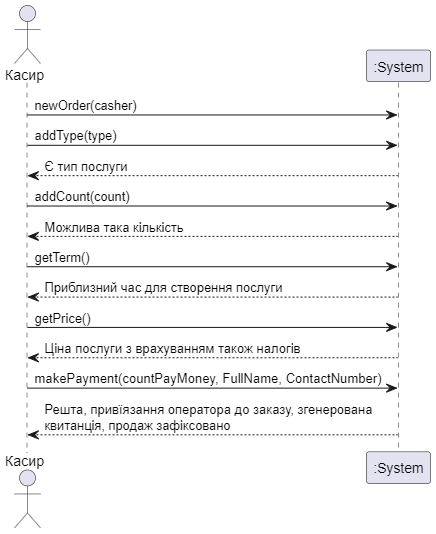
4 ДІАГРАМИ ВЗАЄМОДІЇ

**1 діаграма системних операцій для «важливого» прецедентна:**

Через аналіз пунктів прецеденту «Замовлення послуги» було отримано наступні системні операції з рисунку 1.1.



**Рисунок 1.1  - Системні операції на підставі ВВ «Замовлення послуги»**

**2 обґрунтування вибору класу – контролер:**

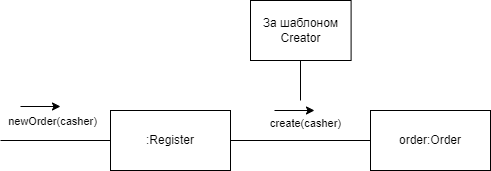
Клас Register підходить для реалізації шаблону проектування «Контролер», через взаємодію з ключовими елементами системи Order, Ticket та PickUpStation.

**3 діаграми взаємодії для кожного пункту сценарію «важливого» прецеденту:**

Через аналіз відношень класів та самих змісту пунктів першого та другого прецеденту, були сформульовані наступні діаграма на рисунку 2.1, 2.2, 2.3, 2.4, 2.5 та 2.6.

1. Проектне рішення newOrder

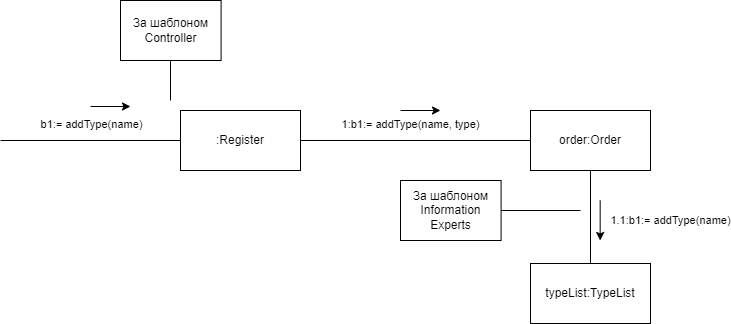
Клієнт звертається за послугую. Касир створює нове замовлення у системі. Система фіксує касира, який наддає замовлення.



**Рисунок 2.1  - Проектне рішення newOrder**

1. Проектне рішення addTypeToOrder

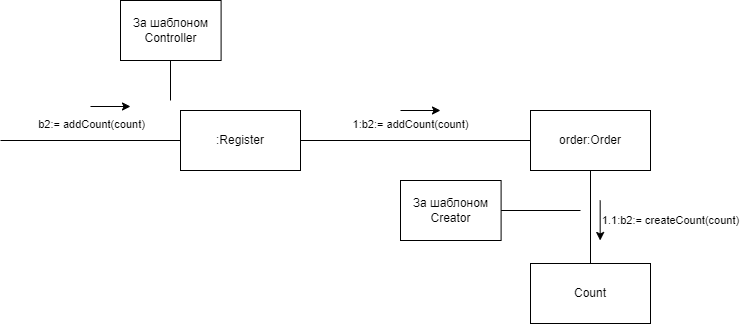
Касир запитує тип послуги. Клієнт говорить тип послуги. Касир вводить у систему сказаний тип послуги. Система підтверджує і фіксує.



**Рисунок 2.2  - Проектне рішення addTypeToOrder**

1. Проектне рішення addCount

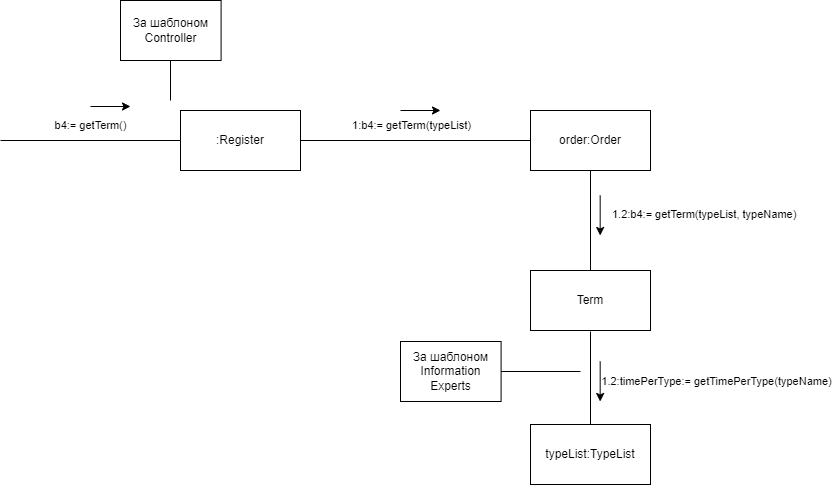
Касир запитує тип послуги. Клієнт говорить тип послуги. Касир вводить у систему сказаний тип послуги. Система підтверджує і фіксує.



