Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліні “ Об’єктно -орієнтоване програмування”

БІРЖА ТОВАРІВ

Керівник , професор Бондарєв В. М.

Студент гр. ПЗПІ-19-4 Уткін Ю.Є.

Комісія:

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

Проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарєв В. М.

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Побіженко І. О.

Харків 2020

**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-19 -4* . Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проект студента***

*Уткін Юрій Євгенович*

(Прізвище, Ім'я, По батькові)

1 Тема проекту: Біржа товарів

2 Термін здачі студентом закінченого проекту: ***“31” - травня - 2020 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 структури проекту, Діаграма класів, алгоритми, приклади екранних форм \_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми | 21-02-2020 р. |
| 2 | Формулювання вимог до програми | 21-04-2020 – 31-04-2020 р. |
| 3 | Розробка бібліотеки класів. | 01-05-2020 – 09-05-2020 р. |
| 4 | Розробка підсистеми реєстрації та авторизації користувачів. | 09-05-2020 – 14-05-2020 р. |
| 5 | Розробка функцій додавання та просмотра товарів | 14-05-2020 – 17-05-2020 р. |
| 6 | Розробка функцій оформлення та просмотра заказів | 17-05-2020 – 23-05-2020 р. |
| 7 | Тестування і доопрацювання розробленої програмної системи. | 23-05-2020 – 26-05-2020 р. |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 26-05-2020 – 31-05-2020 р. |
| 9 | Захист | 01-06-2018 – 19-06-2020 р. |

Студент Уткін Юрій Євгенович

Керівник Бондарєв Володимир Михайлович

« 31 »\_травня\_\_\_\_\_\_\_\_\_\_ 2020 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 29 с., 13 рис., 6 додатків, 5 джерел.

ВІДОМІСТЬ, КЛАС, МОВА ПРОГРАМУВАННЯ C#, НАКЛАДНА, ОБ’ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПРОГРАМА, ПЛАТФОРМА .NET, РЕГІСТР, СКЛАД.

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

Методи розробки базуються на використанні середи розробки Microsoft Visual Studio 2019, Windows Forms, платформи .NET Framework 4.7.2, мови програмування C#.

В результаті отримана програма під назвою “Біржа товарів”, яка дозволяє зберігати список товарів, характеристики кожного товару, такі як: назва, спеціалізація, адреса, телефон, час роботи, вид власності. Є можливість додавати, видаляти та редагувати товари. Також, є функція робити та переглядати замовлення.

**ЗМІСТ**

ВСТУП 5

1 СПЕЦИФІКАЦІЯ ПРОГРАМИ 6

1.1 Користувачі програми 6

1.2 Функції програми 6

1.2.1 Спільні функції 6

1.2.2 Функції покупця 9

1.2.3 Функції продавця 12

2 ПРОЕКТНА СПЕЦИФІКАЦІЯ 16

2.1 Архітектура 16

2.2 Структура проекту 16

2.3 Об’єктна модель 17

2.3.1 Опис 17

2.3.2 Методи 17

2.4 Зберігання та доступ до даних 18

3 ІНСТРУКЦІЯ КОРИСТУВАЧА 19

3.1 Встановлення, запуск та головне вікно 19

3.2 Реєстрація та вхід 19

3.3 Перегляд товарів та оформлення заказу 19

3.4 Додавання та редагування товарів 19

3.5 Закінчення роботи та збереження даних 20

ВИСНОВКИ 21

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 22

ДОДАТОК А Клас User 23

ДОДАТОК Б Клас Product 24

ДОДАТОК В Методи класу TicketPrice 25

ДОДАТОК Г Клас Password 26

ДОДАТОК Д Клас Dao 27

ДОДАТОК Е Посилання на репозиторій Github 28

ВСТУП

Сьогодні купівля та продаж товарів через комп’ютер вже не є чимось надзвичайним. Можливість замовити собі все що завгодно коли і куди ви забажаєте є дуже зручною. Позбавитися старих речей або наладити свій маленький бізнес – також є дуже важними аспектам життя, однак зробити це без спеціалізованої платформи не завжди легко.

