

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ім. Ігоря Сікорського»  
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

**КУРСОВА РОБОТА**

з дисципліни

Програмування та алгоритмічні мови

на тему: Домашня бухгалтерія

Студента 2 курсу групи КА-03  
Галузь знань 124 Системний аналіз  
Захаренко Нікіта Сергійович

Керівник Асистент Древаль Максим  
Михайлович

Національна оцінка \_\_\_\_\_  
Кількість балів: \_\_\_\_\_ Оцінка: ECTS  
\_\_\_\_\_

Члени комісії

_____	_____
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)
_____	_____
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)
_____	_____
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)

Київ – 2021 рік

## **Анотація**

В курсовій роботі представлена програма для спостереження та контролювання власного бюджету користувача. Програма є мультиюзерною, тобто можна створювати безліч акаунтів користувачів зі збереженням усього функціоналу. Продукт надає можливість записувати свої доходи та витрати, після чого дивитися на коротку статистику витрат за вказаний період часу, таку як зміни балансу за цей період та витрати по категоріях. Програма виконана на мові програмування C# в середі розробки Visual Studio 2019.

Кінцевий продукт курсової роботи має статистичний характер та може використовуватися багатьма користувачами на одному пристрої.

## **Summary**

The course work presents a program for monitoring and controlling users' own budget. The program is multi-user, so you can create multiple user accounts while retaining all the features. The product provides an opportunity to record your income and expenses, and then divided into brief statistics of expenses for the specified period of time, as changes in the balance for this period and expenses by category. The program is based on C # language programming in the Visual Studio 2019 development environment.

The final product of the course work is statistical in nature and can be used by many users on one device.

## ВСТУП

Актуальність: Проблема контролю своїх власних витрат для людини успішної є одним з ключових факторів її успіху, тому ця програма є цілком актуальна для людей, які бажають приділяти більше уваги своїм фінансам.

Метою даної курсової роботи було реалізувати алгоритм створення, додавання та змінення баз даних користувачів та їх транзакцій, створити приємний користувацький інтерфейс та додати необхідну статистику для користувача.

Об'єктом дослідження цієї курсової роботи для мене стали бази даних, запити до них та правильне використання Windows Forms в якості інструменту створення інтерфейсу та функціоналу, а також створення ООП структури, яка б дозволяла працювати з базою даних, коректно отримувати та заносити інформацію до неї.

Завдання: На основі власних знань та аналізу літературних джерел була виконана курсова робота. Основні етапи:

- Розробка ідеї програми.
- Розробка баз даних.
- Розробка алгоритму роботи з базами даних.
- Розробка ООП структури.
- Дослідження та втілення інтерфейсу, що використовує Windows Forms.
- Виконання основного дизайну форми.

Були втілені та використані принципи роботи з Forms, що дозволило зробити управління в програмі інтуїтивно зрозумілим та наближеним до реальних десктопних аплікацій.

Практичне значення одержаних результатів: даний продукт може бути використаний для практичного використання багатьма користувачами за одним пристроєм.

Використане програмне забезпечення: При написанні курсової роботи було використане середовище розробки Microsoft Visual Studio та операційна система Windows 11, для підготовки та оформлення курсової роботи використано Microsoft Word.

Структура роботи: Робота складається із вступу, двох розділів, висновків та додатків.

# РОЗДІЛ 1

## Постановка задачі

### 1.1. Основні вимоги для поставленої задачі.

Для коректного користування програмним продуктом такого типу він має мати в собі усі інструменти для додавання та маніпулювання транзакціями, що додав користувач.

До того ж, програма має передбачати створення декількох профілів користувачів, та за потреби мати змогу нагадати користувачеві пароль від його аккаунту, якщо він його раптово забув.

Також, інформація щодо усіх транзакцій користувача має бути доступна тільки йому самому та в повному обсязі, причому в будь-який момент користувач повинен мати змогу передивитись усю доступну статистику його «гаманця».

І, наостанок, програма має бути повністю протестована та позбавлена будь-яких «багів» чи збоїв під час нормального використання її користувачем.

Отже, сумуючи усе вище перелічене, програма повинна:

- Мати інтерфейс входу до системи, реєстрації та нагадування паролю.
- Давати змогу передивлятися усю інформацію та статистику щодо транзакцій користувача в будь-який момент.
- Бути повністю вивіреною, передбачати усі можливі дії користувача в будь-якому місці програми та запобігати помилкам, вказуючи користувачеві на помилку, яку він допустив при користуванні.

### 1.2. Уточнена постановка задачі на розробку програмного забезпечення.

Основна робота програми буде пов'язана з використанням таблиць бази даних, тому потрібно створити базу даних, яка б містила таблиці:

- Користувачів
- Транзакцій

Після цього, потрібно зробити класи-контролери для кожної з цих таблиць, за допомогою яких пізніше буде створюватися зв'язок між додатком та БД. Крім того, слід створити класи User та Transaction, які б мали змогу записувати дані про об'єкти з БД в оперативну пам'ять, задля подальших операцій над ними. Після цього можна переходити до розробки інтерфейсу програми та додавання основних кнопок-контролерів для користувача.

Отже, усі ці пункти мають бути виконані за допомогою мови програмування C#, мати доступний і зрозумілий користувачеві інтерфейс, яким користувач керує мишкою.

## **РОЗДІЛ 2**

### **Розробка програмного продукту**

#### **2.1 Метод розв'язку задачі.**

Для розв'язку даної задачі мною була використана мова C#, інтерфейс був розроблений за допомогою Windows Forms, в якості середовища для бази даних я обрав Microsoft Access. Також, для створення графіків та перегляду статистики був використаний пакет NuGet Package та його бібліотеки під назвою LiveCharts та MaterialSkin.

Найголовнішим серед етапів роботи був етап з підключенням бази даних до програми задля того, щоб вся інформація, яку додає чи змінює користувач, залишалася не в оперативній, а в постійній пам'яті комп'ютера. Для цього використовується клас `ConnectionString`, який містить в собі строку під'єднання до БД.

