|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| СОДЕРЖАНИЕ   |  |  | | --- | --- | |  | лист | | Введение | 4 | | 1 Постановка задачи | 6 | | 1.1 Описание предметной области | 6 | | 1.2 Диаграмма прецедентов предметной области | 7 | | 1.3 Описание входной информации | 8 | | 1.4 Описание выходной информации | 8 | | 1.5 Общие требования к программному продукту | 9 | | 1.6 Описание структуры базы данных | 11 | | 1.7 Контрольный пример | 14 | | 2 Экспериментальный раздел | 18 | | 2.1 Описание программы | 18 | | 2.2 Протокол тестирования программного продукта | 20 | | 2.3 Руководство пользователя | 32 | | Заключение | 43 | | Приложение А. Выходные данные контрольного примера | 44 | | Список использованных источников | 46 | |  |

ВВЕДЕНИЕ

Характерной чертой нашего времени являются информационные технологии. В понятие «информационные технологии» входят компьютерные технологии (оборудование и программное обеспечение) и технологии связи для передачи информации. Необходимость в информатизации обусловливается многими факторами. Огромными потоками информации, которые циркулируют в различных сферах деятельности, временем, затрачиваемым на процесс обработки этой информации и сложностями, связанными с хранением информации.

Внедрение информационной системы позволяет решить эти задачи. В результате чего сокращается время на обработку информации, осуществляется хранение больших объемов информации, ускоряется поиск необходимой информации, и выдача ее в удобном для пользователя виде. В основе любой информационной системы лежит база данных. База данных представляет собой совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов, и их взаимосвязей в рассматриваемой предметной области.

Процесс создания программной системы предполагает несколько этапов:

* исследование предметной области, той сферы, куда будет внедряться информационная система, сбор сведений об объектах, фактах, событиях и процессах, которые участвуют в рассматриваемом виде деятельности;
* проектирование базы данных. Осуществляется выбор модели данных: иерархическая, сетевая или реляционная. Наиболее удобной моделью является реляционная модель. В реляционной базе данных информация организована в виде отношений. Проектирование базы данных предполагает определение отношений и связей между ними.

Основная цель дипломного проекта - сократить временные затраты сотрудника счет автоматизации основных процессов: «Создание договора», «Предоставление товаров для пользователя», «Регистрация товаров».

Для достижения цели необходимо решить следующие задачи:

* исследовать предметную область;
* спроектировать базу данных, которая будет содержать информацию о товарах, заявках, пользователях;
* автоматизировать запись на прием;
* автоматизировать выдачу талона на прием к врачу.

Разработка приложения, для регистрации и обработке товара. Приложение осуществляет следующие функции: добавление информации в базу данных, удаление информации из базы данных, выдача информации в необходимом для пользователя виде, по заданным запросам. Разработка программной системы осуществлялась в Visual studio 2019 community.

1 Постановка задачи

1.1 Описание предметной области

Требуется создать программу для автоматизации информационной системы агентства по обработке и регистрации товаров. Целью задачи является предоставление необходимой информации о существующих на данный момент товаров и их обработка.

Пользователями программы будут являться:

* администратор;
* пользователь;
* поставщик.

В функции администратора входит регистрация товара и обработка заявок от поставщиков.

В функции пользователя входит просмотр и обработка товара.

В функции поставщика входит регистрация и обработка товара.

Администратор регистрирует товар, заполняя следующую информацию:

* номер товара;
* название товара;
* описание товара;
* количество товара на складе;
* фото товара;
* сумма за 1ед товара;
* общую сумму;
* тип товара;
* описание производителя.

Описание данных списка производителя:

* номер производителя;
* производитель;
* модель;
* уникальный номер;
* номер товара;
* номер заявки.

При работе с программой администратор должен иметь возможность решать следующие задачи:

* добавлять, удалять, редактировать записи;
* составлять отчет о регистрации товара.

Входными данными для создания программы являются - данные о товаре.

Выходными данными программы является – договор о купли-продажи.

1.2 Диаграмма прецедентов предметной области

Диаграмма прецедентов предметной области представлена на рисунке 1.2.1

Рисунок 1.2.1 - Функциональная модель предметной области

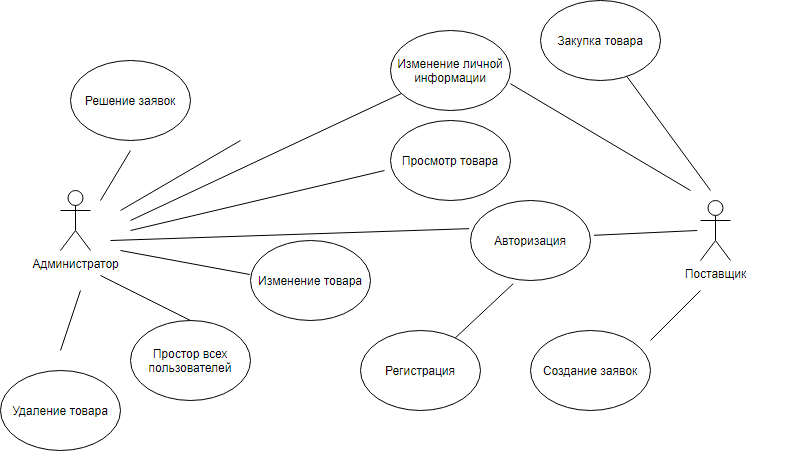


Рисунок 1.2.1 - Диаграмма прецедентов

1.3 Описание входной информации

Входные данные представлены в таблице 1.3.1.

Таблица 1.3.1 – Описание входных документов

|  |  |  |
| --- | --- | --- |
| Наименование документа (шифр) | Периодичность поступления документа | Откуда поступает документ |
| Договор | При введении данных | От приложения |
| Договор о приёме товара | По мере необходимости | От администратора |

1.4 Описание выходной информации

Выходные документы в данной задаче представлены в виде талона на прием к врачу. Описание выходного документа представлено в таблице 1.4.1.

Таблица 1.4.1 – Описание выходного документа

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование документа (шифр) | Периодичность выдачи документа | Кол-во экз. | Куда передаются | Поля сортировки | Поля группировки | Итоги |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Договор о купли-продажи | После покупки товара | 1 | пользователю | – | – | – |

Форма выходного документа представлена на рисунке 1.4.1.

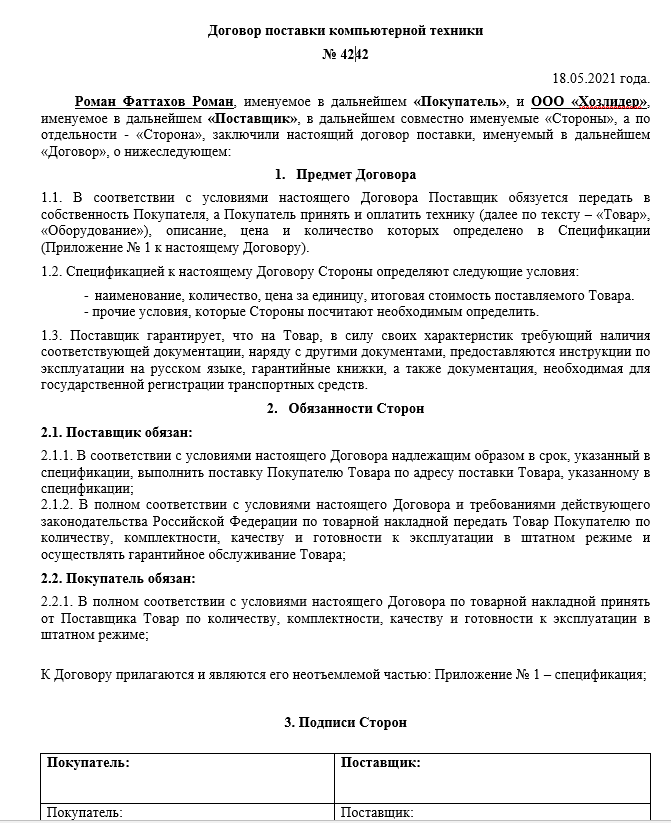


Рисунок 1.4.1 – Договор

1.5 Общие требования к программному продукту

Программа должна упрощать процесс регистрации и обработки товаров.