Саме ці проблеми і вирішує розроблена програма. Користувач зможе виставити на продаж все, що захоче, та придбати все, що є в каталозі. Повний доступ до її функцій отримує авторизований користувач. Доступ до облікового запису диспетчера здійснюється через пароль.

Окрім розробки вище описаної програмної системи метою даної курсової роботи є освоїти мову програмування С#, навчитися об’єктно-орієнтованому програмуванню, розвити навички роботи в Windows Forms.

Задачі роботи:

* Розробка системи зчитування та запису даних;
* Створення логіки роботи системи;
* Робота з користувачем;
* Розробка логіки додавання, пошуку та видалення інформації;
* Створення стійкості програми.

1. СПЕЦИФІКАЦІЯ ПРОГРАМИ
   1. Користувачі програми

Проект має дві категорії користувачів, яким необхідні різні функції програми - це продавці та покупці.

Покупець шукає товари, робить замовлення, переглядає історію своїх покупок.

Продавець додає товари, прибирає продані, приймає замовлення, переглядає історію своїх продажів.

Належність користувача до покупців або продавців визначається в момент оформлення замовлення.

Кожен користувач може виступати як в ролі продавця, так і в ролі покупця в залежності від ситуації.

Питання безпеки, такі як хешування паролів і безпеку даних в програмі не вирішуються.

* 1. Функції програми

Програма надає покупцеві такі можливості.

1. Пошук товару
2. Оформлення замовлення

Програма надає продавцеві наступні можливості.

1. Додавання товару
2. Редагування товару
3. Видалення товару
4. Оновлення інформації про замовлення

Спільні функції

1. Реєстрація
2. Вхід в програму
3. Оновлення.
4. Збереження.
5. Вихід з програми
   * 1. Спільні функції

Функція «Реєстрація»

Користувач реєструє новий обліковий запис та входить в програму.

Імена і паролі користувачів зберігаються в даних програми.

**Основний сценарій**

1. Користувач вводить ім'я, пароль та контактні дані (телефон та електронну пошту) і натискає кнопку "Register" (рис. 1.1).
2. Введені дані перевіряються, і відкриється головне вікно (на вкладці каталог) (рис. 1.4).
3. Якщо дані не пройшли перевірку, на формі введення з'являється повідомлення про це і можна спробувати зареєструватися знову. Число спроб не обмежено.

**Альтернативний сценарій**

1. Якщо користувач вже має обліковий запис, він натискає кнопку "I already have an account".
2. Відкривається панель входу. (рис. 1.2)

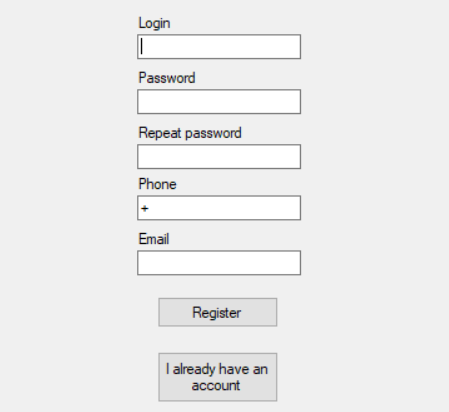


Рисунок 1.1 – Панель реєстрації

Функція «Вхід в програму»

Користувач входить в програму за допомогою імені та пароля.

Імена і паролі користувачів зберігаються в даних програми.

**Основний сценарій**

1. Користувач вводить ім'я та пароль і натискає кнопку "Sign in" (рис. 1.2).
2. Введені дані перевіряються, і відкриється головне вікно (на вкладці каталог) (рис. 1.4).
3. Якщо дані не пройшли перевірку, на формі введення з'являється повідомлення про це і можна спробувати увійти знову. Число спроб не обмежено.