Після того, як БД була під'єднана до проекту, створюємо класи `UserQuery` та `TransactionQuery`, які відповідатимуть за запити до таблиць із БД, використовуючи бібліотеку `OleDb`.

Далі будуються основні форми програми:

- `Login_Form.cs`,
- `MainUserForm.cs`,
- `AdminForm.cs`,
- `AddTransactionForm.cs`,
- `EditTransactionForm.cs`,
- `ForgotPasswordForm.cs`,
- `StatisticsForm.cs`,
- `RegistrationForm.cs`

Кожна з яких має свій власний функціонал в рамках програми та надає користувачеві різні можливості. До того ж, з кожної форми є вихід на іншу.

Також створюються класи для запису та оперування даними з БД в оперативній пам'яті комп'ютера:

- `User.cs`
- `Transaction.cs`

Для початку роботи з програмою, користувач повинен створити свій обліковий запис на формі `RegistrationForm.cs`, ввівши усі необхідні дані, після успішної реєстрації він буде перенаправлений на форму `Login_Form.cs`, у якій він має ввести дані його облікового запису, а саме – логін та пароль. Після вдалого введення, користувач потрапляє на форму `MainUserForm.cs`, на якій

він має змогу виконувати будь-які операції зі своїм акаунтом, а також передивлятися статистику його балансу. Додавання та редагування транзакції відбувається на формах `AddTransactionForm.cs` та `EditTransactionForm.cs` відповідно. Статистику ж користувач може переглянути на формі `StatisticsForm.cs`. Доступ до усіх цих форм відкритий для користувача за допомогою натискання на відповідні кнопки на головній формі. Також існує додаткова форма адміністратора `AdminForm.cs`, на якій можна переглянути інформацію про усіх користувачів одночасно, а за потреби – видалити будь-якого з них, причому доступ до неї відкривається лише при введенні логіну та паролю профіля адміністратора.

В якості форми, що відповідає за відновлення паролю використовується форма `ForgotPasswordForm.cs`, на якій користувачеві задається секретне запитання, на яке він відповідав при реєстрації. При правильній відповіді на нього, користувачеві показується його пароль.

## 2.2. Інтерфейс програми.

На момент запуску програми, користувач потрапляє до форми `Login_Form.cs`, на якій він має ввести дані його облікового запису. (рис. 2.1)

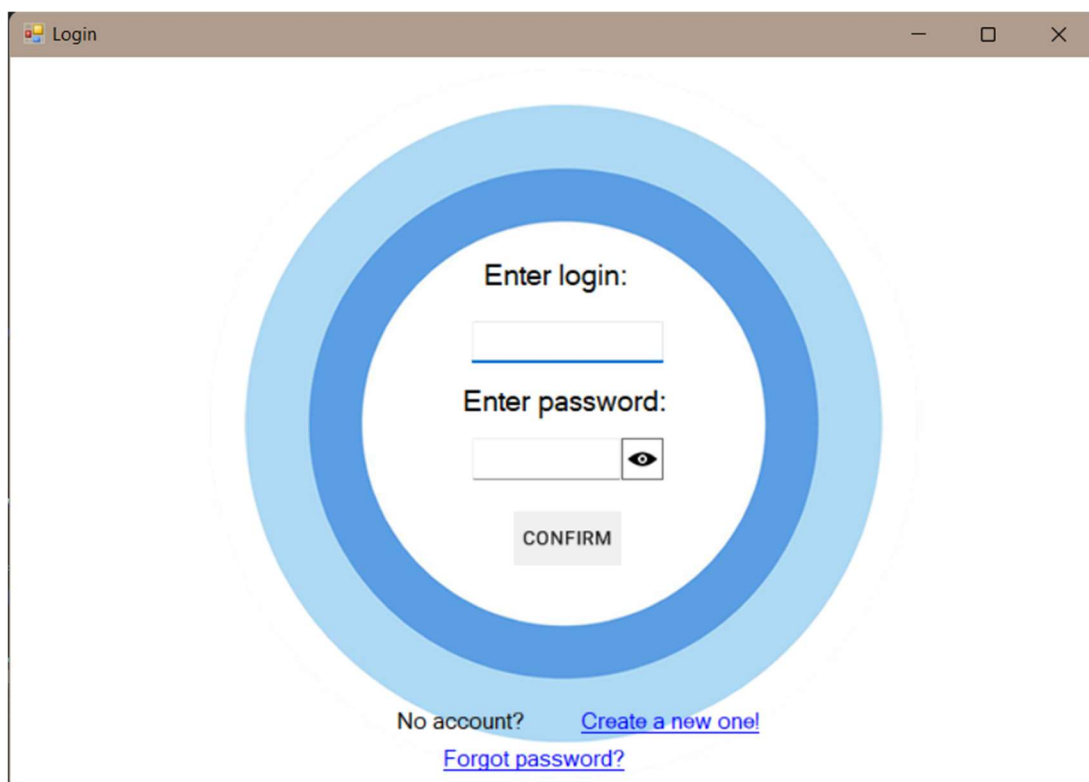


Рис. 2.1

Припустимо, що користувач попадає до програми вперше – в такому випадку він має створити новий акаунт, натисканням на кнопку „Create a new one!“. Після натискання на неї, перед користувачем відкривається форма `RegistrationForm.cs`(рис 2.2):

RegistrationForm

Hi! Let's register!

Enter your name:

Login:

Password:

Confirm your password:

Choose secret question:

Enter answer:

Create

Рис. 2.2

Після введення усіх даних, акаунт буде створено(рис. 2.3):

RegistrationForm

Hi! Let's register!

Enter your name: Nikita

Login: zakharenko

Password: ....

Confirm your password: ....

Choose secret question: The name of your mother?

Enter answer: Natalia

Create

Your account has successfully created!

