появилось сообщение о повышении уровня скидки пользователя.  Открылось окно авторизации пользователя.  Появилось окно проводника.  На рабочем столе появился документ в формате .docx, содержащий корректную информацию. | Пройден |
| 9 | В поле Логин ввести – 1, пароль – 1. | Открылось главное окно программы. В нижней части окна указан уровень пользователя – бронза. | Пройден |
| 10 | В поле «Ключевые слова» начать вводить «чёрн» | В списке товаров отображаются только телевизоры черного цвета. | Пройден |
| 11 | Нажать на кнопку «Сброс» | В списке товаров отображаются все доступные товары. | Пройден |
| 12 | Добавить товар «Xiaomi Mi 4S 50» и «LG 65UP75006LF» в корзину.  Нажать на кнопку корзина. | В корзине отображаются два добавленных товара. Чекбокс «Выбрать все» и чекбоксы, соответствующие товарам отмечены. | Пройден |
| 13 | Нажать на чекбокс «Выбрать все» | С чекбоксов, соответствующим товарам, снялось выделение. Значение количества товаров в корзине – 0, значение суммы – 0.00. | Пройден |
| 14 | Нажать на чекбокс, соответствующий товару «Xiaomi Mi 4S 50» | Значение количества товаров в корзине – 1, значение суммы – 38999.00. | Пройден |

Окончание таблицы 33.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 15 | Нажать на кнопку «+», обозначающую количество единиц товара. | Цифра количества товаров данного типа – 2, Значение количества товаров – 2, значение суммы – 77 998.00. | Пройден |
| 16 | Перезапустить программу. Авторизоваться с логином – 2, паролем – 2. | Открылось главное окно программы. На плитках товаров доступны кнопки «Изменить», «Удалить». В нижней части окна указан уровень доступа [Менеджер] | Пройден |
| 17 | Нажать на кнопку «Удалить», соответствующую товару «Xiaomi Mi 4S 50» | Список товаров обновился, товар «Xiaomi Mi 4S 50» удален. | Пройден |
| 18 | Нажать на кнопку «Изменить», соответствующую товару «LG 65UP75006LF» | Открылось окно редактора товаров. Содержимое полей соответствует характеристикам товара. | Пройден |
| 19 | Изменить в поле «Модель» текст на – abcd999, в выпадающего списка цветов выбрать Белый. Нажать на кнопку Сохранить. | Окно закрылось. Данные о товаре на главном окне обновились и соответствуют новым введенным данным. | Пройден |

## 2.3 Соответствие кода стандартам кодирования

Стандарт оформления кода — набор правил и соглашений, используемых при написании исходного кода на некотором языке программирования. Наличие общего стиля программирования облегчает понимание и поддержание исходного кода, написанного более чем одним программистом, а также упрощает взаимодействие нескольких человек при разработке программного обеспечения.

Рассмотрим некоторые стандарты кодирования.

Соглашения об именах для локальных переменных, глобальных переменных, констант и функций Camel case:

- Содержательное и понятное название переменных помогает любому понять причину его использования.

- Локальные переменные должны быть названы с использованием Camel case, начинающихся с маленькой буквы (например, localData), тогда как имена глобальных переменных должны начинаться с заглавной буквы (например, GlobalData). Имена констант должны быть сформированы только заглавными буквами (например, CONSDATA).

- Лучше избегать использования цифр в именах переменных.

- Названия функции должны быть написаны начиная с маленьких букв.

- Название функции должно четко и кратко описывать причину ее использования.

Рассмотрим данный стандарт в фрагменте кода, представленном на рисунке 11.

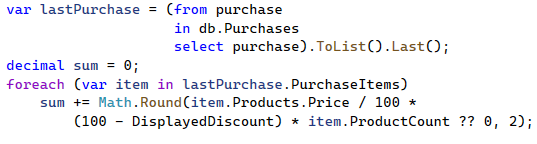


Рисунок 11 – Стандарт Camel case

На данном рисунке все переменные названы в соответствии со стандартом. Так, локальная переменные sum и lastPurchase названы со строчной буквы, второе слово переменной начинается с заглавной буквы. В глобальной переменной DisplayedDiscount - все начальные буквы слов заглавные.

Правильный отступ очень важен для улучшения читабельности кода. Чтобы сделать код читабельным, программисты должны правильно использовать пробелы. Некоторые из интервалов даны ниже:

- После запятой между аргументами функции должен быть пробел.

- Каждый вложенный блок должен иметь правильные отступы и интервалы.

- Все фигурные скобки должны начинаться с новой строки, а код, следующий за окончанием фигурных скобок, также начинается с новой строки.

Рассмотрим данный стандарт по примеру, представленному на рисунке 12.

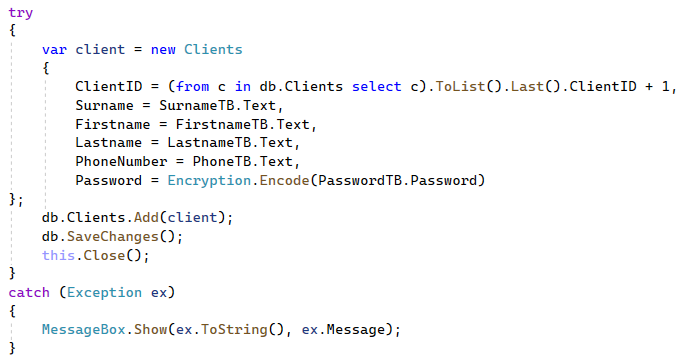


Рисунок 12 – Стандарт отступов и интервалов

Данный стандарт полностью отражается в данном примере.

