

Львівський національний університет імені Івана  
Франка

Звіт до курсової роботи з дисципліни "Програмне забезпечення"

---

ЗВІТ

---

Виконав  
Куновський Р.Л.

Перевірив  
ас. Музичук Ю.А.

14 грудня 2019 р.



# Зміст

1	Постановка завдання	1
1.1	Завдання	1
1.2	Поставлені задачі	1
2	База даних	3
2.1	Опис бази даних	3
2.1.1	Опис таблиць	3
2.1.2	Опис тестових даних	4
2.2	Схема бази даних	5
3	Підключення до бази даних і слої доступу	7
3.1	Вигляд проекту	7
3.2	ConnectionManager	7
3.3	Entity	7
3.4	Dal	8
3.5	DalFactory	8
3.6	Bll	9
3.7	Dto	9
3.8	Visitor	9
3.9	Спілкування рівнів	9
4	ConsoleApp	11
4.1	Menu	11
5	Windows Forms App	13
5.1	Вигляд	13
6	WPF App	17
6.1	Реалізація	17
6.2	Вигляд	17
7	Висновок	19



## Розділ 1

# Постановка завдання

### 1.1 Завдання

Варіант 14: Роль Registered User – логується, попадає на стартову сторінку з можливістю перейти на сторінку своєї ролі за допомогою меню. Доступний перехід на сторінку історії куплених товарів. Може виконувати пошук та сортування. Може розлогуватися. Реалізувати названий функціонал.

### 1.2 Поставлені задачі

Реалізувати завдання, використовуючи C#, а саме реалізувати ConsoleApp, WindowsFormApp і WPFApp. Дані зберігати за допомогою DBMS Microsoft SQL Server.



## Розділ 2

# База даних

### 2.1 Опис бази даних

Для виконання завдання, я використовував DBMS Microsoft SQL Server. Я створив бд з наступними таблицями: tblAdvertising, tblUser, tblPerson, tblSpecialQuestion, tblProduct, tblPurchaseHistory.

#### 2.1.1 Опис таблиць

```
CREATE TABLE tblAdvertising(  
id INT NOT NULL PRIMARY KEY IDENTITY(1,1),  
purchase VARCHAR(30) NOT NULL,  
cost FLOAT,  
type VARCHAR(30) NOT NULL,  
info TEXT,  
picture VARCHAR(100));
```

```
CREATE TABLE tblUser(  
id INT NOT NULL PRIMARY KEY IDENTITY(1,1),  
nickname VARCHAR(20) UNIQUE NOT NULL,  
password VARCHAR(MAX) NOT NULL);
```

```
CREATE TABLE tblPerson (  
user_id INT NOT NULL UNIQUE,  
name VARCHAR(20) NOT NULL,  
surname VARCHAR(20) NOT NULL,  
phone_number INT NOT NULL,  
born_date DATE NOT NULL,  
address VARCHAR(50) NOT NULL,  
FOREIGN KEY (user_id) REFERENCES tblUser(id));
```

```
CREATE TABLE tblSpecialQuestion (  
user_id INT NOT NULL UNIQUE,  
question TEXT NOT NULL,  
answer VARCHAR(20) NOT NULL,  
FOREIGN KEY (user_id) REFERENCES tblUser(id));
```

```
CREATE TABLE tblProduct(  
id INT NOT NULL PRIMARY KEY IDENTITY(1,1),  
name VARCHAR(30) NOT NULL,  
cost FLOAT NOT NULL,  
type VARCHAR(30) NOT NULL,  
brand VARCHAR(30) NOT NULL)
```

```
CREATE TABLE tblPurchaseHistory(  
user_id INT NOT NULL,  
product_id INT NOT NULL,
```

```

date DATE NOT NULL,
FOREIGN KEY (product_id) REFERENCES tblProduct(id),
FOREIGN KEY (user_id) REFERENCES tblUser(id));