**Рисунок 2.3  - Проектне рішення addCount**

1. Проектне рішення getTerm

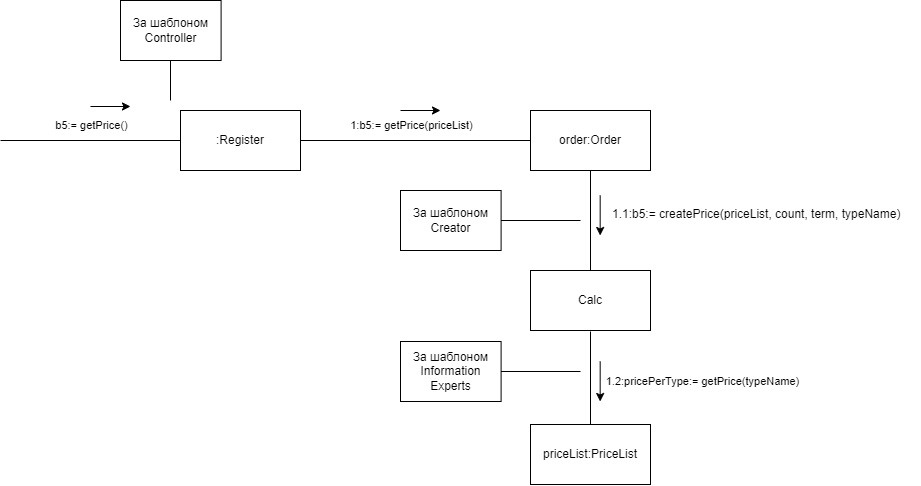
Касир отримує з системи приблизний час виконання послуги. Касир говорить клієнту цей час.



**Рисунок 2.4  - Проектне рішення getTerm**

1. Проектне рішення getPrice

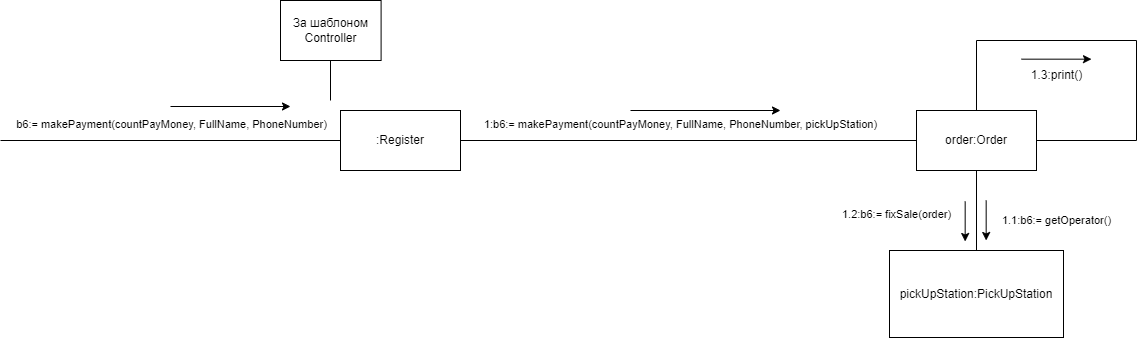
Касир отримує з системи приблизний час виконання послуги. Касир говорить клієнту цей час.



**Рисунок 2.5  - Проектне рішення getPrice**

1. Проектне рішення makePayment

Клієнт дає відповідну кількість грошей, ПІБ, контактний телефон. Касир вводить усе це у систему. Система перевіряє і фіксує замовлення, вибирає оператора, генерує квитанцію і змінює у виробничий стан замовлення.



**Рисунок 2.6  - Проектне рішення makePayment**

**4 обґрунтування прийнятого розподілу обов’язків з посиланням на шаблони проектування:**

У першому був застосований шаблон проектування Create через потребу у створені об’єкту для замовлення і взаємодії х ним за допомогою Register. У другому був застосований шаблон Controller для екземпляру з класом Register та information experts для екземпляру з типом TypeList, до яких проходять методи addType для додавання до екземпляру Order типу послуги. У третьому також використовується контролер. Використовується клас Count для створення та присвоєння order кількість послуги. У четвертому рахується час створення послуги через клас Term та за шаблоном information experts TermList, також використовується контролер Register. Для п’ятого використовується шаблон Creater через клас Calc та information experts через PriceList для отримання вартості за послугу та його розрахунок відповідно замовлення, також використовується Register з шаблоном конструктора для взаємодії з зовнішній системи, а саме ГКІ. У шостому додаються до order ПІБ, телефоний номер клієнта, а також з екземпляру класу PickUpStation, оператора та відправляється їм на виконання, після чого створюється квитанція для замовлення.