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

**Нормативно – правовые акты**

1. Федеральный закон от 27 июля 2006 года № 149-ФЗ «Об информации, информационных технологиях и о защите информации».

**Учебники и учебные пособия**

1. Федорова Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем (2-е изд., стер.) учебник.- М.: Академия, 2018.
2. Кумскова, И.А. Базы данных: учебник / Кумскова И.А. — Москва: КноРус, 2020. — 400 с. — (СПО). — ISBN 978-5-406-07467-1. — URL: https://book.ru/book/932493

**Дополнительные издания**

1. Подбельский, В. В.  Программирование. Базовый курс С# : учебник для среднего профессионального образования / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/456697

**Интернет-ресурсы**

1. Справочник по базам данных <http://msdn.microsoft.com/ru-ru/library/h0y4a0f6.aspx>
2. Пошаговые руководства по C# <http://msdn.microsoft.com/ru-ru/library/1dbsh6t3.aspx>
3. Работа с базами данных на языке C# <http://sernam.ru/book_cbd.php>

# ПРИЛОЖЕНИЕ А

**Техническое задание**

**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Саратовский государственный технический университет имени Гагарина Ю.А.»**

**(СГТУ имени Гагарина Ю.А.)**

**ПРОФЕССИОНАЛЬНО-ПЕДАГОГИЧЕСКИЙ КОЛЛЕДЖ**

**Информационной система магазина бытовой техники**«Appliance Store»

Техническое задание

RU.03654343.00019 - 01 90 01

На 8 листах

Выполнил студент

группы ИСП-941

Свириденко Даниил Дмитриевич

Саратов 2022

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc96917554)

[1 Наименование программы и область применения 3](#_Toc96917555)

[1.2 Область применения 3](#_Toc96917556)

[2 Требования к программе 3](#_Toc96917557)

[2.1 Требования к функциональным характеристикам 3](#_Toc96917558)

[2.2 Требования к надежности 4](#_Toc96917559)

[2.2.1 Требования к обеспечению надежного функционирования программы 4](#_Toc96917560)

[2.2.2 Время восстановления после отказа 4](#_Toc96917561)

[2.2.3 Отказы из-за некорректных действий оператора 5](#_Toc96917562)

[3 Условия эксплуатации 5](#_Toc96917563)

[3.1 Климатические условия эксплуатации 5](#_Toc96917564)

[3.2 Требования к квалификации и численности персонала 5](#_Toc96917565)

[3.3 Требования к составу и параметрам технических средств 6](#_Toc96917566)

[3.4 Требования к информационной и программной совместимости 7](#_Toc96917567)

[3.4.1 Требования к исходным кодам и языкам программирования 7](#_Toc96917568)

[3.4.2 Требования к программным средствам, используемым программой 7](#_Toc96917569)

[4 Требования к программной документации 8](#_Toc96917570)

[4.1 Предварительный состав программной документации 8](#_Toc96917571)

# ВВЕДЕНИЕ

Информационная система «Appliance Store» создана для автоматизации оформления заказов клиентами целевого магазина бытовой техники.

## 1 Наименование программы и область применения

Предметом разработки является информационная система магазина бытовой техники.

Наименование программы:

Информационная система магазина бытовой техники «Appliance Store».

## 1.2 Область применения

Данная АС разработана для внутреннего пользования на целевом предприятии и используется для автоматизации составления заказов на бытовую технику, контроля ассортимента магазина, создания электронной документации и отчетов.

Областью применения программного продукта является сфера предоставления услуг целевого предприятия.

# 2 Требования к программе

## 2.1 Требования к функциональным характеристикам

Информационная система «Appliance Store» предназначена для обеспечения доступа клиентам целевой компании к информации о товарах и возможности совершить заказ с использованием терминала, находящегося в торговом зале. Для сотрудников компании информационная система должна обеспечить доступ к ассортименту магазина, возможности его редактировать, добавлять, изменять и удалять позиции товаров, а также составлять отчеты по продажам за определенный период времени.

## 2.2 Требования к надежности

Под надежностью ИС «Appliance Store» следует понимать комплексное свойство системы сохранять во времени в установленных нормативно-технической и/или конструкторской документацией пределах значения параметров, характеризующих способность системы выполнять свои функции, определяемые ее назначением, режимами и условиями эксплуатации.

### 2.2.1 Требования к обеспечению надежного функционирования программы

ИС «Appliance Store» должна обеспечивать:

− целостность и корректность данных при разрыве соединения во время взаимодействия со смежными системами;

− ИС «Appliance Store» не должна терять работоспособность в случае возникновения сбоев, аварий и отказов, возникающих на рабочих станциях сотрудников и терминалах в торговом зале;

− ИС «Appliance Store» в целом должна сохранять работоспособность при некорректных действиях пользователей;

Минимальный срок эксплуатации системы не менее 5 лет, отдельных подсистем – не менее 5 лет.

### 2.2.2 Время восстановления после отказа

Время восстановления после сбоя Системы должно определяться как промежуток времени в минутах, в течение которого обслуживающий персонал, в случае необходимости производит восстановление базы данных Системы из резервной копии, и запустит комплекс программ. Время восстановления после сбоя должно быть не более 60 минут.

### 2.2.3 Отказы из-за некорректных действий оператора

После запуска программы отказ программы вследствие некорректных действий оператора должен быть исключён. В том числе должна быть исключена возможность непреднамеренного выключения программы, не связанного с техническими неполадками сервера.

# 3 Условия эксплуатации

## 3.1 Климатические условия эксплуатации

Необходимо сухое, регулярно убираемое помещение, в котором соблюдаются все требования по электробезопасности.