```

### 2.1.2 Опис тестових даних

```

INSERT INTO tblAdvertising(purchase,cost,type,info,picture) VALUES
('Pasta',25,'macaroni','Delicious pasta ...','/Pasta.jpeg'),
('Mivina',5,'macaroni','Potato powder...','/Mivina.jpeg'),
('Funta',15,'water','Bright, bubbly ...','/Funta.jpg'),
('Sprite',16,'water','Sprite - це вибуховий, ... ','/Sprite.jpg'),
('Roshen',30,'sweet','(Roshen) - українська ...','/Roshen.jpg'),
('Svitoch',30,'sweet','Історія Світоч ...','/Swetoch.jpg');

```

```

INSERT INTO tblUser(nickname,password) VALUES
('kunorom','dd26ea8b2bc48d1b4038b99e4fdf612a'),
('ivan228','2c42e5cf1cdbafea04ed267018ef1511'),
('dima7','0f5b25cd58319cde0e6e33715b66db4c');

```

```

INSERT INTO tblPerson(name,surname,phone_number,born_date,address,user_id) VALUES
('Roman','Kunovskyi',0987241896,'2000-2-5','I Frunka 4',1),
('Ivan','Vynavskyi',0984354365,'2000-2-12','S Mazepy 7',2),
('Dmitro','Levkovich',0678903534,'2000-11-20','Sholom 3',3);

```

```

INSERT INTO tblSpecialQuestion(quetion,answer,user_id) VALUES
('Where are you study?','LNU',1),
('Who are you?','Dodeck',2),
('What is your favorite song?','Umorist',3);

```

```

INSERT INTO tblProduct(name,cost,type,brand) VALUES
('Pasta',2.5,'macaroni','Boroshno'),
('Mivina',0.5,'macaroni','Mivina'),
('Sprite',3,'water','Coca-Cola'),
('Fanta',3.1,'water','Coca-Cola'),
('Coca-Cola',2.9,'water','Coca-Cola'),
('Chokolate',4.5,'sweet','Roshen'),
('Chokolate',4,'sweet','Svitoch');

```

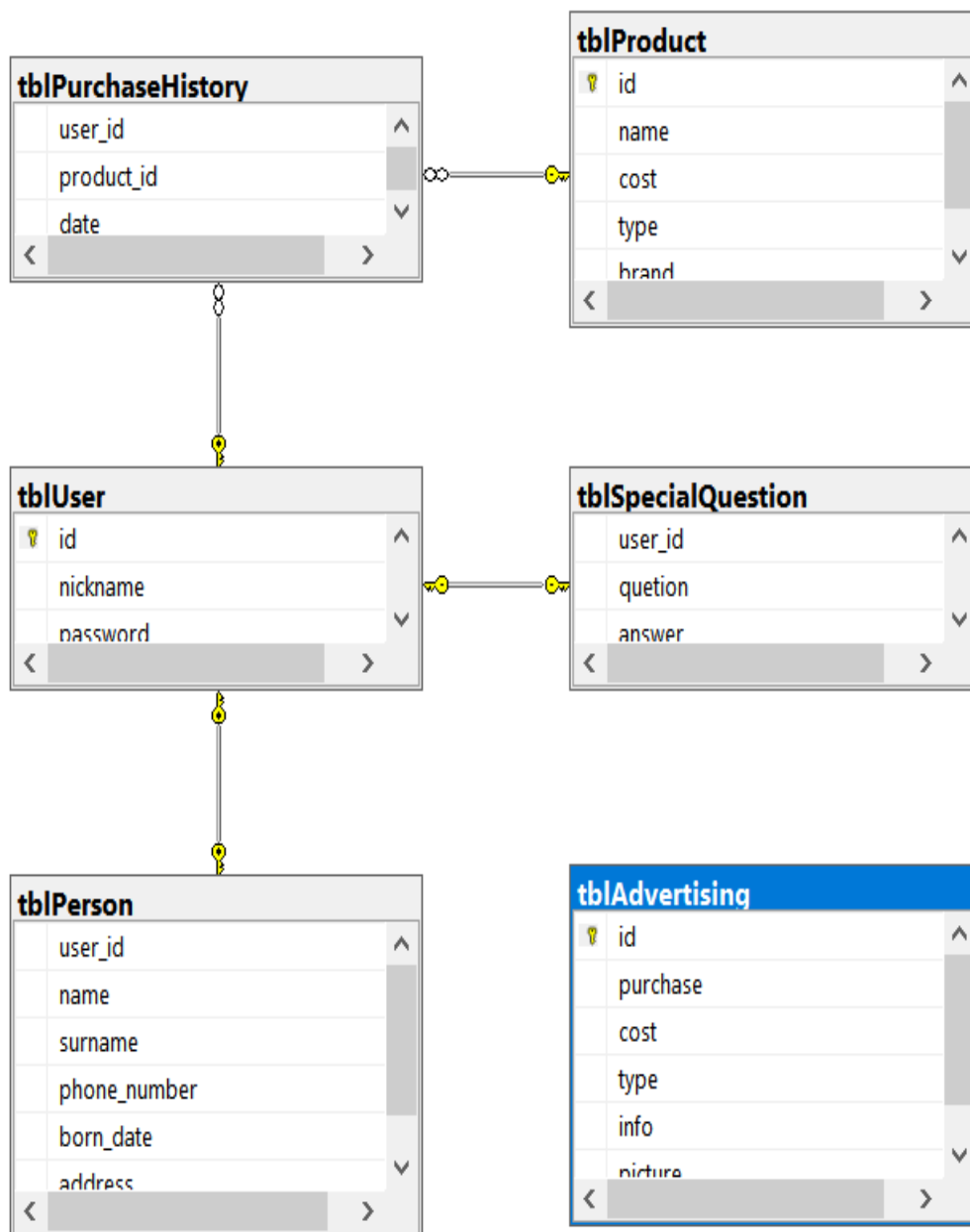
```

INSERT INTO tblPurchaseHistory(date,user_id,product_id) VALUES
('2010-2-5',2,7),
('2010-4-4',2,4),
('2010-5-5',2,6),
('2010-6-7',2,5),
('2010-8-9',3,2),
('2012-3-10',3,1),
('2012-12-15',3,1),
('2012-2-5',1,5),
('2014-4-4',3,2),
('2015-2-2',2,3),
('2016-5-6',1,4),
('2017-1-1',1,7),
('2018-1-2',1,6);

