Министерство образования и науки Украины

Харьковский национальный университет радиоэлектроники

Кафедра Программной инженерии

КУРСОВАЯ РАБОТА

ОБЪЯСНИТЕЛЬНАЯ ЗАПИСКА

По дисциплине “Объектно-ориентированное программирование”

«Склад»

Руководитель: Бондарев В. М.

Студентка гр. ПЗПИ-19-2 Жила М. Г.

Комиссия:

Ст. Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю. Ю.

Проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарев В. М.

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Побеженко И. А.

Харьков 2020

**ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ**

Кафедра: ***Программной инженерии***

Дисциплина: ***Объектно-ориентированное программирование***

Специальность: ***121 Инженерия программного обеспечения***

Образовательная программа: ***Программная инженерия***

Курс 1 . Группа ПЗПИ-19-2 . Семестр 2 .

***ЗАДАНИЕ***

***на курсовой проект студентки***

***Жилы Марии Георгиевной***

1. Тема проекта: *«Склад»*
2. Термин сдачи студентом законченного проекта: ***“31” – мая – 2020 г.***
3. Исходные данные к проекту:

*Методические указания к выполнению курсовой работы*

1. Содержание расчетно-пояснительной записки:

*Вступление, спецификация программы, проектная спецификация, инструкция пользователя, тестирование данных, выводы.*

1. Перечень графического материала:

*Рисунки, примеры экранных форм.*

**КАЛЕНДАРНЫЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Название этапа* | *Термин выполнения* |
| 1 | Выдача, согласование и утверждение темы. | 30-03-2020 р. |
| 2 | Составление спецификации | 31-03-2020 – 06-04-2020 г. |
| 3 | Проектирование | 07-04-2020 – 13-04-2020 г. |
| 4 | Программирование модели | 14-04-2020 – 04-05-2020 г. |
| 5 | Программирование пользовательского интерфейса | 05-05-2020 – 18-05-2020 г. |
| 6 | Тестирование и работа над деталями | 19-05-2020 – 24-05-2020 г. |
| 7 | Написание пояснительной записки | 25-05-2020 – 31-05-2020 г. |
| 8 | Защита | 01-06-2020 – 01-07-2020 г. |

Студентка Жила Мария Георгиевна

Руководитель Бондарев Владимир Михайлович

*« 30 » марта 2020 года*

**РЕФЕРАТ**

Пояснительная записка к курсовой работе: 48 с, 45 рис.

КЛАСС, ЯЗЫК ПРОГРАММИРОВАНИЯ С#, НАКЛАДНАЯ, ОБЪЕКТНО-ОРИЕНТИРОВАНИЕ ПРОГРАММИРОВАНИЕ, ПРОГРАММИРОВАНИЕ, ПРОГРАММА, ТОВАР, СКЛАД, ПЛАТФОРМА .NET, WINDOWS FORMS.

Целью данной работы есть разработка программы “Склад” на основе объектно-ориентированного подхода.

Методы разработки базируются на использовании среды разработки Microsoft Visual Studio 2019, Windows Forms, платформы .NET Framework 4.7.2, языка программирования C#.

В результате получена программа под названием “Склад”, которая позволяет сохранять список товаров, которые находятся на складе, получать информацию о них, такую как: наименование, единица измерения, количество, артикул, цена; делать их переучет; оформлять заказ, а также подтверждать его; просматривать накладные, истории заказов.

**СОДЕРЖАНИЕ**

Вступление………………………………………………………………………........6

1. Спецификация программы…………………………………………………….…..7

1.1. Задание……………………………………………………………………….…...7

1.2. Основные положения………………………………………………….………...7

1.3. Функции программы…………………………………………………………….8

1.4. Общие функции………………………………………………………………….8

1.5. Функции заведующего складом………………………………………………...9

1.6. Функции клиента……………………………………………………………….17

2. Проектирование…………………………………………………………………..23

2.1. Архитектура…………………………………………………………………….23

2.2. Структура проекта……………………………………………………………...24

2.3. Объектная модель………………………………………………………………25

2.4. Методы………………………………………………………………………….27

2.5. Диаграммы……………………………………………………………………...27

2.6. Хранение и доступ к данным…………………………………………………..29

3. Инструкция пользователю……………………………………………………….30

3.1. Установка программы………………………………………………………….30

3.2. Работа с программой. Инструкция заведующему складом………………….30

3.3. Работа с программой. Инструкция клиенту…………………………………..39

4. Тестирование данных…………………………………………………………….47

Выводы………………………………………………………………………………48

**ВСТУПЛЕНИЕ**

Целью данной курсовой работы является закрепление знаний, полученных при изучении дисциплины “Объектно-ориентированное программирование”, а также приобретение навыков для разработки программных продуктов.

Программа должна быть написана с использованием объектно-ориентированного подхода. Язык, на котором требуется написать данный проект - С#. В ходе курсовой работы также предполагается, что будут изучены некоторые тонкости данного языка.

Программный продукт, который нужно разработать в рамках данного курсового проекта, имеет тему “Склад”. Данный продукт должен содержать базу товаров, которые содержатся на складе. В нем нужно реализовать возможность создания новой поставки с формированием приходной накладной и отгрузки (продажи) с формированием расходной накладной. Также заведующий складом должен иметь возможность проводить инвентаризацию.

Создание курсовой будет разделяться на несколько этапов, такие как: составление спецификации, проектирование, программирование модели, программирование пользовательского интерфейса, написание пояснительной записки. В конце данную работу требуется защитить, проявив такие качества как уверенность в собственных знаниях, умение отстаивать собственное мнение, умение выступать перед аудиторией.

Продукт, полученный в результате курсовой работы, предназначен для складских хозяйств. Его главное применение состоит в учете товара на складе и мониторинге действий, которые были совершенны по отношению к данному товару. Использовать данное приложение могут, как и большие предприятия, так и малые.

1. **СПЕЦИФИКАЦИЯ ПРОЕКТА**
   1. ***Задание.***

На складе имеются товары, которые изготовили на фабрике.

Товар периодически приходит на склад.

Заведующий складом вносит его в программу, после чего формируется приходная накладная. Также одобряет закупку, после этого формируется расходная накладная. Проводит инвентаризацию. Регистрирует новых пользователей.

Клиенты выбирают товар. Создают заказ, который подтверждает заведующий складом. Смотрят неподтвержденные заказы, а также историю покупок.

О товаре известно: имя, единица измерения, цена единицы, кол-во, артикул, дата последнего завоза.

О покупателях - имя предприятия, почтовый адрес, логин и пароль.

* 1. ***Основные положения***

Пользователи программы.

У проекта имеются две категории пользователей, которым необходимы разные функции программы — это заведующий склада и клиенты.

Клиент отыскивает товары, добавляет их в корзину, создает заказ, который должен одобрить заведующий складом, а также просматривает историю своих прошлых заказов.

Заведующий складом вносит товар на склад, подтверждает заказы, просматривает отчеты, а именно приходную и расходную накладные, добавляет пользователей, проводит инвентаризацию.

Для заведующего складом и клиентов существует два разных приложения. При входе в приложение нужно пройти процесс аутентификации.

Вопросы безопасности, такие как хеширование паролей и безопасность данных в программе не решаются.

* 1. ***Функции программы***

Программа предоставляет клиенту следующие возможности.

1. Вход в программу
2. Добавление товара в корзину
3. Редактирование корзины
4. Создание заказа
5. Просмотр деталей о неподтвержденных заказах
6. Удаление неподтвержденных заказов
7. Просмотр истории покупок
8. Выход из программы

Программа предоставляет заведующему склада следующие возможности.

1. Вход в программу
2. Внесение товара в программу
3. Просмотр деталей заказа
4. Подтверждение заказов
5. Добавление пользователей
6. Удаление пользователей
7. Инвентаризация товара
8. Получение отчетов
9. Выход с программы
   1. ***Общие функции***
      * 1. Загрузка данных из постоянной памяти в оперативную.
        2. Сохранение текущего состояния данных в постоянной памяти.
        3. Генерация тестовых данных
   2. ***Функции заведующего складом***

*Функция “Вход в программу”.*

Заведующий складом входит в свое отдельное приложение. Вводит имя и пароль. Имя и пароль заведующего складом сохраняются в данных программы.

*Основной сценарий*

Заведующий складом открывает программу, открывается окно аутентификации. (рис. 1)

Заведующий складом вводит логин и пароль и нажимает кнопку "OK".

Введенные данные проверяются, и открывается панель администратора – главное окно заведующего складом. (рис. 2)

Если данные не прошли проверку, на форме ввода появляется сообщение об этом и можно попытаться войти снова. Число попыток не ограничено.