**Альтернативний сценарій**

1. Якщо користувач не має облікового запису, він натискає кнопку "I do not have an account".
2. Відкривається панель реєстрації. (рис. 1.1)

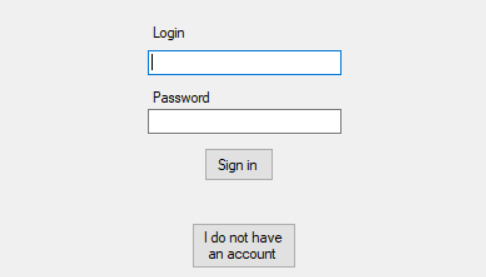


Рисунок 1.2 – Панель входу

Функція «Оновити»

Користувач завантажуэ дані з постійної пам'яті в оперативну.

**Основний сценарій**

1. Користувач натискає кнопку "Refresh" в меню.
2. Дані в оперативній пам’яті замінюються даними з постійної пам’яті.

**Альтернативний сценарій**

1. Якщо користувач змінював дані в оперативній пам’яті, він побачить вікно, яке сповіщаэ про те, що зміни будуть втрачені. Користувачу пропонується зберегти їх. (рис 1.3)
2. Якщо користувач натискає кнопку "Так", зміни будуть збережені.
3. Якщо користувач натискає кнопку "Ні", всі зміни будуть втрачені і виконається основний сценарій.
4. Якщо користувач натискає кнопку "Відміна", операція буде відмінена.

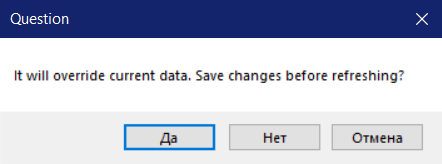


Рисунок 1.3 – Вікно підтвердження

Функція «Зберегти»

Користувач завантажуэ дані з оперативної пам'яті в постійну.

**Основний сценарій**

1. Користувач натискає кнопку "Save" в меню.
2. Дані в постійній пам’яті замінюються даними з оперативної пам’яті.

Функція «Вихід»

Користувач виходить з програми.

**Основний сценарій**

1. Користувач натискає кнопку "Exit" в меню.
2. Програма закривається.

**Альтернативний сценарій**

1. Якщо користувач змінював дані в оперативній пам’яті, він побачить вікно, яке сповіщаэ про те, що зміни будуть втрачені. Користувачу пропонується зберегти їх. (рис 1.3)
2. Якщо користувач натискає кнопку "Так", зміни будуть збережені і програма закриється.
3. Якщо користувач натискає кнопку "Ні", всі зміни будуть втрачені і програма закриється.
4. Якщо користувач натискає кнопку "Відміна", операція буде відмінена.
   * 1. Функції покупця

### Функція «Пошук товару»

На головному вікні у вкладці «Каталог» користувач бачить список товарів, який він може листати за допомогою скролера. (рис 1.4) Натискаючи на один з товарів, або вибираючи пункт меню "Show product", користувач побачить вікно з додатковою інформацією про товар. (рис 1.5)

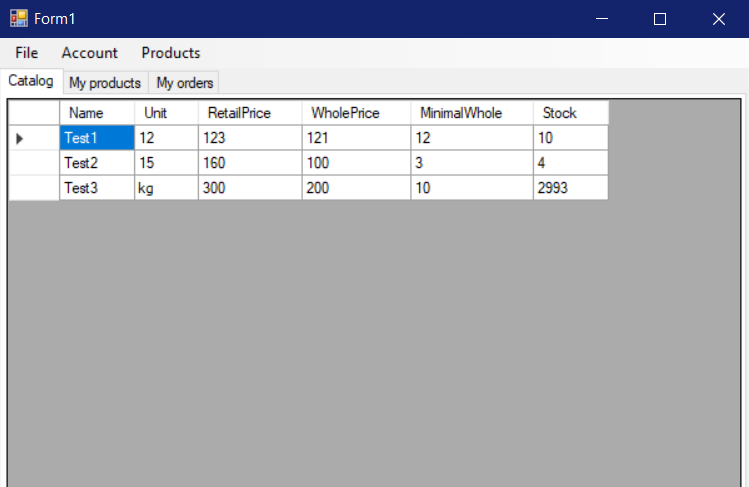


Рисунок 1.4 – Головне вікно, вкладка «Каталог»