## 3.2 Требования к квалификации и численности персонала

Для эксплуатации ИС «Appliance Store» должны быть предусмотрены следующие роли пользователей:

- Системный администратор;

- Менеджер;

- Покупатель.

Основными обязанностями системного администратора являются:

- Модернизация, настройка и мониторинг работоспособности комплекса технических средств (серверов, рабочих станций);

- Установка, модернизация, настройка и мониторинг работоспособности системного и базового программного обеспечения;

- Установка, настройка и мониторинг прикладного программного обеспечения;

- Ведение учетных записей пользователей системы.

Системный администратор должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию программных и технических средств, применяемых в системе.

Основными обязанностями менеджера являются:

- Добавление новых позиций товара в информационную систему;

- Редактирование существующих позиций товара;

- Удаление не актуальных позиций товара.

Менеджер должен обладать уверенным навыком владения ПК и уметь выполнять свои обязанности, используя интерфейс информационной системы на рабочей станции.

Основными задачами покупателя являются:

- Регистрация нового аккаунта пользователя и авторизация в существующем;

- Сортировка товара по параметрам цены, производителя, названия и частным параметрам для отдельных категорий товаров;

- Добавление интересующего товара в корзину;

- Оформление заказа с последующей автоматической печатью чека.

## 3.3 Требования к составу и параметрам технических средств

В состав технических средств должны входить несколько персональных компьютеров, выполняющих роль рабочих станции, включающих в себя:

- Процессор Intel Core I3-3,0Hz, не менее;

- Оперативную память объемом, 4Гигабайт, не менее;

- Операционную систему Windows 10.

Также один персональный компьютер, выполняющий роль сервера, включающий в себя:

- Процессор Intel Core I5-2,6GHz, не менее;

- Оперативную память объемом, 16 Гигабайт, не менее;

- Операционную систему Windows 10;

- Microsoft SQL Server 2017.

## 3.4 Требования к информационной и программной совместимости

Системные программные средства, используемые программным заказом, должны быть представлены лицензионной локализованной версией операционной системы Windows 10 и наличие системных библиотек Microsoft Framework 4.

### 3.4.1 Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть созданы на языке программирования C# в системе WPF и применением системы управления базами данных Microsoft SQL Server.

### 3.4.2 Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы не ниже Windows 10. На системе должны быть установлены системные библиотек Microsoft Framework 4.

# 4 Требования к программной документации

## 4.1 Предварительный состав программной документации

В состав программной документации должны входить:

* Техническое задание;
* Программа и методика испытаний;
* Описание рабочих окон программы;
* Сведения о структуре используемой базы данных.

**ПРОГРАММНАЯ ЧАСТЬ**

**1 Скрипт создания базы данных и ее объектов**

**2 Код программы**

AuthorizationWindow

public partial class AuthorizationWindow : Window

{

ApplianceStoreEntities db = new ApplianceStoreEntities();

public AuthorizationWindow()

{

InitializeComponent();

WelcomeMessage.Visibility = Visibility.Hidden;

}

private async void LoginButton\_Click(object sender, RoutedEventArgs e) //Обработчик события нажатия на конпку войти

{

try //Попытка авторизоваться

{

string \_login = loginTextBox.Text;

string \_password = PasswordBox.Password;

try

{

var \_client = (from client in db.Clients

where \_login == client.PhoneNumber && \_password == client.Password

select client).Single();

WelcomeMessage.Text += "\n" + \_client.Firstname + ' ' + \_client.Lastname + '!';

WelcomeMessage.Visibility = Visibility.Visible;

LoginButton.IsEnabled = false;

//await Task.Delay(2000);

Catalog c = new Catalog();

c.Client = \_client;

c.Show();

this.Close();

}

catch

{

var \_manager = (from manager in db.Managers

where \_login == manager.Login && \_password == manager.Password

select manager).Single();

WelcomeMessage.Text = \_manager.FullName;

WelcomeMessage.Visibility = Visibility.Visible;

LoginButton.IsEnabled = false;

//await Task.Delay(2000);

Catalog c = new Catalog();

c.IsAdmin = true;

c.AddProductButton.Visibility = Visibility.Visible;

c.Client = new Clients { Surname = \_manager.FullName, DiscountLevels = new DiscountLevels { Name = " [Менеджер]" } };

c.Show();

this.Close();

}

}

catch //Если пара логина и пароля не найдена в базе данных

{

LoginLabel1.Foreground = Brushes.Red;

LoginLabel2.Foreground = Brushes.Red;

await Task.Delay(1000);

LoginLabel1.Foreground = (Brush)new BrushConverter().ConvertFrom("#BF000000");

LoginLabel2.Foreground = (Brush)new BrushConverter().ConvertFrom("#BF000000");

}

}

private void RegisterLink\_MouseUp(object sender, MouseButtonEventArgs e)

{

RegistrationWindow r = new RegistrationWindow(db);

r.ShowDialog();

db = new ApplianceStoreEntities();

}

}

Catalog

public partial class Catalog : Window, INotifyPropertyChanged

{

public ApplianceStoreEntities db = new ApplianceStoreEntities();

List<Products> DisplayedProducts = new List<Products>();

private Clients \_Client;

public Clients Client { get => \_Client; set => Set(ref \_Client, value); }

public bool IsAdmin = false;

Categories \_SelectedCategory;

public Categories SelectedCategory { get => \_SelectedCategory; set => Set(ref \_SelectedCategory, value); }

private ObservableCollection<Products> \_ShoppingCartList = new ObservableCollection<Products>();

public ObservableCollection<Products> ShoppingCartList { get => \_ShoppingCartList; set => Set(ref \_ShoppingCartList, value); }

