Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліні "Об’єктно-орієнтоване програмування"

ІНТЕРНЕТ-МАГАЗИН З ОДНИМ ПРОДАВЦЕМ

Керівник, Ст. викл Черепанова Ю.Ю.

Студент гр. ПЗПІ-18-7 Владимиров І.М.

Комісія:

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кравець Н.С.

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мар'їн С.О.

Ст. викл. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю.Ю.

Харків 2019

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ**

Кафедра: ***Програмної інженерії***

Дисципліна: ***Об’єктно-орієнтоване програмування***

Спеціальність: ***121***  ***Інженерія програмного забезпечення***

Освітня програма: ***Програмна інженерія***

Курс 1 .Група *ПЗПІ-18-7* . Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проект студента***

*Владимиров Іван Максимович*

(Прізвище, Ім'я, По батькові)

1 Тема проекту: Інтернет-магазин з одним продавцем

2 Термін здачі студентом закінченого проекту: ***“13” - червня - 2019 р***.

3 Вихідні дані до проекту:

*Специфікація програми, методичні вказівки до виконання курсової роботи*

4 Зміст розрахунково-пояснювальної записки:

*Вступ,* *специфікація програми, проектна специфікація, інструкція користувача, висновки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Перелік графічного матеріалу:

*Схема об’єктної моделі, приклади екранних форм, вигляд спроектованного для проекта сайту, структура бази данних*.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми | 19-02-2019 р. |
| 2 | Формулювання вимог до програми | 19-02-2018 – 30-03-2019 р. |
| 3 | Розробка структури бази данних. | 30-03-2019 – 02-04-2019 р. |
| 4 | Розробка підсистеми роботи з базою данних з використанням WinForms. | 02-04-2019 – 10-04-2019 р. |
| 5 | Розробка сайту інтернет магазину з викроистанням технології Razor Pages | 10-04-2019 – 20-04-2019 р. |
| 6 | Розробка підсистеми обслуговування сайту з використанням APS.NET. | 15-04-2019– 25-04-2019 р. |
| 7 | Тестування і доопрацювання розробленої програмної системи. | 25-04-2019 – 11-05-2019 р. |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 11-05-2019 – 10-06-2019 р. |
| 9 | Захист | 10-06-2019 – 15-06-2019 р. |

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Прізвище, Ім'я, По батькові)

« 18 »\_лютого\_\_\_\_\_\_\_\_\_\_ 2019 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 82 с., 17 рис., 1 додаток, 5  джерел.

КЛАС, МОВА ПРОГРАМУВАННЯ C#, ОБ’ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПРОГРАМА, ПЛАТФОРМА .NET, БАЗА ДАННИХ MYSQL, ASP.NET, RAZOR PAGES, ІНТЕРНЕТ МАГАЗИН, WINDOWS FORMS.

Метою роботи є розробка програми “SpearShop” на засадах об'єктно-орієнтованого програмування.

Методи розробки базуються на використанні середи розробки Microsoft Visual Studio 2019, EntityFrameworkCore [4], Windows Forms, платформи .NET Framework 4.8 [1], мови програмування C# [2], реляційної бази данних MySql, технології APS.NET [3] та Razor Pages [5].

В результаті отримана програма під назвою “SpearShop”, яка складється з двох частин. Перша, що працює на Windows Forms, дозволяє створювати, редагувати та видаляти контент що буде відображатися на сайті інтернет магазину. Друга частина працює на сайті у веб-браузері, на якому користувач може зручно переглянути товари, заказати їх, переглянути інформацію щодо них, наприклад зображення, опис, відео та інше. Також, в обох частинах є функції пошуку критеріями та сортування відображення, наприклад за ціною.

**ЗМІСТ**

Вступ 7

1 Специфікація програми 8

* + 1. Загальний вигляд головного

вікна програми(настільна частина) 8

1.1.2 Сторінка “Goods” 9

1.1.2.1 Основний функціонал 9

1.1.2.2 Таблиця товарів 10

1.1.2.3 Вікно редагування товару 10

1.1.2.4 Операція пошуку 11

1.1.2.5 Операція сортування 11

1.1.3 Сторінка “Orders” 12

1.1.3.1 Основний функціонал 12

1.1.3.2 Таблиця заказів 13

1.1.3.3 Вікно редагування заказу 13

1.1.3.4 Операція пошуку 14

1.1.3.5 Операція сортування 14

1.1.4 Сторінка “Customers” 15

1.1.4.1 Основний функціонал 15

1.1.4.2 Таблиця покупців 15

1.1.4.3 Вікно редагування покупця 16

1.1.5 Сторінка “Categories” 17

1.1.5.1 Основний функціонал 17

1.1.5.2 Таблиця категорій 18

1.1.5.3 Вікно редагування категорії 18

1.1.6 Сторінка “Manufacturers” 19

1.1.6.1 Основний функціонал 19

1.1.6.2 Таблиця виробників 20

1.1.6.3 Вікно редагування виробника 20

1.1.7 Сторінки “Delivery types” 21

1.1.7.1 Основний функціонал 21

1.1.7.2 Таблиця типів доставки 22

1.2.1 Загальний вигляд головного вікна програми (веб частина) 23

1.2.2 Загальний вигляд вікна товару(веб частина) 24

2 Проектна специфікація 25

2.1 Загальне проектування 25

2.2 Об’єктна структура програми 25

2.2.1 Модель програми 25

2.2.2 Форми програми 28

2.2.3 Контролери 30

2.2.4 RazorPages 30

3 Інструкція користувача 32

3.1 Системні вимоги 32

3.2 Встановлення та запуск 32

3.3 Навігація у програмі 32

3.4 Редагування полів 33

3.5 Перегляд інформації що буде використовуватися на сайті 33

3.6 Сторінка елементу 34

Висновки 35

Перелік джерел посилання 36

Додаток А Код програми.. 37

**ВСТУП**

У наш час дуже важливу роль відіграє інтернет-комерція.

Метою виконання курсової роботи є створення програми під назвою “SpearShop”, яка дозволяє змінювати наповнення сайту інтернет-магазину, Результатами роботи мають стати також закріплення знань, набутих при вивченні дисципліни "Об’єктно-орієнтоване програмування", а головне, придбання навичок у розробці програмних продуктів з використанням популярних технологій.

Розроблювана програма призначена для власників інтернет-магазинів, яким потрібно контролювати товари.

Для задоволення потреб користувачів розроблені функції додавання, редагування та видалення елементів сайту. Усім цим легко користуватися, а результати дій є наочними.

Інтерфейс веб-сайту є інтуітивно зрозумілим для користувача. При цьому реалізована адаптивна структура, що робить відображення сайту коректним на різних пристроях таких як телефони, настільни комп’ютери, планшети та інше.

Закази на сайті оброблюються автоматично та автоматично змінюється інформація у базі данних. Отже від користувача необхідно лише оброблювати закази(відправлення, зміна статуса та інше).

Програма має властивість легко змінювати свій масштаб, тобто при збільшенні потреб сайту, завдяки гнучкої структури, доступно малозатратне розширення функціоналу програми. Але й ця версія досить гарно задовольняє потреби невеликих інтернет-магазинів.

**1 СПЕЦИФІКАЦІЯ ПРОГРАМИ**

1.1.1 Загальний вигляд головного вікна програми (настільна частина)

Увесь основний функціонал програми повинен бути зібраний в одному програмному вікні, зміст якого повинен змінюватися у вигляді окремих вкладок (див. рис. 1.1.1).

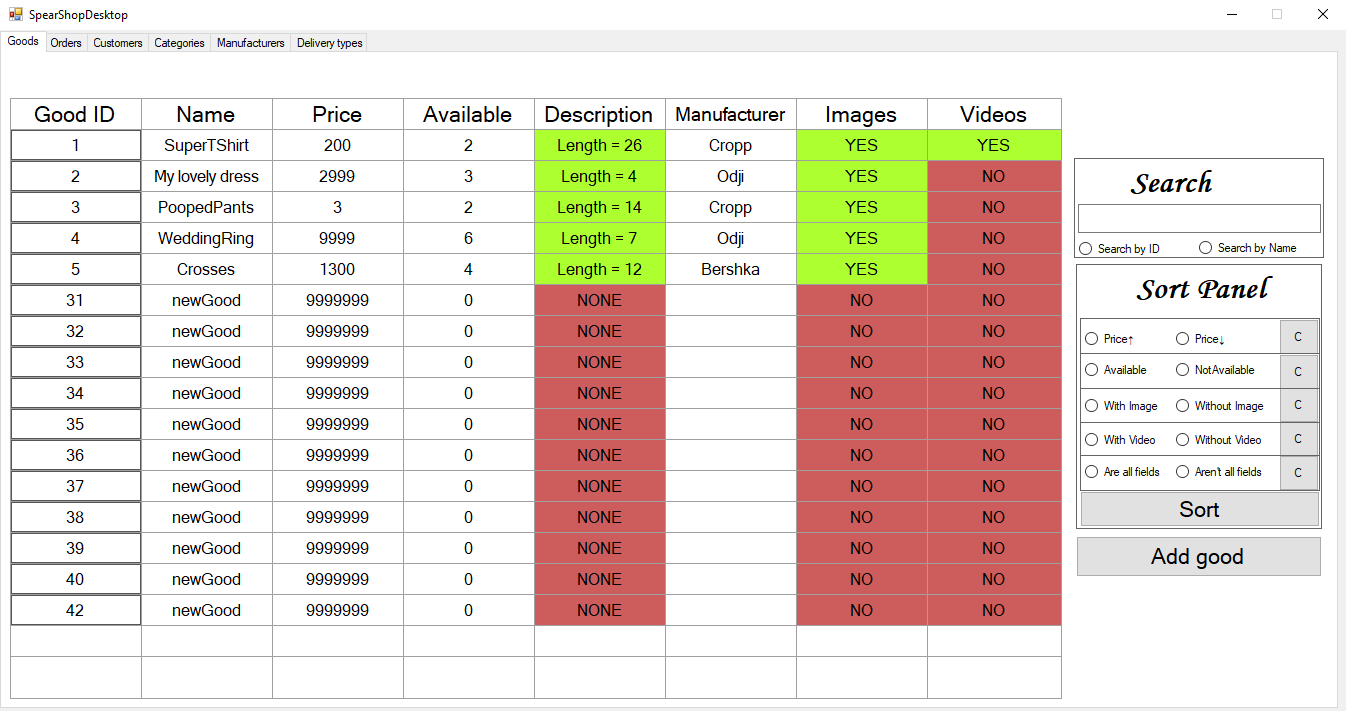


Рисунок 1.1.1 – Головне вікно програми

Перехід між сторінками повинен здійснюватися за допомогою вкладок, що розміщуються з верху вікна. При цьому, вкладка, яка відповідає за активну сторінку повинна бути світліша за кольором.

Стандартною сторінкою є сторінка товарів(Goods).

1.1.2 Сторінка “Goods”

1.1.2.1 Основний функціонал

Головна сторінка програми, яка називається “Goods”, має виконувати функцію відображення таблиці товарів, пошуку товару, та сортування відображення товарів. (див. рис. 1.1.2.1).