Разрабатываемое ПО должно обеспечивать:

* корректную регистрацию товаров;
* программа должна сохранять и отображать необходимую информацию.

Программный продукт должен обладать высокой степенью надежности. Во всем программном продукте должно предусмотрено:

* ограничение ввода данных в поля;
* проверка введенной информации на соответствие;
* блокировку некорректных действий пользователя при работе с системой.

Для корректной работы программы компьютер должен соответствовать следующим минимальным требованиям:

* процессор 2 ГГц;
* 1 ГБ ОЗУ;
* 50 МБ свободного пространства на диске.

Для нормальной работы приложения необходимо наличие на компьютере пользователя операционной системы Microsoft Windows XP и выше.

Функциональные возможности:

* программный продукт должен формировать и отображать выходной документ;
* должен быть обеспечен просмотр базы данных с возможностью добавления, редактирования;
* должен быть обеспечен вывод данных в MS Excel, MS Word.

Требования к надежности:

* программа должна обрабатывать ошибочные действия пользователя и сообщать ему об этом;
* программа должна обеспечивать контроль входной и выходной информации.

Требования к практичности - программа должна работать в многооконном графическом режиме и поддерживать работу, как клавиатуры, так и манипулятора типа "мышь", сенсорного экрана.

Требования к информационной и программной совместимости: обеспечить работу с таблицами баз данных типа СУБД MS SQL и совместимых типов в качестве входной и выходной информации.

1.6 Описание структуры базы данных

Структура база данных представлена в таблице 1.6.1.

Таблица 1.6.1 – Структура базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Описание поля | Тип данных | Размер поля | Тип ключа[[1]](#footnote-1) |
| 1 | 2 | 3 | 4 | 5 |
| login (Пользователь) | | | | |
| ID | Номер пользователя | INT | 4 | PK |
| Login | Логин | VARCHAR | 30 |  |
| Pass | Пароль | VARCHAR | 30 |  |
| Email | email | VARCHAR | 150 |  |
| Phone | Телефон | INT | 4 |  |
| DateReg | Дата регистрации | DATE | 6 |  |
| Adres | Адрес | VARCHAR | 300 |  |
| Datezax | Последний актив | DATETIME | 6 |  |

Продолжение таблицы 1.6.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Royale | Должность | VARCHAR | 10 |  |
| kat | Количество купленных товаров | INT | 4 |  |
| zav | Количество оставленных заявок | INT | 4 |  |
| Foto | Фото профиля | VARCHAR | 300 |  |
| Tovar(Товары) | | | | |
| ID | Номер товара | INT | 4 | PK |
| Title | Название товара | VARCHAR | 300 |  |
| Comment | Описание товара | VARCHAR | 1000 |  |
| Cost | Количество товара | INT | 4 |  |
| Foto | Фото товара | VARCHAR | 255 |  |
| Sum | Цена за 1ед товара | VARCHAR | 150 |  |
| TotalSum | Полная сумма товара | INT | 4 | PK |
| Type | Тип товара | INT | 4 | FK |
| ID\_man | Номер производителя | INT | 4 | FK |
| Zavki(Заявки) | | | | |
| ID | Номер заявки | INT | 4 | PK |
| Title | Название | VARCHAR | 300 |  |

Продолжение таблицы 1.6.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Comment | Описание | VARCHAR | 1000 |  |
| Cost | Количество товара | INT | 4 |  |
| Foto | Фото заявки | VARCHAR | 255 |  |
| Sum | Сумм за 1 ед | VARCHAR | 150 |  |
| TotalSum | Полная сумма | INT | 4 |  |
| Print | Печать | INT | 4 |  |
| ID\_man | Номер производителя | INT | 4 | FK |
| Type | Номер типа | INT | 4 | FK |
| Status | Статус | INT | 4 |  |
| Kat1(Закупки) | | | | |
| ID | Номер закупки | INT | 4 | PK |
| ID\_TOVAR | Номер товара | INT | 4 | FK |
| Sum | Сумма за 1 ед | VARCHAR | 150 |  |
| TotalSum | Полная сумма | INT | 4 |  |
| ID\_user | Номер пользователя | INT | 4 | FK |
| Cost | Количество товара | INT | 4 |  |
| Status(Статус) | | | | |
| ID | Номер статуса | INT | 4 | PK |
| Status | Название статуса | VARCHAR | 100 |  |

Продолжение таблицы 1.6.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Foto | Фото статуса | VARCHAR | 200 |  |
| Type(Тип товара) | | | | |
| ID | Номер типа | INT | 4 | PK |
| Title | Название типа | VARCHAR | 300 |  |

Схема отношений представлена на рисунке 1.6.1

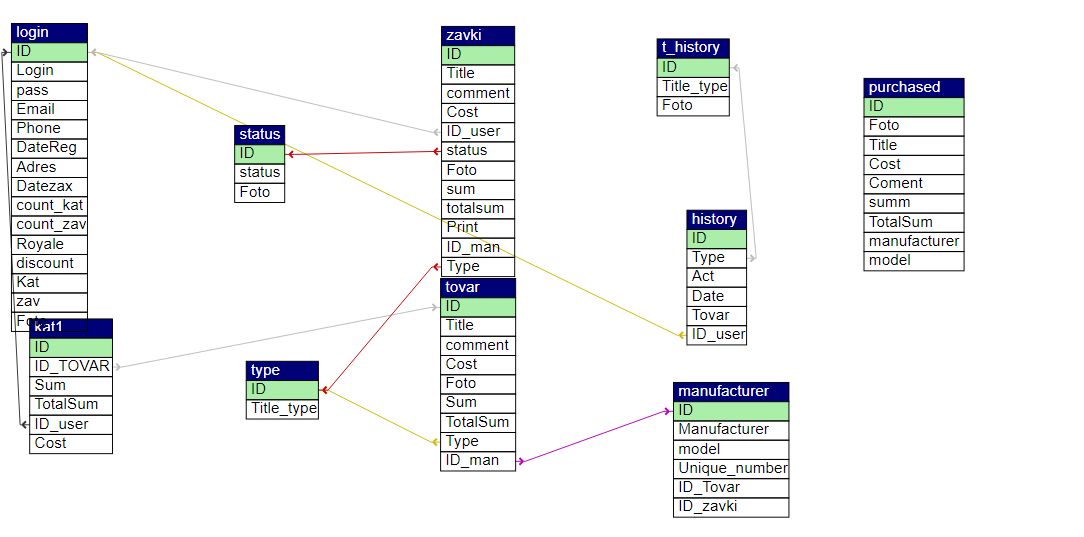


Рисунок 1.6.1 – Схема отношения

1.7 Контрольный пример

При построении контрольного примера входные данные и предполагаемые результаты задаются в виде таблиц. Эти результаты в дальнейшем должны совпадать с результатами программного продукта.

Входная информация контрольного примера представлена в таблицах 1.7.1 и 1.7.7.