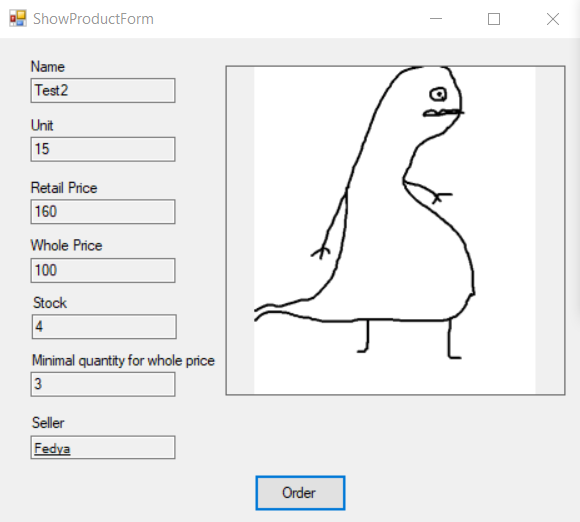


Рисунок 1.5 – Вікно з докладним описом товару

### Функція «Оформлення замовлення»

При натисканні на товар або у пункт меню «Show current» відкривається вікно з його докладним описом. У ньому користувач може оформити замовлення - вибрати кількість товару та спосіб оплати.

**Основний сценарій**

1. Користувач натискає на продукт, що його цікавить або у пункт меню «Show current».
2. Відкривається вікно з докладним описом товару. (рис 1.5).
3. Користувач натискає кнопку «Order».
4. Відкривається вікно оформлення замовлення (рис 1.7).
5. Користувач вводить кількість товару на замовлення і спосіб оплати та натискає кнопку "Order".
6. Введені дані перевіряються, і відкривається вікно товару. (рис. 1.5).
7. Якщо дані пройшли перевірку, створюється замовлення, яке можна побачити у вкладці «Мої замовлення» (рис. 1.6).

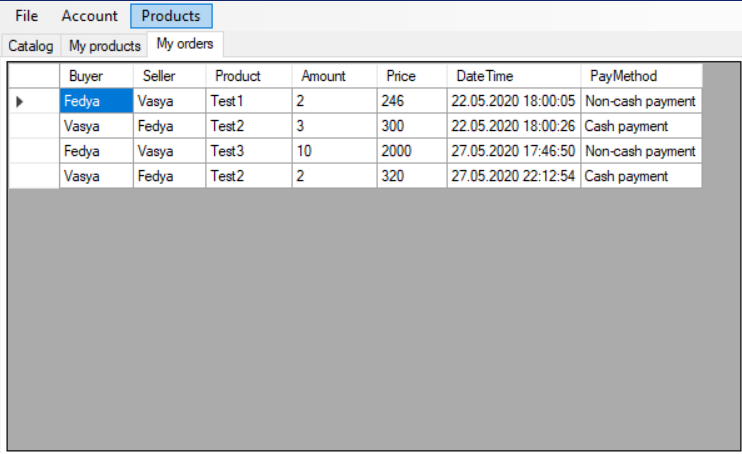


Рисунок 1.6 – Головне вікно, вкладка «Мої замовлення»

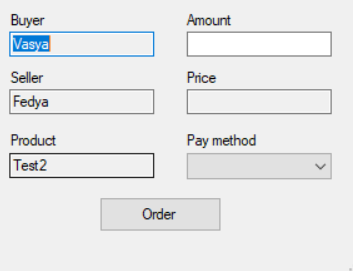


Рисунок 1.7 – Вікно оформлення замовлення

* + 1. Функції продавця

Функція «Додавання товару»