#region PropertyChanged

public event PropertyChangedEventHandler PropertyChanged;

private void OnPropertyChanged([CallerMemberName] string PropertyName = null)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(PropertyName));

}

private bool Set<T>(ref T field, T value, [CallerMemberName] string PropertyName = null)

{

if (Equals(field, value)) return false;

field = value;

OnPropertyChanged(PropertyName);

return true;

}

#endregion

public Catalog()

{

DataContext = this;

InitializeComponent();

ChangeCategory(1);

Refresh();

}

public void Refresh()

{

foreach (var c in db.Categories)

{

Button btn = new Button { Margin = new Thickness(5, 5, 5, 0), Content = c, Tag = c.CategoryID };

btn.Click += new RoutedEventHandler(CtgrsClick);

CategoriesPanel.Children.Add(btn);

}

}

public void CtgrsClick(object sender, RoutedEventArgs e)

{

ChangeCategory(int.Parse(((Button)sender).Tag.ToString()));

}

public void ChangeCategory(int categoryID)

{

DisplayedProducts.Clear();

SelectedCategory = (from c in db.Categories

where c.CategoryID == categoryID

select c).Single();

DisplayedProducts = (from p in db.Products

where p.Category == SelectedCategory.CategoryID

select p).ToList();

DisplayProducts();

}

public void DisplayProducts()

{

if (DisplayedProducts != null)

{

MainBox.Children.Clear();

foreach (var p in DisplayedProducts)

{

ProductPreview pw = new ProductPreview(p, this);

if (ShoppingCartList.Contains(p))

pw.IsButtonEnabled = false;

MainBox.Children.Add(pw);

}

}

}

public void FilterProducts(string filter)

{

if (filter == null)

DisplayProducts();

else

{

MainBox.Children.Clear();

foreach (var p in DisplayedProducts)

{

ProductPreview pw = new ProductPreview(p, this);

if (pw.ProductTitle.ToLower().Contains(filter.ToLower()))

{

if (ShoppingCartList.Contains(p))

pw.IsButtonEnabled = false;

MainBox.Children.Add(pw);

}

}

}

}

public void AddProductToDisplay(Products product)

{

if (product != null)

{

ProductPreview pw = new ProductPreview(product, this);

MainBox.Children.Add(pw);

}

}

private void ShoppingCartButton\_Click(object sender, RoutedEventArgs e)

{

ShoppingCart s = new ShoppingCart(ShoppingCartList, Client) { Owner = this };

s.Show();

}

private void AddProductButton\_Click(object sender, RoutedEventArgs e)

{

EditWindow ew = new EditWindow(new Products(), db);

ew.Show();

}

private void DropFilterButton\_Click(object sender, RoutedEventArgs e)

{

FilterProducts(null);

FilterTB.Clear();

}

private void FilterTB\_TextChanged(object sender, TextChangedEventArgs e)

{

if(FilterTB.Text != string.Empty)

{

DropFilterButton.Visibility = Visibility.Visible;

FilterProducts(FilterTB.Text);

}

else

{

FilterProducts(null);

DropFilterButton.Visibility = Visibility.Collapsed;

}

}

}

Detailed Product View

public partial class DetailedProductView : Window, INotifyPropertyChanged

{

#region PropertyChanged

public event PropertyChangedEventHandler PropertyChanged;

private void OnPropertyChanged([CallerMemberName] string PropertyName = null)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(PropertyName));

}

private bool Set<T>(ref T field, T value, [CallerMemberName] string PropertyName = null)

{

if (Equals(field, value)) return false;

field = value;

OnPropertyChanged(PropertyName);

return true;

}

#endregion

private string \_ImagePath;

public string ImagePath { get => \_ImagePath; set => Set(ref \_ImagePath, value); }

private string \_ProductTitle;

public string ProductTitle { get => \_ProductTitle; set => Set(ref \_ProductTitle, value); }

private decimal? \_Price;

public decimal? Price { get => \_Price; set => Set(ref \_Price, value); }

private string \_Description;

public string Description { get => \_Description; set => Set(ref \_Description, value); }

private bool \_IsButtonEnabled = true;

public bool IsButtonEnabled { get => \_IsButtonEnabled; set => Set(ref \_IsButtonEnabled, value); }

private Products Product;

ProductPreview owner;

public DetailedProductView(Products product, ProductPreview pw)

{

Product = product;

this.owner = pw;

DataContext = this;

InitializeComponent();

ImagePath = product.Image;

ProductTitle = pw.ProductTitle;

Price = product.Price;

Description = product.Description;

IsButtonEnabled = pw.IsButtonEnabled;

DataGrid dg = new DataGrid();

dg.Columns.Add(new DataGridTextColumn { Binding = new Binding() { Path = new PropertyPath("P") }, Width = 400 });

dg.Columns.Add(new DataGridTextColumn { Binding = new Binding() { Path = new PropertyPath("V") } });

dg.Items.Add(new { P = "Тип", V = product.Categories.CategoryName });

dg.Items.Add(new { P = "Производитель", V = product.Manufacturers.CompanyName });

dg.Items.Add(new { P = "Модель", V = product.Model });

dg.Items.Add(new { P = "Основной цвет", V = product.Colors.ColorName });

ParametersPanel.Children.Add(dg);

if (product.Category == 1)

{

TextBlock tb = new TextBlock() { Style = (Style)FindResource("ParametersHeader"), Text = "Экран" };