```



## 2.2 Схема бази даних

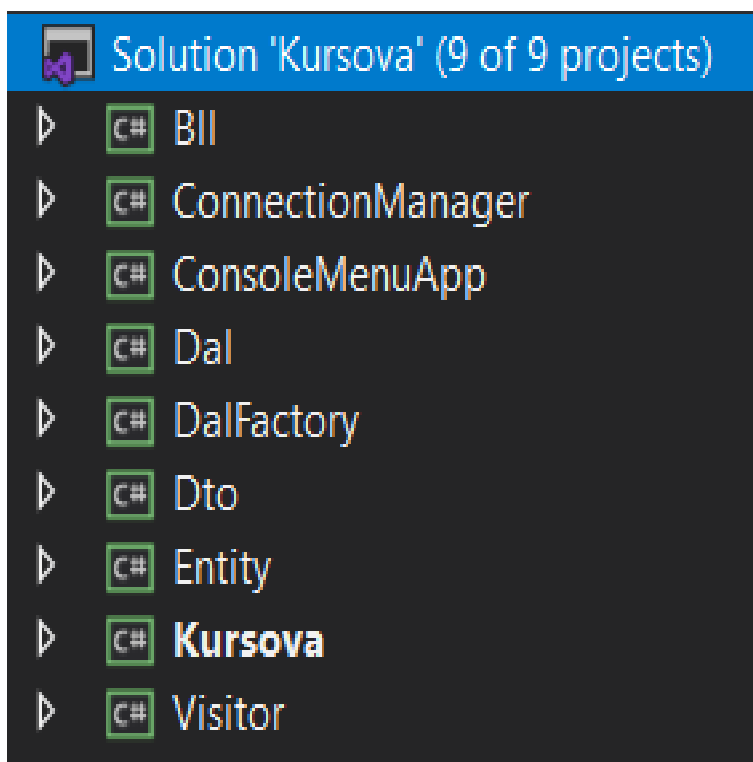




## Розділ 3

# Підключення до бази даних і слої доступу

### 3.1 Вигляд проекту



### 3.2 ConnectionManager

Підключення до бази даних відбувається за допомогою конекшин стрінгу, який зберігається в окремому класі в окремій бібліотеці. Також, там знаходиться клас "Сінгтон який роздає крнекти потокам, які до нього звертаються.