Додає товар до каталога, після чого інші користувачі зможуть його замовити

**Основний Сценарій**

1. Користувач натискає на пункт меню «Add product».
2. Відкривається форма додавання товару. Поля форми порожні. (рис. 1.8)
3. Користувач заповнює поля форми, вибирає файл зображення.
4. Користувач натискає кнопку «Create».
5. Введені дані перевіряються, і відкривається головне вікно.
6. Якщо дані пройшли перевірку, створюється товар, який можна побачити у вкладці «Каталог» та у вкладці «Мої товари» (рис. 1.9).



Рисунок 1.8 –Вікно додавання товару

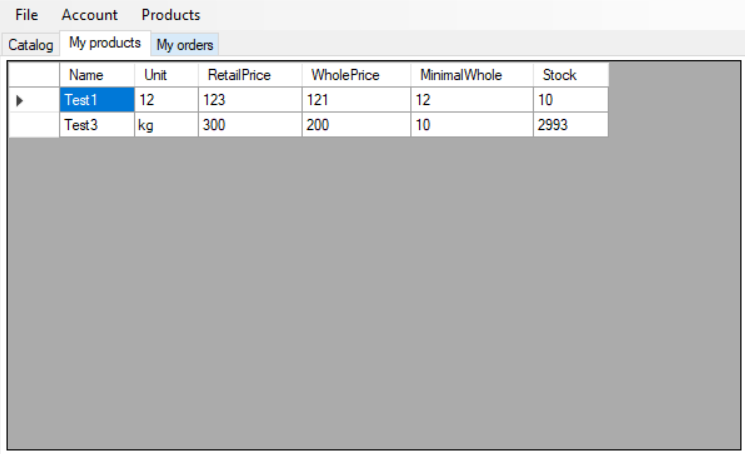


Рисунок 1.9 – Головне вікно, вкладка «Мої товари»

Функція «Редагування товару»

Продавець має можливість у будь-який момент змінити інформацію про свій товар.

**Основний Сценарій**

1. Продавець обирає товар для редагування з вкладки «Каталог» або «Мої товари».
2. Відкривається вікно з докладним описом товару.(рис. 1.10).
3. Продавець натискає кнопку «Edit».
4. Продавець змінює поля форми, змінює файл зображення.
5. Продавець натискає кнопку «Save».



Рисунок 1.10 – Вікно з його докладним описом товару

Функція «Видалення товару»

Продавець має можливість у будь-який момент видалити свій товар з каталогу.

**Основний Сценарій**

1. Продавець обирає товар для видалення з вкладки «Каталог» або «Мої товари».
2. Відкривається вікно з докладним описом товару.(рис. 1.10).
3. Продавець натискає кнопку «Delete».
4. Продавець змінює поля форми, змінює файл зображення.
5. Продавець натискає кнопку «Save».
6. Продавець побачить вікно з підтвердженням видалення (рис 1.11).
7. Після підтвердження товар видаляється, каталог оновлюється.

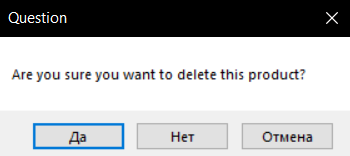


Рисунок 1.11 – Вікно з підтвердженням видалення товару

Функція «Оновлення інформації про замовлення»

Після того, як хтось замовить товар, продавець побачить це замовлення у вкладці «Мої замовлення».

1. ПРОЕКТНА СПЕЦИФІКАЦІЯ
   1. Архітектура

Програма буде створюватися на мові C # з використанням Microsoft Visual Studio 2019 на платформі .Net Framework 4.7.2 з використанням інтерфейсу програмування Windows Forms.

Вибір мови обумовлений об'єктно-орієнтованим підходом до написання програми. На даний момент мова C # є однією з найпопулярніших, зручних і повноцінних серед об'єктно-орієнтованих мов програмування.