ParametersPanel.Children.Add(tb);

dg = new DataGrid();

dg.Columns.Add(new DataGridTextColumn { Binding = new Binding() { Path = new PropertyPath("P") }, Width = 400 });

dg.Columns.Add(new DataGridTextColumn { Binding = new Binding() { Path = new PropertyPath("V") } });

dg.Items.Add(new { P = "Тип подсветки", V = product.BacklightTypes.BacklightTypeName });

dg.Items.Add(new { P = "Диагональ экрана (дюйм)", V = product.ScreenSizes.ScreenSizeInInches + '"' });

dg.Items.Add(new { P = "Диагональ экрана (сантиметр)", V = product.ScreenSizes.ScreenSizeInCentimeters + " см" });

dg.Items.Add(new { P = "Разрешение экрана", V = product.ScreenResolutions.ScreenResolution });

if (product.HDRSupport == false || product.HDRSupport == null)

dg.Items.Add(new { P = "Поддержка HDR", V = "Нет" });

else

dg.Items.Add(new { P = "Поддержка HDR", V = "Есть" });

ParametersPanel.Children.Add(dg);

tb = new TextBlock() { Style = (Style)FindResource("ParametersHeader"), Text = "Смарт-функции" };

ParametersPanel.Children.Add(tb);

dg = new DataGrid();

dg.Columns.Add(new DataGridTextColumn { Binding = new Binding() { Path = new PropertyPath("P") }, Width = 400 });

dg.Columns.Add(new DataGridTextColumn { Binding = new Binding() { Path = new PropertyPath("V") } });

if (product.SmartTVSupport == false || product.SmartTVSupport == null)

dg.Items.Add(new { P = "Поддержка SmartTV", V = "Нет" });

else

{

dg.Items.Add(new { P = "Поддержка SmartTV", V = "Есть" });

dg.Items.Add(new { P = "Операционная система", V = product.OperatingSystems.OperatingSystemName });

}

ParametersPanel.Children.Add(dg);

}

}

private void AddToShoppingCartButton\_Click(object sender, RoutedEventArgs e)

{

owner.AddToShoppingCartButton\_Click(sender, e);

}

}

Edit Window

public partial class EditWindow : Window, INotifyPropertyChanged

{

ApplianceStoreEntities db;

private Products \_Product;

public Products Product { get => \_Product; set => Set(ref \_Product, value); }

#region PropertyChanged

public event PropertyChangedEventHandler PropertyChanged;

private void OnPropertyChanged([CallerMemberName] string PropertyName = null)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(PropertyName));

}

private bool Set<T>(ref T field, T value, [CallerMemberName] string PropertyName = null)

{

if (Equals(field, value)) return false;

field = value;

OnPropertyChanged(PropertyName);

return true;

}

#endregion

public EditWindow(Products product, ApplianceStoreEntities DB)

{

db = DB;

DataContext = this;

\_Product = product;

InitializeComponent();

Refresh();

}

private void Refresh()

{

CategoriesCB.ItemsSource = db.Categories.ToArray();

ManufacturersCB.ItemsSource = db.Manufacturers.ToArray();

ColorsCB.ItemsSource = db.Colors.ToArray();

if (Product.Category != 0)

{

CategoriesCB.SelectedIndex = Product.Category - 1;

ManufacturersCB.SelectedIndex = (int)Product.Manufacturer - 1;

ColorsCB.SelectedIndex = (int)Product.Color - 1;

ImageTB.Text = Product.Image.Substring(22);

}

}

private void LoadImageButton\_Click(object sender, RoutedEventArgs e)

{

OpenFileDialog openDialog = new OpenFileDialog();

openDialog.Filter = "Файлы изображений|\*.bmp;\*.png;\*.jpg";

if (openDialog.ShowDialog() != true)

return;

int found;

found = openDialog.FileName.Replace('\\', '/').IndexOf("/Data/Images/");

ImageTB.Text = openDialog.FileName.Replace('\\', '/').Substring(found);

}

private void SaveButton\_Click(object sender, RoutedEventArgs e)

{

try

{

if (CategoriesCB.SelectedIndex != -1 && ManufacturersCB.SelectedIndex != -1 && ModelTB.Text != string.Empty)

{

Product.Category = ((Categories)CategoriesCB.SelectedItem).CategoryID;

Product.Manufacturer = ((Manufacturers)ManufacturersCB.SelectedItem).ManufacturerID;

Product.Model = ModelTB.Text;

Product.Price = decimal.Parse(PriceTB.Text.Replace('.',','));

Product.Width = decimal.Parse(WidthTB.Text.Replace('.', ','));

Product.Height = decimal.Parse(HeightTB.Text.Replace('.', ','));

Product.Depth = decimal.Parse(DepthTB.Text.Replace('.', ','));

Product.Warranty = short.Parse(WarrantyTB.Text);

Product.Color = ((Models.Colors)ColorsCB.SelectedItem).ColorID;

Product.Image = "pack://application:,,," + ImageTB.Text;

db.SaveChanges();

MessageBox.Show("Успешно!");

}

else

MessageBox.Show("Заполните обязательные поля", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

catch

{

MessageBox.Show("Введите корректные данные", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

}

RegistrationWindow

public partial class RegistrationWindow : Window

{

ApplianceStoreEntities db;

public RegistrationWindow(ApplianceStoreEntities context)

{

InitializeComponent();

db = context;

}

private void regButton\_Click(object sender, RoutedEventArgs e)

{

if (SurnameTB.Text == string.Empty || FirstnameTB.Text == string.Empty || PhoneTB.Text == string.Empty || PasswordTB.Password == string.Empty)

