**Архітектура**

Програма буде створюватися на мові С # з використанням середовища розробки Visual Studio 2019 на платформі .NЕТ Framework 4.7.2 з використанням інтерфейсу програмування додатків Windows Forms.

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

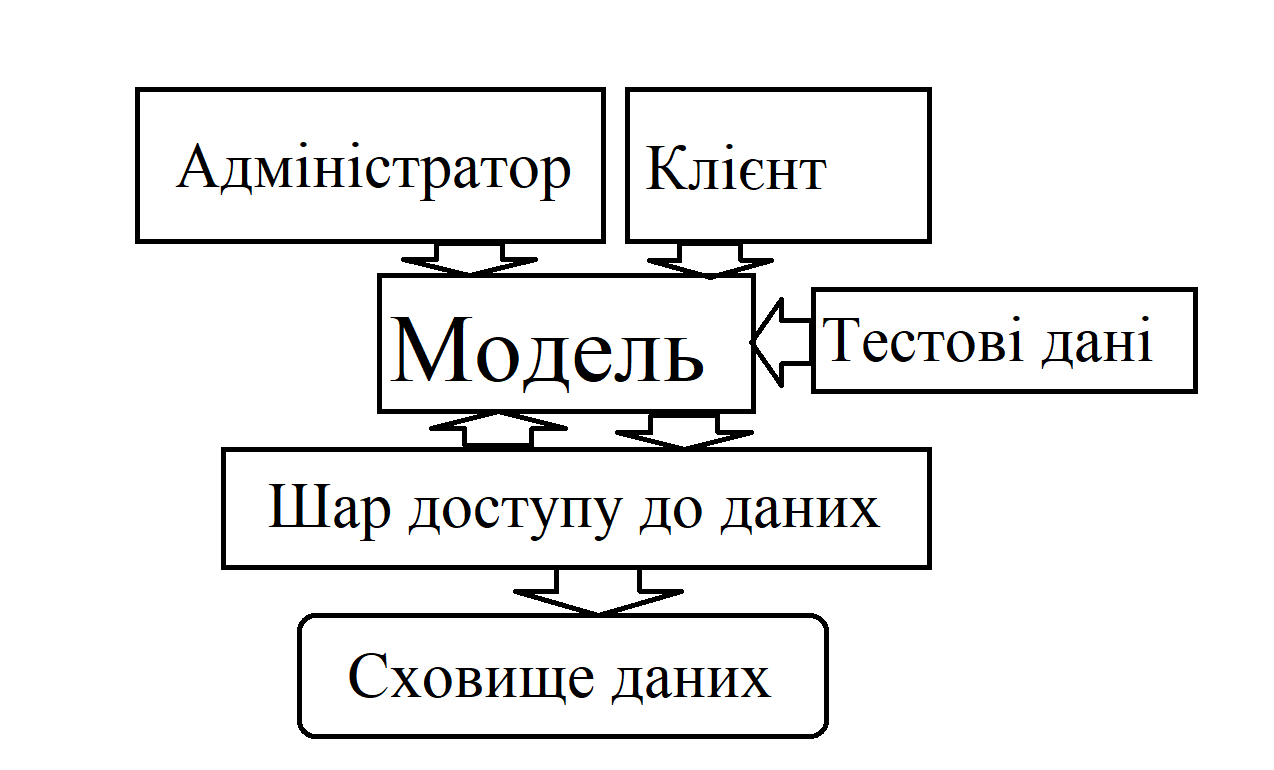
В процесі проектування розглядалися 2 платформи, з якими мова С # має найкраще взаємодію, а саме .NЕТ Core і .NЕТ Framework. Наш вибір припав на останню, через збільшеної кількості керуючих елементів і більш високого рівня розвитку платформи. На жаль, це означає що додаток буде працювати виключно на машинах з операційною системою Windows.

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

**Структура проекту**

Буде розумно поділити проект, описаний в специфікації, на два окремих додатки: для адміністратора і для клієнтів, оскільки функції цих типів користувачів між собою не перетинаються.

Крім того, для тестування програми на етапі програмування доречно створити кілька тестових даних для перевірки правильності роботи програми.

На цій підставі, можемо скласти наступну структуру проекту:

**Об'єктна модель**

**Зміст:**

Клієнт обирає бажані товари, тим самим наповнюючи свою корзину. Закінчує сеанс створенням замовлення.

Адміністратор додає нові товари, редагує вже створенні найменування, отримує звіти про наявність та продажах за певний період.

• Про товар відомо: назва, фото, ідентифікатор, наявна кількість, кількість придбань, одиниця виміру.

• Про клієнта відомо - ім'я, пароль, історія купівель.

• Про адміністратора відомо - ім'я, пароль.

• Кошик - це колекція покупок.

• Купівля - це товар + його кількість.

• Замовлення - це кошик + покупець + дата замовлення.

• Поставка - це дата + колекція порцій товару.

• Порція - це товар + його кількість.

• Магазин - це кілька колекцій: товарів, покупців, поставок, замовлень.