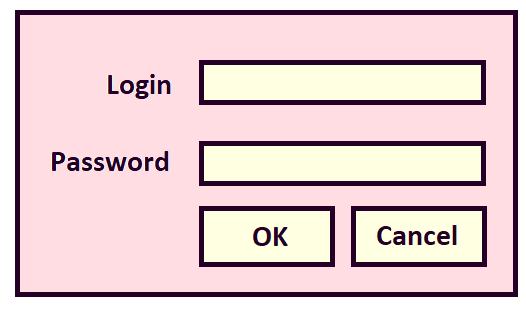


Рисунок 1 - Панель входа заведующего складом

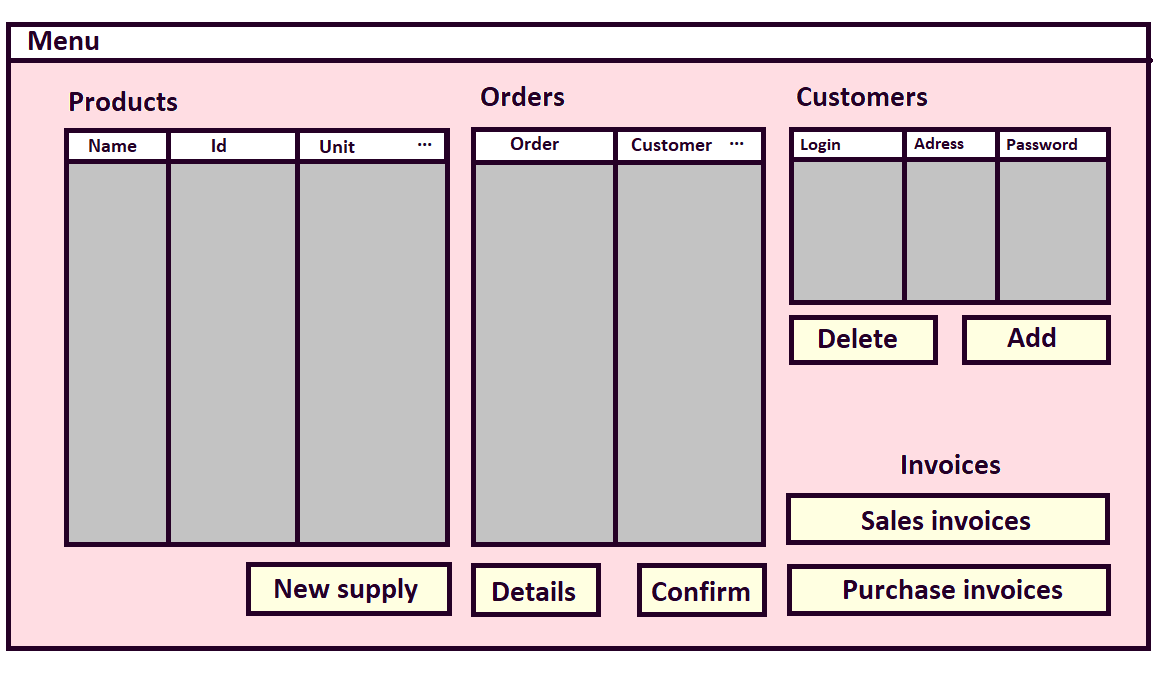


Рисунок 2 - Панель администратора

*Функция “Внесение товара в программу”.*

Когда товар приходит на склад, заведующий складом должен внести его в программу. Товар приходит поставками, это значит, что на склад приходит не один вид товара. Для этого будет создано отдельное окно новой поставки, в котором администратор будет вносить товар, который поступил. В данном окне администратор будет вводить всю информацию о товаре. Нельзя действовать безошибочно, поэтому должна быть возможность не только пополнять список поставки, но и исправлять ошибки, которые были совершены при внесении товара, для этого будет создана кнопка для удаления товара с поставки. Когда составление поставки будет завершено и проверено администратор может подтвердить поставку.

*Основной сценарий*

* + - 1. На панели администратора заведующий складом нажимает кнопку "New supply". (рис. 2)
      2. Открывается окно ввода новой поставки. (рис. 3)
      3. Составляется список товаров, которые входят в данную поставку. Элементы в списке можно добавлять и удалять.
      4. Когда список поставки окончательно сформирован, заведующий складом нажимает кнопку "Confirm".
      5. Список товаров в магазине пополняется новыми товарами, а также формируется приходная накладная.

*Сценарий «Добавление товара в поставку»*

В окне новой поставки на полях ввода заведующий складом заполняет информацию о товаре, который хочет добавить. (рис. 3)

Когда вся информация о товаре введена, заведующий складом нажимает кнопку "Add".

Список товаров в поставке пополняется новым товаром.

*Сценарий «Удаление товара из поставки»*

В окне новой поставки заведующий складом выбирает товар, который хочет удалить из поставки и кликает по этому выбранному товару мышкой. (рис. 3)

Далее заведующий складом нажимает "Delete" и подтверждает действие.

Список товаров в поставке обновляется.

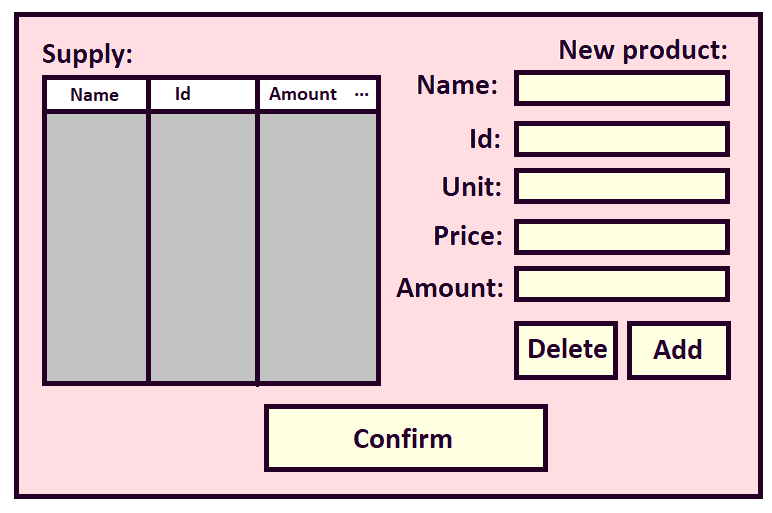


Рисунок 3 – Окно новой поставки

*Функция “Просмотр деталей заказа”*

На панели администратора отображены только основные детали заказа, для получения детальной информации о заказе заведующий складом обращается к данной функции.

*Основной сценарий*

Заведующий складом на панели администратора нажимает кнопку "Details". (рис. 2)

Открывается окно деталей данного заказа. (рис. 4)

Для вывода детальной информации о конкретном продукте заведующий складом должен кликнуть по ячейке с данным продуктом в таблице заказа.

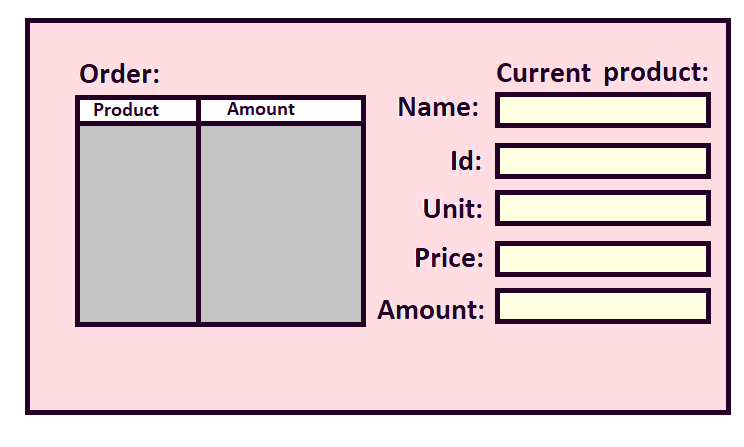


Рисунок 4 - Окно деталей заказа

*Функция “Подтверждение заказа”*

Одна из главных функций заведующего складом – это подтверждение заказов.

*Основной сценарий*

На панели администратора заведующий складом нажимает кнопку "Confirm" и подтверждает действие. (рис. 2)

Заказ исчезает с таблицы заказов, происходят изменения данных о товаре, формируется расходная накладная. Также данный заказ будет доступен покупателю в его истории покупок

*Функция “Добавление пользователя”*

Поскольку количество клиентов непостоянно, то соответственно нужно реализовать такую функцию, которая позволила бы добавлять новых клиентов.

*Основной сценарий*

На панели администратора заведующий складом нажимает кнопку "Add". (рис. 2)

Открывается окно регистрации клиента (рис. 5)

Заведующий складом вводит информации о клиенте.

Когда информация введена заведующий складом нажимает кнопку "Confirm".

Информация проверяется на корректность, если все правильно, то в программе начинает числиться новый клиент.

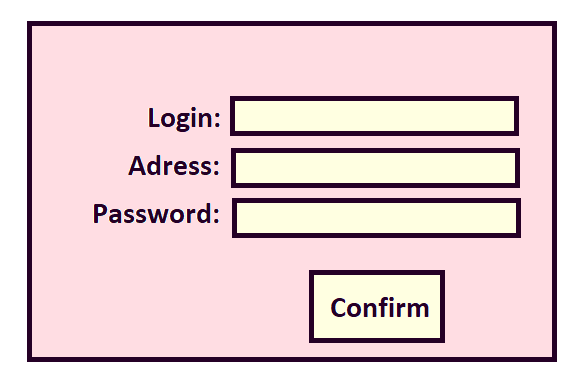


Рисунок 5 – Окно регистрации пользователя

*Функция “Удаление пользователя”*

Данная функция нужна случае ошибки регистрации пользователя или желании просто убрать клиента из программы.

*Основной сценарий*

На панели администратора заведующий складом нажимает кнопку "Delete" (рис. 2) и подтверждает свое действие.

Список клиентов в магазине обновляется.

*Функция “Инвентаризация товара”*

Товар может быть списан в следствии порчи. При инвентаризации происходит пересчет товара. Если товар действительно был списан, то заведующему склада нужно будет изменить текущее значение товара на новое.

*Основной сценарий*

Заведующий складом на вкладке меню, которая находится на панели администратора нажимает кнопку "Inventory". (рис. 2)

Открывается окно инвентаризации. (рис. 6)

Изменяет количество товаров на складе, подтверждая данное действие кнопкой "OK".

Когда все изменения внесены заведующий складом нажимает кнопку "Confirm".

Изменяется количество товара на складе

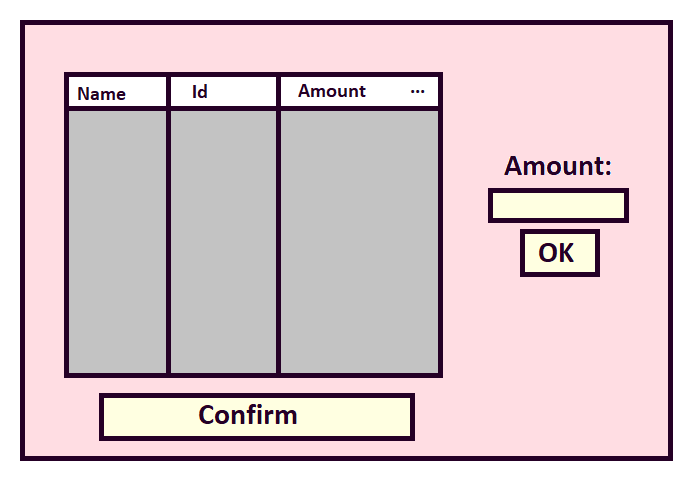


Рисунок 6 – Окно инвентаризации товара

*Функция “Получение отчетов”*

Данная функция отвечает за получение расходной и приходной накладных. Сценарий и вид окна у данных двух накладных почти одинаковый, поэтому распишем в общем случае.

*Основной сценарий*

Заведующий складом на вкладке меню, которая находится на панели администратора нажимает кнопку "Purchase invoices" / "Sales invoices". (рис. 2)

Открывается окно накладной (рис. 7)

Для получения детальной информации пользователь кликает по сеткам.