### 3.3 Entity

Entity представляє собою сутність, яка відповідає спобцям таблиці в БД. Приклад (tblPerson):

```
public enum EPersonEntity
{
    ID = 0,
    NAME,
    SURNAME,
```

```

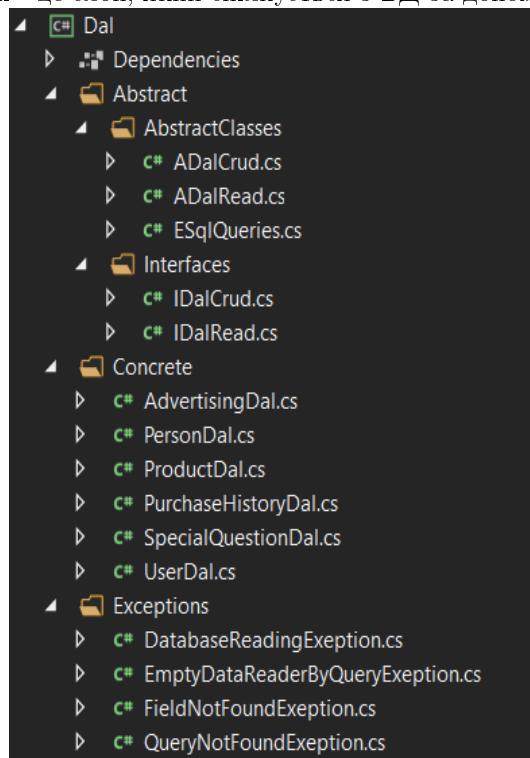
        PHONE_NUMBER,
        BORN_DATE,
        ADDRESS
    }
    public sealed class PersonEntity:IEntity
    {
        public long? ID { get; set; }
        public string Name { get; set; }
        public string Surname { get; set; }
        public long? PhoneNumber { get; set; }
        public DateTime BornDate { get; set; }
        public string Address { get; set; }
        public PersonEntity(long id,string name, string surname, long phoneNumber, DateTime bornDate)
        {
            ID = id; Name = name; Surname = surname; PhoneNumber = phoneNumber; BornDate = bornDate;
        }
    }

```

IEntity - це інтерфейс з властивістю ID, його реалізує кожна з Entity.

### 3.4 Dal

Dal - це слой, який спілкується з БД за допомогою SQL query.



Тут абстрактні класи(Read - лише для зчитування таблиць, Crud - для зчитування і редагування таблиць) реалізують відповідні інтерфейси. Ці абстрактні класи наслідуються класами, які безпосередньо спілкуються з БД(кожною таблицею окремо). Класи "Експешини"представляють собою класи, які обробляють помилки, що можуть виникнути при роботі Dal рівню.

### 3.5 DalFactory

Це сінглтон клас,який звертається до всіх класів Dal рівню, що знаходяться в DalFactory. Це доволі зручний спосіб, адже, підключивши DalFactory, ми маємо доступ до всіх таблиць нашої БД.

## 3.6 Bll

Цей рішчень реалізує логіку, яка є необхідною для функціонування нашої програми. Також варто зазначити, що Bll приймає Entity і перезаписує їх в Dto. Також тут відбувається хешування паролей.

## 3.7 Dto

Це класи моделі, які призначені для зберігання даних і транспортування їх по проекту. Приклад:

```
class UserDto
{
    public string Nickname { get; set; }
    public string Password { get; set; }
    public string Name { get; set; }
    public string Surname { get; set; }
    public long? PhoneNumber { get; set; }
    public DateTime BornDate { get; set; }
    public string Address { get; set; }
    public string Question { get; set; }
    public string Answer { get; set; }
    public UserDto(string nickname, string password, string name, string
        surname, long phoneNumber, DateTime bornDate, string address, string
        question, string answer)
    {
        Nickname = nickname; Password = password; Name = name;
        Surname = surname; PhoneNumber = phoneNumber;
        BornDate = bornDate; Address = address;
        Question = question; Answer = answer;
    }
    private string ToString()
    {
        return "User:\n"
            + "\t(nickname = " + Nickname + ",\n"
            + "\tPassword = " + Password + "\n"
            + "\tName = " + Name + "\n"
            + "\tSurname = " + Surname + "\n"
            + "\tPhoneNumber = " + PhoneNumber + "\n"
            + "\tBornDate = " + BornDate + "\n"
            + "\tAddress = " + Address + "\n"
            + "\tQuestion = " + Question + "\n"
            + "\tAnswer = " + Answer + ")";
    }
}
```