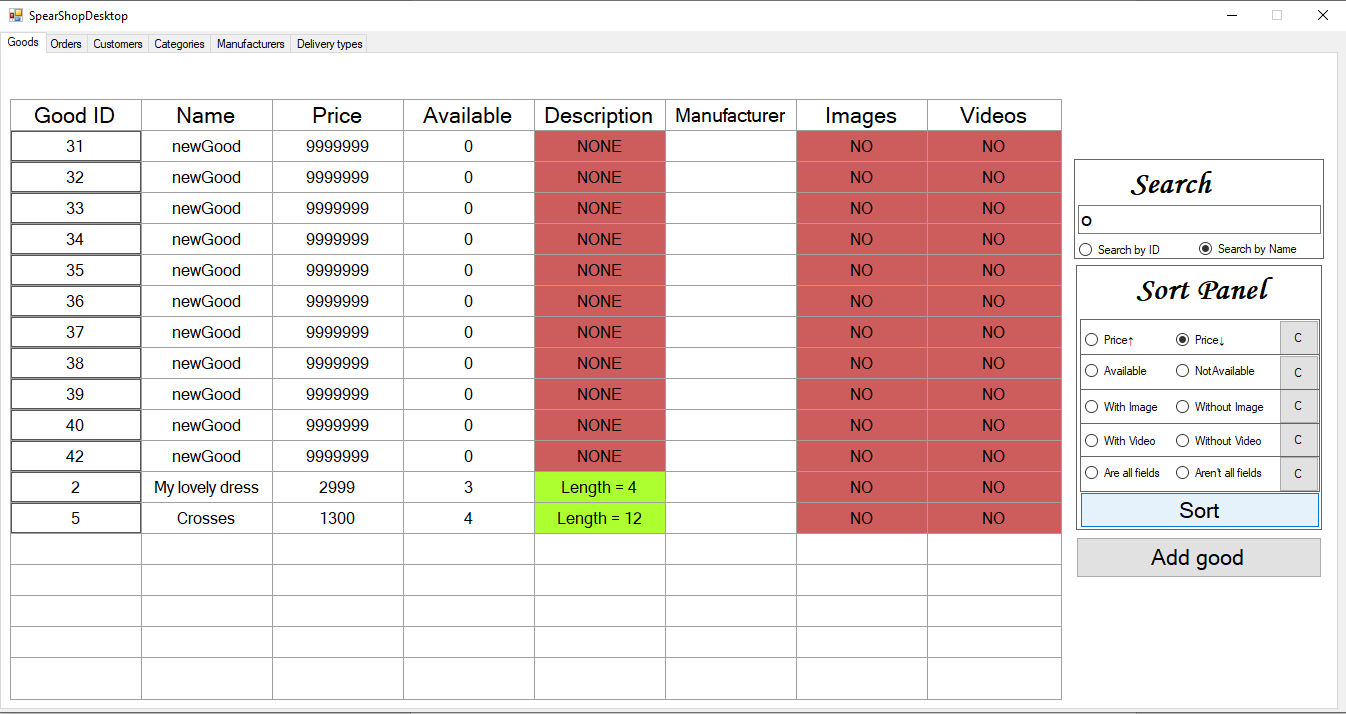


Рисунок 1.1.2.1 – Сторінка “Goods”

Ця сторінка повинна бути наочною та містити функції, які користувач буде використовувати найчастіше: перегляд, пошук, сортування.

1.1.2.2 Таблиця товарів

Основним джерелом інформації на цій сторінці, має бути таблиця, яка містить товари та інформацію про них. Рядками цієї таблиці є товари, а стовпцями – інформація про них.

1.1.2.3 Вікно редагування товару

Вікно редагування товару має містити основні функції для роботи з товаром, такі як встановлення чи зміна категорій, зображень, відео, опису, назви, ціни, кількості на складі, виробника, а також кнопки видалення товару та збереження. (див. рис. 1.1.2.3)

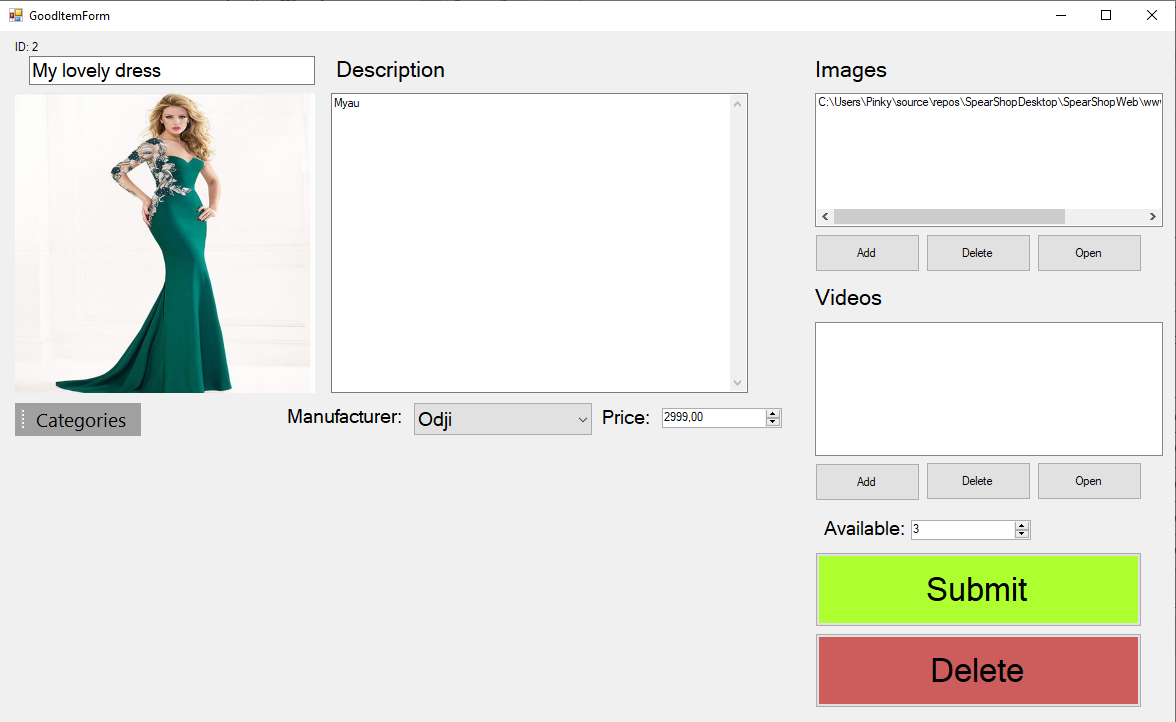


Рисунок 1.1.2.3 – Сторінка товару

Ця сторінка слугуватиме для відображення детальної інформації про товар та зміни цієї інформації. Кнопка “Submit” буде виконувати функцію збереження, а кнопка “Delete” – видалення. Кнопки у блоках Videos та Images “Add”, “Delete”, “Open” добавляють, видаляють чи відкривають відео чи зображення відповідно. У випадаючих меню “Manufacturer” та “Categories” можна вибрати виробника та категорії відповідно. Текстове поле над зображенням слугує для зміни назви товару. Поле під заголовком “Description” для зміни опису товару. Числові поля “Price” та “Available” характеризують ціну та наявність відповідно.

1.1.2.4 Операція пошуку

Така операція пошуку використовується й на інших сторінках. Пошук можна робити чи по імені чи по id, для вказання цього слугує дві радіокнопки під строкою пошуку “Search by name” та “Search by ID” відповідно.

1.1.2.5 Операція сортування

.Сортування таблиці відбувається шляхом вибору деяких фільтрів що представлені у блоці сортування. Доступно сортування по натупник признакам: ціна, наявність на складі, наявність відео, наявність зображення та наявність інформації в усіх полях. Для відміни фільтру необхідно натиснути кнопку “С” у відповідному рядку.

1.1.3 Сторінка “Orders”

1.1.3.1 Основний функціонал

Головна сторінка програми, яка називається “Orders”, має виконувати функцію відображення таблиці товарів, пошуку товару, та сортування відображення заказів. (див. рис. 1.1.3.1).

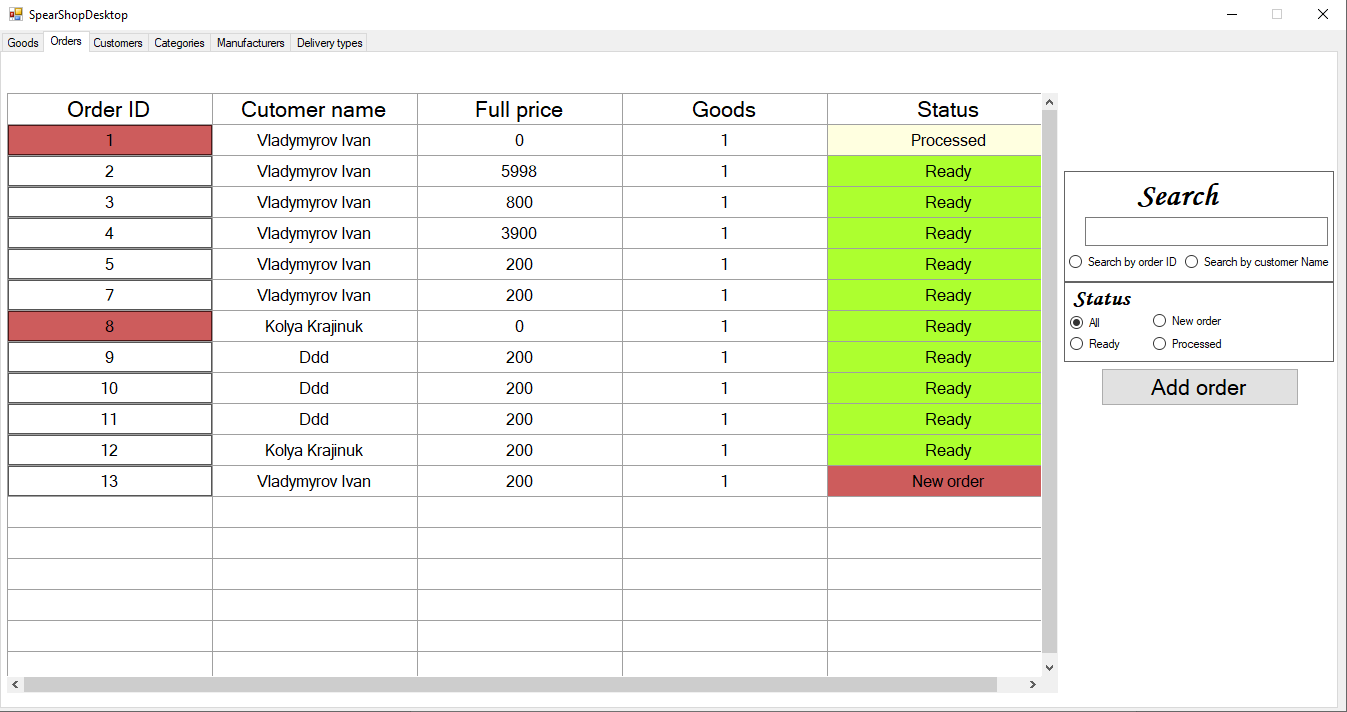


Рисунок 1.1.3.1 – Сторінка “Orders”

Ця сторінка повинна бути наочною та містити функції, які користувач буде використовувати найчастіше: перегляд, пошук, сортування.

1.1.3.2 Таблиця заказів

Основним джерелом інформації на цій сторінці, має бути таблиця, яка містить закази та інформацію про них. Рядками цієї таблиці є закази, а стовпцями – інформація про них.