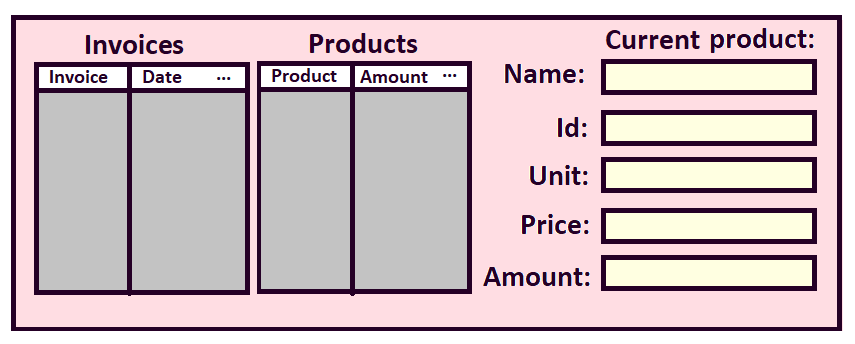


Рисунок 7 – Окно накладной

*Функция “Выход из программы”*

В процессе работы пользователь изменяет данные, которые находятся в оперативной памяти.

Если данные изменены, но не сохранены в постоянной памяти, то перед закрытием программы пользователь получает предложение сохранить данные.

Сценарий должен быть таким же, как при завершении работы с любым текстовым редактором, поэтому здесь не приводится.

* 1. ***Функции клиента***

*Функция “Вход в программу”.*

Клиент входит в свое отдельное приложение. Вводит имя и пароль. Имя и пароль клиентов склада хранятся в данных программы.

*Основной сценарий*

Клиент открывает программу, открывается окно аутентификации. (рис. 8)

Клиент вводит логин и пароль и нажимает кнопку "OK".

Введенные данные проверяются, и открывается панель клиента – главное окно клиента. (рис. 9)

Если данные не прошли проверку, на форме ввода появляется сообщение об этом и можно попытаться войти снова. Число попыток не ограничено.

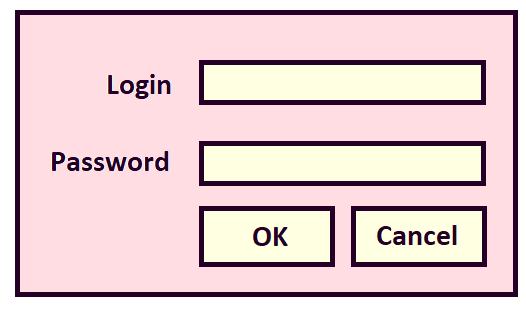


Рисунок 8 - Панель входа клиента

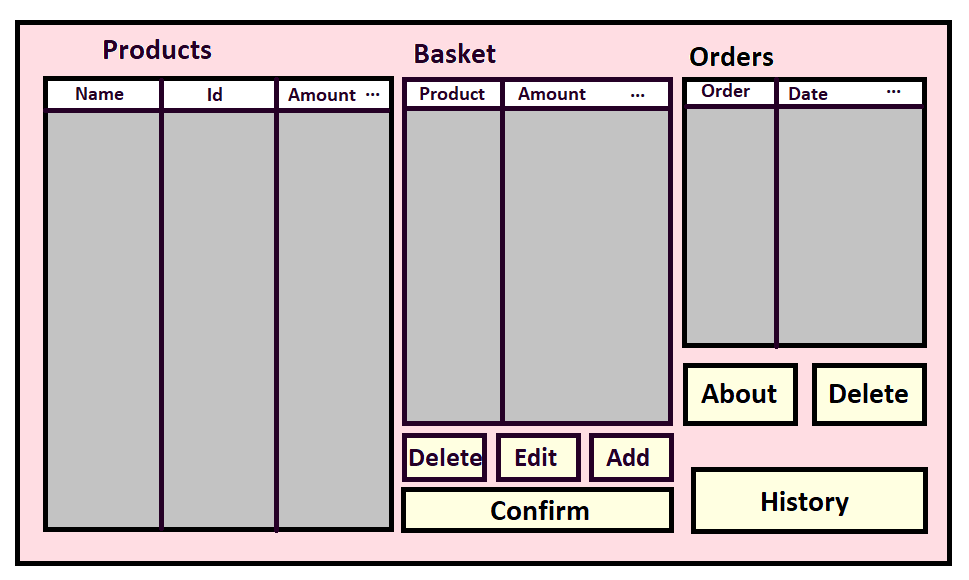


Рисунок 9 – Панель клиента

*Функция “Добавление товара в корзину”.*

Когда клиент только выбирает товары для своего заказа, он вносит их в корзину.

*Основной сценарий*

На панели клиента пользователь нажимает кнопку "Add". (рис. 9)

Открывается окно добавления товара в корзину. (рис. 10)

Клиент вводит количество.

Для подтверждения нужно нажать кнопку "Confirm".

Товар добавляется в корзину.

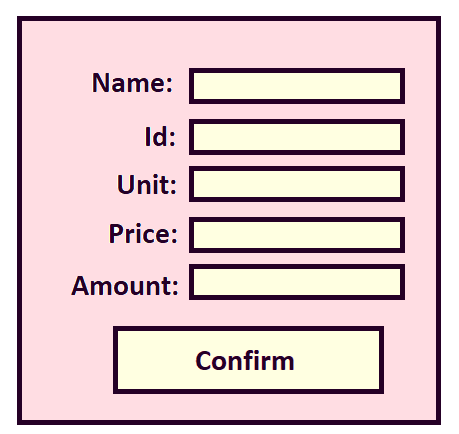


Рисунок 10 – Окно добавления / редактирования товара

*Функция “Редактирование корзины”.*

Клиент должен иметь возможность не только добавлять в корзину, но и вносить правки.

*Основной сценарий*

На панели клиента пользователь нажимает кнопку "Edit". (рис. 9)

Открывается окно изменения товара в корзине. (рис. 10)

Клиент вводит количество.

Для подтверждения нужно нажать кнопку "Confirm".

Товар в корзине обновляется.

*Функция “Удаление товара из корзины”.*

В некоторых случаях клиенту нужно иметь возможность вовсе удалить товар из корзины.

*Основной сценарий*

На панели клиента пользователь нажимает кнопку "Delete". (рис. 9)

Подтверждает свое действие.

*Функция “Создание заказа”.*

Отправка запроса на покупку заказа.

*Основной сценарий*

На панели клиента клиент нажимает кнопку "Confirm". (рис. 9)

Подтверждает свое действие.

*Функция “Просмотр деталей о неподтвержденном заказе”.*

Если покупатель забыл, что заказывал, то он может посмотреть список товаров, которые он заказал в данном разделе.

*Основной сценарий*

1. На панели клиента пользователь нажинает кнопку "Details". (рис. 9)
2. Открывается окно просмотра деталей заказа. (рис. 11)
3. Для дополнительной информации клиент щелкает по ячейкам в таблице

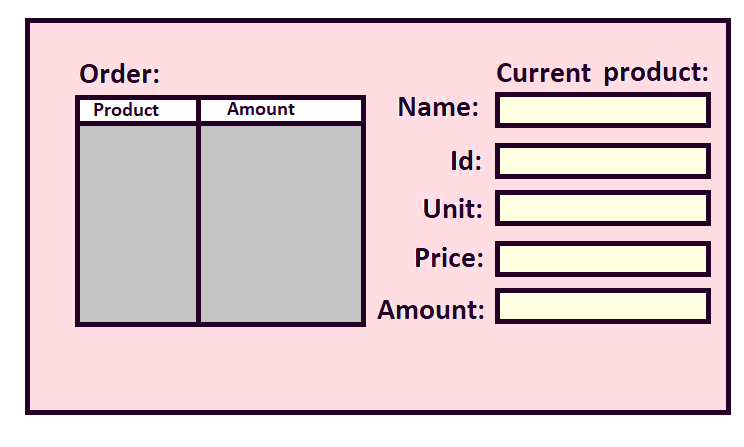


Рисунок 11 – Окно деталей заказа

*Функция “Удаление неподтвержденных заказов”.*

В некоторых случаях клиенту нужно иметь возможность удалить заказ с запроса. Данная функция доступна до тех пор, пока заказ находится в обработке.

*Основной сценарий*

На панели клиента пользователь нажимает кнопку "Delete". (рис. 9)

Подтверждает свое действие.

*Функция “Просмотр истории покупок”.*

Дает возможность клиенту смотреть одобренные заказы.

*Основной сценарий*

На панели клиента пользователь нажимает кнопку "History". (рис. 9)

Открывается панель истории покупок (рис. 12)

Для более подробной информации клиент щелкает по ячейкам.

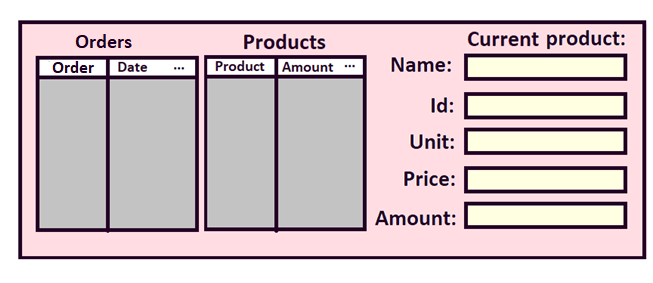


Рисунок 12 – Панель истории покупок

*Функция “Выход из программы”*

В процессе работы пользователь изменяет данные, которые находятся в оперативной памяти.

Если данные изменены, но не сохранены в постоянной памяти, то перед закрытием программы пользователь получает предложение сохранить данные.

Сценарий должен быть таким же, как при завершении работы с любым текстовым редактором, поэтому здесь не приводится.

1. **ПРОЕКТИРОВАНИЕ**
   1. ***Архитектура***

Программа будет создаваться на языке С# с использованием среды разработки Microsoft Visual Studio 2019 на платформе .NET Framework 4.7.2. с использованием интерфейса программирования приложений Windows Forms.

Для написания курсового проекта был выбран объектно-ориентированный подход, в котором мы концентрируемся на концепции представления, продумывая архитектуру нашей программы и только после этого приступаем к программированию. Язык для написания – С#, который является одним из самых популярных, удобных и полноценных объектно-ориентированных языков программирования. В нем удачно сочетаются испытанные средства программирования с самыми последними новшествами и предоставляется возможность для эффективного и очень практичного написания программ.