OK

Рис. 2.3



Після натискання ОК, користувач повернеться на форму логіну, де він має ввести дані аккаунту, який він щойно створив(рис. 2.4):

Enter login:

zakharenko

Enter password:

....

CONFIRM

No account? [Create a new one!](#)

[Forgot password?](#)

Рис. 2.4

Після натискання кнопки «CONFIRM», користувач потрапляє на основну форму програми – MainUserForm.cs(рис. 2.5):

Hi, Nikita!

Your balance: 0 uah.

Add Transaction

Statistics

Delete Edit

Exit

Here are your last transactions:

Category	Value	Date
----------	-------	------

Рис. 2.5

Тепер користувач може починати додавати транзакції, натиснувши кнопку „Add Transaction“, тим самим відкриваючи форму AddTransactionForm.cs(рис. 2.6):

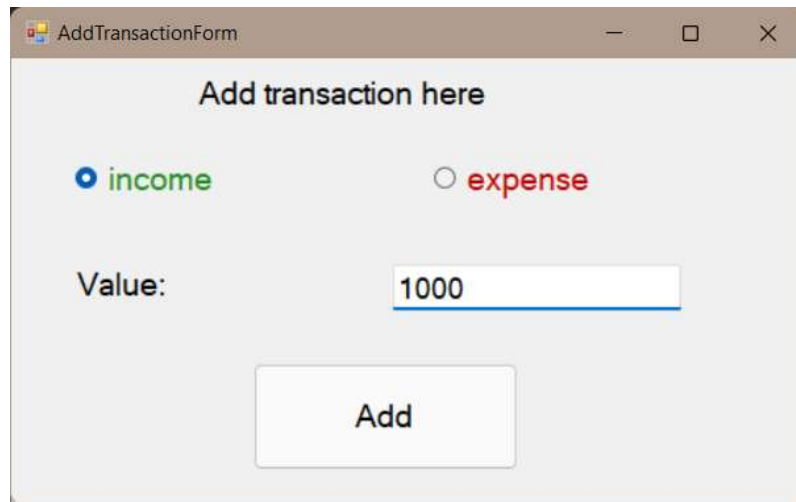
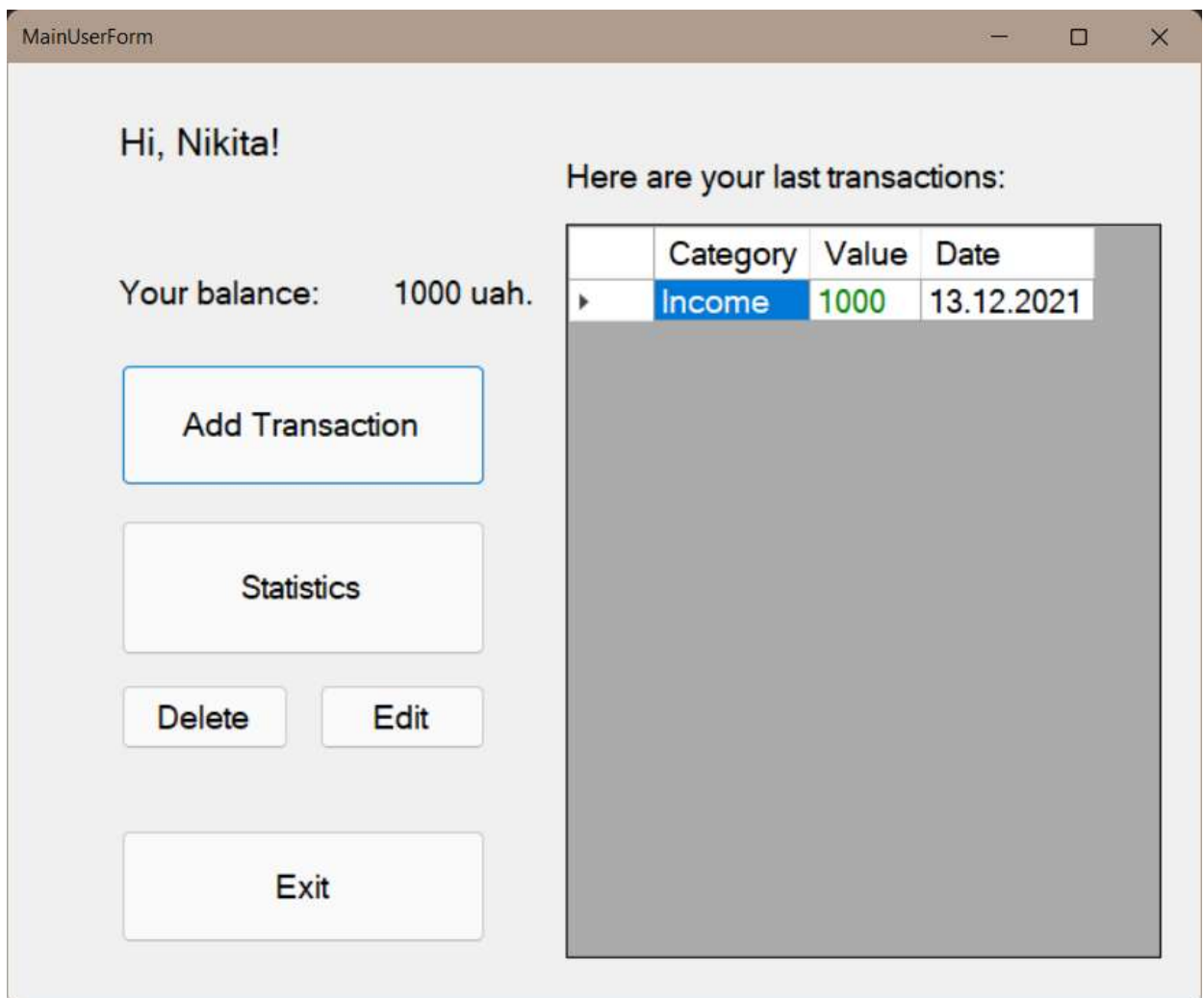