Таблица 1.7.1 – Входная информация по пользователям

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Логин | Пароль | Email | Телефон | Дата регистрации | Адрес | | Последняя дата | | | | должность | Покупок | | заявок | | | Фото |
| 1 | 2 | 3 | 4 | 5 | 6 | | | | 7 | | 8 | 9 | | 10 | | | 11 |
| Продолжение таблицы 1.7.1 | | | | | | | | | | | | | | | | | |
| Roma | 123 | dasdas@kdas.ru | 241452551515 | 2021-05-14 | | dfsdfsdfsdfsd@das.ru | | 2021-05-16 12:22:56 | | user | | | 4 | | 0 | 16147553091060.jpg | |
| Admin | admin | dsasda@as.ru | 241452551515 | 2021-05-14 | | dfsdfsdfsdfsd@das.ru | | 2021-05-16 12:22:56 | | a | | | 0 | | 0 | 16147553091060.jpg | |
| Roma\_P | 123p | dasdasd@mail.ru | 241452551515 | 2021-05-14 | | dfsdfsdfsdfsd@das.ru | | 2021-05-16 12:22:56 | | postav | | | 5 | | 6 | 16147553091060.jpg | |

Таблица 1.7.2 – Входная информация товара

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Название | Описание | Кол-во | Фото | Сумма за 1 ед | Полная сумма | Тип |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Футболка s размера | Эта вещь нужна каждому | 100 | syringe\_medical\_5333.png | 250 | 25000 | 5 |
| Игрушечная машинка | Идеально пойдёт ребёнку | 25 | saw\_5333.png | 1000 | 25000 | 2 |
| Пачка бумаги A4 | Бумага | 300 | load\_433.png | 300 | 90000 | 5 |

Таблица 1.7.3 – Входная информация заявки

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Название | Описание | Кол-во | Фото | Сумма за 1 ед | Полная сумма | Тип | Статус |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Игровой монитор | Высококачественный кожаный игровой монитор | 2 | mon\_5333.png | 20000 | 40000 | 1 | 1 |
| сухофрукты | Ням ням | 10000 | eat\_523.png | 150 | 1500000 | 2 | 2 |
| Mustang Energy | безалкогольный тонизирующий (энергетический) | 58 | Energy.png | 100 | 58000 | 6 | 3 |

Таблица 1.7.4 – Входная информация закупки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер товара | Сумма | Полная сумма | Номер пользователя | Количество |
| 1 | 2 | 3 | 4 | 5 |
| 4 | 1000 | 15000 | 14 | 5 |
| 3 | 500 | 500 | 12 | 1 |
| 88 | 128 | 128000 | 2 | 1000 |
| 24241 | 100 | 5000 | 15 | 50 |

Таблица 1.7.5 – Входная информация статуса

|  |  |
| --- | --- |
| Статус | Фото |
| 1 | 2 |
| Ожидание | Ожидание.png |
| Отменено | Отмена.png |
| Подтверждено | Принятие.png |

Таблица 1.7.6 – Входная информация типа

|  |  |
| --- | --- |
| Номер | Описание |
| 1 | 2 |
| 1 | Одежда, обувь, аксессуары |
| 2 | Продукты питания, напитки |
| 3 | Строительство и ремонт |
| 4 | Сырье и материалы |

Таблица 1.7.7 – Входная информация статуса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Производитель | Модель | Уникальный номер | Номер товара | Номер заявки |
| Китай | шёлк | 1241241 | 15 | 10 |
| Nvdia | Rx3080 | 2141241 | 14 | 11 |
| LG | Xl21f | 1516352 | 11 | нет |
| Очаково | S2 | 1456366 | 5 | 1 |

Выходная информация контрольного примера представлена в приложении A.

2 Экспериментальный раздел

2.1 Описание программы

Программа имеет модульную структуру. При ее запуске выполняется проект Tovar.exe.

Все процедуры можно разделить на основные и сервисные. К сервисным процедурам относятся процедуры визуализации форм, контроль ввода, активация компонентов Devexpress.

При запуске программы вызывается модуль Program.cs, форма Main.cs является главным окном программы. Из модуля Unit1 производится вызов всех модулей, в которых выполняются основные функции программы.

На рисунке 2.1.1 представлена модульная схема программы

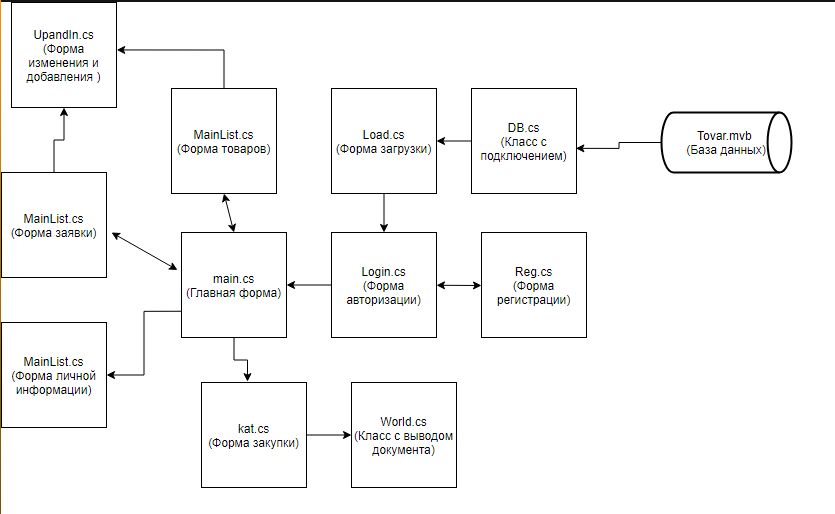


Рисунок 2.1.1 – Модульная схема программы

Описание модулей представлено в таблице 2.1.1.

Таблица 2.1.1 — Описание модулей

|  |  |
| --- | --- |
| Модуль | Назначение |
| 1 | 2 |
| Login.cs(авторизация) | |
| Button\_Click | Переход на главную форму |
| materialLabel1\_Click | Переход на форму «регистрация пользователя» |
| Check2\_OnChange | Сохранение данных |
| Reg.cs(Регистрация) | |
| bunifuThinButton21\_Click | Выбор фотографии |
| TForm3. BitBtn1Click(Sender: TObject) | Добавление пользователя |
| bunifuThinButton22\_Click | Подтверждение аккаунта |
| Maint(Главная форма) | |
| Foto | Загрузка фотографии |
| Chose | Очистка информации |
| Form3\_Activated | Загрузка сценариев |
| button1\_Click | Переход на форму «товары» |
| button2\_Click | Переход на форму «личный кабинет» |
| button4\_Click | Переход на форму «закупки» |
| button3\_Click | Переход на форму «заявки» |
| bunifuFlatButton4\_Click | Переход на форму «пользователи» |
| Mainlist.cs(Товары) | |
| loaded | Загрузка страницы |
| gClick | Переход на форму «добавление товара» |
| imageComboBoxEdit1\_SelectedIndexChanged | Поиск по фильтрации |
| UpandIn(Добавление и изменение) | |
| button1\_Click | Возвращение на прошлую форму |
| Form2\_Load | Загрузка страницы |
| materialFlatButton3\_Click | Выбор фотографии |
| materialFlatButton2\_Click | Добавление товара |
| Info(личный кабинет) | |

Продолжение таблицы 2.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| materialFlatButton3\_Click | Выбор фотографии |
| materialFlatButton2\_Click | Изменение информации |
| SecondList(Закупки) | |
| Form4\_Load | Загрузка формы |
| gClick1 | Переход на форму «добавление заявки» |
| gClick | Рассмотрение заявок |
| gClick2 | Переход на форму «Информация» |
| ViewList(Пользователи) | |
| comboBox1\_SelectedIndexChanged | Фильтрация |
| bunifuCustomDataGrid1\_CellClick | Выбор конкретного пользователя |
| materialFlatButton1\_Click | Возвращение на предыдущая форму |

2.2 Протокол тестирования программного продукта

В ходе тестирования приложения на корректных и некорректных данных не было обнаружено ошибок, которые влияли бы на работу самого приложения и всей системы.

Данное приложение удовлетворяет всем предъявленным требованиям, имеет комфортный интерфейс и интуитивно понятный функционал, исключает появление системных ошибок.

В таблицах 2.2 – 2.8 представлены протоколы тестирования.