В процесі проектування розглядалося 2 платформи, з якими мова C # має найкраще взаємодію, а саме .Net Core і .Net Framework. Наш вибір припав на останню через збільшеної кількості керуючих елементів і більш високого рівня розвитку платформи. На жаль, це означає, що додаток буде працювати тільки на машинах з ОС Windows.

Для реалізації призначеного для користувача інтерфейсу було вирішено використовувати графічний інтерфейс, т. к. він простий для сприйняття і інтуїтивно зрозумілий. Для реалізації інтерфейсу ми будемо використовувати інтерфейс Windows Forms, який є частиною платформи .Net Framework. З його допомогою у нас буде можливість створити повноцінний інтерфейс для користувача програми.

* 1. Структура проекту

Буде нерозумно ділити проект, описаний в специфікації на два окремі проекти, оскільки функції продавця і покупця перетинаються.

На цій підставі можемо скласти наступну структуру проекту:



Рисунок 2.1 – Структура проекту

* 1. Об’єктна модель
     1. Опис

На біржі товарів є товари.

Користувач додає нові товари і оформляє замовлення.

Про товар відомо: ідентифікатор, найменування, одиниця виміру, роздрібна ціна, оптова ціна, мінімальне замовлення для опту, запас, продавець.

Про користувача відомо: ідентифікатор, ім'я, пароль, номер телефону, електронна пошта.

Про замовлення відомо: номер замовлення, покупець, продавець, товар, кількість, дата і час замовлення, спосіб оплати.

Товар містить тип даних «Користувач». Замовлення містить типи даних «Товар» і «Користувач» відповідно.

До простих типів не належать:

База товарів, товар, список користувачів, користувач, база замовлень, замовлення.

З них не є колекціями і колекцій не містять:

Товар, користувач, замовлення.

База товарів - це набір товарів.

База замовлень - це колекція замовлень.

Список користувачів - це колекція користувачів.

* + 1. Методи

На данном этапе к главным методам можно отнести:

1. Пошук користувача по Id.
2. Пошук товару по Id.
3. Пошук замовлення по Id

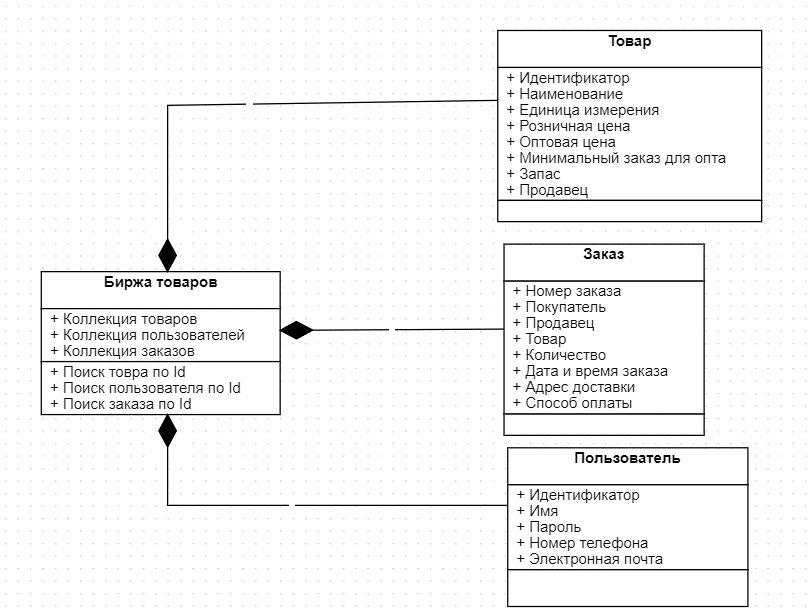


Рисунок 2.2 – Діаграма класів

* 1. Зберігання та доступ до даних

Всі дані будуть зберігатися в єдиному бінарному файлі «exchange.bin». Дані будуть переводитися в потік байтів за допомогою серіалізації з використанням вбудованої бібліотеки BinaryFormatter. Захист даних в рамках роботи не розглядається.