Рис. 2.6

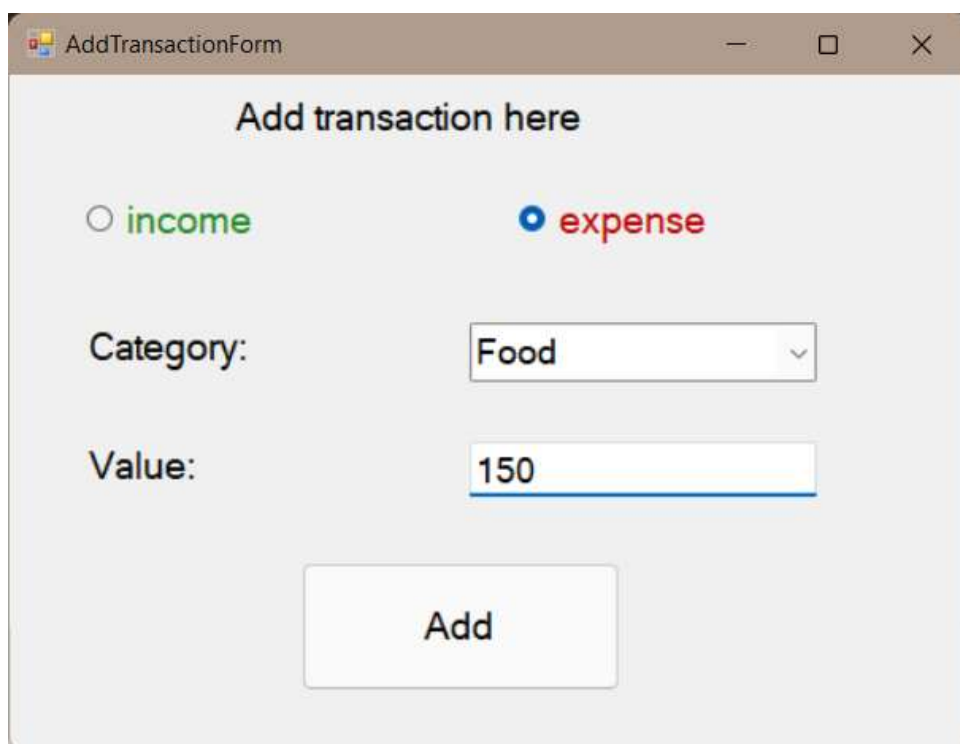
Після натиснення кнопки „Add“, користувач повертається на основну форму(рис. 2.7):



	Category	Value	Date
►	Income	1000	13.12.2021

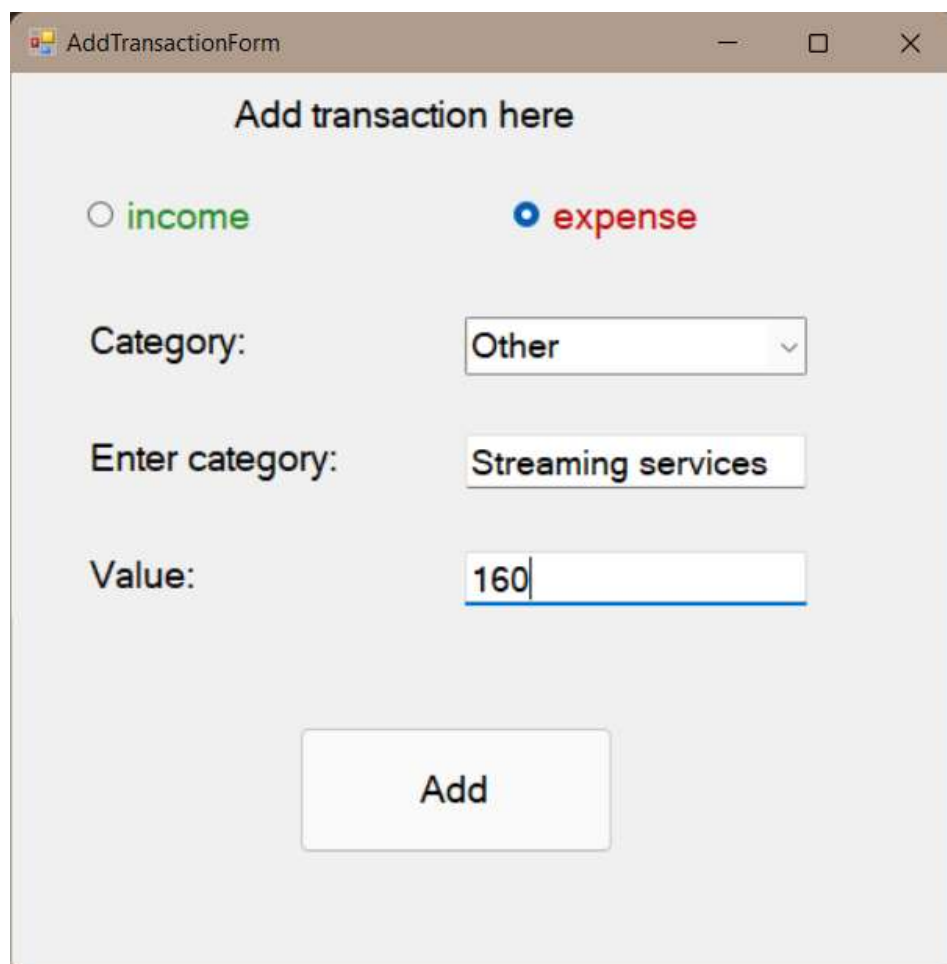
Рис. 2.7

Після додавання ще декількох транзакцій з поміткою «expense»(рис. 2.8, рис. 2.9):



The screenshot shows a window titled "AddTransactionForm" with a light gray background. At the top, it says "Add transaction here". Below this, there are two radio buttons: "income" (green) and "expense" (red). The "expense" radio button is selected. Below the radio buttons, there are two labels: "Category:" and "Value:". The "Category:" label is followed by a dropdown menu showing "Food". The "Value:" label is followed by a text input field containing "150". At the bottom, there is a large button labeled "Add".