## 3.8 Visitor

Це клас для зберігання можливостей і інформації юзера. В цьому проекті можливі два види юзера: RegisteredUser, FreeVisitor.

## 3.9 Спілкування рівнів

ConnectionManeger конектится до БД. Про ConnectionManeger знає Dal, який звертається до БД через відповідні квері і за допомогою Entity передає інформацію з БД.

DalFactory знає про всі класи Dal і зберігає їх екзмпляри. Bll знає про DalFactory і через неї доступаться до Dal рівню. Також він перезаписує дані з Entity в Dto і відправляє їх на вищій рівні.

## Розділ 4

# ConsoleApp

### 4.1 Menu

Це клас який реалізовує консольне меню, за допомогою printMethods і ControlMethods.  
Приклад:

```
private void PurchaseHistorySortMenu()
{
    Console.WriteLine("_____Sort_____");
    Console.WriteLine("Sort_by_cost_____1");
    Console.WriteLine("sort_by_purchase_____2");
    Console.WriteLine("sort_by_type_____3");
    Console.WriteLine("sort_by_date_____4");
    Console.WriteLine("sort_by_brand_____5");
    Console.WriteLine("Exid_____0");
}

private void PurchaseHistorySort()
{
    ConsoleKeyInfo key;
    do
    {
        Console.Clear();
        PurchaseHistorySortMenu();
        key = Console.ReadKey(true);
        switch (key.Key)
        {
            case ConsoleKey.D1:
                Visitor.GetPurchaseHistoryBll().SortByCost();
                return;
            case ConsoleKey.D2:
                Visitor.GetPurchaseHistoryBll().SortByName();
                return;
            case ConsoleKey.D3:
                Visitor.GetPurchaseHistoryBll().SortByType();
                return;
            case ConsoleKey.D4:
                Visitor.GetPurchaseHistoryBll().SortByDate();
                return;
            case ConsoleKey.D5:
                Visitor.GetPurchaseHistoryBll().SortByBrand();
                return;
        }
    } while (key.Key != ConsoleKey.D0);
}
}}
```

Де `PurchaseHistorySortMenu` реалізовує консольне `View`, а `PurchaseHistorySort` логіку переходів і звернення до `ВІІ` для отримання або обробки даних.



## Розділ 5

# Windows Forms App

Windows Forms — інтерфейс програмування додатків (API), відповідальний за графічний інтерфейс користувача і є частиною Microsoft .NET Framework. Даний інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за допомогою створення обгортки для Win32 API в керованому коді.