1.1.3.3 Вікно редагування заказу

Вікно редагування товару має містити основні функції для роботи з товаром, такі як зміна статусу, товарів, а також кнопки видалення та збереження товару, переходу на сторінку покупця. (див. рис. 1.1.3.3)

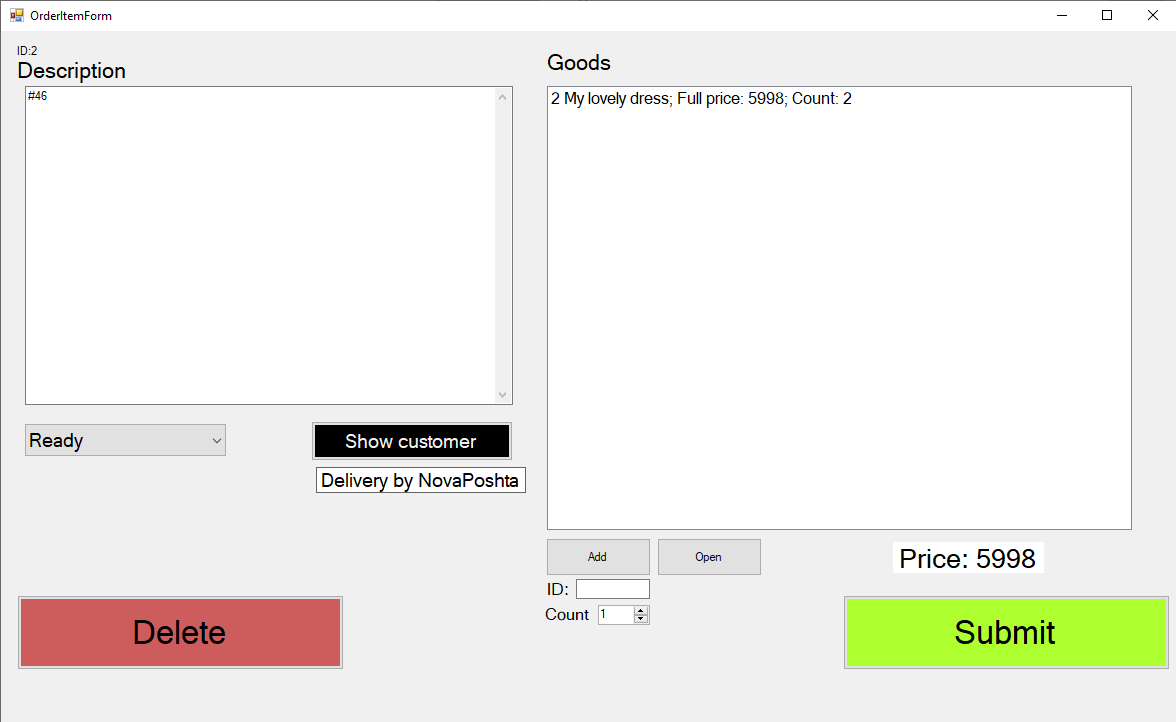


Рисунок 1.1.3.3 – Сторінка заказу

Ця сторінка слугуватиме для відображення детальної інформації про заказ та зміни цієї інформації. Кнопка “Submit” буде виконувати функцію збереження, а кнопка “Delete” – видалення. У випадаючому меню можна вибрати статус заказу. Поле під заголовком “Description” для зміни опису заказу. Лист “Goods” відображає заказані товари та деяку інформацію про них. Кнопки “Add” та “Open” можуть додавати чи відкривати товар відповідно.

1.1.3.4 Операція пошуку

Така операція пошуку використовується й на інших сторінках. Пошук можна робити чи по імені чи по id, для вказання цього слугує дві радіокнопки під строкою пошуку “Search by customer name” та “Search by order ID” відповідно.

1.1.3.5 Операція сортування

Сортування таблиці відбувається шляхом вибору деяких фільтрів що представлені у блоці “Status”. Доступно сортування по статусу: “All”(любий), “Ready”(готовий), “New Order”(новий) та “Processed”(в обробці).

1.1.4 Сторінка “Customers”

1.1.4.1 Основний функціонал

Головна сторінка програми, яка називається “Customers”, має виконувати функцію відображення таблиці покупців. (див. рис. 1.1.4.1).

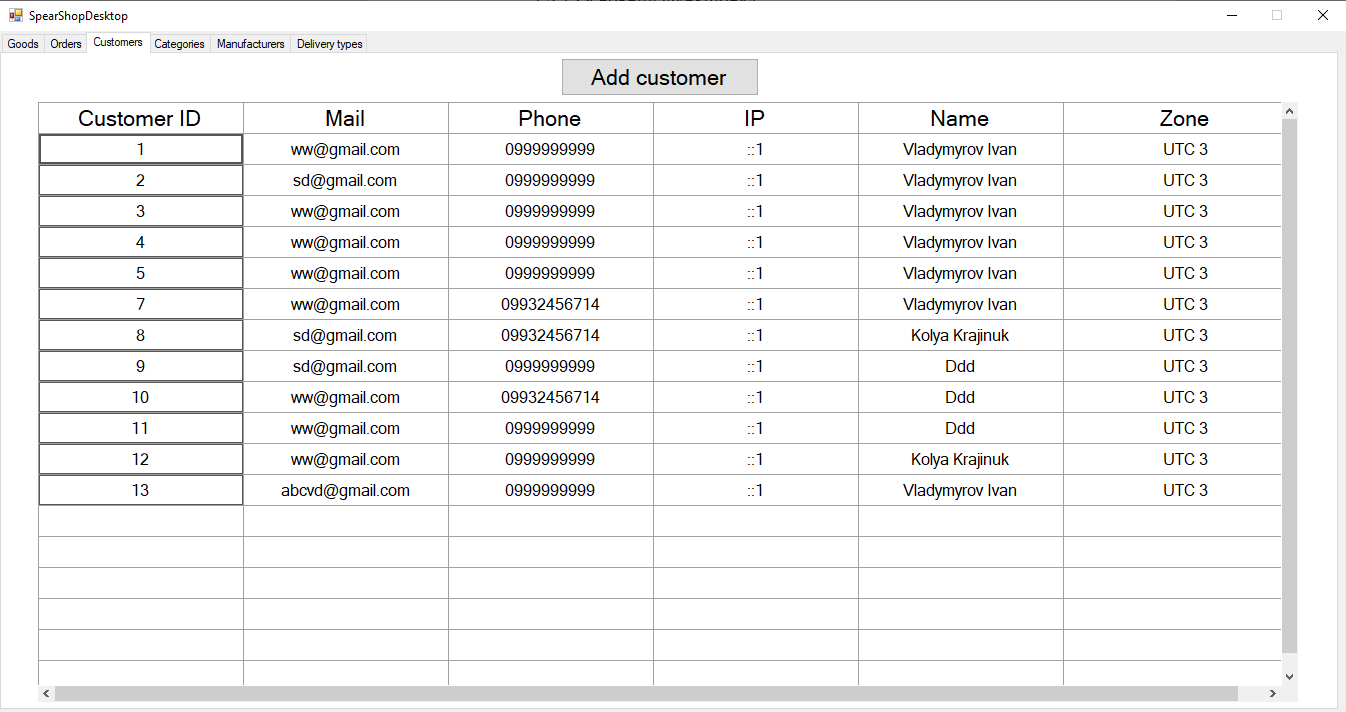


Рисунок 1.1.4.1 – Сторінка “Customers”

Ця сторінка повинна бути наочною та містити функції, які користувач буде використовувати найчастіше: перегляд.

1.1.4.2 Таблиця покупців

Основним джерелом інформації на цій сторінці, має бути таблиця, яка містить покупців та інформацію про них. Рядками цієї таблиці є покупці, а стовпцями – інформація про них.

1.1.4.3 Вікно редагування покупця

Вікно редагування покупця має містити основні функції для роботи з покупцем, такі як встановлення чи зміна ім’я, почти, телефону. (див. рис. 1.1.4.3)

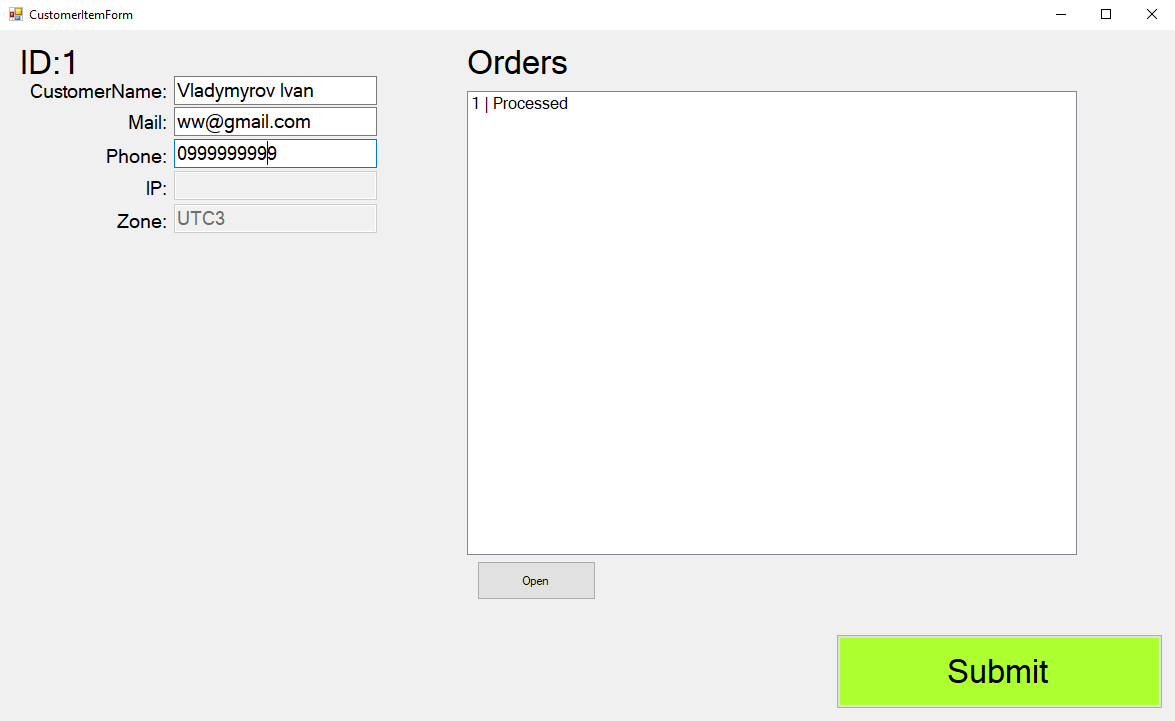


Рисунок 1.1.4.3 – Сторінка покупця

Ця сторінка слугуватиме для відображення детальної інформації про товар та зміни цієї інформації. Кнопка “Submit” буде виконувати функцію збереження. У листі заказів(Orders) можна переглянути закази покупця, натиснувши спочатку на заказ а потім на кнопку “Open”.

1.1.5 Сторінка “Categories”

1.1.5.1 Основний функціонал

Головна сторінка програми, яка називається “Categories”, має виконувати функцію відображення таблиці категорій. (див. рис. 1.1.5.1).