MessageBox.Show("Заполните необходимые поля", "Обязательные поля не заполнены", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

try

{

var client = new Clients

{

ClientID = (from c in db.Clients select c).ToList().Last().ClientID + 1,

Surname = SurnameTB.Text,

Firstname = FirstnameTB.Text,

Lastname = LastnameTB.Text,

PhoneNumber = PhoneTB.Text,

Password = Encryption.Encode(PasswordTB.Password)

};

db.Clients.Add(client);

db.SaveChanges();

this.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), ex.Message);

}

}

}

}

ShoppingCart

public partial class ShoppingCart : Window, INotifyPropertyChanged

{

#region PropertyChanged

public event PropertyChangedEventHandler PropertyChanged;

private void OnPropertyChanged([CallerMemberName] string PropertyName = null)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(PropertyName));

}

private bool Set<T>(ref T field, T value, [CallerMemberName] string PropertyName = null)

{

if (Equals(field, value)) return false;

field = value;

OnPropertyChanged(PropertyName);

return true;

}

#endregion

private decimal DisplayedSum = 0;

private int DisplayedNum = 0;

private decimal DisplayedDiscount = 0;

public List<ShoppingCartProduct> ShoppingCartList = new List<ShoppingCartProduct>();

public ObservableCollection<Products> Products = new ObservableCollection<Products>();

ApplianceStoreEntities db = new ApplianceStoreEntities();

public ShoppingCart(ObservableCollection<Products> products, Clients client)

{

InitializeComponent();

Products = products;

DisplayedDiscount = client.DiscountLevels.PercentDiscount;

DiscountTextBlock.Text = DisplayedDiscount.ToString();

UpdateProducts();

}

public void UpdateProducts()

{

ProductsPanel.Children.Clear();

ShoppingCartList.Clear();

foreach (var product in Products)

{

var p = new ShoppingCartProduct(product, this);

ProductsPanel.Children.Add(p);

ShoppingCartList.Add(p);

}

UpdateInfo();

}

public void UpdateInfo()

{

DisplayedSum = 0;

DisplayedNum = 0;

if (ShoppingCartList.Count != 0)

{

foreach (var item in ShoppingCartList)

{

if (item.IsSelected)

{

DisplayedNum += item.ProductCount;

DisplayedSum += item.Product.Price / 100 \* (100 - DisplayedDiscount) \* item.ProductCount ?? 0;

}

NumTextBox.Text = DisplayedNum.ToString() + ' ' + RussianCases.GenerateNumEnding(DisplayedNum, "товар", "товара", "товаров");

SumTextBox.Text = String.Format("на {0:#.00} ₽", DisplayedSum);

}

}

}

private void SelectAllCheckBox\_Click(object sender, RoutedEventArgs e)

{

if (((CheckBox)sender).IsChecked == true)

{

foreach (var item in ShoppingCartList)

{

item.IsSelected = true;

}

}

else if (((CheckBox)sender).IsChecked == false)

{

foreach (var item in ShoppingCartList)

{

item.IsSelected = false;

}

}

else

{

foreach (var item in ShoppingCartList)

{

item.IsSelected = true;

}

}

UpdateInfo();

}

public void CheckCheckBoxes()

{

for (int i = 0; i < ShoppingCartList.Count - 1; i++)

{

if (ShoppingCartList[i].IsSelected == ShoppingCartList[i + 1].IsSelected)

continue;

else

{

SelectAllCheckBox.IsChecked = null;

return;

}

}

SelectAllCheckBox.IsChecked = ShoppingCartList[0].IsSelected;

}

private void CheckoutButton\_Click(object sender, RoutedEventArgs e)

{

if (ShoppingCartList.Count != 0)

{

if (MessageBox.Show("Вы уверены что хотите распечатать этот заказ?", "Подтвердите действие",

MessageBoxButton.OKCancel, MessageBoxImage.Question) == MessageBoxResult.OK)

{

try

{

int \_ClientID;

if (((Catalog)Owner).IsAdmin)

\_ClientID = 0;

else

\_ClientID = ((Catalog)Owner).Client.ClientID;

var purchase = new Purchases

{

PurchaseID = (from p in db.Purchases

select p.PurchaseID).ToList().Last() + 1,

ClientID = \_ClientID

};

db.Purchases.Add(purchase);

db.SaveChanges();

decimal sum = 0;

foreach (ShoppingCartProduct p in ShoppingCartList)

{

if (p.IsSelected)

{

PurchaseItems item = new PurchaseItems

{

PurchaseID = (from pur in db.Purchases

select pur).ToList().Last().PurchaseID,

ProductID = p.Product.ProductID,

ProductCount = p.ProductCount

};

db.PurchaseItems.Add(item);

sum += item.ProductCount \*

(from product in db.Products

where product.ProductID == item.ProductID

select product)

.Single().Price ?? 0;

}

}

var client = (from c in db.Clients

where c.ClientID == \_ClientID

select c)

.Single();

client.Account += sum;

List<DiscountLevels> lvls = (from l in db.DiscountLevels select l).ToList();

bool IsUpgraded = false;

if (client.DiscountLevel != lvls.Count - 1)

{

for (int i = 0; i < lvls.Count; i++)

{

if (client.Account >= lvls[i].AmountOfPurchases)

{

IsUpgraded = true;

client.DiscountLevel = lvls[i].LevelID;

}

}

if (IsUpgraded)

MessageBox.Show("Для получения товарного чека подойдите на кассу. " +

"\nУровень повышен! Теперь уровень вашего дисконта - " + lvls[client.DiscountLevel].Name +

".\nВаша скидка составляет " + lvls[client.DiscountLevel - 1].PercentDiscount + "%.",