1. ІНСТРУКЦІЯ КОРИСТУВАЧА
   1. Встановлення, запуск та головне вікно

Для користування програмою її необхідно завантажити на ваш комп’ютер. Після цього необхідного відкрити файл за шляхом \CommodityExchange\ExchangeApp\bin\Debug\ExchangeApp.exe.

Після запуску програми відкривається вікно входу (рис. 1.2).

* 1. Реєстрація та вхід

При першому вході в програму вам потрібно буде створити обліковий запис. Для цього натисніть кнопку «I do not have an account». Відкриється вікно реєстрації (рис. 1.1).

Введіть свої дані та нажміть кнопку «Register».

При наступному вході вже не треба реєструвати новий обліковий запис, а просто зайти зі старого. Для цього введіть своє ім’я та пароль і нажміть кнопку «Sign in». Після цього відкриється головне вікно (рис. 1.4)

* 1. Перегляд товарів та оформлення заказу

На вкладці «Каталог» ви бачите всі товари, що продаються на даний момент. Можете вибрати будь-який з них та відкрити додаткову інформацію про нього, зробивши подвійний клік (рис 1.5).

Для оформлення заказу виберіть товар та натисніть кнопку «Order». Відкриється вікно оформлення замовлення (рис. 1.7). Введіть бажану кількість товару та спосіб оплати, потім натисніть кнопку «Order».

* 1. Додавання та редагування товарів

Щоб додати товар виберіть пункт меню «Add product». Відкриється вікно додавання товару (рис 1.8). Введіть інформацію про товар, виберіть файл зображення та натисніть кнопку «Create». Товар з’явиться у вкладках «Каталог» та «Мої товари».

Щоб змінити інформацію про товар, виберіть його зі списку ті відкрийте вікно з додатковою інформацією. Натисніть кнопку «Edit». Зробіть бажані зміни та натисніть кнопку «Save». Товар змінився.

Щоб видалити товар, виберіть його зі списку ті відкрийте вікно з додатковою інформацією. Натисніть кнопку «Delete». З’явиться вікно з підтвердженням видалення (рис. 1.11). Натисніть кнопку «Так». Товар видалено.

* 1. Закінчення роботи та збереження даних

Після закінчення роботи з програмою потрібно зберегти зміни. Для цього натисніть кнопку «Save» в меню.

Програму можна закрити натиснувши кнопку «Exit» в меню або хрестик у правому верхньому куті вікна.

ВИСНОВКИ

У ході виконання курсової роботи були освоєні методи створення програмних систем на засадах об’єктно-орієнтованого програмування, набулися навички роботи з програмною системою, створення в’язків між об’єктами.

У результаті отримана програма під назвою «Біржа товарів», яка дозволяє звичайному користувачу переглядати товари, робити замовлення, виставляти на продаж та редагувати свої товари.

Програма написана на мові програмування C# на платформі .NET Framework 4.7.2 з використанням інтерфейсу програмування Windows Forms.

Усі дані програми зберігаються локально у файлі exchange.bin. Програма може бути поліпшена застосуванням справжньої бази даних, яка синхронізується з локальними даними біржі по мережі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Полное руководство по языку программирования С# 7.0 и платформе .NET 4.7 [Електронний ресурс] – Режим доступу до ресурсу: https://metanit.com/sharp/tutorial/.
2. Бондарев В. М. Объектно-ориентированное программирование на С# [Текст] : учеб. Пособие / В. М. Бондарев. – Харьков : СМИТ, 2009. –224 с. – Библиогр. : ISBN 978-966-2028-34-8.
3. Microsoft Developer Network. Библиотека MSDN / MSDN – сеть разработчиков Microsoft [Електронний ресурс] – Режим доступу до ресурсу: https://msdn.microsoft.com/ru-ru/.
4. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5, 6-е изд. [Текст]: пер. с англ./ Э. Троелсен.- М.: Вильямс, 2013 - 1312 с. – ISBN 978-5-8459-1814-7, 978-1-43-024233-8.
5. Либерти Дж. Программирование на С# [Текст] : пер. с англ. / Дж. Либерти. – 2-е изд. – СПб. ; М. : Символ-Плюс, 2003. – 688 с.