Таблица 2.1 – Общая информация о тестировании

|  |  |
| --- | --- |
| Название проекта | Hozlider |
| Номер версии | 1.0 |
| Имя тестера | Фаттахов Роман |
| Даты тестирования | 17.05.2021 |

Таблица 2.2 – Протокол тестирования авторизации на корректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| 1 | 2 |
| Наименование | Описание |

Продолжение таблицы 2.2

|  |  |
| --- | --- |
| 1 | 2 |
| Test Case # | Hozlider \_A\_1 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка авторизации в модуле login.cs («Авторизация») с корректными пользовательскими данными |
| Резюме испытания | Необходимо добиться корректного поведения приложения при вводе корректных данных |
| Шаги тестирования | Ввод корректных данных в текстовые поля;  нажатие кнопки «Войти». |
| Данные тестирования | Логин: admin;  Пароль: admin. |
| Ожидаемый результат | Приложение должно открыть форму модуля Main.cs («Администратор») |
| Фактический результат | В результате тестирования программа открыла форму MainList.cs («Администратор») |
| Предпосылки | Открытие страницы login.cs («Авторизация»). |
| Постусловия | Система не зависает, находится в состоянии полной работоспособности. |
| Статус (Pass/Fail) | Pass |

Проведём тестирование авторизации с правильным логином и паролем администратора. Результаты изображены на рисунках 2.1 – 2.2.

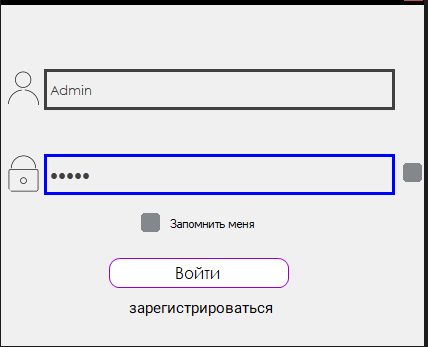


Рисунок 2.1 – Авторизация с вводом правильных данных

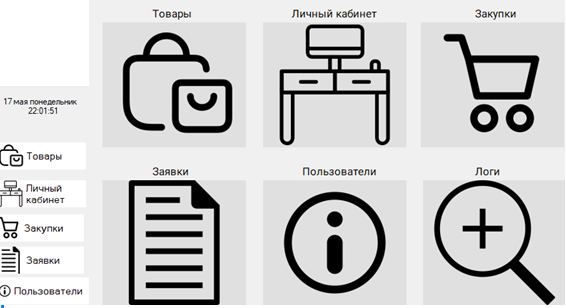


Рисунок 2.2 – Результаты исследований

Таблица 2.3 – Протокол тестирования авторизации на некорректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| 1 | 2 |
| Наименование | Описание |
| Test Case # | Hozlider\_A\_2 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка авторизации в модуле login.cs («Авторизация») с некорректными пользовательскими данными |
| Резюме испытания | Необходимо добиться корректного поведения приложения при вводе неправильного логина или пароля |
| Шаги тестирования | Ввод неправильного логина и пароля в текстовые поля;  нажатие кнопки «Войти». |
| Данные тестирования | Логин: das;  Пароль: asd. |
| Ожидаемый результат | Приложение должно сообщить пользователю о том, что введены неправильные данные |

Продолжение таблицы 2.3

|  |  |
| --- | --- |
| 1 | 2 |
| Фактический результат | В результате тестирования приложение вывело окно с сообщением «Неверные логин или пароль» |
| Предпосылки | Открытие страницы login.cs («Авторизация») |
| Постусловия | Система не зависает, находится в состоянии полной работоспособности. |
| Статус (Pass/Fail) | Pass |

Проведём тестирование авторизации с неправильным логином и паролем пациента. Результаты изображены на рисунках 2.3 – 2.4.

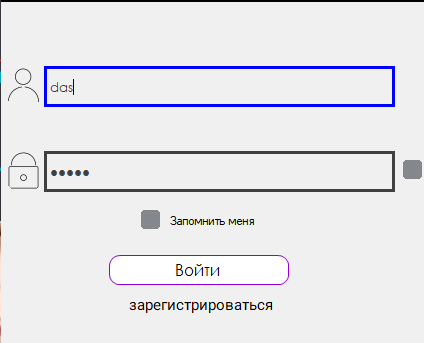


Рисунок 2.3 – Авторизация с вводом неправильных данных

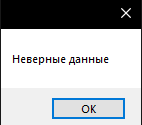


Рисунок 2.4 – «Ошибка в случае зарезервированного времени»

Таблица 2.4 – Протокол тестирования авторизации на некорректных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Наименование | Описание |
| Test Case # | Hozlider\_ND\_3 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка авторизации в модуле login.cs («Авторизация») с незаполненными пользовательскими данными |
| Резюме испытания | Необходимо добиться корректного поведения приложения при незаполненных полях логина и пароля |
| Шаги тестирования | Текстовые поля пустые;  нажатие кнопки «Войти». |
| Данные тестирования | – |
| Ожидаемый результат | Веб-приложение должно сообщить пользователю о том, что поля не заполнены |
| Фактический результат | В результате тестирования приложение вывело окно с сообщением «Заполните поля логина и пароля» |
| Предпосылки | Открытие формы login.cs («Авторизация»). |
| Постусловия | Система не зависает, находится в состоянии полной работоспособности. |
| Статус (Pass/Fail) | Pass |

Проведём тестирование авторизации при незаполненных полях логина и пароля. Результаты изображены на рисунках 2.5 – 2.6.

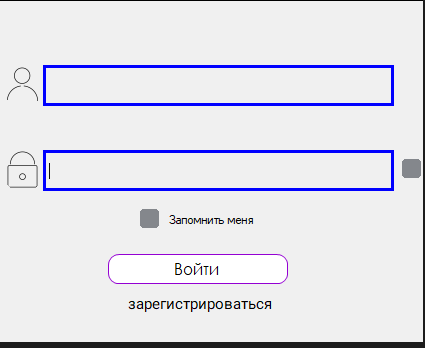


Рисунок 2.5 – Авторизация при незаполненных полях

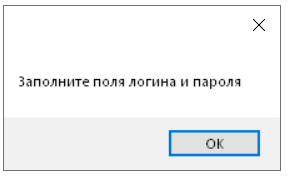


Рисунок 2.6 – Сообщение об ошибке

Таблица 2.5 – Протокол тестирования на данных контрольного примера

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| 1 | 2 |
| Наименование | Описание |
| Test Case # | Hozlider\_ADD\_4 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка добавления новой записи на прием для клиента в модуле MainList.cs с корректными данными |
| Резюме испытания | Необходимо добиться корректного поведения программы при добавлении новой записи |
| Шаги тестирования | Запуск программы;  Авторизация;  Открыты модули «MAIN\_form.cs» и «MainList.cs». |
| Данные тестирования | Название: MP4 плеер;  Тип: Электроника;  Цена за 1ед: 250;  Производитель: Sony  Модель: Xlss |
| Ожидаемый результат | Приложение должно вывести сообщение об успешном добавлении услуги и добавить запись в базу |
| Фактический результат | В результате тестирования приложение вывело сообщение товар добавлен и добавило данные в базу |
| Предпосылки | Запуск программы;  Авторизация;  Открыты модули «Main.cs» и «MainList.cs». |

Продолжение таблицы 2.5

|  |  |
| --- | --- |
| 1 | 2 |
| Постусловия | Система не зависает, находится в состоянии полной работоспособности. |
| Статус (Pass/Fail) | Pass |

Проведём тестирование добавления новой записи товара. Результаты изображены на рисунках 2.7-2.9.