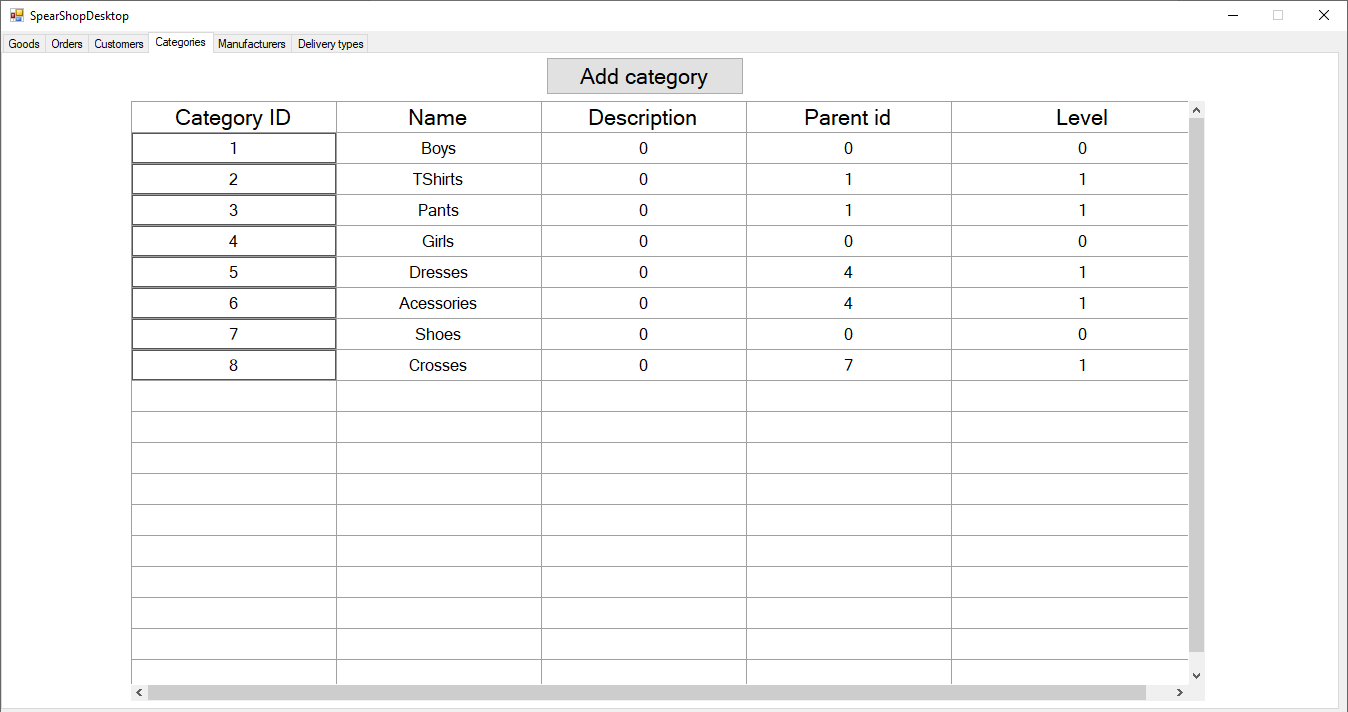


Рисунок 1.1.5.1 – Сторінка “Categories”

Ця сторінка повинна бути наочною та містити функції, які користувач буде використовувати найчастіше: перегляд.

1.1.5.2 Таблиця категорій

Основним джерелом інформації на цій сторінці, має бути таблиця, яка містить категорії та інформацію про них. Рядками цієї таблиці є категорії, а стовпцями – інформація про них.

1.1.5.3 Вікно редагування категорії

Вікно редагування категорії має містити основні функції для роботи з товаром, такі як зміна імені, опису, а також кнопки видалення та збереження, кнопки вибора, зміни, видалення батьківької категорії. (див. рис. 1.1.5.3)

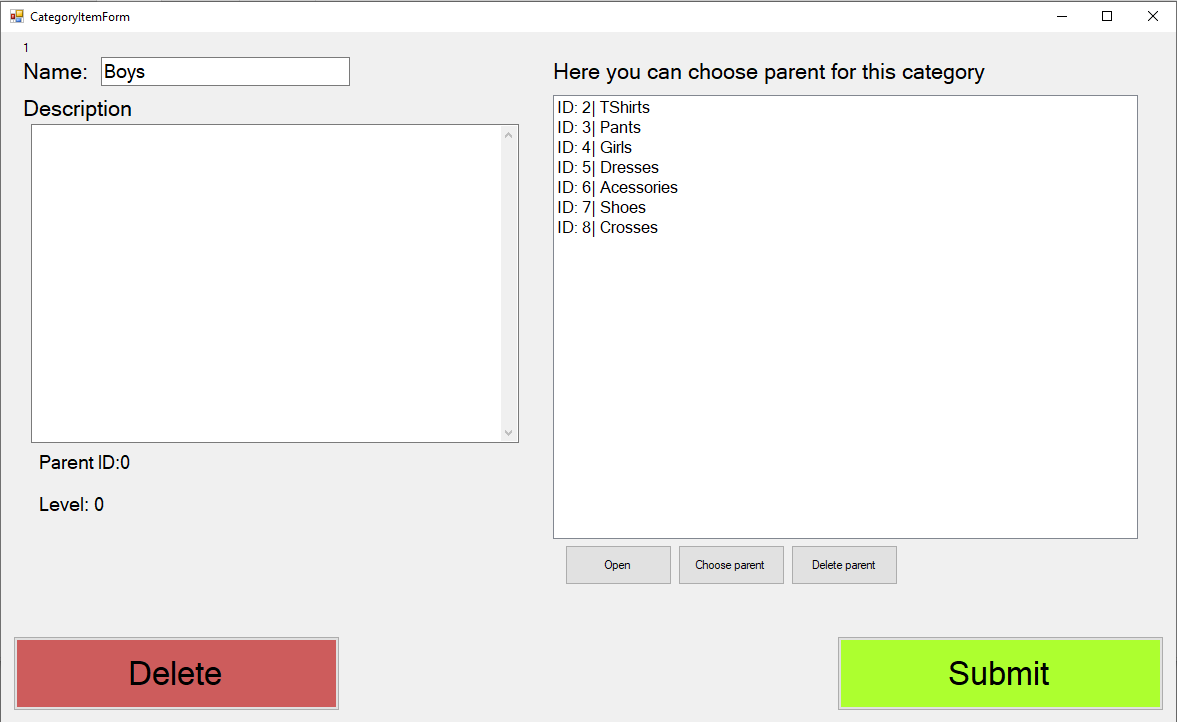


Рисунок 1.1.5.3 – Сторінка заказу

Ця сторінка слугуватиме для відображення детальної інформації про категорію та зміни цієї інформації. Кнопка “Submit” буде виконувати функцію збереження, а кнопка “Delete” – видалення. Поле під заголовком “Description” для зміни опису категорії. Лист відображає усі категорії. Кнопки “Сhoose parent” та “Delete parent” можуть змінювати чи видаляти батьківську категорію. Кнопка “Open” відкриває вибрану категорію.

1.1.6 Сторінка “Manufacturers”

1.1.6.1 Основний функціонал

Головна сторінка програми, яка називається “Manufacturers”, має виконувати функцію відображення таблиці виробників. (див. рис. 1.1.6.1).

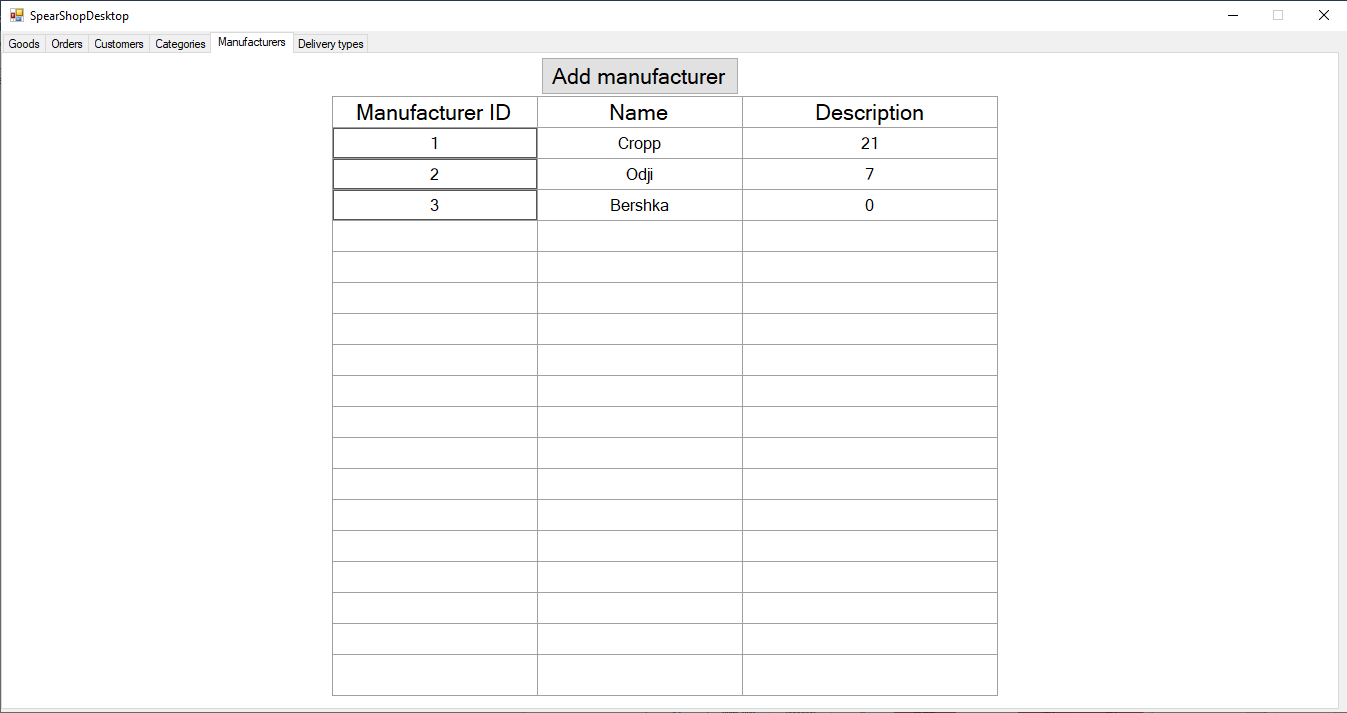


Рисунок 1.1.6.1 – Сторінка “Manufacturers”

Ця сторінка повинна бути наочною та містити функції, які користувач буде використовувати найчастіше: перегляд.

1.1.6.2 Таблиця виробників

Основним джерелом інформації на цій сторінці, має бути таблиця, яка містить виробників та інформацію про них. Рядками цієї таблиці є виробники, а стовпцями – інформація про них.