В качестве платформы для написания курсового перед нами встал выбор между .NET Core и .NET Framework. У каждой из этих платформ есть свои преимущества, проанализировав их мы предпочли преимущества .NET Framework, а именно большее количество управляющих элементов, чем в .NET Core, а также мы учли тот факт, что выбранная нами платформа является более зрелой, то есть та нету ошибок. Вообще, С# изначально предназначен для создания кода, который должен выполняться в данной среде. Ну и важно отметить, что библиотеки, используемые в С# определены в среде .NET Framework. Данная среда обеспечивает безопасность, переносимость программ и общую модель программирования для платформы Windows, однако наше приложение не сможет работать на машинах с другой операционной системой.

Для реализации нашего приложения был выбран графический интерфейс, поскольку нашей программой будут пользоваться люди не связанные со сферой программирования, соответственно им нужен интуитивно понятный интерфейс, в котором не нужно запоминать лишнего, а также который будет простым для восприятия. Мы будем использовать интерфейс Windows Forms, который же является наиболее простым и удобным и принадлежит платформе .NET Framework.

* 1. ***Структура проекта***

Если мы рассмотрим функционал нашей спецификации, то увидим, что нашей программой пользуется два пользователя у которых нет общих функций, поэтому нет смысла все размещать в одной программе, и будет целесообразно поделить наше приложение на два приложения, а именно: для покупателя и заведующего склада. Общей же частью данных приложений является модель, слой доступа к данным и хранилище, поэтому данную часть будет логично вынести отдельно. Поскольку модель и слой доступа к данным не имеют пользовательского интерфейса, то верным решением будет придать им форму библиотеки, с которой будут работать два наши приложения. Безусловно мы нуждаемся в тестировании наших данных, хорошим решением в нашем проектировании будет создать отдельную программу, отвечающую за тестирование данных.

Подытожим наши рассуждения структурой проекта:

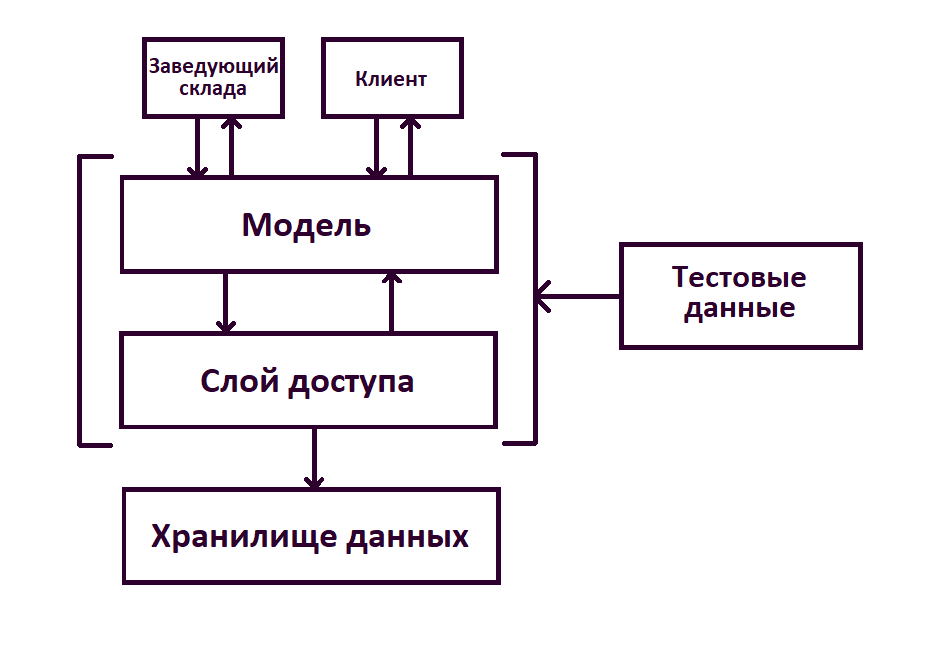


Рисунок 13 – Структура проекта

* 1. ***Объектная модель***

На **складе** имеются **товары,** которые изготовили на фабрике.

Товар периодически приходит на склад.

**Заведующий складом** вносит его в программу, после чего формируется **приходная накладная**. Также одобряет **закупку**, после этого формируется **расходная накладная**. Проводит инвентаризацию. Регистрирует новых пользователей.

**Клиенты** выбирают товар. Создают **заказ**, который подтверждает заведующий складом. Смотрят неподтвержденные заказы, а также **историю покупок.**

О товаре известно: **имя**, **единица измерения**, **цена единицы**, **кол-во**, **артикул**, **дата последнего завоза**.

О покупателях - **имя предприятия**, **почтовый адрес**, **логин** и **пароль**.

Выбираем те, что не относятся к простым типам:

Склад, товар, приходная накладная, заведующий склада, закупка, заказ, покупатели (клиенты), история покупок.

Их этих определяем те, что не являются коллекциями и коллекций не содержат:

Это – товар, клиент, заведующий складом.

Это будут первые пользовательские типы, т. е. классы:

Товар – это наименование + артикул + единица измерения + цена единицы + количество + дата последнего завоза.

Клиент – это имя предприятия (сделаем имя предприятия логином) + почтовый адрес + пароль.

Заведующий складом – это логин + пароль.

Имя, ед. измерения, цена единицы, количество, артикул, логин, почтовый адрес, пароль – это величины стандартных типов, т.е числа и строки.

Остальные термины можно определить так:

Порция – это товар + его количество

Закупка (Заказ) – это коллекция порций

Склад – это коллекция товаров, покупателей, продаж, поставок, заказов.

Также мы имеем приходную накладную (поставку) и расходную накладную (продажу). В своей сущности – это дата + коллекция порций товаров.

***Фасадный класс – склад.***

* 1. ***Методы***

К главным методам относятся:

1. Регистрация наших клиентов. Метод работает с данными коллекции “клиент”, которая находиться в фасадном классе, относим к фасадном классу
2. «Подтверждения пользователя» (при входе в программу). Данный метод будет принадлежать и классу Зав. склада и классу Клиент. Поскольку мы не впервые замечаем пересечение методов/полей/свойств у данных классов, то хорошим решением будет создать им общего предка, допустим, абстрактный класс User.
3. Создание заказа. Будет относиться к классу заказ, возможно, будет являться конструктором данного класса.
4. Одобрение заказа (Учет продаж товаров). Вполне логично, что тоже будет относиться к данному классу.
5. Просмотр истории покупок. Относиться к покупателю.
6. Учет поставок. Этим занимается администратор, однако данный метод меняет и товар в магазине и создает отчет. Возможно, данный метод стоит сделать методом фасадного класса.
7. Получение отчетов (наших накладных). Отнесем тоже к фасадному классу.

В дальнейшем методы будут добавляться по мере необходимости, возможно, некоторые наши методы будут в какой-то мере изменены.

* 1. ***Диаграммы***

После обсуждения следует изложить наши соображения в виде диаграмм для наглядности.

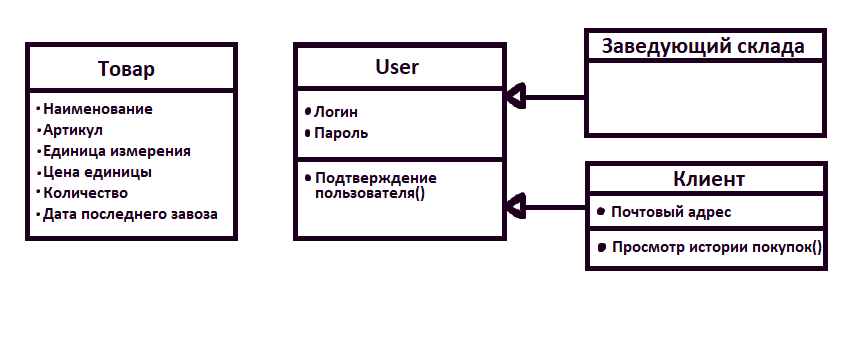


Рисунок 14 - Диаграммы

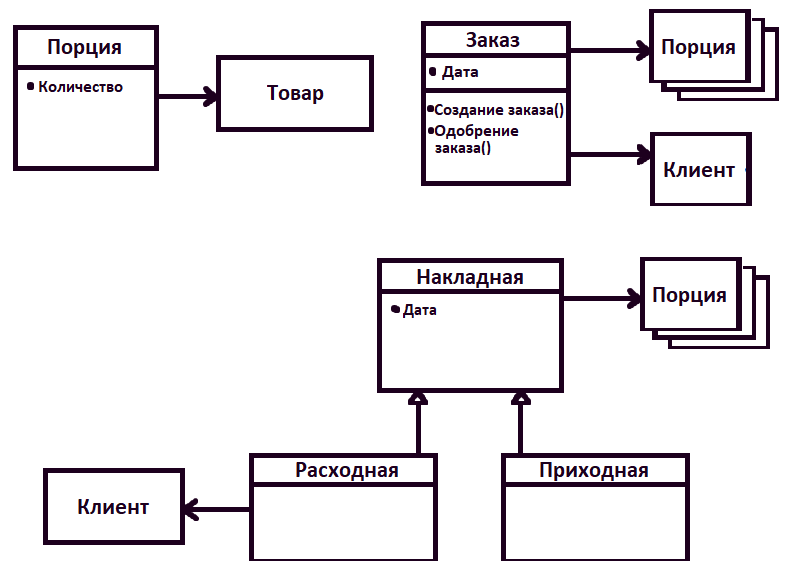


Рисунок 15 - Диаграммы

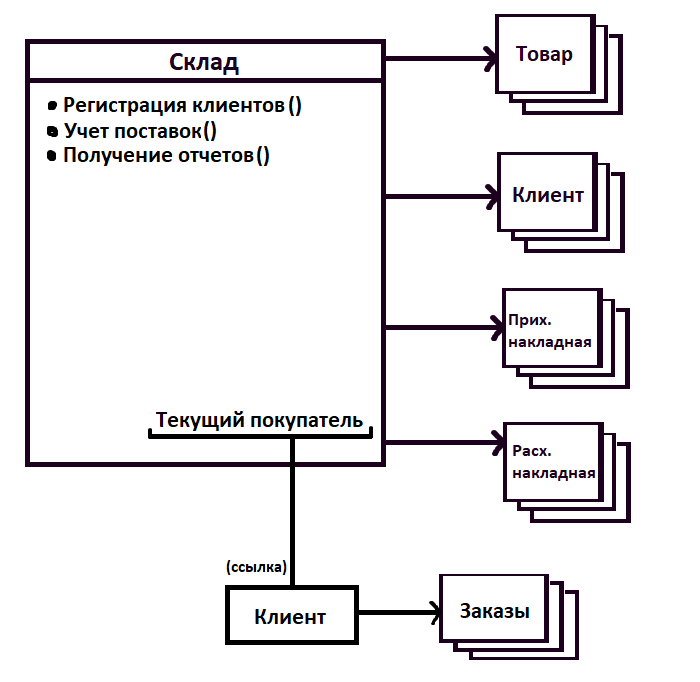


Рисунок 16 - Диаграммы

* 1. ***Хранение и доступ к данным***

Все наши данные, отмеченные словами «база» или «список», будут представлены в виде отдельных бинарных файлов в каталоге данного проекта. Данные будут переводиться из текстовых в поток байтов при помощи сериализации с использованием встроенной библиотеки BinaryFormatter, конечно, мы могли использовать библиотеку XmlSerializer как альтернативу нашей, но бинарные файлы занимают меньше места, а также мы не посылаем поток по сети, соответственно целесообразно будет использовать библиотеку BinaryFormatter. Все данные будут храниться в бинарном файле warehouse.bin.