Рис. 2.8



The screenshot shows the same "AddTransactionForm" window. The "expense" radio button is still selected. The "Category:" dropdown menu now shows "Other". Below the "Category:" label, there is a new label "Enter category:" followed by a text input field containing "Streaming services". The "Value:" text input field now contains "160". The "Add" button remains at the bottom.

Рис. 2.9

Користувач буде бачити ось таку картину(рис. 2.10):

The screenshot shows a window titled "MainUserForm" with a light gray background. On the left side, there is a greeting "Hi, Nikita!" followed by "Your balance: 690 uah.". Below this, there are four buttons: "Add Transaction", "Statistics", "Delete", and "Exit". The "Delete" and "Edit" buttons are positioned side-by-side. On the right side, there is a heading "Here are your last transactions:" above a table. The table has four columns: "Category", "Value", and "Date". The first row is "Streaming services" with a value of "-160" and date "13.12.2". The second row is "Food" with a value of "-150" and date "13.12.2". The third row is "Income" with a value of "1000" and date "13.12.2". The table is partially obscured by a large gray rectangle.

	Category	Value	Date
▶	Streaming services	-160	13.12.2
	Food	-150	13.12.2
	Income	1000	13.12.2

Рис. 2.10

І зможе подивитися статистику щодо його витрат, натисненням на кнопку "Statistics", та потрапляючи на форму StatisticsForm.cs (рис. 2.11):

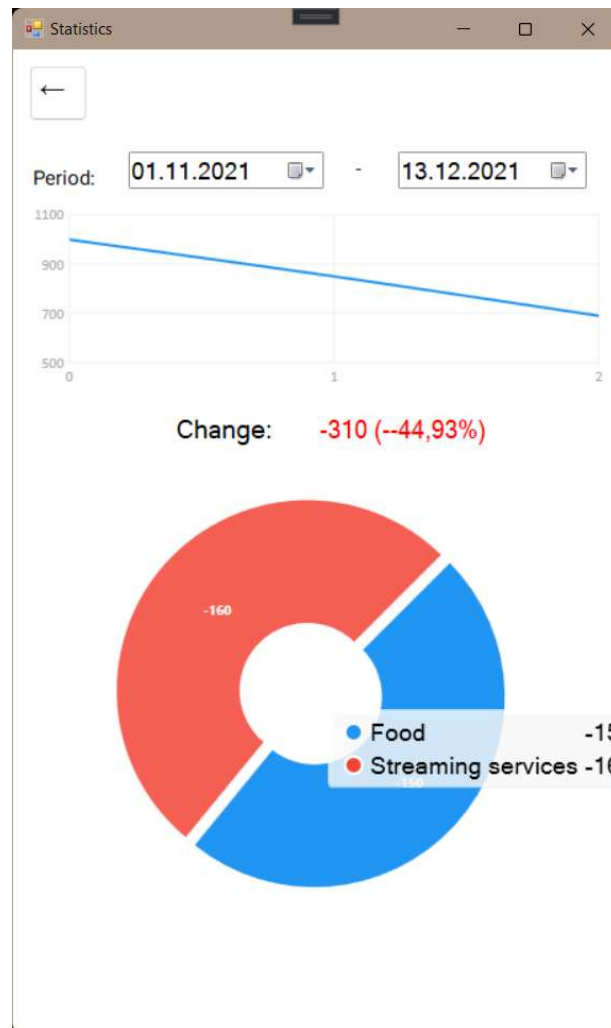


Рис 2.11

Також користувач має змогу змінити дані транзакції, виділивши її та натиснувши кнопку "Edit", відкривши тим самим форму EditTransaction.cs (рис 2.12 ):