1.1.6.3 Вікно редагування виробника

Вікно редагування товару має містити основні функції для роботи з товаром, такі як встановлення чи зміна назви та опису, а також кнопки видалення товару та збереження. (див. рис. 1.6.3)

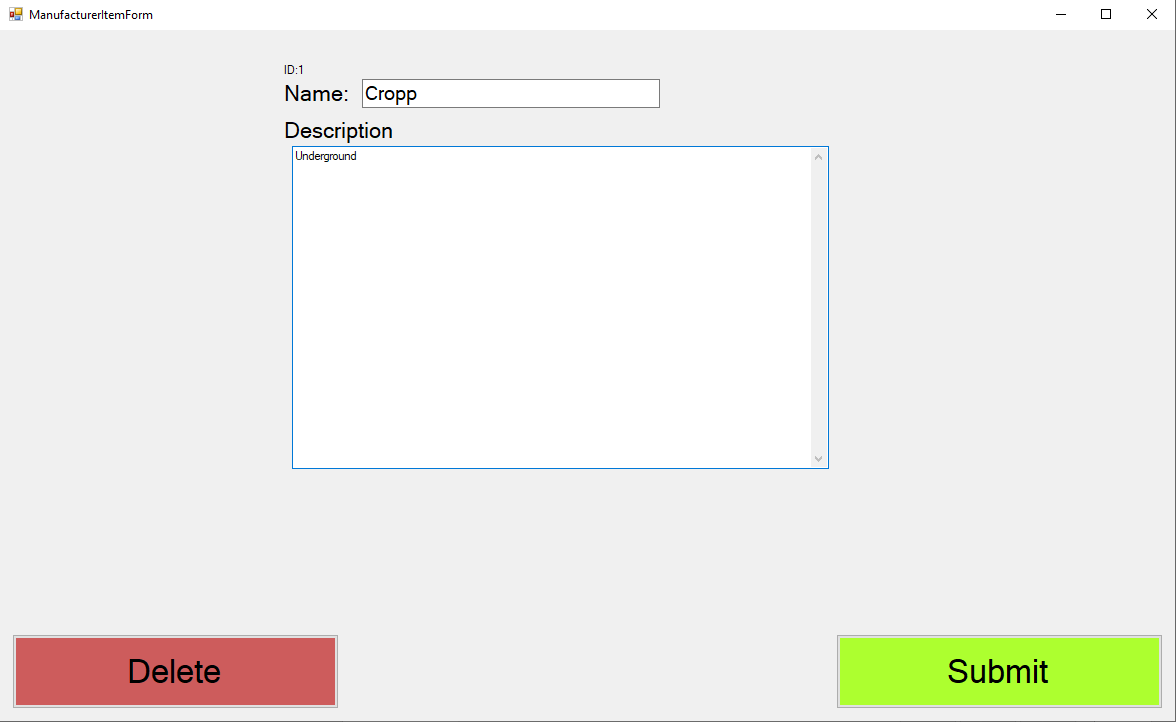


Рисунок 1.1.6.3 – Сторінка виробника

Ця сторінка слугуватиме для відображення детальної інформації про виробника та зміни цієї інформації. Кнопка “Submit” буде виконувати функцію збереження, а кнопка “Delete” – видалення. Поле під заголовком “Description” для зміни опису виробника. Поле під заголовком “Name” призначено для назви.

1.1.7 Сторінка “Delivery types”

1.1.7.1 Основний функціонал

Головна сторінка програми, яка називається “Delivery types”, має виконувати функцію відображення таблиці типів доставки, та зміни цих типів. (див. рис. 1.1.7.1).

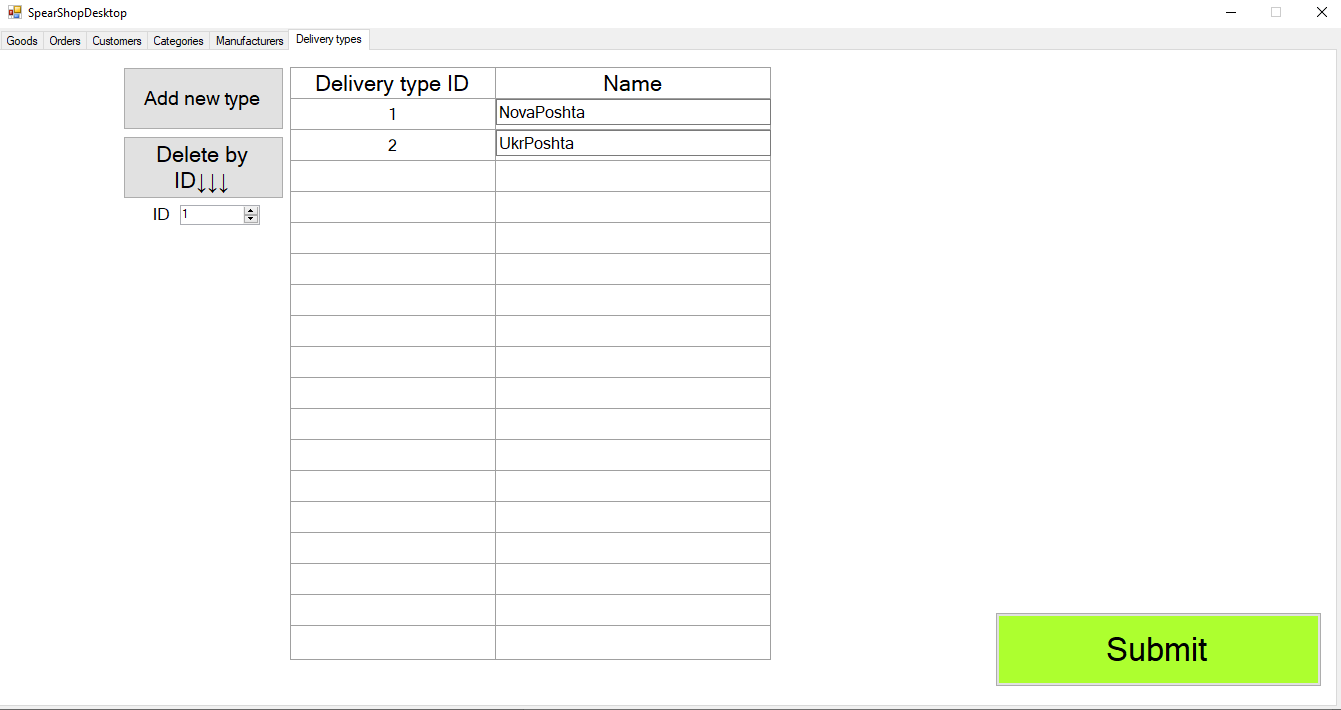


Рисунок 1.1.7.1 – Сторінка “Delivery types”

Ця сторінка повинна бути наочною та містити функції, які користувач буде використовувати найчастіше: перегляд та зміна.

1.1.7.2 Таблиця типів доставки

Основним джерелом інформації на цій сторінці, має бути таблиця, яка містить закази та інформацію про них. Рядками цієї таблиці є закази, а стовпцями – інформація про них. Стовпець “Name” можна змінювати тим самим змінюючи назву способу доставки. Додавання нових типів доставки реалізовано через кнопку “Add new type”. Також є можливість видалення типу: треба написати id того типу що хочете видалити, натиснути кнопку “Delete by ID”. У кінці роботи з цією таблицею потрібно натиснути кнопку “Submit” що буде виконувати функцію збереження

1.2.1 Загальний вигляд головного вікна програми (веб частина)

Увесь основний функціонал програми повинен бути зібраний в одному вікні, наповнення якого повинно змінюватися у вигляді окремих вкладок (див. рис. 1.2.1).

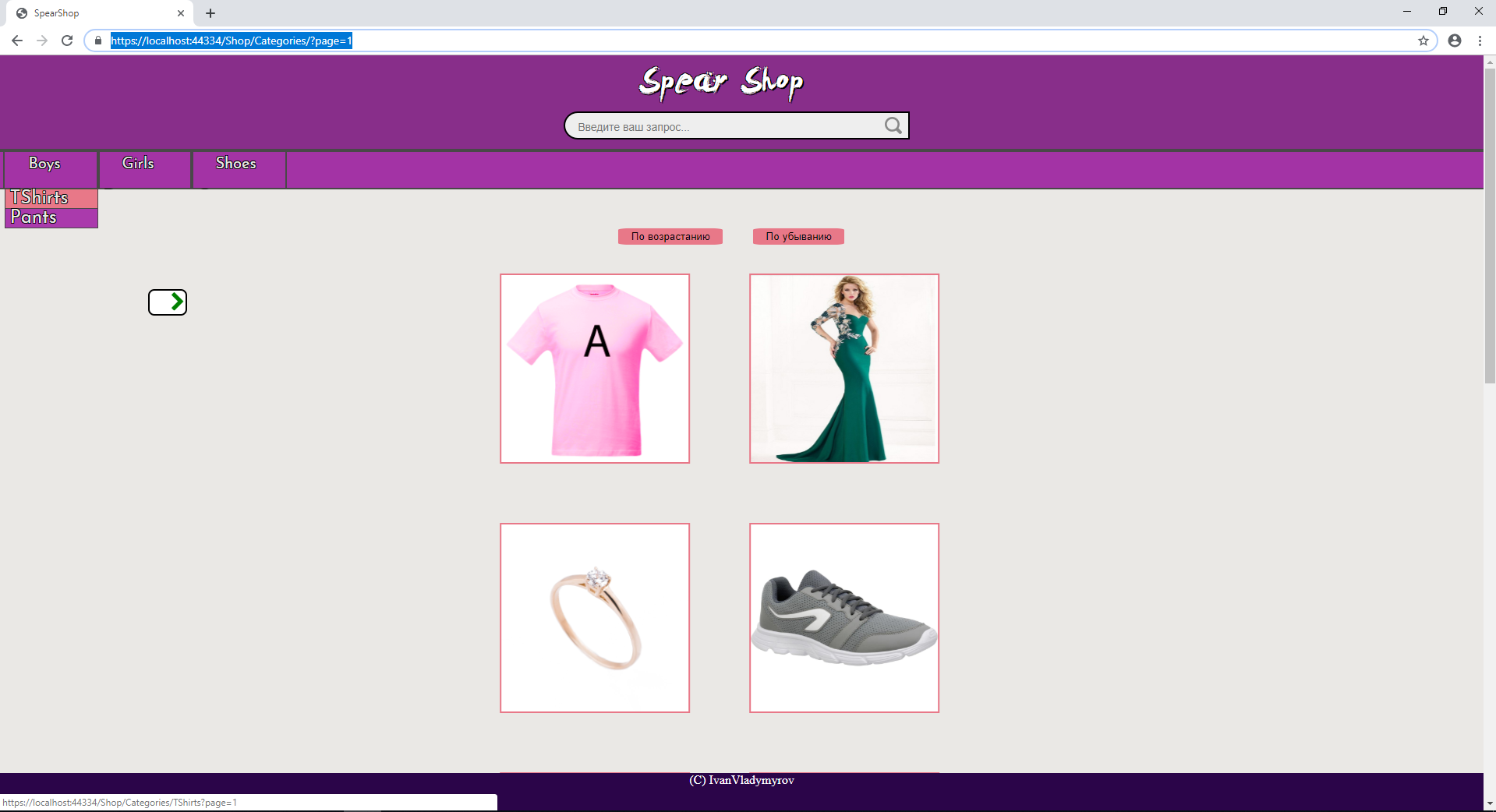


Рисунок 1.2.1 – Головна сторінка магазину

З цієї сторінки можна змінювати відображення товарів(вибираючи певну категорію, чи вибираючи сортування за ціною), робити пошук товара за ім’ям чи id шляхом написання у рядок пошуку слів та натискання на іконку пошуку. Переходити на сторінку товара (навести на зображення, та натиснути на відповідну іконку) (див. рис. 1.2.2).