1. **ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЮ**
   1. ***Установка программы.***

Процедура установки протекает таким образом:

* + - 1. Заходим по ссылке <https://github.com/u3o8/warehouse>.
      2. Загружаем проект на компьютер.
      3. Распаковываем архив.
      4. Открываем папку “App” запускаем нужное приложение. Для заведующего складом AdminApp, для клиента – CustomerApp. (рис. 17)

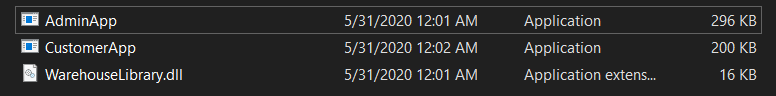


Рисунок 17 – Вид в папке

* 1. ***Работа с программой. Инструкция заведующему складом***

Запускаем программу. Открывается окно аутентификации (рис. 18). Вводим логин и пароль и нажимаем кнопку “OK”. Произойдет переход на главную страницу заведующего складом.

При первом входе по умолчанию логин и пароль стоят admin1. В дальнейшем пароль можно сменить на свое усмотрение.

Если вы считаете, что вам ненужно совершать вход в программу, то закройте программу с помощью кнопки “Cancel” или крестика в левом верхнем уголке.

На данной форме пароль скрыт при его введении, однако его можно запросто отобразить, используя кнопку с изображением глаза. Скрыть обратно пароль можно теми же манипуляциями.

Заранее стоит оговорить, что верхняя панель работает также как и во всех приложениях, а именно вы можете скрыть, расширить и закрыть окно. С каждой формы можно выйти нажатием крестика в левом верхнем уголке.

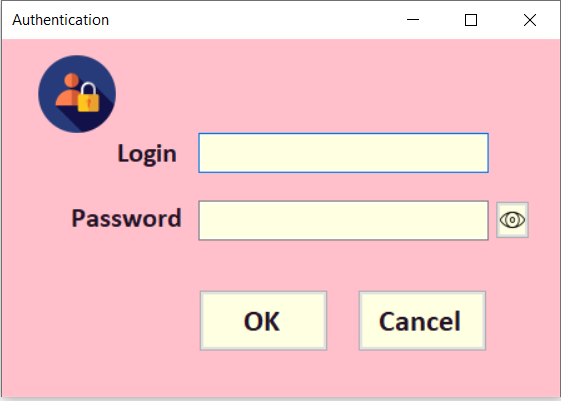


Рисунок 18 – Окно аутентификации

После входа в программу вы увидите главную страницу заведующего складом (рис. 19). Отсюда проводятся все основные манипуляции.

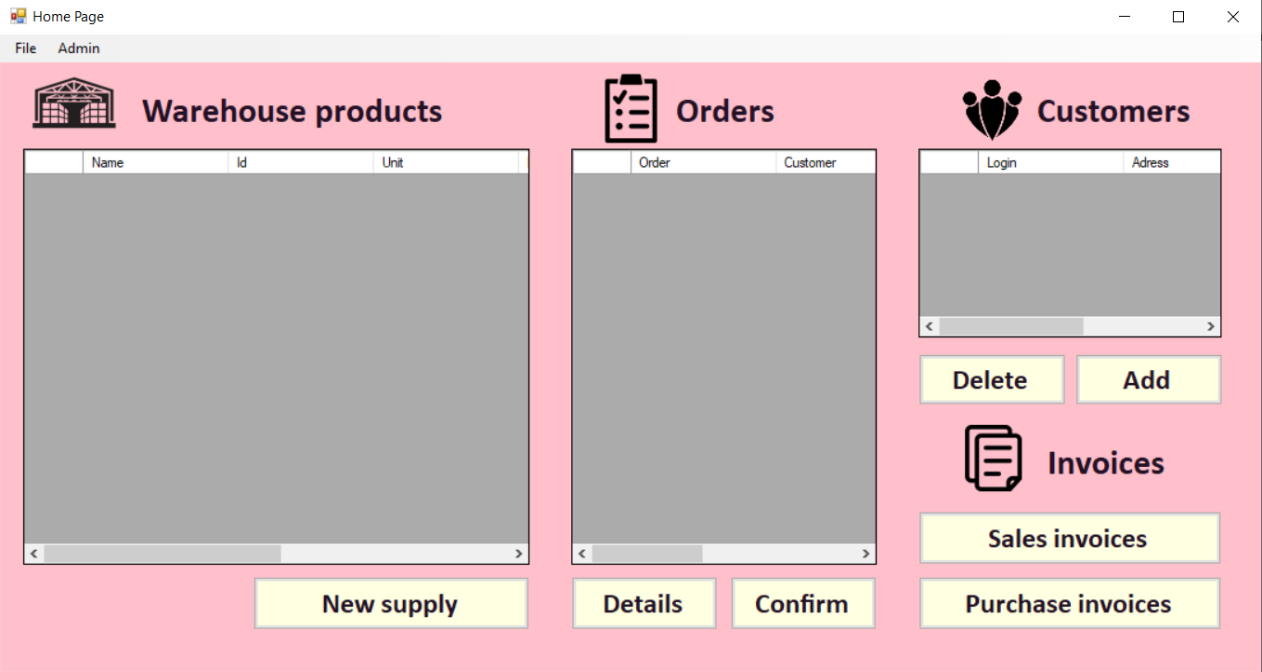


Рисунок 19 – Главная страница заведующего складом

Мы видим на данной странице три сетки, или же таблицы. Первая таблица содержит информацию о товаре на складе, подписана она “Warehouse products”. Вторая таблица содержит информацию о неподтвержденных заказах от покупателей, которые заведующий складом может подтвердить, подписана она “Orders”. Третья таблица, которая подписана “Customers” содержит информацию о клиентах, которые числятся. Сверху на панели содержится выпадающее меню. Также тут размещены разные кнопки, предназначение которых будет оговорено дальше в инструкции к данной программе.

Начнем с того, как вести учет поставок. Для того, что зарегистрировать новую поставку нужно для начала нажать кнопку “New supply”.

После этого вы увидите окошко, в котором составляется новая поставка.

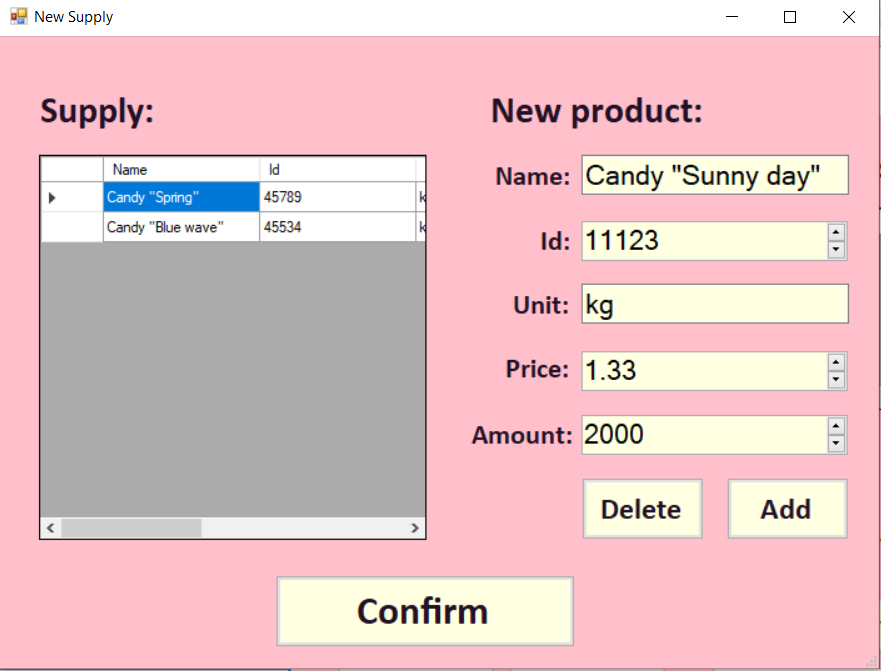


Рисунок 20 – Окно новой поставки

Остается только ввести информацию о товаре и нажать “Add”. После этого товар появится в таблице данной поставки. Следующие товары, которые вы хотите внести в проставку, добавляются таким же образом. При надобности можно удалить товар из поставки. Для этого в таблице нужно нажать на товар, который вы хотите удалить и нажать на кнопку “Delete”.

Когда поставка составлена нажмите на кнопку “Confirm”. Это действие внесет продукты в программу, а вас вернет на главную страницу заведующего складом.