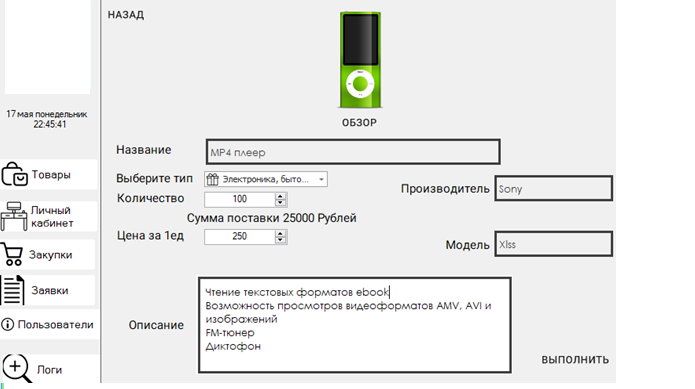


Рисунок 2.7 – Добавление товара

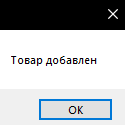


Рисунок 2.8 – Сообщение об успешном добавлении записи

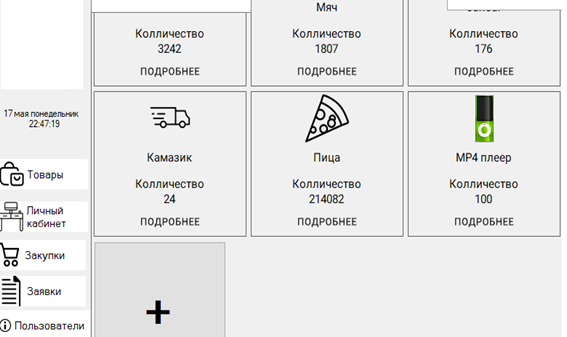


Рисунок 2.9 – Товар добавлен

Таблица 2.6 – Протокол тестирования на данных контрольного примера

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Наименование | Описание |
| Test Case # | Hozlider\_INSERT\_5 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка изменения информации в товаре в модуле MainList.cs с корректными данными |
| Резюме испытания | Необходимо добиться корректного поведения приложения при незаполненных полях логина и пароля |
| Шаги тестирования | Запуск программы;  Авторизация;  Открыты модули «MAIN\_form.cs» и «MainList.cs». |
| Данные тестирования | Название: MP4 плеер;  Тип: Электроника;  Цена за 1ед: 250;  Производитель: ASUS  Модель: Xl  Описание: На запччасти |

Продолжение таблицы 2.6

|  |  |
| --- | --- |
| 1 | 2 |
| Ожидаемый результат | Приложение должно сообщить пользователю о том, что товар изменён. |
| Фактический результат | В результате тестирования приложение вывело окно с сообщением «Товар изменён». |
| Предпосылки | Открытие формы UpandIN.cs («Добавление»). |
| Постусловия | Система не зависает, находится в состоянии полной работоспособности. |
| Статус (Pass/Fail) | Pass |

Проведём тестирование добавления новой записи на прием для клиента. Результаты изображены на рисунках 2.10-2.12.

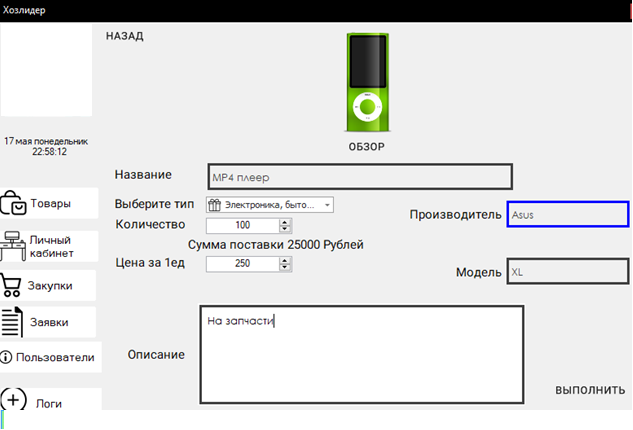


Рисунок 2.10 – Изменение товара

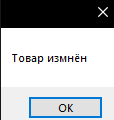


Рисунок 2.11 – Сообщение об успешном изменении записи

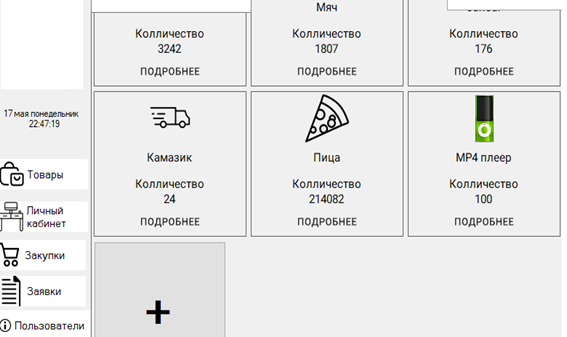


Рисунок 2.12 – Товар изменён

Таблица 2.7 – Протокол тестирования регистрации при незаполненных данных

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Наименование | Описание |
| Test Case # | Hozlider\_LG\_6 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка авторизации в модуле Reg.cs («Регистрация») с незаполненными пользовательскими данными |
| Резюме испытания | Необходимо добиться корректного поведения приложения при незаполненных полях логина и пароля |

Продолжение таблицы 2.7

|  |  |
| --- | --- |
| Шаги тестирования | Запуск программы;  Регистрация;  Открыты модули «Reg.cs». |
| Данные тестирования | Login:1  Password:1 |
| Ожидаемый результат | приложение должно сообщить пользователю о том, что поля не заполнены |
| Фактический результат | В результате тестирования приложение вывело окно с сообщением «Заполните все поля» |
| Предпосылки | Открытие формы Reg.cs («Регистрация»). |
| Постусловия | Система не зависает, находится в состоянии полной работоспособности. |
| Статус (Pass/Fail) | Pass |

Проведём тестирование добавления новой записи на прием для клиента. Результаты изображены на рисунках 2.13-2.14.

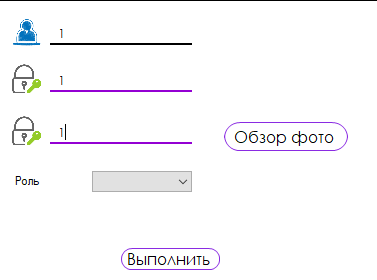


Рисунок 2.13 – Авторизация при незаполненных полях

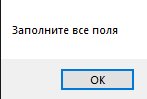
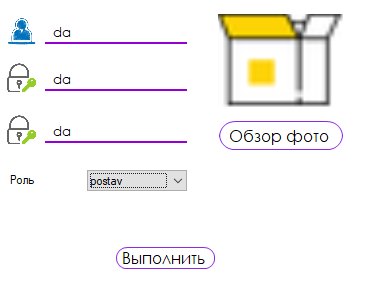


Рисунок 2.14 – Авторизация при незаполненных полях

Таблица 2.8 – Протокол тестирования авторизации с незаполненными данными

|  |  |
| --- | --- |
| Описание информационных полей для тестирования | |
| Наименование | Описание |
| Test Case # | DentystryPro\_LG\_7 |
| Приоритет тестирования | Высокий |
| Название тестирование | Проверка авторизации в модуле login.cs («Авторизация») с незаполненными пользовательскими данными |
| Резюме испытания | Необходимо добиться корректного поведения приложения при незаполненных полях логина и пароля |
| Шаги тестирования | Запуск программы;  Регистрация;  Открыты модули «Reg.cs». |
| Данные тестирования | Login:da;  Passwodr:da;  Royale:postav; |
| Ожидаемый результат | Приложение должно сообщить пользователю о том, профиль добавлен. |
| Фактический результат | В результате тестирования приложение вывело окно с сообщением «Заполните поля логина и пароля» |
| Предпосылки | Открытие формы login.cs («Авторизация»). |
| Постусловия | Система не зависает, находится в состоянии полной работоспособности. |
| Статус (Pass/Fail) | Pass |

Проведём тестирование добавления новой записи на прием для клиента. Результаты изображены на рисунках 2.15-2.16.