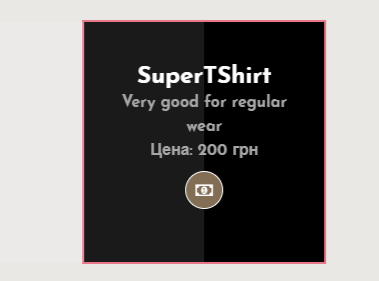


Рисунок 1.2.1 – Задня частина зображення з іконкою для переходу на сторінку товару

Переходити на начальну сторінку сайту( натискаючи на логотип SpearShop), чи переходити на сторінку уперед чи на сторінку назад натискаючи відповідні кнопки, що знаходиться зліва.

1.2.2 Загальний вигляд вікна товару(веб частина)

Сторінка товару існує для того щоб там можна було зробити заказ. Для цього на ній існує форма заповнення інформації про заказ. На цій сторінці також є опис товару, його назва. Також передбачена можливість переходу на сторінку де-якої категорії чи головну сторінку сайту(див. рис. 1.2.2). Також можна переглянути відео натиснувши на відповідну іконку що з’являється при наведенні на зображення товару.

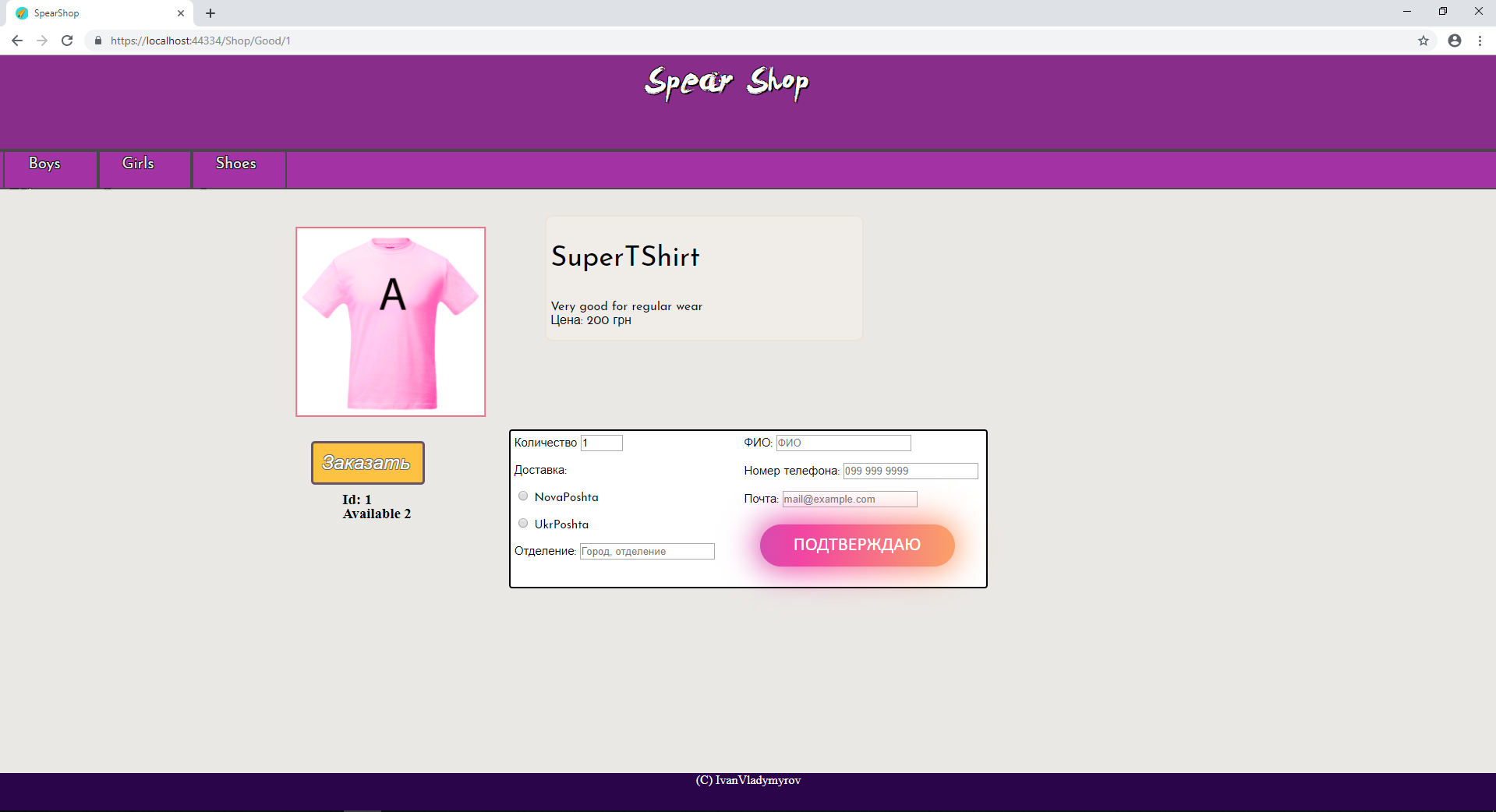


Рисунок 1.2.2 – Сторінка товару

Форма валідується на стороні клієнта за допомогою JS та регулярних виразів. Якщо поля введені некоректно, форма не буде відправлена.

**2 ПРОЕКТНА СПЕЦИФІКАЦІЯ**

2.1 Загальне проектування

Для реалізації першої частини програми “SpearShop” була обрана настільна архітектура з графічним інтерфейсом Windows Forms на платформі .NET Framework 4.8 та мова C#, платформа розробки веб-програм ASP.NET та її додатки були використани для створення другої частини програми.

В основному, для редагування моделі даних використовується MySqlEntityFrameworkCore. Дані у програмі представлені у вигляді моделі з классів, робота з якими, завдяки MySqlEntityFrameworkCore змінює й дані в базі даних. На цих об’єктах будується подальший розвиток програми. На підставі цієї інформації, створюються таблиці, які потім й складають графічний інтерфейс. У веб частині є класи контролерів які забезпечують роботу за HTTP.

2.2 Об'єктна структура програми

2.2.1 Модель програми

Головним елементом програми, навколо якого будуються усі інші компоненти є модель, яка містить наступні класи: Категорія (Category.cs), Покупець (Customer.cs), Тип доставки (Delivery\_Type.cs), Товар (Good.cs), Виробник (Manufacturer.cs), Замовлення (Order.cs), Зображення (Img.cs), Відео (Video.cs) та класи CategoryGood та OrderGood, які є таблицями пошука для зв’язку ManyToMany.

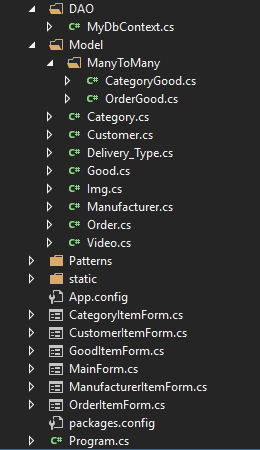


Рисунок 2.2.1 – Класи моделі

Category (Categrory.cs) – Він має наступні публічні властивості: ідентифікатор(id), рівень, опис, ім’я, посилання(об’єкт посилкового типу) на батька(та категорія для якої данна є вкладенною), та лист що характеризує зв’язок ManyToMany у базі з таблицею Goods. Відповідні стовпці має й таблиця categories в базі даних. Ідентифікатор – 32-бітне ціле число зі знаком, яке використовується для збереження і відновлення зв'язків підчас збереження та завантаження із файлу. Воно потрібне для економії пам'яті, бо на один екземпляр категорії може посилатися велика кількість інших об'єктів. Клас має лише стандартний конструктор, бо цей клас оброблюється MySQLEntityFrameworkCore.

Покупець (Customer.cs) – Він має наступні публічні властивості Customer\_Id, Mail, Phone, Ip, Zone, Name, та лист Orders(зв’язок OneToMany), що вказує на закази які зробив покупець. Відповідні стовпці має й таблиця customers в базі даних.

Тип доставки (Delivery\_Type.cs) – Він має наступні публічні властивості Type\_Id, Name, та лист Orders(зв’язок OneToMany), що вказує на закази у яких є такий тип доставки. Відповідні стовпці має й таблиця delivery\_type в базі даних.

Товар (Good.cs) – класс що представляє той елемент, що є головною частиною магазину. Він має наступні властивості: id, ім’я, кількіть доступних, ціна, опис, а також поля Manufacturer(зв’язок OneToMany), CategoryGoods(зв’язок ManyToMany) OrderGoods(зв’язок ManyToMany) Imgs(зв’язок OneToMany), Videos(зв’язок OneToMany), з якими працює MySQLEntityFrameworkCore. Відповідні стовпці має й таблиця goods в базі даних.

Зображення (Img.cs) – клас що має наступні властивості: id, та шлях до файлу зображення, а також ідентифікаційний номер до якого прив’язано це зображення Good\_Id та властивість Good(зв’язкок ManyToOne). Відповідні стовпці має й таблиця imgs в базі даних.

Виробник (Manufacturer.cs) – клас що має наступні властивості: id, опис, шлях до логотипу виготовника, та властивість Goods(зв’язок OneToMany). Відповідні стовпці має й таблиця manufacturers в базі даних.

Замовлення (Order.cs) – клас що має наступні властивості: id, опис, повна ціна, статус, id користувача який зробив це замовлення, id типу доставки, а також властивості які заповнюються автоматично завдяки MySQLEntityFrameworkCore Customer(OneToOne), Delivery\_Type(OneToOne), OrderGoods(ManyToMany). Відповідні стовпці має й таблиця orders в базі даних.

Відео (Video.cs) – клас що має наступні властивості: id, та шлях до файлу відео, а також ідентифікаційний номер до якого прив’язано це відео Good\_Id та властивість Good(зв’язкок ManyToOne). Відповідні стовпці має й таблиця videos в базі даних.

З усіма класами, що наведені вище, працює MySQLEntityFrameworkCore, отже вони є моделями.

Окрім головних, є також і службові типи, які призначені для об'єднання інформації. Клас доступу до бази данних (MyDbContext.cs) успадкований від стадартного класу DbContext має поля сетів(стандартний клас DbSet) відповідні до усіх класів що наведені вище: CategoryGoods, OrderGoods, Goods, Categories, Manufacturers, Imgs, Videos, Customers, Orders та Delivery\_Types. А також методи конфігурції: OnConfiguring(що ініціалізує доступ до бази даних на MySql сервері) та OnModelCreating(що завдає додаткові параметри для ініціалізації моделей, а саме зв’язків ManyToMany що були використані).

2.2.2 Форми програми

Вікнами програми слугують форми Windows Forms (див. рис. 2.2.2). Між вікнами можна переходити натискаючи відповідні кнопки. Переходи між вікнами заначені стрілочками(див. рис. 2.2.2).