Теперь рассмотрим, как зарегистрировать клиента. Для этого нужно нажать на кнопку “Add”, которая находится под таблицей пользователей на главной странице. Откроется окно для регистрации пользователя. (рис. 21)

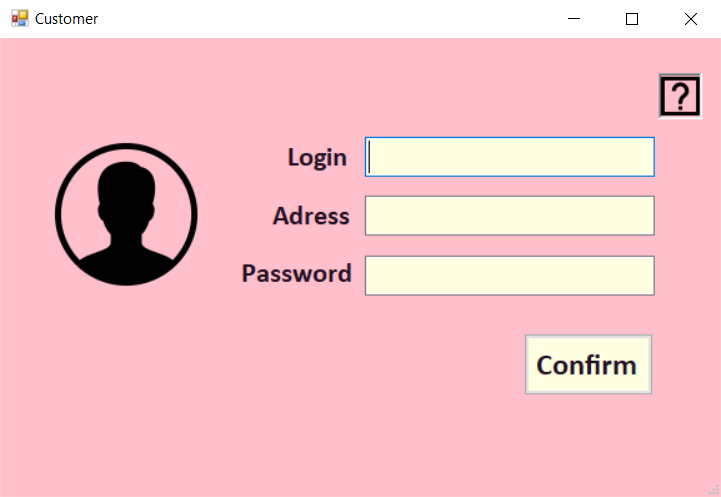


Рисунок 21 – Окно регистрации клиента

Для регистрации нового клиента нужно ввести данные и нажать “Confirm”. Стоит обратить внимание на предупреждение (рис. 22)

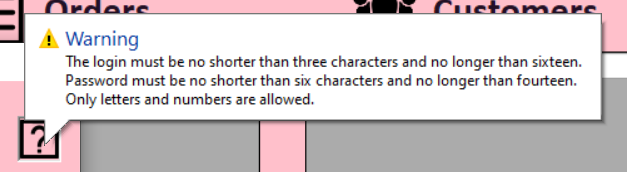


Рисунок 22 - Предупреждение

Если ввести данные некорректно, то пользователь не будет зарегистрирован в системе, а программа выведет сообщение о некорректности ввода данных (рис. 23).

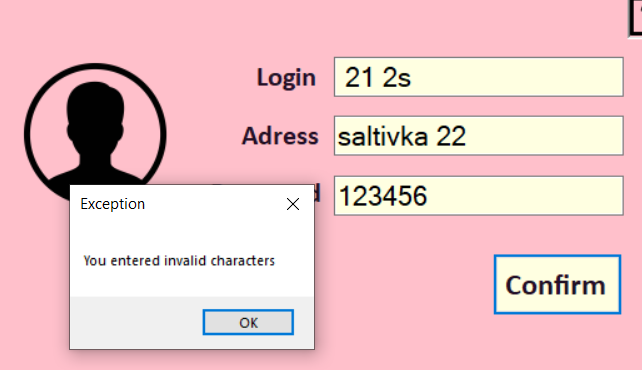


Рисунок 23 – Сообщение о некорректном вводе данных

После успешной регистрации пользователь появится в таблице, а вы вернетесь на главную страницу.

Для удаления клиента нужно на главной странице выбрать его в таблице и нажать кнопку “Delete” и подтвердить действие. (рис. 24)

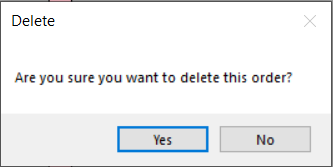


Рисунок 24 – Подтверждение удаления

Перейдем к работе с заказами, для того, чтоб подтвердить заказ нужно выбрать заказ в таблице и нажать на кнопку “Confirm”, которая расположена на главной странице и подтвердить действие. После этого заказ считается обслуженным и удаляется с таблицы. Для просмотра деталей заказа, прежде чем вы его захотите подтвердить, выберите заказ и нажмите кнопку “Details”. Откроется окно с деталями данного заказа. (рис. 25)

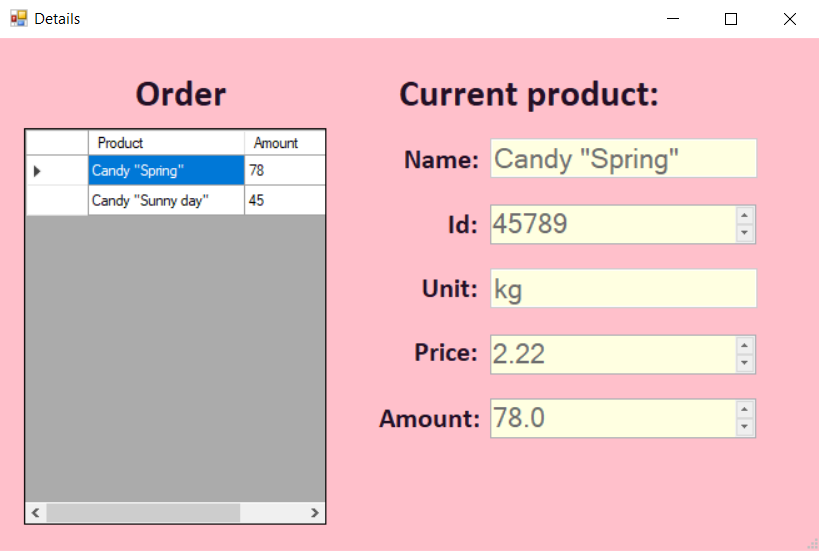


Рисунок 25 – Детали заказа

Переключаться между продуктами можно кликая на них в таблице. Выход из данного окна осуществляется нажатием крестика в левом верхнем углу.

Для просмотра накладных нужно нажать на соответствующую кнопку.

Для приходных накладных – “Purchase invoices”. Откроется окно для просмотра (рис. 26)

Для расходных накладных - “Sales invoices”. Откроется окно для просмотра (рис. 27)

С помощью кликов по таблицам можно получать более подробную информацию для активного элемента.

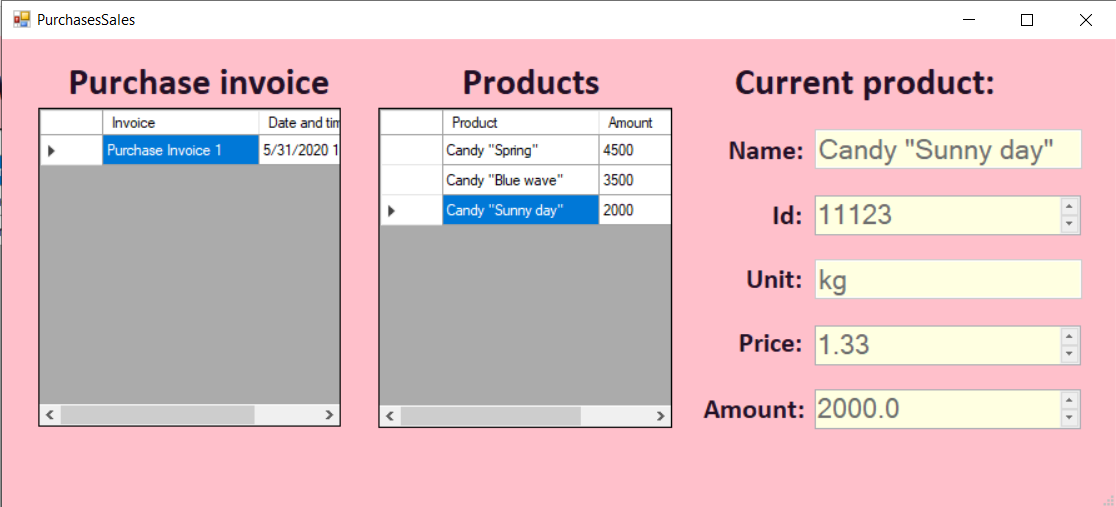


Рисунок 26 – Окно для просмотра приходных накладных

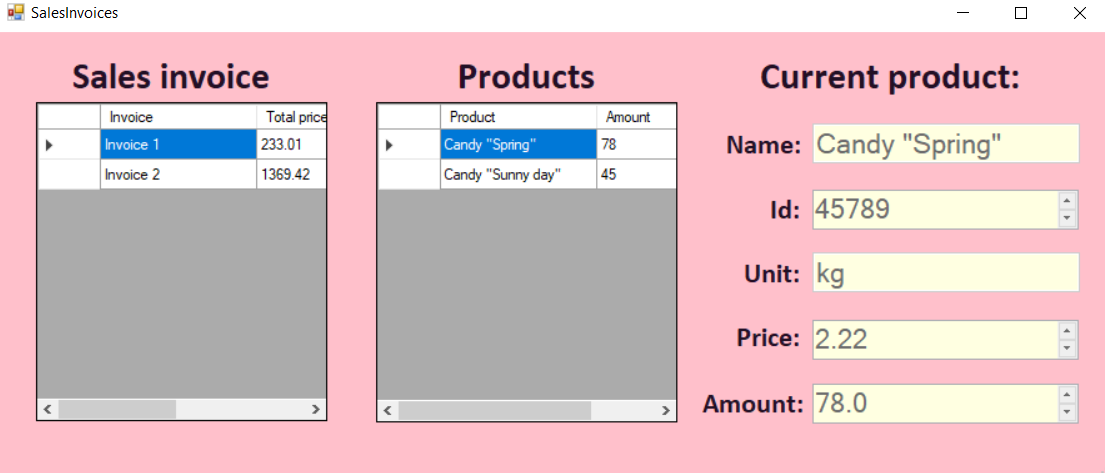


Рисунок 27 – Окно для расходных накладных

Теперь рассмотрим работу с выпадающим меню на главной странице. (рис. 28)

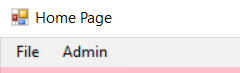


Рисунок 28 – Выпадающее меню

Нажмем кнопку “Admin”. Увидим, что доступно еще две кнопки. (рис. 29)

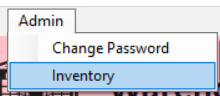


Рисунок 29 – Нажатие кнопки “Admin”

Нажав на “Change password” откроется окно, в котором можно сменить пароль. (рис. 30)

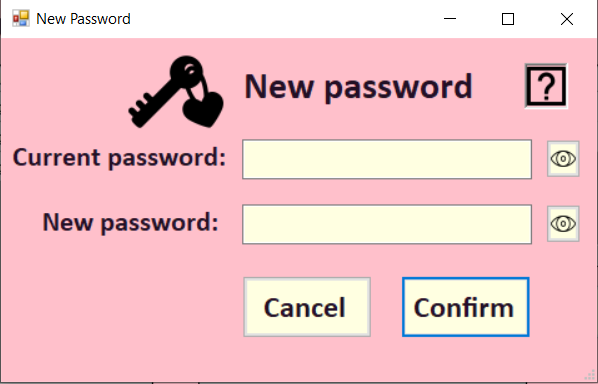


Рисунок 30 - Окно для смены пароля

Следует снова обратить внимание на предупреждение. (рис. 31)

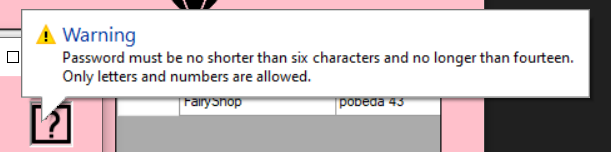


Рисунок 31 – Предупреждение

Как и в главном меню пароль можно отобразить, нажав на кнопку с изображением глаза.