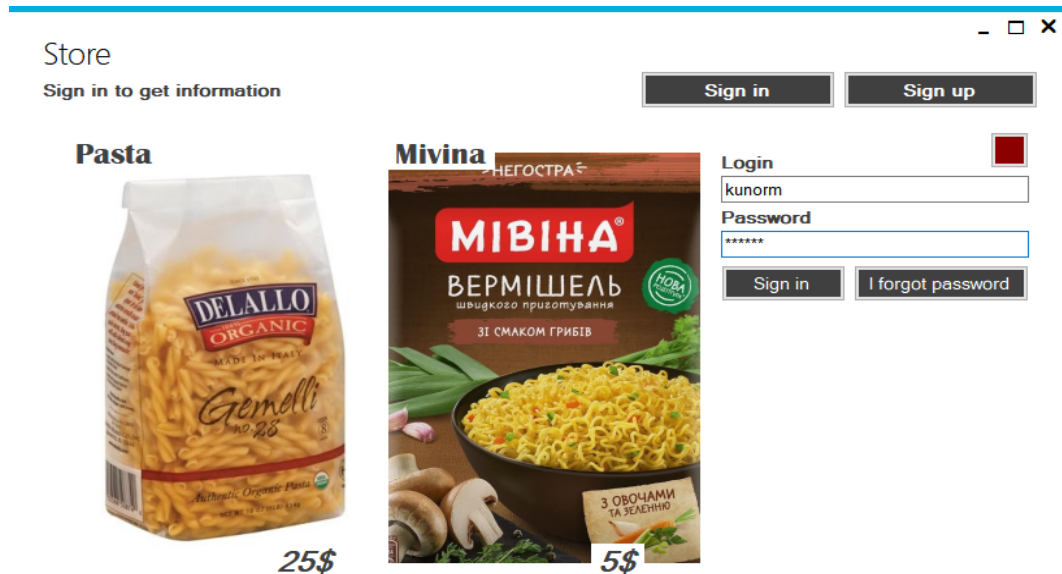
Всередині .NET Framework, Windows Forms реалізується в межах простору імен System.Windows.Forms.

### 5.1 Вигляд

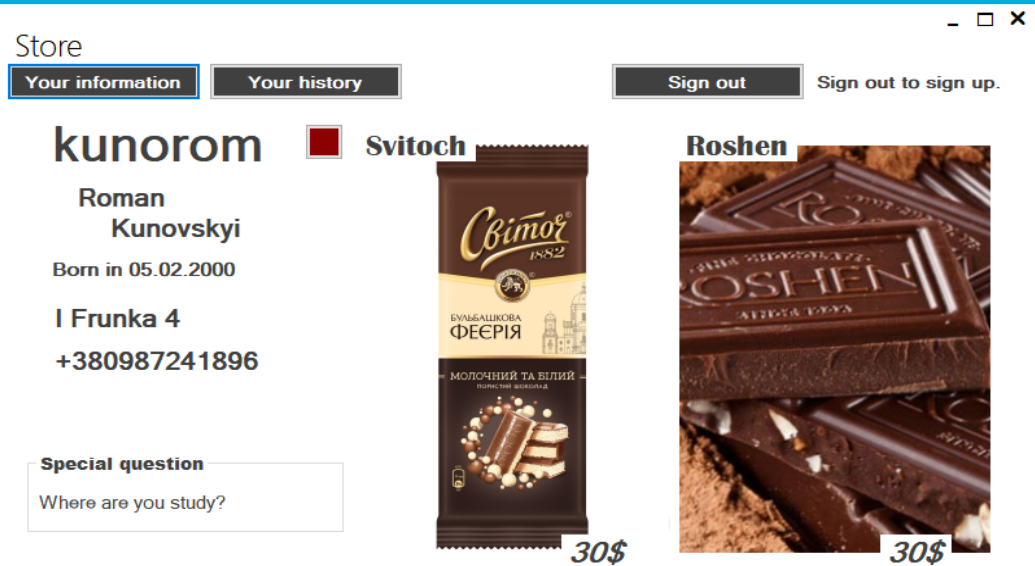
Запустимо програму. Користувач бачить рекламу і має можливість увійти в акаунт (відновлення паролю передбачене) або зареєструватися.



Ввійдемо в акаунт...



Реклама змінилася, адже була проаналізована історія покупок користувача.



Переглянемо історію покупок.

Store

Sort by

Type

Name

Cost

Date

Brand

Filter by

Filter

Clear filters

Product name

Product type

Product brand

Product cost

from

to

	Name	Cost	Type	Barand	Date
▶	Null7476476	23\$	Null11	Null11	06.12.2019 0:00:00
	NewP	32\$	Null123	Null123	06.12.2019 0:00:00
	Coca-Cola	2,9\$	spilling water	Coca-Cola	05.02.2012 0:00:00
	Fanta	3,1\$	water123	Coca-Cola	06.05.2016 0:00:00
	Milk Chokolate1	4\$	sweet	Svitoch	01.01.2017 0:00:00
	qwer	22\$	qwer	qwre	23.11.2019 0:00:00
*					

6

Тут можна сортувати і фільтрувати дані про історію покупок.