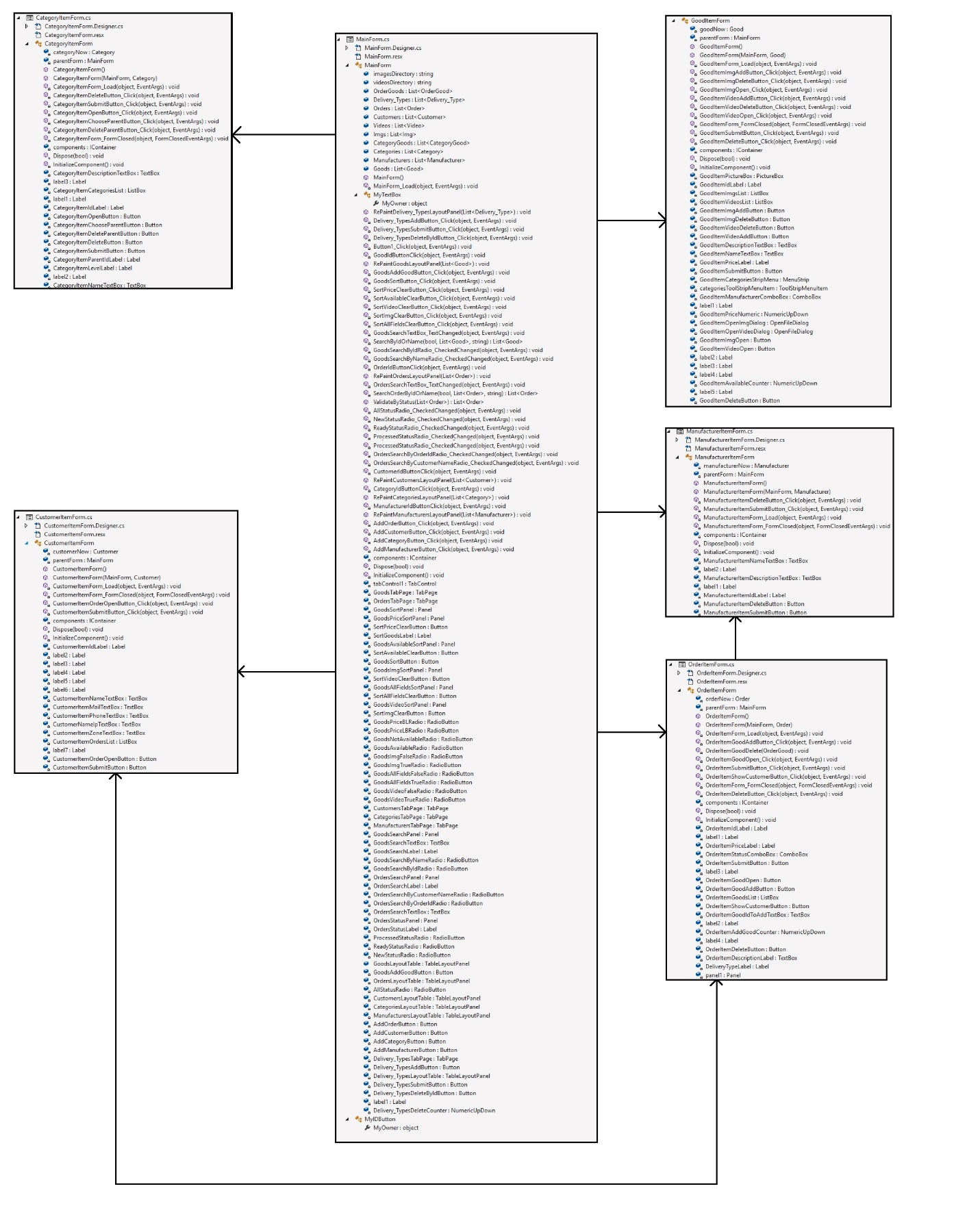


Рисунок 2.2.2 – Діаграма класів форм програми та переходів між ними

Головна форма (MainForm) містить вкладки: Goods, Customers, Orders, Categories, Delivery\_Types, Manufacturers; на якиї відображаються відповідні таблиці з інформацією, та елементи що дозволяють робити пошук чи сортування відповідних елементів. У першому рядку кожної таблиці знаходиться кнопка, натиснувши на яку можна перейти на відповідну форму подробної інформації за елементом, де її можна буде змінити, чи видалити елемент.

Усі інші форми містять подробну інформацію про елемент, що був вибраний у таблиці головної форми. Наповнення цих форм вже описано(див. розділ 1).

2.2.3 Контролери

Для обробки HTTP запросів використовується контроллер (ShopController.cs). Під час роботи в URL храняться параметри: sort(яка була обрана сортировка), searchstr(рядок пошуку), page(номер сторінки). Під час переходу на сторінку товару id товару зберігається через знак “/”.

Основні Get запроси

а) Categories­ – видає сторінку з товарами певної категорії.

б) Good – видає сторінку з певним товаром.

Основні Post запроси

а) Good – змінює базу данних відповідно до заказу

2.2.4 Razor Pages

Для динамічного відтворення HTML сторінок використовуються Razor Pages: Good.cshtml(сторінка товару), Categories.cshtml(сторінка товарів за категоріями). Об’єкти що будуть відтворені на сторінці передаються в ViewBag з яким й працює .cshtml файл

**3 ІНСТРУКЦІЯ КОРИСТУВАЧА**

3.1 Системні вимоги

Для запуску програми “SpearShop” на персональному комп'ютері потрібне виконання таких системних вимог:

а) операційна система Windows 7, Windows 8, Windows 10;

б) центральний процесор частотою 1 ГГц;

в) обсяг оперативної пам'яті – 512 Мб.

г) доступ к мережі Інтернет

д) встановлений MySql сервер з відлагодженим дампом бази даних.

3.2 Встановлення та запуск програми

Програма є портативною, тому її встановлення не потрібне. Запуск програми здійснюється за допомогою відкриття файлу SpearShopDesktop.exe, що знаходиться у головному каталозі програми. Для запуску веб частини потрібно відкривати файл SpearShopWeb.exe.

3.3 Навігація у програмі

Увесь функціонал програми розподілений по вкладкам. Щоб перейти на іншу сторінку, натисніть на потрібну кнопку у верхній частині вікна програми (далі – панель навігації). Активна сторінка виділиться світлішим кольором, а зміст вікна зміниться.

Щоб вийти з програми, закрийте її будь-яким зручним для вас способобом.

3.4 Редагування полів

Щоб додавати інформацію в програму, вам знадобиться вводити її з клавіатури.

Для редагування текстового поля, натисніть на нього та з допомогою клавіатури введіть потрібний текст.

Для редагування числового поля, натисніть на нього та введіть за допомогою клавіатури числове значення. Також ви можете збільшити або зменшити величину поля, натиснувши відповідно на стрілку уверх чи вниз цього поля.

Змінюючи поля у таблицях у SpearShopDesktop будуть відбуватися зміни на сайті, але для цього потрібно перезапустити SpearShopWeb.exe.

3.5 Перегляд інформації що буде використовуватися на сайті

Ви можете переглянути усі таблиці елементів переходячи між відповідими вкладками. Початковою вкладкою є вкладка товарів.

Також можно сортувати відображення елементів таблиці використовуючи характерний блок “Sort” чи “Status” (для вкладки “Orders”). Після вибору параметрів за якими буде проходити сортування потрбно натиснути кнопку “Sort”. Для зняття деяких параметрів сортування потрбно натиснути кнопку “C” у рядку, параметри якого потрібно відключитит. Доступен живий пошук за параметром який ви виберете у блоці “Search”.

3.6 Сторінка елементу

Сторінка елементу дозволяє вам переглядати додаткову інформацію щодо елементу. Можуть бути присутні текстові поля, вводити інформацію у які можна використовуючи клавіатуру, чисельні поля, ввод інформація за допомогою клавітури чи характерних кнопок що позначаються стрілочками уверх чи униз. Якщо на сторінці передбачене додавання зображень чи відео, на сторінці буде характерні блоки “Videos” та “Images”. На є інтуитивно зрозумілі кнопки “Open” (відкрити елемент для передбачення), “Add” (додати елемент), “Delete” (видалити елемент). Якщо передбачається що файл знаходиться на комп’ютері, то відкриється характерне вікно вибору файлу у файловій системі.

**ВИСНОВКИ**

Кінцевим результатом виконання курсової роботи з дисципліни “Об'єктно-орієнтоване програмування” є програма “SpearShop”, яка відповідає усім вимогам справжнього програмного продукту. Була досягнена головна мета: створення простої в користуванні програми, призначеної для широкого кола осіб. Але найголовніше ­– був набутий безцінний досвід самостійної та повної розробки настільної та веб програми із використанням сучасних технологій та загальних принципів об'єктно-орієнтованого програмування, який ще неодноразово знадобиться у майбутньому.

Розроблена програма хоч і є завершеним програмним продуктом, але має декілька цікавих шляхів вдосконалення. Наприклад можна додати систему аутентифікації клієнтів та розробити на сайті сторінку особистого кабінету. Це дало б змогу упорядкувати інформацію щодо клієнтів, та відкрило б нові шляхи вдосконалення такі як створення корзини покупок, системи бонусів для клієнта та інше. Незважаючи на те що програма є закінченою, її потрібно поставити на хостинг, щоб вона виконувала свою головну мету, а саме продаж товарів у мережі Інтернет. Окрім цього, структура сайту дозволяє наповнювати його будь-якими товарами, та якщо додати деякі зручні інструменти для зміни оформленя сайту(назва, колір елементів та т.п.), можна використовувати цей проект як шаблон для готових продуктів, що також би мали цінність.

Але навіть зараз завершена програма є досить корисною: Як результат – програмний продукт відповідає усім сучасним вимогам та готовий пристосовуватися до їх змін.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Документация по .NET Framework. URL: https://docs.microsoft.com/ru-ru/dotnet/framework/ (дата звернення: 14.04.2019).

2. Бондарев В. М. Объектно-ориентированное программирование на С#. – Х.: Компания СМИТ, 2009. – 224 с.

3. Metanit ASP.NET. URL: https://metanit.com/sharp/aspnet\_webapi/ (дата звернення: 08.06.2019).

4. Metanit Entity Framework Core. URL: https://metanit.com/sharp/entityframeworkcore/ (дата звернення 10.04.2019).

5. Metanit Razor Pages. URL: https://metanit.com/sharp/aspnet5/29.1.php (дата звернення 10.06.2019).

ДОДАТОК А

Код програми

1. Код файлу MainForm.cs

using SpearShopDesktop.Model;

using System;

using System.Collections.Generic;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Windows.Forms;

namespace SpearShopDesktop

{

public partial class MainForm : Form

{

public string imagesDirectory;

public string videosDirectory;

public List<OrderGood> OrderGoods;

public List<Delivery\_Type> Delivery\_Types;

public List<Order> Orders;

public List<Customer> Customers;

public List<Video> Videos;

public List<Img> Imgs;

public List<CategoryGood> CategoryGoods;

public List<Category> Categories;

public List<Manufacturer> Manufacturers;

public List<Good> Goods;

public MainForm()

{

InitializeComponent();

}

private void MainForm\_Load(object sender, EventArgs e)

{

try

{

System.IO.Directory.SetCurrentDirectory(System.IO.Directory.GetParent(System.IO.Directory.GetCurrentDirectory()).FullName);

System.IO.Directory.SetCurrentDirectory(System.IO.Directory.GetParent(System.IO.Directory.GetCurrentDirectory()).FullName);

videosDirectory = Path.GetFullPath(Path.Combine(System.IO.Directory.GetCurrentDirectory(),@"..\SpearShopWeb\wwwroot\videos"));

imagesDirectory = Path.GetFullPath(Path.Combine(System.IO.Directory.GetCurrentDirectory(), @"..\SpearShopWeb\wwwroot\images"));