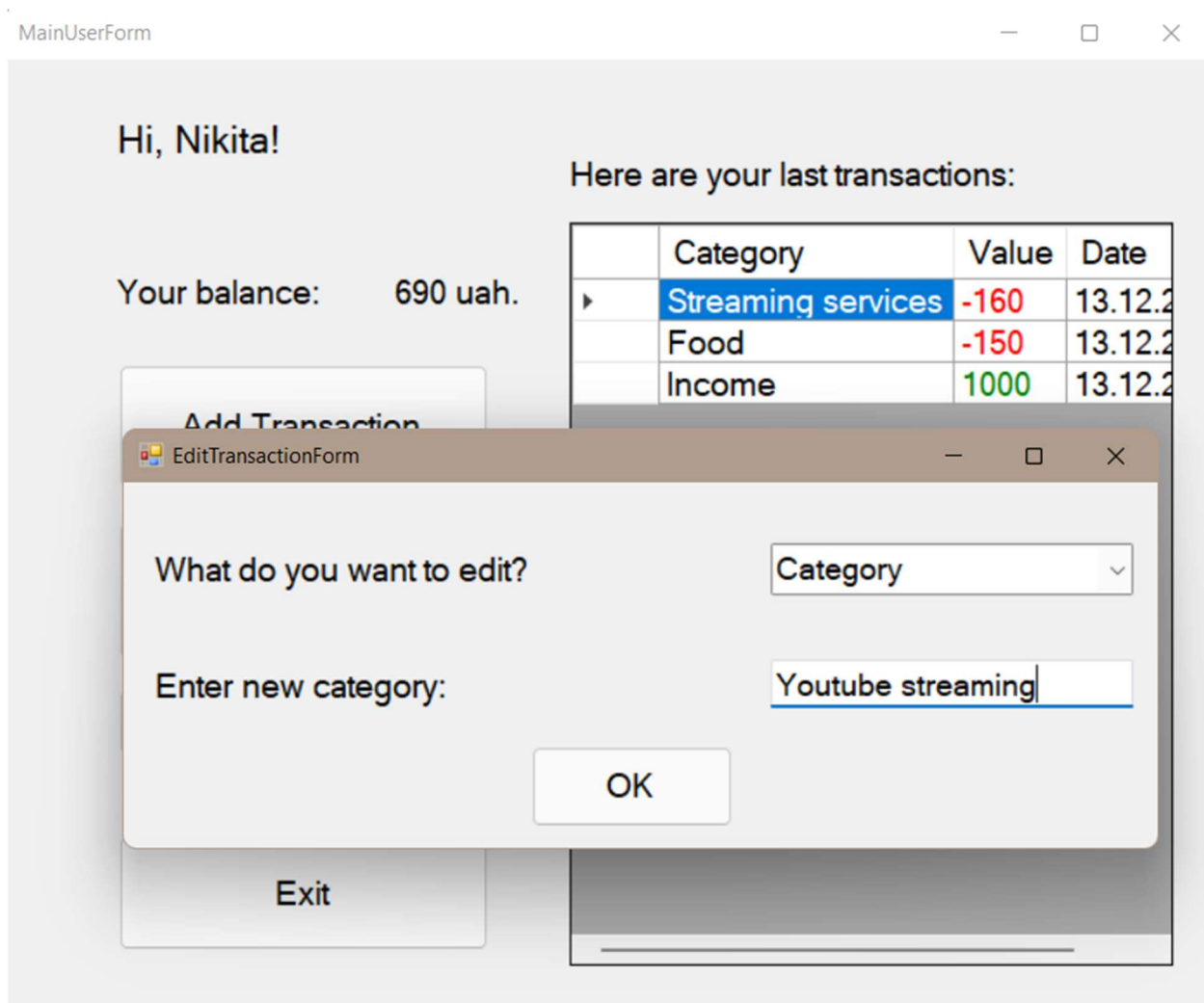


Рис 2.12

Після чого він повернеться на основну сторінку, на якій дані вже будуть змінені(рис. 2.13):

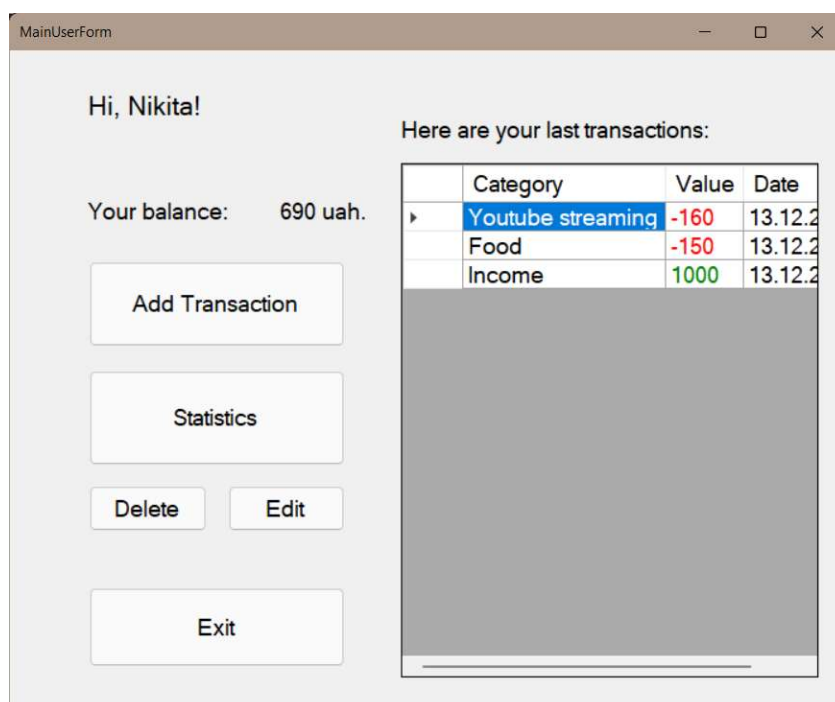


Рис 2.13

Таким же чином, виділивши в таблиці потрібну транзакцію, користувач може її видалити(рис 2.14, 2.15):