Для успешного изменения пароля нужно ввести новый пароль, учитывая все требования и верный старый пароль. После нажать “Confirm”.

Провести инвентаризацию можно нажав сначала на кнопку “Admin”, а потом на “Inventory”. После появится окно инвентаризации (рис. 32)

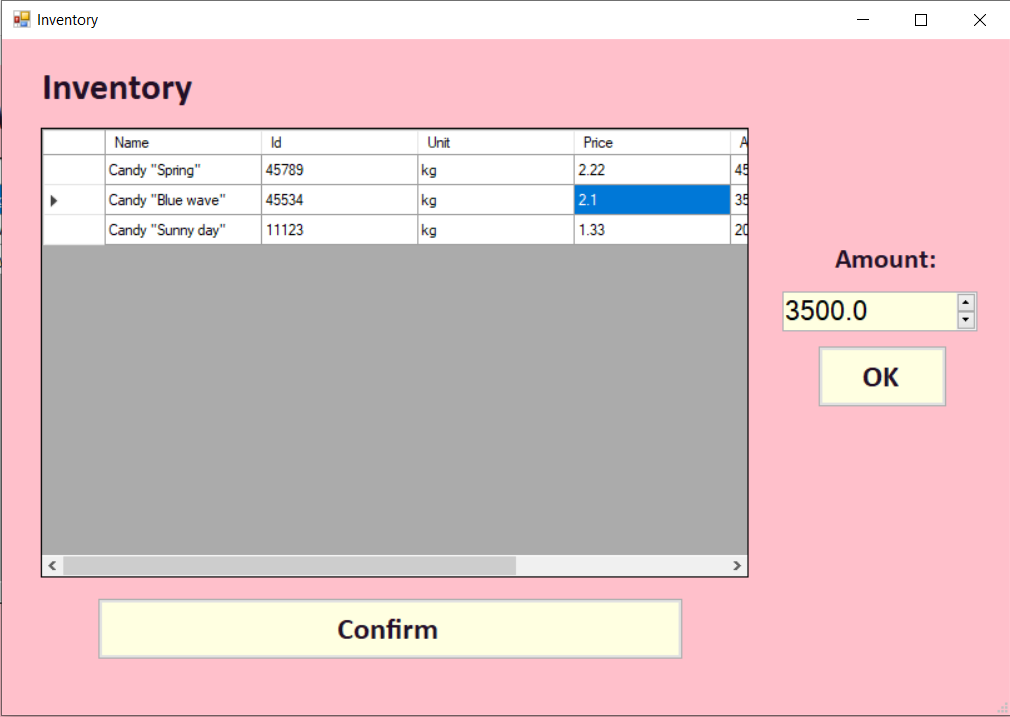


Рисунок 32 – Окно инвентаризации

Кликая на продукт мы делаем его текущем и можем поменять его стоимость введя его количество и нажав кнопку “OK”.

Когда инвентаризация проведена нажимаем кнопку “Confirm”, чтобы внести изменения в продукты.

Теперь рассмотрим в заключении как сохранять и загружать данные.

Сохранить изменения, которые мы внесли в программу в течении сеанса можно с помощь нажатий клавиш “File” -> “Save”. (рис. 33)

Загрузить последние сохраненные данные – “File” -> “Load” (рис. 33)

Выйти из программы можно нажатием крестика в левом верхнем углу или нажатием кнопок “File” -> “Exit” (рис. 33)

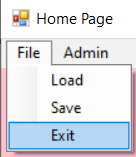


Рисунок 33 – Нажатие на кнопку “File”

Когда вы будете выходить программа также позаботиться о том, чтобы вы сохранили данные (рис. 34)

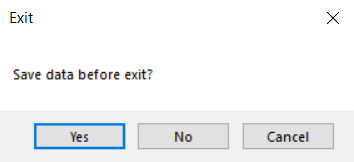


Рисунок 34 – Выход

* 1. ***Работа с программой. Инструкция клиенту***

Запускаем программу. Открывается окно аутентификации (рис. 35). Вводим логин и пароль и нажимаем кнопку “OK”. Произойдет переход на главную страницу клиента.

Для того, чтобы узнать ваш логин и пароль обратитесь к заведующему склада. При потребности в регистрации – тоже следует обратиться к заведующему складом.

Если вы считаете, что вам ненужно совершать вход в программу, то закройте программу с помощью кнопки “Cancel” или крестика в левом верхнем уголке.

На данной форме пароль скрыт при его введении, однако его можно запросто отобразить, используя кнопку с изображением глаза. Скрыть обратно пароль можно теми же манипуляциями.

Заранее стоит оговорить, что верхняя панель работает также как и во всех приложениях, а именно вы можете скрыть, расширить и закрыть окно. С каждой формы можно выйти нажатием крестика в левом верхнем уголке.

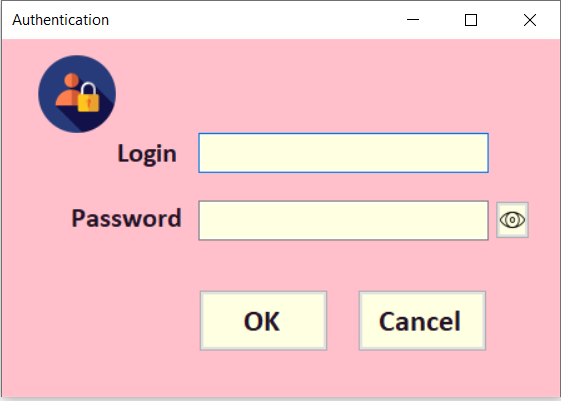


Рисунок 35 – Окно аутентификации

После входа в программу вы увидите главную страницу клиента (рис. 36). Отсюда проводятся все основные манипуляции.

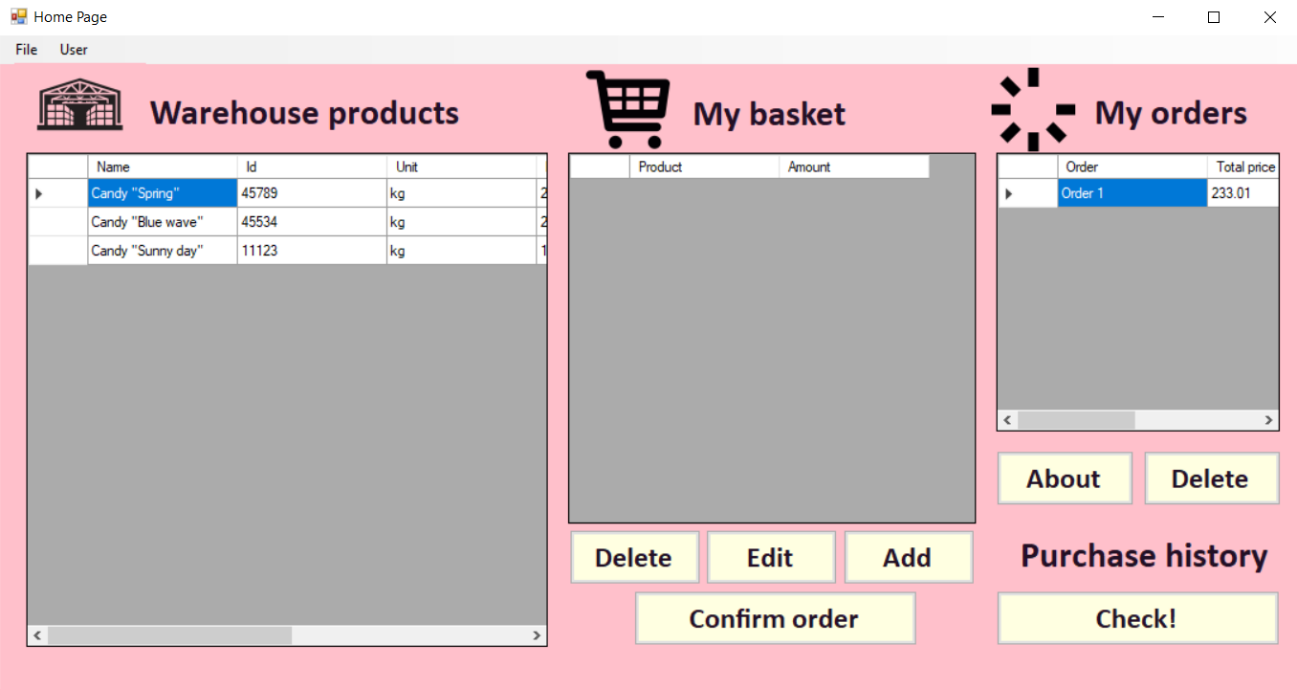


Рисунок 36 – Главная страница клиента

Мы видим на данной странице три сетки, или же таблицы. Первая таблица содержит информацию о товаре на складе, подписана она “Warehouse products”. Вторая таблица, которая подписана “Basket” является корзиной, тут будут отображены товары, которые мы хотим внести в заказ. Третья таблица, которая подписана “My Orders” содержит информацию о неподтвержденных заказах. Также на странице размещены разные кнопки, предназначение которых будет оговорено дальше в инструкции к данной программе.