}

catch (Exception ex) { MessageBox.Show("Img or videos founding problem: "+ex.Message); }

try

{

using (MyDbContext db = new MyDbContext())

{

Delivery\_Types = new List<Delivery\_Type>(db.Delivery\_Types.ToList());

Orders = new List<Order>(db.Orders.ToList());

Customers = new List<Customer> (db.Customers.ToList());

Videos = new List<Video>(db.Videos.ToList());

Imgs = new List<Img>(db.Imgs.ToList());

CategoryGoods = new List<CategoryGood>( db.CategoryGoods.ToList());

Categories = new List<Category>(db.Categories.ToList());

Manufacturers = new List<Manufacturer>(db.Manufacturers.ToList());

OrderGoods = new List<OrderGood>(db.OrderGoods.ToList());

Goods = new List<Good>(db.Goods.ToList());

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

Application.Exit();

}

RePaintOrdersLayoutPanel(Orders);

RePaintGoodsLayoutPanel(Goods);

RePaintCustomersLayoutPanel(Customers);

RePaintCategoriesLayoutPanel(Categories);

RePaintManufacturersLayoutPanel(Manufacturers);

RePaintDelivery\_TypesLayoutPanel(Delivery\_Types);

}

public class MyTextBox : TextBox

{

public object MyOwner { get; set; }

}

public void RePaintDelivery\_TypesLayoutPanel(List<Delivery\_Type> dtps)

{

Delivery\_TypesLayoutTable.Visible = false;

Delivery\_TypesLayoutTable.Controls.Clear();

Label goodsColumn1 = new Label();

goodsColumn1.Margin = new Padding(0, 0, 0, 0);

goodsColumn1.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn1.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn1.Font = new Font("Microsoft Sans Serif", 16);

goodsColumn1.Text = "Delivery type ID";

Label goodsColumn2 = new Label();

goodsColumn2.Margin = new Padding(0, 0, 0, 0);

goodsColumn2.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn2.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn2.Font = new Font("Microsoft Sans Serif", 16);

goodsColumn2.Text = "Name";

Delivery\_TypesLayoutTable.Controls.AddRange(new Control[] { goodsColumn1, goodsColumn2});

foreach(Delivery\_Type dt in Delivery\_Types)

{

Label column2Val = new Label();

column2Val.Margin = new Padding(0, 0, 0, 0);

column2Val.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

column2Val.TextAlign = ContentAlignment.MiddleCenter;

column2Val.Font = new Font("Microsoft Sans Serif", 12);

column2Val.Text = Convert.ToString(dt.Type\_Id);

MyTextBox column3Val = new MyTextBox();

column3Val.MyOwner = dt;

column3Val.Margin = new Padding(0, 0, 0, 0);

column3Val.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

column3Val.Font = new Font("Microsoft Sans Serif", 12);

column3Val.Text = Convert.ToString(dt.Name);

Delivery\_TypesLayoutTable.Controls.AddRange(new Control[] {column2Val, column3Val });

}

Delivery\_TypesLayoutTable.Visible = true;

}

private void Delivery\_TypesAddButton\_Click(object sender, EventArgs e)

{

Delivery\_Type newDt = new Delivery\_Type();

Label column2Val = new Label();

column2Val.Margin = new Padding(0, 0, 0, 0);

column2Val.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

column2Val.TextAlign = ContentAlignment.MiddleCenter;

column2Val.Font = new Font("Microsoft Sans Serif", 12);

column2Val.Text = "AUTO";

MyTextBox column3Val = new MyTextBox();

column3Val.MyOwner = newDt;

column3Val.Margin = new Padding(0, 0, 0, 0);

column3Val.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

column3Val.Font = new Font("Microsoft Sans Serif", 12);

column3Val.Text = "newType";

Delivery\_TypesLayoutTable.Controls.AddRange(new Control[] { column2Val, column3Val });

}

private void Delivery\_TypesSubmitButton\_Click(object sender, EventArgs e)

{

using (MyDbContext db = new MyDbContext())

{

foreach (object myTb in Delivery\_TypesLayoutTable.Controls)

{

if(myTb.GetType()== typeof(MyTextBox))

{

Delivery\_Type t = db.Delivery\_Types.FirstOrDefault(s => s.Type\_Id == ((Delivery\_Type)(((MyTextBox)myTb).MyOwner)).Type\_Id);

if (t != null)

{

t.Name = ((Delivery\_Type)(((MyTextBox)myTb).MyOwner)).Name;

db.SaveChanges();

}

else

{

if (((MyTextBox)myTb).Text != null && ((MyTextBox)myTb).Text.Length > 0)

{

Delivery\_Type nDt = new Delivery\_Type { Name = ((MyTextBox)myTb).Text };

db.Delivery\_Types.Add(nDt);

db.SaveChanges();

}

}

}

else { }

}

}

using (MyDbContext db = new MyDbContext())

{

OrderGoods = db.OrderGoods.ToList();

Delivery\_Types = db.Delivery\_Types.ToList();

Orders = db.Orders.ToList();

Customers = db.Customers.ToList();

Videos = db.Videos.ToList();

Imgs = db.Imgs.ToList();

CategoryGoods = db.CategoryGoods.ToList();

Categories = db.Categories.ToList();

Manufacturers = db.Manufacturers.ToList();

Goods = db.Goods.ToList();

RePaintDelivery\_TypesLayoutPanel(Delivery\_Types);

RePaintOrdersLayoutPanel(Orders);

RePaintGoodsLayoutPanel(Goods);

RePaintCustomersLayoutPanel(Customers);

RePaintCategoriesLayoutPanel(Categories);

RePaintManufacturersLayoutPanel(Manufacturers);

}

}

private void Delivery\_TypesDeleteByIdButton\_Click(object sender, EventArgs e)

{

using (MyDbContext db = new MyDbContext())

{

int idToDel = (int)Delivery\_TypesDeleteCounter.Value;

if (db.Delivery\_Types.FirstOrDefault(s => s.Type\_Id == idToDel) != null)

{

db.Delivery\_Types.Remove(db.Delivery\_Types.FirstOrDefault(s => s.Type\_Id == idToDel));

db.SaveChanges();

OrderGoods = db.OrderGoods.ToList();

Delivery\_Types = db.Delivery\_Types.ToList();

Orders = db.Orders.ToList();

Customers = db.Customers.ToList();

Videos = db.Videos.ToList();

Imgs = db.Imgs.ToList();

CategoryGoods = db.CategoryGoods.ToList();

Categories = db.Categories.ToList();

Manufacturers = db.Manufacturers.ToList();

Goods = db.Goods.ToList();

RePaintDelivery\_TypesLayoutPanel(Delivery\_Types);

RePaintOrdersLayoutPanel(Orders);

RePaintGoodsLayoutPanel(Goods);

RePaintCustomersLayoutPanel(Customers);

RePaintCategoriesLayoutPanel(Categories);

RePaintManufacturersLayoutPanel(Manufacturers);

}

else

{

}

}

}

private void Button1\_Click(object sender, EventArgs e)

{

using (MyDbContext db = new MyDbContext())

{

db.Database.EnsureCreated();

Console.WriteLine("Объекты успешно сохранены");

//Sorted

//goods=goods.OrderBy(g => g.Price).ToList();

//goods=goods.OrderByDescending(g => g.Price).ToList();

//Adding

//Good god1 = new Good { Name = "TomasMraz11", Available = 63, Price = 2008, Manufacturer = users[0], CategoryGoods = new List<CategoryGood>() };

//db.Goods.Add(god1);

//db.SaveChanges();

//god1.CategoryGoods.Add(new CategoryGood { Category\_Id = categories[0].Category\_Id, Good\_Id = god1.Good\_Id });

//db.SaveChanges();

//Deleting

//Good student = db.Goods.FirstOrDefault();

//db.Goods.Remove(student);

}

}

////

///

//

/\*GOOD PART\*/

//

//

//

private void GoodIdButtonClick(object sender, EventArgs e)

{

try

{

Good goodNow = ((Good)((MyIDButton)sender).MyOwner);

GoodItemForm itemForm = new GoodItemForm(this, goodNow);

itemForm.Show();

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

public void RePaintGoodsLayoutPanel(List<Good> goodsToPaint)

{

GoodsLayoutTable.Visible = false;

GoodsLayoutTable.Controls.Clear();

Label goodsColumn1 = new Label();

goodsColumn1.Margin = new Padding(0, 0, 0, 0);

goodsColumn1.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn1.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn1.Font = new Font("Microsoft Sans Serif", 16);

goodsColumn1.Text = "Good ID";

Label goodsColumn2 = new Label();

goodsColumn2.Margin = new Padding(0, 0, 0, 0);

goodsColumn2.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn2.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn2.Font = new Font("Microsoft Sans Serif", 16);

goodsColumn2.Text = "Name";

Label goodsColumn3 = new Label();

goodsColumn3.Margin = new Padding(0, 0, 0, 0);

goodsColumn3.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn3.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn3.Font = new Font("Microsoft Sans Serif", 16);

goodsColumn3.Text = "Price";

Label goodsColumn4 = new Label();

goodsColumn4.Margin = new Padding(0, 0, 0, 0);

goodsColumn4.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn4.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn4.Font = new Font("Microsoft Sans Serif", 16);

goodsColumn4.Text = "Available";

Label goodsColumn5 = new Label();

goodsColumn5.Margin = new Padding(0, 0, 0, 0);

goodsColumn5.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn5.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn5.Font = new Font("Microsoft Sans Serif", 16);

goodsColumn5.Text = "Description";

Label goodsColumn6 = new Label();

goodsColumn6.Margin = new Padding(0, 0, 0, 0);

goodsColumn6.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn6.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn6.Font = new Font("Microsoft Sans Serif", 14);

goodsColumn6.Text = "Manufacturer";

Label goodsColumn7 = new Label();

goodsColumn7.Margin = new Padding(0, 0, 0, 0);

goodsColumn7.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn7.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn7.Font = new Font("Microsoft Sans Serif", 16);

goodsColumn7.Text = "Images";

Label goodsColumn8 = new Label();

goodsColumn8.Margin = new Padding(0, 0, 0, 0);

goodsColumn8.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

goodsColumn8.TextAlign = ContentAlignment.MiddleCenter;

goodsColumn8.Font = new Font("Microsoft Sans Serif", 16);

goodsColumn8.Text = "Videos";

GoodsLayoutTable.Controls.AddRange(new Control[] { goodsColumn1, goodsColumn2, goodsColumn3, goodsColumn4, goodsColumn5, goodsColumn6, goodsColumn7, goodsColumn8 });

GoodsLayoutTable.Visible = false;

foreach (Good good in goodsToPaint)

{

MyIDButton idButton = new MyIDButton();

idButton.MyOwner = good;

idButton.Cursor = Cursors.Hand;

idButton.Text = Convert.ToString(good.Good\_Id);

idButton.Margin = new Padding(0, 0, 0, 0);

idButton.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right | AnchorStyles.Left | AnchorStyles.Top);

idButton.Font = new Font("Microsoft Sans Serif", 12);

idButton.FlatStyle = FlatStyle.Popup;

idButton.Height = 30;

idButton.Click += GoodIdButtonClick;

Label column2Val = new Label();

column2Val.Margin = new Padding(0, 0, 0, 0);

column2Val.Anchor = (AnchorStyles.Bottom | AnchorStyles.Right |

Подальший код наведено на диску