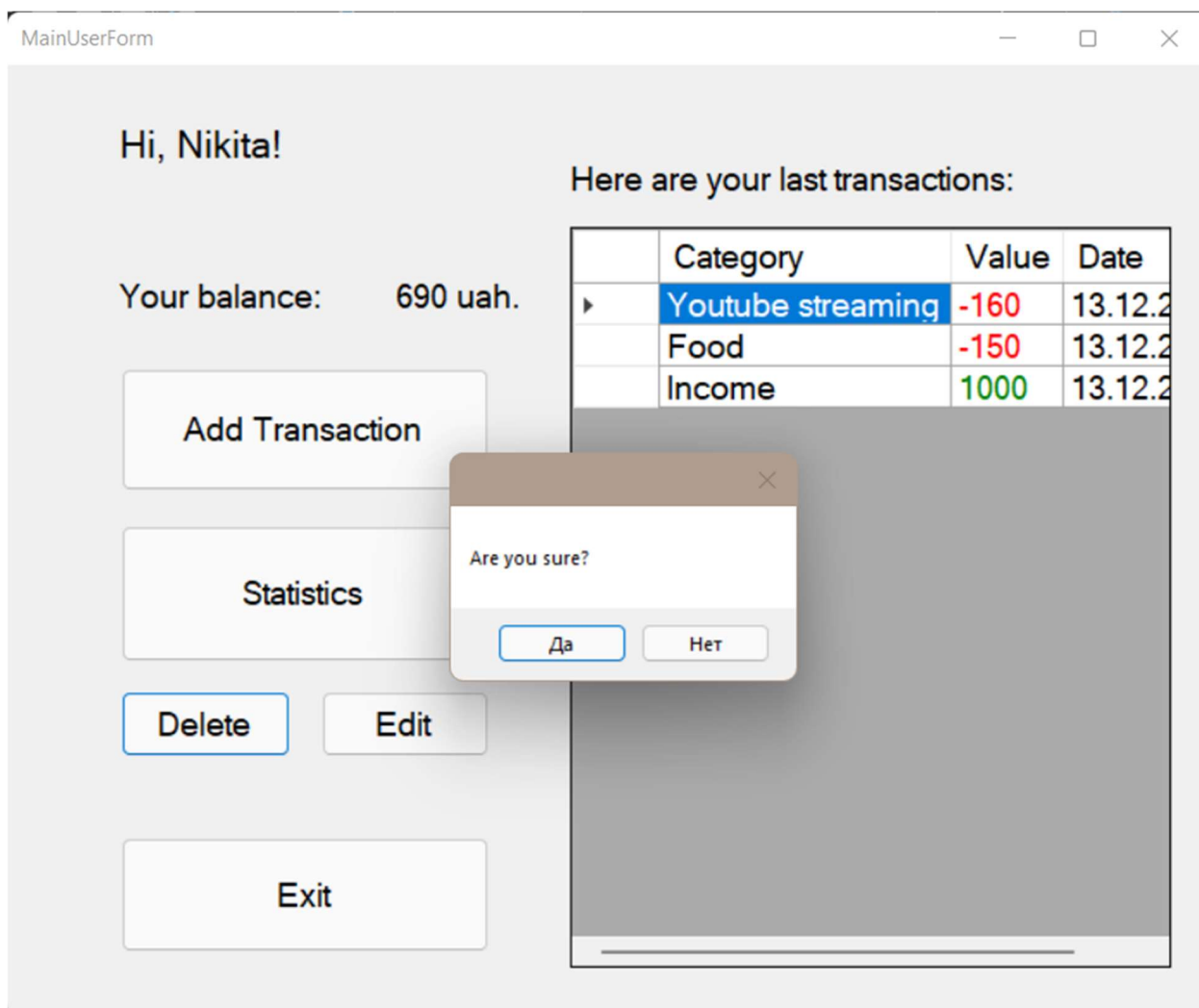


Рис 2.14

MainUserForm

Hi, Nikita!

Your balance: 850 uah.

Add Transaction

Statistics

Delete Edit

Exit

Here are your last transactions:

	Category	Value	Date
►	Food	-150	13.12.2021
	Income	1000	13.12.2021

Рис 2.15

І, як приклад облікового запису, яким користувався я власноруч для контролю своїх витрат протягом останнього місяця (рис 2.16, 2.17):



MainUserForm

Hi, Nikita!

Your balance: 3366 uah.

Add Transaction

Statistics

Delete

Edit

Exit

Here are your last transactions:

	Category	Value	Date
▶	UserTransaction	-3200	12.12
	Income	20	24.11
	Food	-20	24.11
	Food	-50	24.11
	Income	10	24.11
	Income	50	24.11
	Income	100	24.11
	Food	-200	24.11
	Income	500	23.11
	Income	2500	23.11
	Income	1000	23.11
	Flowers	-346	23.11
	anime dokimakura	-1500	22.11
	AutoService	-230	22.11
	Health	-500	22.11
	Food	-300	22.11