# Рисунок 2.15 – Регистрация при Корректных данных

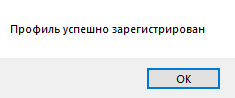


Рисунок 2.16 – Запись об успешной операции

2.3 Руководство пользователя

Доступ к программе осуществляется посредством обычного запуска приложения «Hozlider.exe».

При запуске файла «Hozlider.exe» открывается окно авторизации (рисунок 2.3.1).

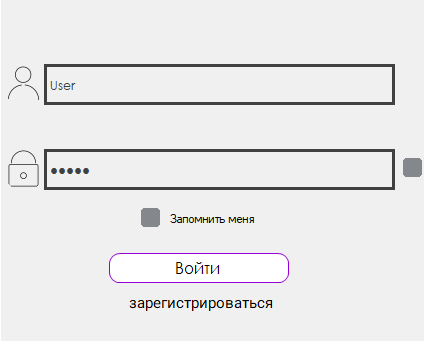


Рисунок 2.3.1 — «Форма авторизации»

Если пользователь уже имеет аккаунт в приложении, то для использования вам необходимо ввести логин и пароль от него. Далее необходимо нажать на кнопку «Войти».

Если пользователь не имеет аккаунта, то ему необходимо перейти на форму регистрации. Для этого нужно нажать на кнопку «Зарегистрироваться». После этого пользователя перенаправит на форму регистрации. Страница регистрации представлена на рисунке 2.3.3.

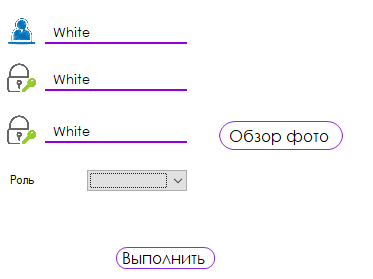


Рисунок 2.3.2 — «Форма регистрации»

После ввода логина и пароля открывается «Главное меню» количество функций зависит от того, какой пользователь вошёл в систему. После необходимо нажать кнопку сменить пользователя, чтобы перейти в главную форму (рисунки 2.3.3-2.3.5).

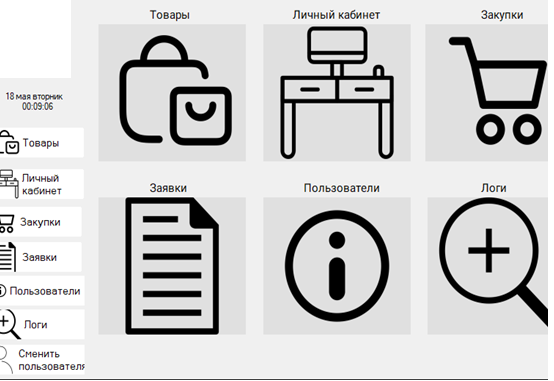


Рисунок 2.3.3 — «Главное меню администратора»

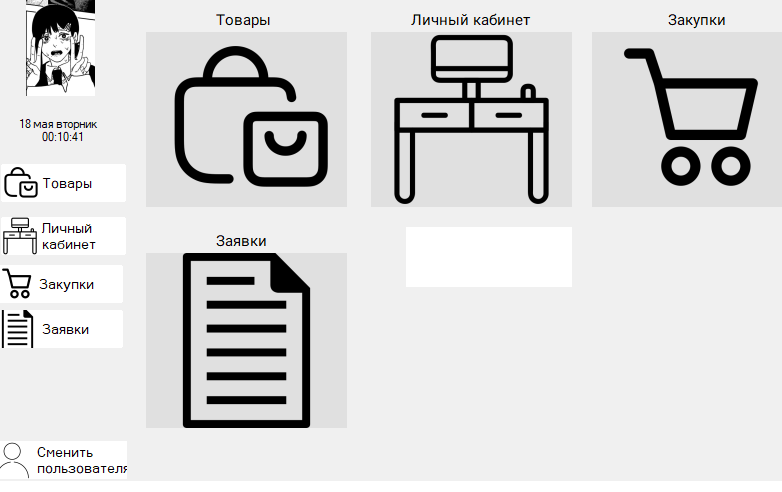


Рисунок 2.3.4 — «Главное меню поставщика»

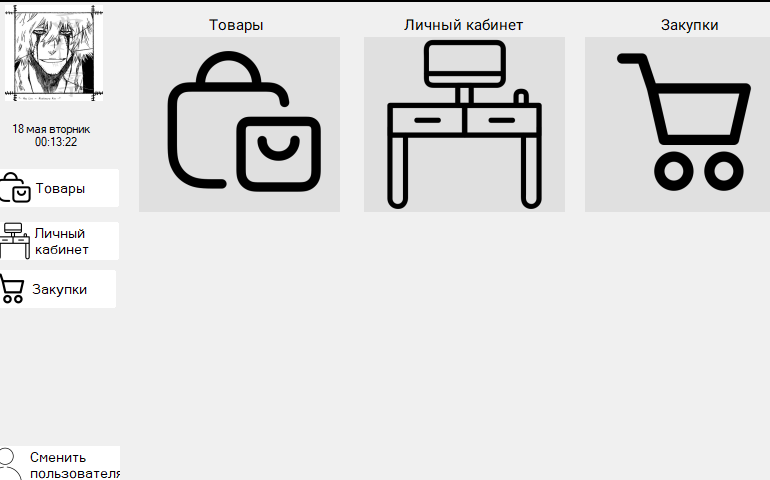


Рисунок 2.3.5 — «Главное меню пользователя»

При нажатии на кнопку «Товары» осуществляется переход на форму «Список товаров» (рисунок 2.3.6).



Рисунок 2.3.6 — «Список товаров»

При нажатии на кнопку «Подробнее» осуществляется переход на форму «Информация о товаре» (рисунок 2.3.7).

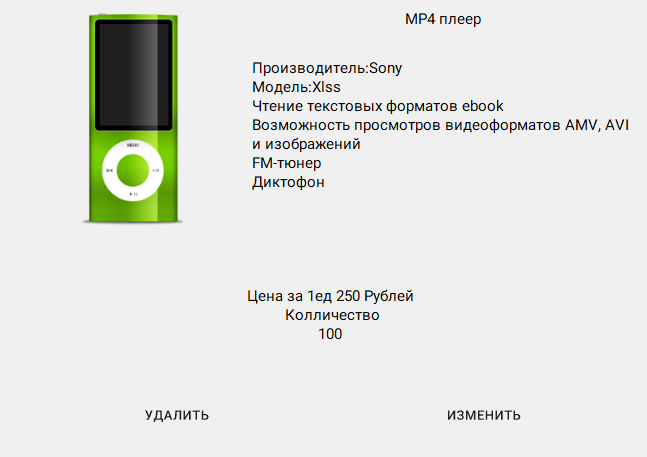


Рисунок 2.3.7 – «Информация о товаре»

На форме расположены четыре кнопки:

– «Добавить» – переход на форму добавления/изменения товара (рисунок 2.3.8);

– «Изменить» – переход на форму добавления/изменения товара (рисунок 2.3.8);

– «Удалить» – удаление выбранного товара;

– «Назад» – переход на прошлую страницу.