Якщо авторизований користувач закриє програму, його його логін і пароль серіалізуються за допомогою JSON, і при його поверненні увійде в акаунт автоматично.



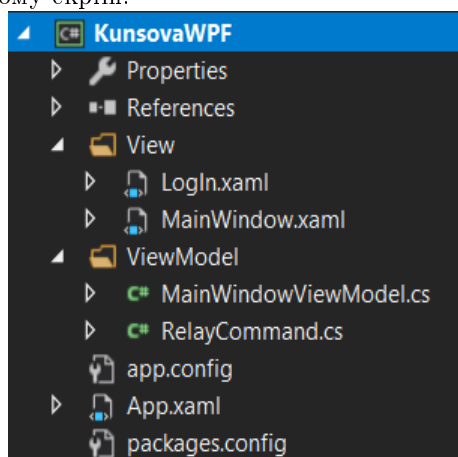
## Розділ 6

# WPF App

Windows Presentation Foundation (WPF, кодова назва — Avalon) — графічна (презентаційна) підсистема (аналог WinForms), яка починаючи з .NET Framework 3.0 в складі цієї платформи. Має пряме відношення до XAML. WPF разом з .NET Framework 3.0 вбудована в Windows Vista, а також доступна для установки в Windows XP Service Pack 2 і Windows Server 2003.

### 6.1 Реалізація

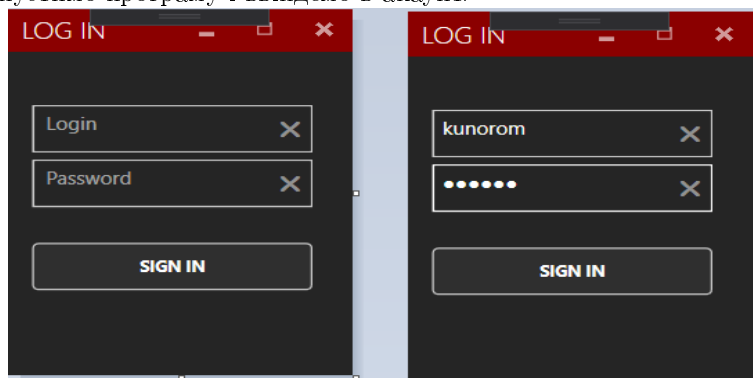
Я реалізував WPF app за допомогою паттерну MVVM. Це можна побачити на наступному скріні.



Класи ViewModel реалізують інтерфейс `INotifyPropertyChanged`. Всі івенти при нажиманні кнопок обробляються за допомогою класу `RelayCommand`, який наслідує інтерфейс `ICommand`.

### 6.2 Вигляд

Запустимо програму і ввійдемо в акаунт.



Ми бачимо меню історії покупок, де ми можемо видаляти, додавати і редагувати дані.

HISTORY

kunorom

Name

×

Cost

×

Type

×

Brand

×

Date

ADD

DELETE

Total Records : 6

NAME	COST	TYPE	BRAND	DATE
Null7476476	23	Null11	Null11	12/6/2019 1
NewP	32	Null123	Null123	12/6/2019 1
Coca-Cola	2.9	spilling water	Coca-Cola	2/5/2012 12
Fanta	3.1	water123	Coca-Cola	5/6/2016 12
Milk Chokolate1	4	sweet	Svitoch	1/1/2017 12
qwer	22	qwer	qwre	11/23/2019

◀

▶

## Розділ 7

# Висновок

Під час виконання цього завдання я навчився користуватися СУБД MS SQL, також, ознайомився і в певній мірі реалізував багаторівневу програму, яка працює з БД. Навчився працювати з Windows Forms і Wpf проектами, а також хешувати дані, серіалізувати за допомогою JSON, використовувати валідацію моделей, реалізовувати різні паттерни проектування і багато іншого... Варто зазначити, що виконання цього завдання зайняло мені доволі багато часу, який я не вважаю витраченим марно, бо я зіткнувся з великою кількістю проблем, які було цікаво вирішувати. Отже, цей проект був хорошим випробуванням для моїх знань в C# і СУБД.