ДОДАТОК А  
Клас User

public class User

{

public Guid Id { set; get; }

public string Name { set; get; }

public string Password { set; get; }

public string Phone { set; get; }

public string Email { set; get; }

public User() { }

public User(string name, string password, string phone, string email)

{

Id = Guid.NewGuid();

Name = name;

Password = password;

Phone = phone;

Email = email;

}

}

ДОДАТОК Б  
Клас Product

public class Product

{

public Guid Id { set; get; }

public string Name { set; get; }

public string Unit { set; get; }

public double RetailPrice { set; get; }

public double WholePrice { set; get; }

public double MinimalWhole { set; get; }

public double Stock { set; get; }

public User Seller { set; get; }

public Image Image { set; get; }

public Product(string name, string unit, double retailPrice, double wholePrice, double minimalWhole, double stock, User seller, Image image)

{

Id = Guid.NewGuid();

Name = name;

Unit = unit;

RetailPrice = retailPrice;

WholePrice = wholePrice;

MinimalWhole = minimalWhole;

Stock = stock;

Seller = seller;

Image = image;

}

}

ДОДАТОК В  
Методи класу TicketPrice

public class Order

{

public Guid Id { set; get; }

public User Buyer { set; get; }

public User Seller { set; get; }

public Product Product { set; get; }

public double Amount { set; get; }

public double Price { set; get; }

public DateTime DateTime { set; get; }

public string PayMethod { set; get; }

public Order(User buyer, User seller, Product product, double amount, double price, string paymethod, DateTime? t = null)

{

Id = Guid.NewGuid();

Buyer = buyer;

Seller = seller;

Product = product;

Amount = amount;

Price = price;

PayMethod = paymethod;

if (t == null)

DateTime = DateTime.Now;

else

DateTime = (DateTime)t;

}

}

ДОДАТОК Г  
Клас Password

public class Exchange

{

public List<Product> Products { private set; get; }

public List<User> Users { private set; get; }

public List<Order> Orders { private set; get; }

public Exchange()

{

Products = new List<Product>();

Users = new List<User>();

Orders = new List<Order>();

}

public void Save()

{

new Dao(this).Save();

}

public void Load()

{

new Dao(this).Load();

}

public Product FindProductById(Guid Id)

{

foreach (Product product in Products)

{

if (product.Id == Id)

return product;

}

throw new Exception("Product Not Found");

}

public User FindUserById(Guid Id)

{

foreach (User user in Users)

{

if (user.Id == Id)

return user;

}

throw new Exception("User Not Found");

}

public Order FindOrderById(Guid Id)

{

foreach (Order order in Orders)

{

if (order.Id == Id)

return order;

}

throw new Exception("ProductNotFound");

}

}

ДОДАТОК Д  
Клас Dao

public class Dao

{

Exchange exchange;

const string filePath = "exchange.bin";

public Dao(Exchange exchange)

{

this.exchange = exchange;

}

public void Save()

{

using (Stream stream = File.Create(filePath))

{

var serializer = new BinaryFormatter();

serializer.Serialize(stream, exchange);

}

}

public void Load()

{

using (Stream stream = File.OpenRead(filePath))

{

var serializer = new BinaryFormatter();

Exchange ex = (Exchange)serializer.Deserialize(stream);

Copy(ex.Products, exchange.Products);

Copy(ex.Users, exchange.Users);

Copy(ex.Orders, exchange.Orders);

}

void Copy<T>(List<T> from, List<T> to)

{

to.Clear();

to.AddRange(from);

}

}

}

ДОДАТОК Е  
Посилання на репозиторій GitHub

<https://github.com/Yurambolo/CommodityExchange>