"Заказ оформлен успешно.", MessageBoxButton.OK, MessageBoxImage.Information);

else

MessageBox.Show("Для получения товарного чека подойдите на кассу. " +

"\nДо повышения уровня осталось приобрести товаров на " +

(lvls[client.DiscountLevel + 1].AmountOfPurchases - client.Account),

"Заказ оформлен успешно.", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

MessageBox.Show("Для получения товарного чека подойдите на кассу",

"Заказ оформлен успешно.", MessageBoxButton.OK, MessageBoxImage.Information);

db.SaveChanges();

this.Close();

AuthorizationWindow a = new AuthorizationWindow();

a.Show();

((Catalog)Owner).Close();

PrintToWord();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Ошибка - " + ex.Message);

}

}

else

MessageBox.Show("Заказ отменён");

}

else

MessageBox.Show("Корзина пуста!");

}

private void PrintToWord()

{

try

{

Word.Application app = new Word.Application();

string source = Environment.CurrentDirectory + @"\CheckoutTemplate.dotx";

Word.Document doc = app.Documents.Add(source);

doc.Activate();

Word.Bookmarks bookmarks = doc.Bookmarks;

var lastPurchase = (from purchase

in db.Purchases

select purchase).ToList().Last();

decimal sum = 0;

foreach (var item in lastPurchase.PurchaseItems)

sum += Math.Round(item.Products.Price / 100 \*

(100 - DisplayedDiscount) \* item.ProductCount ?? 0, 2);

foreach (Word.Bookmark bookmark in bookmarks)

{

switch (bookmark.Name)

{

case "Адрес": bookmark.Range.Text = "Адрес"; break;

case "Дата": bookmark.Range.Text = (from p in db.Purchases select p.PurchaseDate).ToList().Last().ToString(); break;

case "КоличествоНаименований": bookmark.Range.Text = lastPurchase.PurchaseItems.Count.ToString(); break;

case "НомерЧека": bookmark.Range.Text = lastPurchase.PurchaseID.ToString(); break;

case "Продавец": bookmark.Range.Text = "Наименование магазина - продавца"; break;

case "Сумма": bookmark.Range.Text = sum.ToString(); break;

case "СуммаТекстом": bookmark.Range.Text = RussianCases.RubPhrase(sum); break;

case "Телефон": bookmark.Range.Text = "Контактный телефон"; break;

}

}

Word.Table table = doc.Tables[1];

int i = 2;

foreach(var item in lastPurchase.PurchaseItems)

{

table.Cell(i, 1).Range.Text = (i - 1).ToString();

table.Cell(i, 2).Range.Text = item.ProductID.ToString();

table.Cell(i, 3).Range.Text = new ShoppingCartProduct((from p in Products

where p.ProductID == item.ProductID

select p).Single(), this).ProductTitle;

table.Cell(i, 4).Range.Text = item.Products.Price.ToString();

table.Cell(i, 5).Range.Text = item.ProductCount.ToString();

table.Cell(i, 6).Range.Text = DisplayedDiscount.ToString() + '%';

table.Cell(i, 7).Range.Text = Math.Round((item.Products.Price / 100 \* (100 - DisplayedDiscount) \* item.ProductCount ?? 0), 2).ToString();

table.Rows.Add();

i++;

}

table.Rows[i].Cells[1].Merge(table.Rows[i].Cells[6]);

table.Rows[i].Range.Font.Bold = 1;

table.Cell(i, 2).Range.Text = sum.ToString();

doc.Close();

doc = null;

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString(), "Ошибка - " + ex.Message);

}

}

}

ProductPreview

public partial class ProductPreview : UserControl, INotifyPropertyChanged

{

#region PropertyChanged

public event PropertyChangedEventHandler PropertyChanged;

private void OnPropertyChanged([CallerMemberName] string PropertyName = null)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(PropertyName));

}

private bool Set<T>(ref T field, T value, [CallerMemberName] string PropertyName = null)

{

if (Equals(field, value)) return false;

field = value;

OnPropertyChanged(PropertyName);

return true;

}

#endregion

ApplianceStoreEntities db = new ApplianceStoreEntities();

private Products \_Product;

public Products Product { get => \_Product; set => Set(ref \_Product, value); }

private string \_ProductTitle;

public string ProductTitle { get => \_ProductTitle; set => Set(ref \_ProductTitle, value); }

private bool \_IsButtonEnabled = true;

public bool IsButtonEnabled { get => \_IsButtonEnabled; set => Set(ref \_IsButtonEnabled, value);}

Catalog Owner;

public ProductPreview(Products product, Catalog owner)

{

Owner = owner;

Product = product;

DataContext = this;

InitializeComponent();

if (Product != null)

{

ManufacturerName.Text = Product.Manufacturers.CompanyName;

Warranty.Text = Product.Warranty.ToString() + " месяцев";

if (Product.Category == 1)

{

ProductTitle = Product.ScreenSizes.ScreenSizeInInches.ToString() + "\" (" +

Product.ScreenSizes.ScreenSizeInCentimeters.ToString() + " см) " +

"Телевизор " + Product.Manufacturers.CompanyName + ' ' + Product.Model + ' ' +

Product.Colors.ColorName + " [" + Product.ScreenResolutions.ScreenResolutionName + ", " +

Product.ScreenResolutions.ScreenResolution + ", " + Product.BacklightTypes.BacklightTypeName + ", " +

Product.OperatingSystems.OperatingSystemName + ']';

Property1.Text = "Разрешение экрана";

Property2.Text = "Поддержка SmartTV";

Property3.Text = "Поддержка HDR";

Value1.Text = Product.ScreenResolutions.ScreenResolutionName + " (" + Product.ScreenResolutions.ScreenResolution + ')';

if (Product.SmartTVSupport == false || Product.SmartTVSupport == null)