Перед тем как отправить заказ на запрос нужно заполнить корзину. Добавить товар в корзину можно с помощью кнопки “Add”, которая находится на главной странице. Откроется окно добавления товара, где можно ввести количество. (рис. 37)

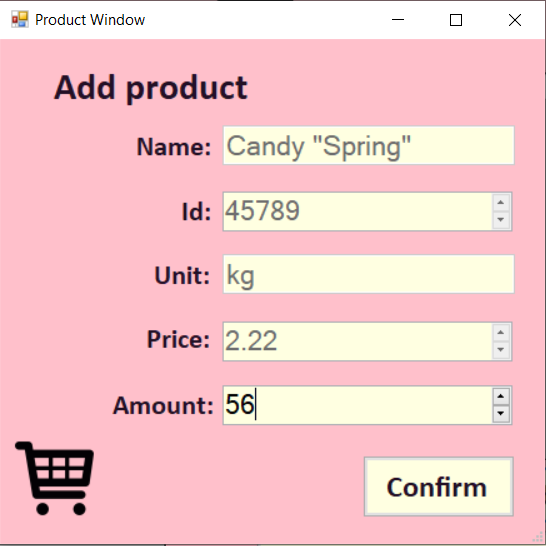


Рисунок 37 – Окно для добавления товара в корзину

Для подтверждения внесения товара нужно нажать кнопку “Confirm”.

Нельзя действовать всегда безошибочно, поэтому нужно иметь возможность редактировать товар в корзине, а также его удалять. За это отвечают кнопки “Delete” и “Edit” на главной странице. При нажатии на кнопку “Delete” нужно будет подтвердить свое действие (рис. 38). При нажатии “Edit” вы увидите окно для редактирования количества товара (рис. 39), введите новое значение количества и нажмите кнопку “Confirm”.

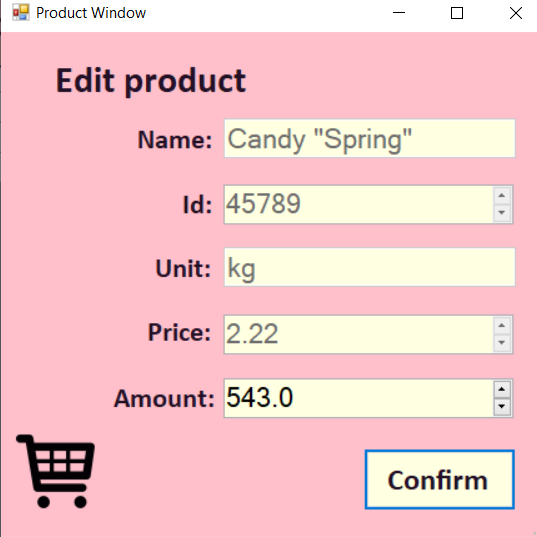


Рисунок 38 – Форма для редактирования товара

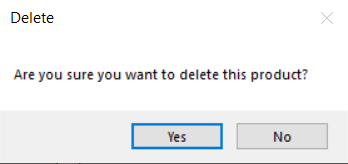


Рисунок 39 – Форма подтверждения удаления

Когда вы внесете все желанные товары в корзину, отправьте запрос на одобрение заказа с помощью кнопки “Confirm”, которая находится на главной странице. Данное действие тоже нужно подтвердить.

Для просмотра детальной информации о неподтвержденных заказах выбираете заказ и нажимаете кнопку “About” на главной странице. Откроется окно дополнительной информации (рис. 40). Кликая по товару, можно получить дополнительную информацию.

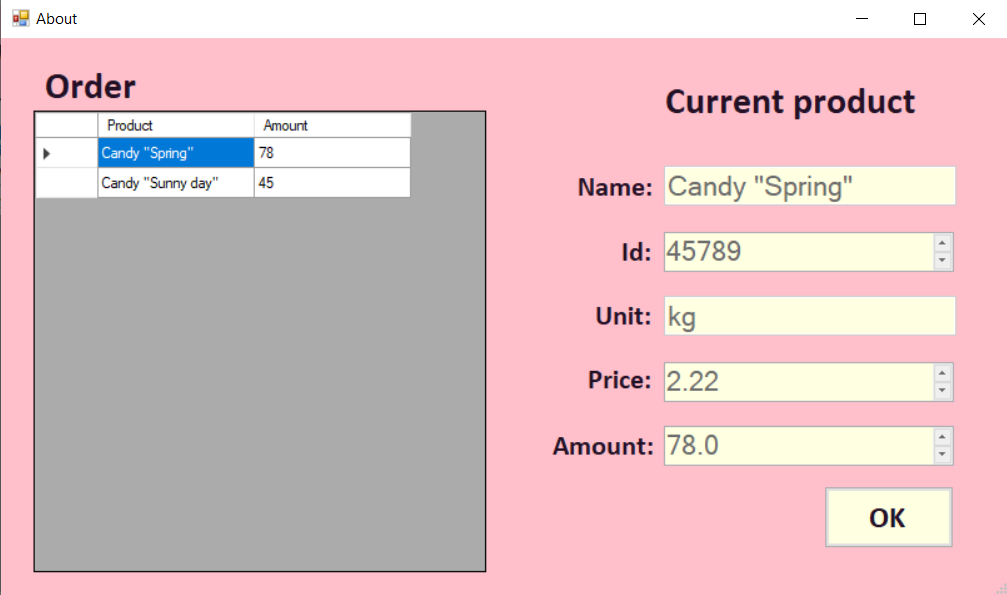


Рисунок 40 – Окно дополнительной информации

Удалить можно выбрав заказ и нажав кнопку “Delete”. Программа запросит у вас подтверждение данного действия.

Просмотр истории покупок доступен по нажатию на кнопку “Check!”. Откроется окно для просмотра истории покупок (рис. 41) Дополнительную информацию можно получить, кликая по ячейкам таблицы.

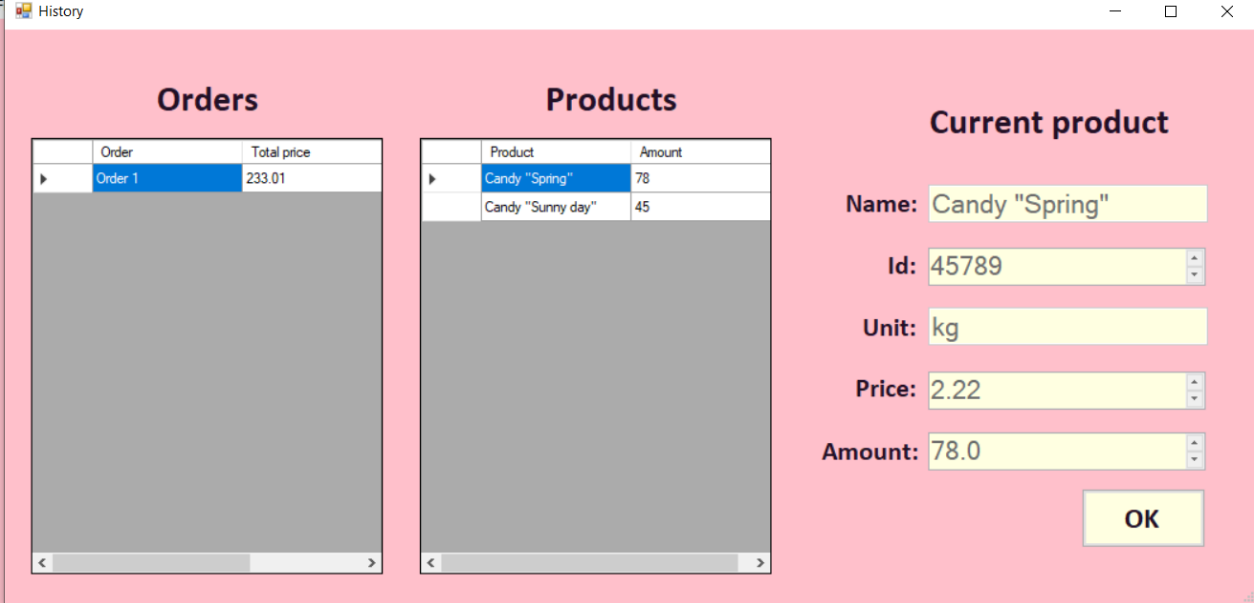


Рисунок 41 – Окно истории покупок

Теперь рассмотрим работу с выпадающим меню на главной странице. (рис. 42)

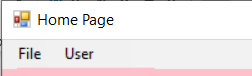


Рисунок 42 – Выпадающее меню

Нажмем кнопку “User”. Там есть только одна кнопка “About”, которая выводит информацию о пользователе

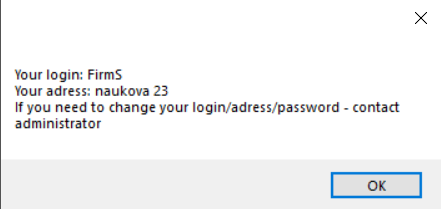


Рисунок 43 – Меню с информацией о клиенте

Теперь рассмотрим в заключении как сохранять и загружать данные.

Сохранить изменения, которые мы внесли в программу в течении сеанса можно с помощь нажатий клавиш “File” -> “Save”. (рис. 44)

Загрузить последние сохраненные данные – “File” -> “Load” (рис. 44)

Выйти из программы можно нажатием крестика в левом верхнем углу или нажатием кнопок “File” -> “Exit” (рис. 44)

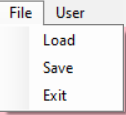


Рисунок 44 – Нажатие на кнопку “File”

Когда вы будете выходить программа также позаботиться о том, чтобы вы сохранили данные (рис. 45)

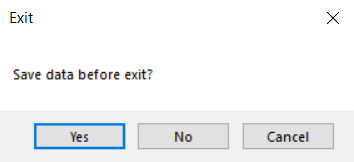


Рисунок 45 - Выход

1. **ТЕСТИРОВАНИЕ ДАННЫХ**

В данном проекте существует тестирующая система, которая имеет 23 автоматических независимых теста, для тестирования методов по отдельности. Данные тесты вынесены в отдельную программу. Для ручного тестирования можно сгенерировать данные, запустив тест GenerateData, или же просто скопировать из папки TestData warehouse.bin и добавить данный файл в папку App.

**ВИСНОВКИ**

Целью данной курсовой работы являлось создание проекта. Создание данного проекта сопровождалось несколькими этапами. Вначале была создана спецификация, которую на следующих этапах требовалось реализовать. На этапе проектирования мы определили структуру программы, спроектировали объектную модель и описали, как будут храниться данные в постоянном хранилище. Следующим делом мы программировали модель и пользовательский интерфейс. Важно отметить, что то, что мы указали на этапе, проектирования подверглось неким изменениям в силу того, что довольно-таки сложно с первого раза безошибочно составить проектную спецификацию, однако основные моменты, которые были описаны в ней были соблюдены. Последним этапом выполнения курсового проекта, было составление пояснительной записки. В результате мы получили полноценный программный продукт.