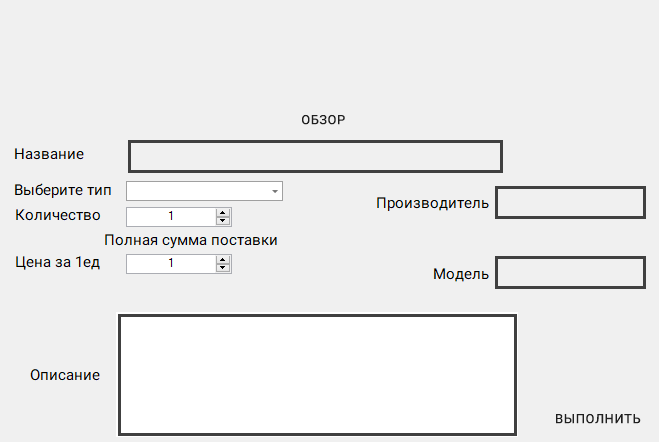


Рисунок 2.3.8 – «Добавление-изменение товара»

При нажатии на кнопку «Личный кабинет» осуществляется переход на форму «Личный кабинет» (рисунок 2.3.9):

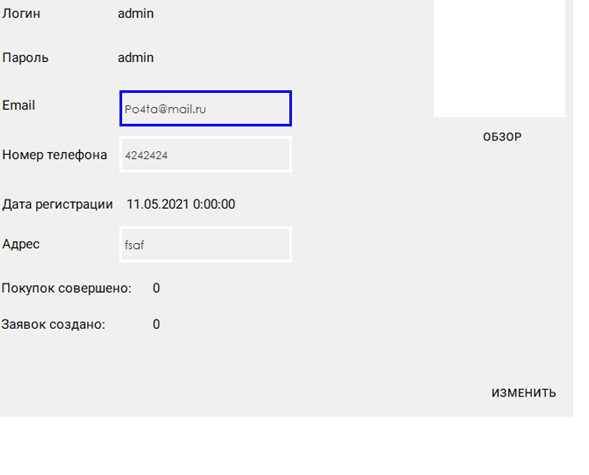


Рисунок 2.3.9 – «Личный кабинет»

На форме расположена кнопка «изменить» - изменения данных пользователя.

При нажатии на кнопку «Закупки» осуществляется переход на форму «Закупи» (рисунок 2.3.10).

Форма «Закупки» выводит данные только в том случае если пользователь внёс товар в корзину.

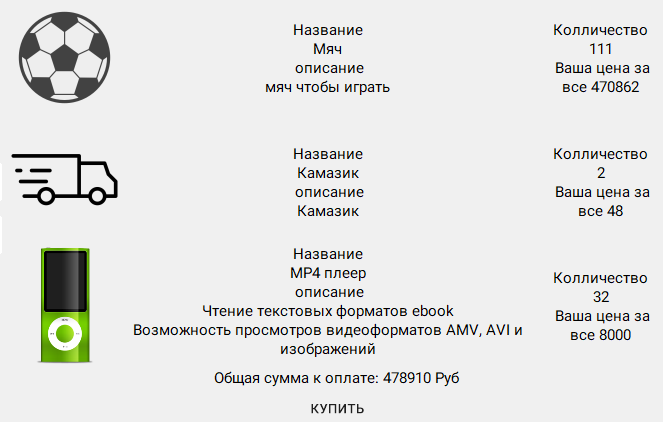


Рисунок 2.3.10 – «Закупки»

На форме расположена кнопка «купить», которая выводит договор о купли-продажи и Приложение 1 (рисунки 2.3.11-.2.3.12).

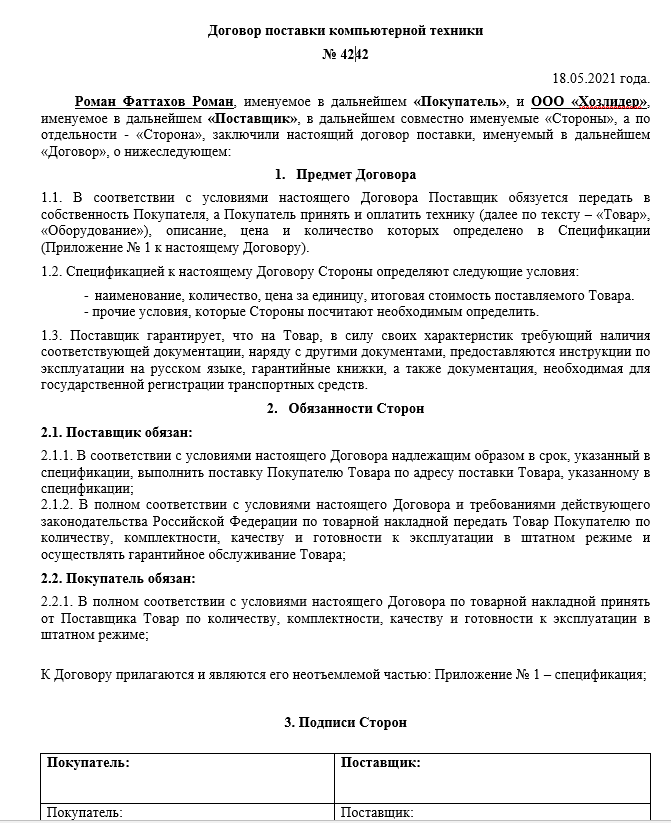


Рисунок 2.3.11 – «Договор о купли-продажи»

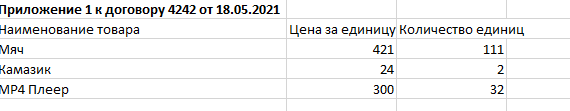


Рисунок 2.3.12 – «Приложение 1»

При нажатии на кнопку «Заявки» у поставщика осуществляется переход на форму «Заявки» (рисунок 2.3.13).

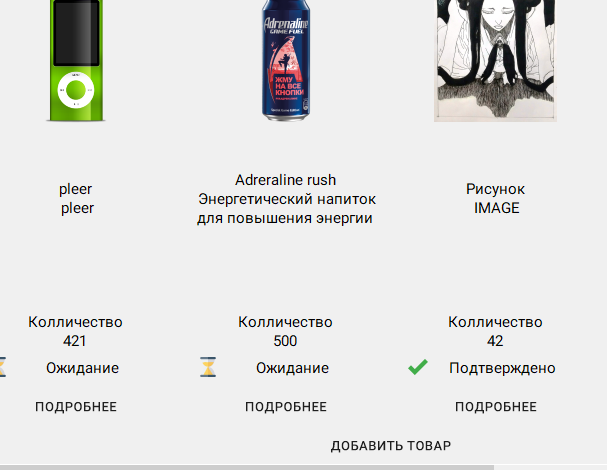


Рисунок 2.3.13 – «Заявки» поставщика

На форме расположена кнопка «Добавить» – переход на форму добавления заявки (рисунок 2.3.8);

При нажатии на кнопку «Заявки» у администратора осуществляется переход на форму «Заявки» (рисунок 2.3.13).

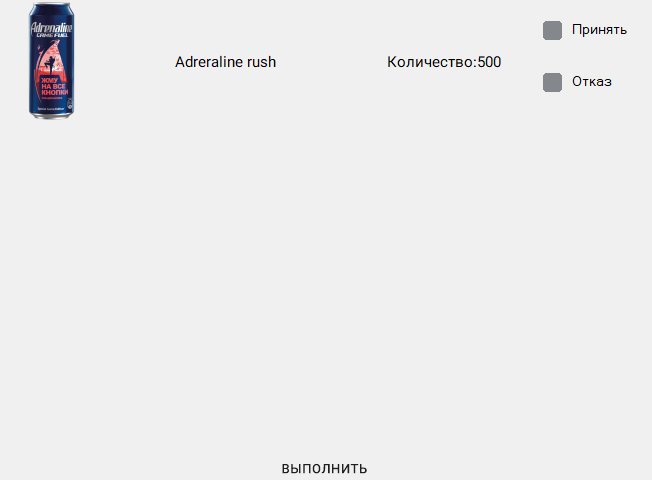


Рисунок 2.3.14 – «Заявки администратора»

На форме расположены три кнопки:

* «Принять» - Добавление товара из определённой заявки;
* «Отказ» - Отказ от заявки;
* «Выполнить» – Просмотр заявок;

При нажатии на кнопку «Пользователи» у администратора осуществляется переход на форму «Пользователи» (рисунок 2.3.15).