У моделі зручно виділити кореневої клас, який представляє модель в цілому.

Інші частини програми будуть працювати з моделлю через методи і властивості кореневого класу.

Він ніби є фасадом, за яким ховаються інші класи моделі.

У нашому прикладі очевидно, що це Магазин.

**Методи класів:**

На даному етапі основні методи магазину мають такий вигляд:

Save()

Load()

AddProduct()

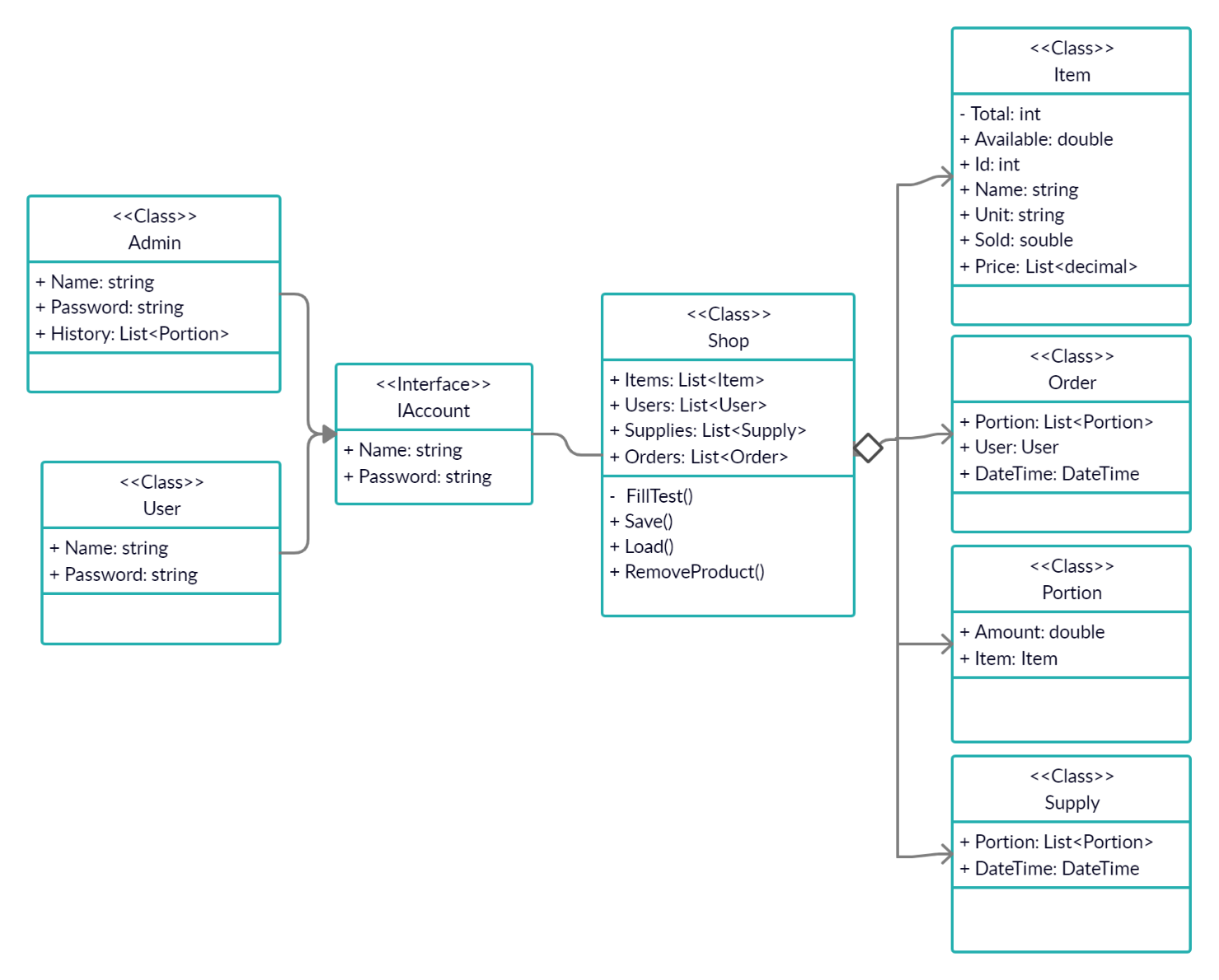
RemoveProduct()

FindProduct()

RegisterBuyer()

AddOrder()

AddSupply()

[](https://app.creately.com/diagram/iM2K4YTepd2/)**Діаграма класів**

**Сховище і доступ до даних:**

Всі дані, зазначені в роботі словом «магазин», а також «список користувачів» будуть представлені у вигляді окремих бінарних файлів в каталозі проекту. Дані будуть переводитися з текстових в потік байтів за допомогою сериализации з використанням вбудованої бібліотеки ВinaryFormatter. Файл під назвою «store.bin» розташований за адресою - «\ bin \ Debug». Захист даних в рамках роботи не розглядається.

<https://github.com/RuslanProgrammer/MyShop>