Value2.Text = "Нет";

else

Value2.Text = "Есть";

if (Product.HDRSupport == false || Product.HDRSupport == null)

Value3.Text = "Нет";

else

Value3.Text = "Есть";

}

else if (Product.Category == 2)

{

ProductTitle = "Холодильник " + Product.FreezerLocations.FreezerLocationName + ' ' + Product.Manufacturers.CompanyName + ' ' +

Product.Model + ' ' + Product.Colors.ColorName + " [" + (Product.RefrigeratorVolume + Product.FreezerVolume).ToString() + " л, " +

Product.Width + " см x " + Product.Height + " см x " + Product.Depth + " см]";

Property1.Text = "Компоновка";

Property2.Text = "Инверторный компрессор";

Property3.Text = "Зона свежести";

TextInfo ti = CultureInfo.CurrentCulture.TextInfo;

Value1.Text = ti.ToTitleCase(Product.FreezerLocations.FreezerLocationName);

if (Product.InverterCompressor == false || Product.InverterCompressor == null)

Value2.Text = "Нет";

else

Value2.Text = "Есть";

if (Product.FreshnessZone == false || Product.FreshnessZone == null)

Value3.Text = "Нет";

else

Value3.Text = "Есть";

}

else

{

ProductTitle = Product.Manufacturers.CompanyName + ' ' + Product.Model + ' ' + Product.Colors.ColorName +

" [" + Product.Width + " см x " + Product.Height + " см x " + Product.Depth + " см]";

Property1.Text = "Категория";

Property2.Text = "Энергетический класс";

Property3.Text = "Потребление энергии";

Value1.Text = Product.Categories.CategoryName;

Value2.Text = Product.EnergyClasses.EnergyClassName;

Value3.Text = Product.PowerConsumption.ToString();

}

if(Owner.IsAdmin)

{

EditButton.Visibility = Visibility.Visible;

DeleteButton.Visibility = Visibility.Visible;

}

}

}

private void MainBorder\_MouseUp(object sender, System.Windows.Input.MouseButtonEventArgs e)

{

DetailedProductView d = new DetailedProductView(Product, this);

d.Owner = Owner;

d.Show();

}

public void AddToShoppingCartButton\_Click(object sender, RoutedEventArgs e)

{

Owner.ShoppingCartList.Add(Product);

IsButtonEnabled = false;

}

private void EditButton\_Click(object sender, RoutedEventArgs e)

{

EditWindow ew = new EditWindow(Product, Owner.db);

ew.Show();

}

private void DeleteButton\_Click(object sender, RoutedEventArgs e)

{

if(MessageBox.Show("Вы уверены, что хотите удалить данный товар?", "Требуется подтверждение",

MessageBoxButton.YesNo, MessageBoxImage.Exclamation) == MessageBoxResult.Yes)

{

var deleted\_product = (from p in db.Products where Product.ProductID == p.ProductID select p).First();

db.Products.Remove(deleted\_product);

db.SaveChanges();

Owner.ChangeCategory(Owner.SelectedCategory.CategoryID);

}

}

}

ShoppingCartProduct

public partial class ShoppingCartProduct : UserControl, INotifyPropertyChanged

{

#region PropertyChanged

public event PropertyChangedEventHandler PropertyChanged;

private void OnPropertyChanged([CallerMemberName] string PropertyName = null)

{

PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(PropertyName));

}

private bool Set<T>(ref T field, T value, [CallerMemberName] string PropertyName = null)

{

if (Equals(field, value)) return false;

field = value;

OnPropertyChanged(PropertyName);

return true;

}

#endregion

private short \_ProductCount = 1;

public short ProductCount { get => \_ProductCount; set => Set(ref \_ProductCount, value); }

private string \_ProductTitle;

public string ProductTitle { get => \_ProductTitle; set => Set(ref \_ProductTitle, value); }

private bool \_IsSelected = true;

public bool IsSelected { get => \_IsSelected; set => Set(ref \_IsSelected, value); }

private Products \_Product;

public Products Product { get => \_Product; set => Set(ref \_Product, value); }

ShoppingCart Owner;

public ShoppingCartProduct(Products product, ShoppingCart owner)

{

DataContext = this;

Owner = owner;

Product = product;

InitializeComponent();

if (Product.Category == 1)

{

ProductTitle = Product.ScreenSizes.ScreenSizeInInches + "\" " + "Телевизор ";

}

else if (Product.Category == 2)

{

ProductTitle = "Холодильник " + Product.FreezerLocations.FreezerLocationName + ' ';

}

else { ProductTitle = "fucked up"; }

ProductTitle += Product.Manufacturers.CompanyName + ' ' + Product.Model + ' ' + Product.Colors.ColorName;

}

private void PlusButton\_Click(object sender, RoutedEventArgs e)

{

ProductCount++;

}

private void MinusButton\_Click(object sender, RoutedEventArgs e)

{

if (ProductCount != 1)

{

ProductCount--;

}

}

private void CountTextBlock\_TextChanged(object sender, TextChangedEventArgs e)

{

if(ProductCount == 0)

IsSelected = false;

else

IsSelected = true;

Owner.UpdateInfo();

}

private void DeleteButton\_Click(object sender, RoutedEventArgs e)

{

Owner.Products.Remove(Product);

Owner.UpdateProducts();

((Catalog)Owner.Owner).DisplayProducts();

}

private void CheckBox\_Click(object sender, RoutedEventArgs e)

{

Owner.UpdateInfo();

Owner.CheckCheckBoxes();

}

}