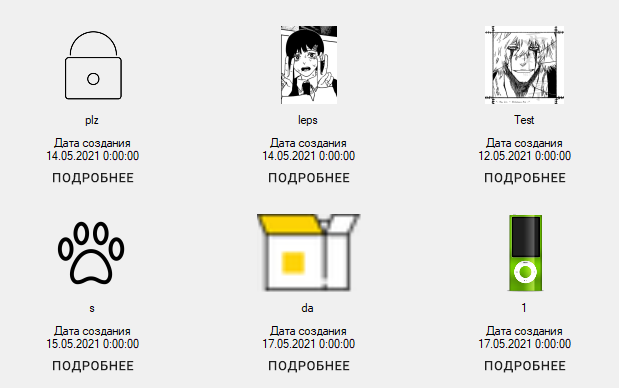


Рисунок 2.3.15 – «Пользователи»

На форме расположена кнопка «Подробнее» – переход на форму просмотра информации (рисунок 2.3.16);

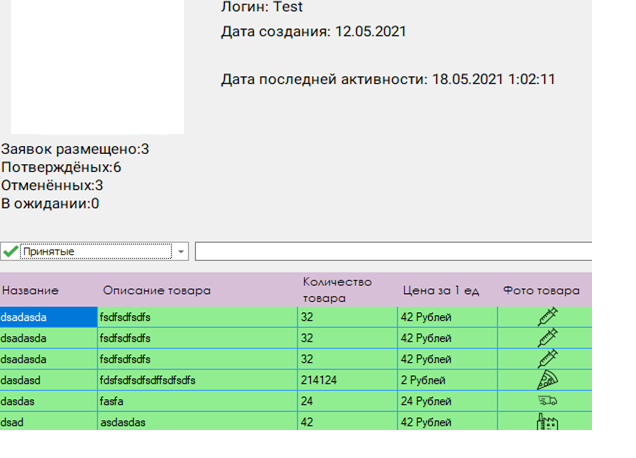


Рисунок 2.3.16 – «Просмотр информации»

ЗАКЛЮЧЕНИЕ

В результате выполнения проекта было разработано приложение, которое упрощает процесс регистрации и обработки товаров в агентстве по продаже товаров. В ходе анализа возможностей разработанного приложения был сделан вывод о том, что полноценная реализация подобного приложения требует более глубокой детализации предметной области.

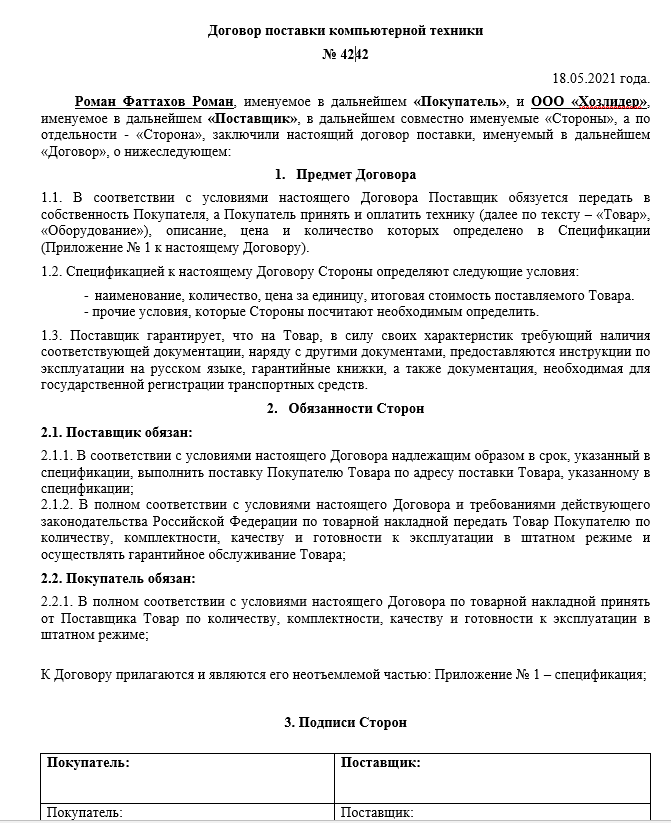
Применение базы данных в качестве хранилища информации позволяет оптимально и эффективно хранить информацию, а также структурировать ее.

Программа написана на языке C# в среде программирования Visual studio 2019 community. В ходе разработки программного продукта были решены все поставленные задачи.

Программный продукт имеет дружественный пользователю интерфейс, для работы с программой не требуется дополнительное обучение.

В ходе работы был получен законченный программный продукт, позволяющий организации облегчить процесс регистрации и обработке товаров в агентстве по продаже товаров.

Приложение А



СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Федеральный закон от 12.01.96 № 7­ФЗ «О коммерческих организациях»
2. Федеральный закон от 19.05.95 № 82­ФЗ «Об общественных объединениях»
3. ГОСТ 2.104-2006 ЕСКД. Основные надписи;
4. ГОСТ 19.701-90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения;
5. ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам;
6. ГОСТ 7.32-2001 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления;
7. Архангельский, А.Я. Delphi 7: Справочное пособие [Текст] / А.Я. Архангельский - М.:Бином-Пресс., 2004. - 1024 с. Санкт-Петербург: Питер, 2013 – 240 с.
8. Гагарина, Л.Г. Разработка и эксплуатация автоматизированных информационных систем [Текст] / Л.Г.Гагарина, Д.В. Киселев, Е.Л. Федотова, 2014 г. – 383 с.;
9. Голицына, О. Л. Информационные системы : Учебное пособие [Текст] / О. Л. Голицына, Н. В. Максимов, И. И. Попов.-2-е изд.-М.: Форум: НИЦ ИНФРА-М, 2014.-448 с.
10. Голицына, О. Л. Основы проектирования баз данных: Учебное пособие [Текст] / О.Л.Голицына, Т.Л. Партыка, И.И. Попов,-2-е изд.- М.: Форум, НИЦ ИНФРА-М, 2016.-416 с.
11. Диго С. Базы данных: проектирование и создание [Текст] / С. Диго - М.:ЕАОИ, 2013. - 171 с.
12. Емельянова, Н.З. Проектирование информационных систем :Учебное пособие [Текст] / Н. З. Емельянова, Т. Л. Пактыко, И. И. Попов, -М.: Форум, 2013.-432 с.
13. Заботина, Н.Н. Проектирование информационных систем :Учебное пособие [Текст] / Н. Н.. Заботина. -М.: НИЦ ИНФРА-М, 2013.-331 с.
14. Исаев, Г.Н. Проектирование информационных систем: Учебное пособие [Текст] / Г. Н.Исаев. - М.: Омега - Л, 2013.-424 с.
15. Карпова, И.П. Базы данных: Учебное пособие [Текст] / И.П. Карпова.- Санкт-Петербург: Питер,2013.-240 с.
16. Кириллов, В.В. Ведение в реляционные базы данных [Текст] / В.В. Кириллов, Г.Ю. Громов.- Санкт-Петербург: БХВ- Петербург,2012.- 464 с.
17. Коваленко, В.В. Проектирование информационных систем : Учебное пособие [Текст] / В.В. Коваленко. –М.: Форум, 2013.-424 с.
18. Кузин, А.В. Базы данных: Учебное пособие для студентов высших учебных заведений [Текст] / А.В.Кузин, С.В. Левонисова.-М.: ИЦ Академия, 2012. - 320с.
19. Мезенцев, К.Н. Автоматизированные информационные системы: Учебник для студентов учреждений среднего профессионального образования [Текст] / К.Н. Мезенцев. – М.:ИЦ Академия,2013.-176 с.
20. Немцова, Т.И. Программирование на языке Object C#: Учебное пособие [Текст] / Т.И. Немцова. – М.:ИД ФОРУМ: НИЦ ИНФРА – М, 2013.-496 с.

1. PK-первичный ключ

   FK-внешний ключ [↑](#footnote-ref-1)