Рис. 2.16

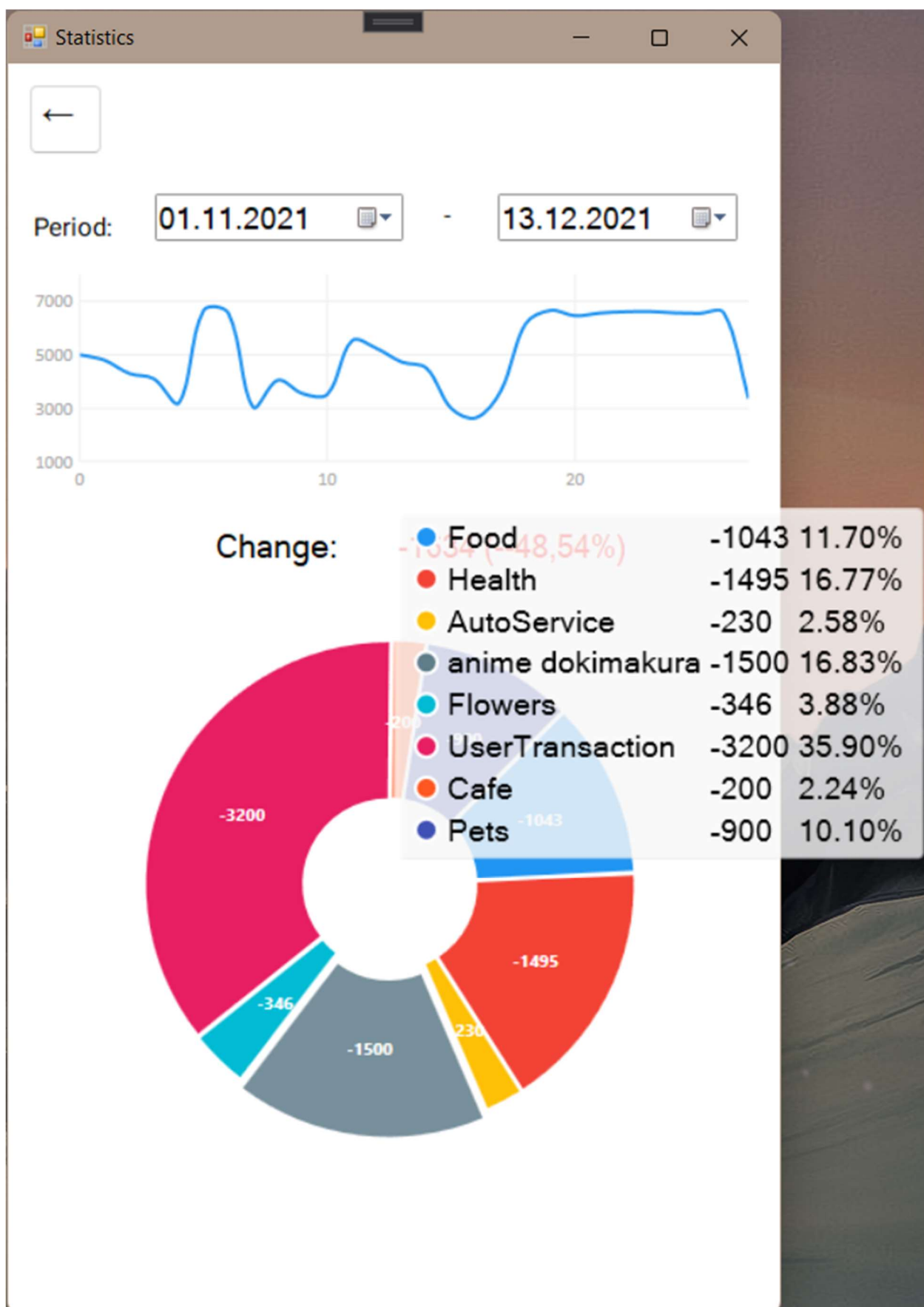


Рис. 2.17

### 2.3. Діаграма класів.

Нижче приведена діаграма класів(рис. 2.18):

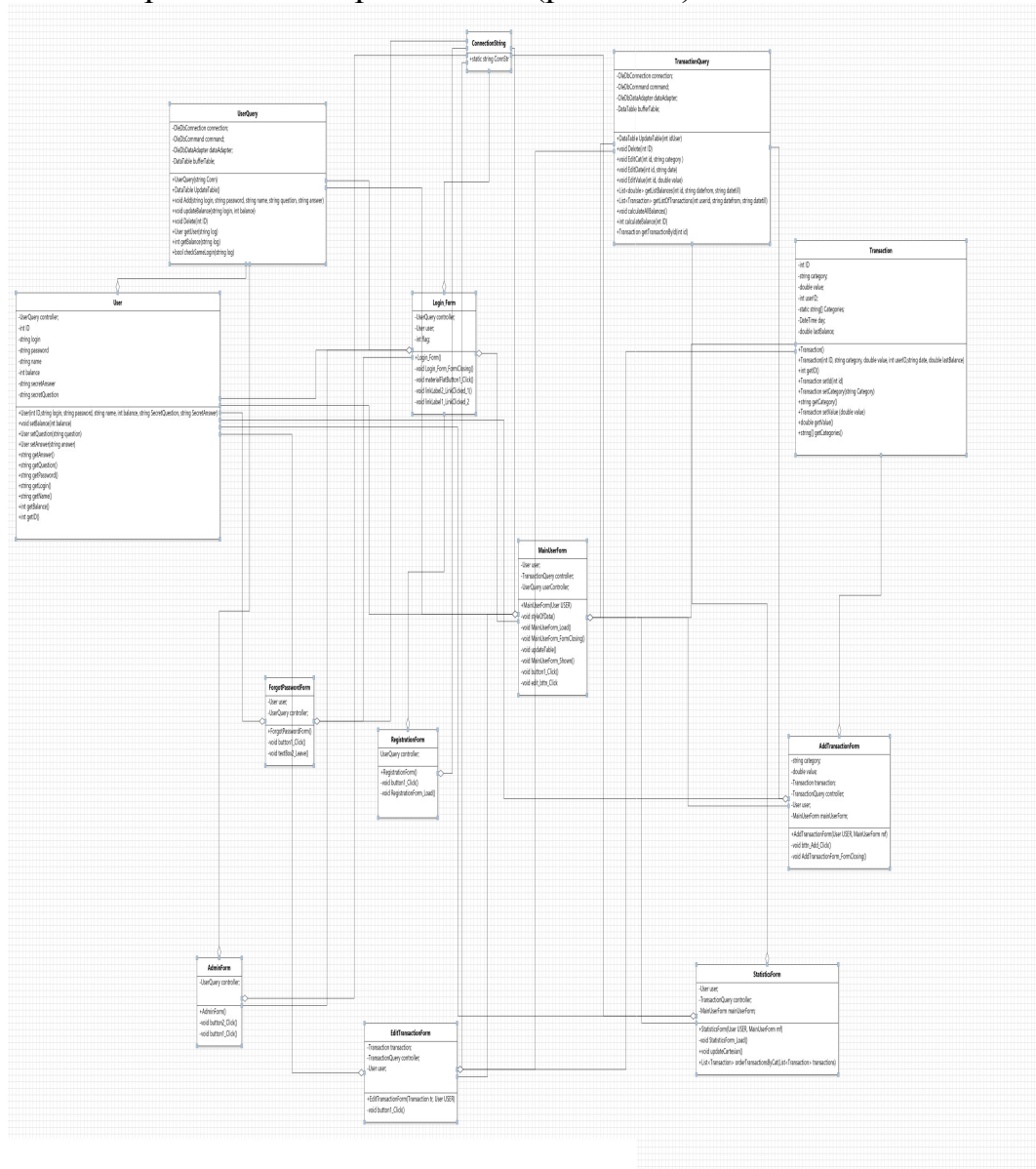


Рис. 2.18

(копія більшого розширення у додатках)

## **Висновки**

Під час розробки даного програмного продукту, я вдосконалив навички використання мови C# для розробки багатокористувацьких десктопних додатків з GUI, що використовує для комунікації з користувачем не клавіатуру, а мишу. Також навчився працювати із базами даних використовуючи бібліотеку OleDb та середовище для баз даних MS Access.

За допомогою мови програмування C# вдалося реалізувати легкий і доступний графічний інтерфейс для користувача. Основним середовищем розробки було Microsoft Visual Studio.

Також отримав багато досвіду в розробці структури моєї програми, використовуючи знання з дисципліни «ООП», та багато досвіду в написанні запитів мовою „SQL“, оскільки програма орієнтована на роботу з базою даних.