1. ДІАГРАМИ ВЗАЄМОДІЇ

4.1 діаграма системних операцій для «важливого» прецедентна:

Через аналіз пунктів прецеденту «Замовлення послуги» було отримано наступні системні операції з рисунку 3.1.

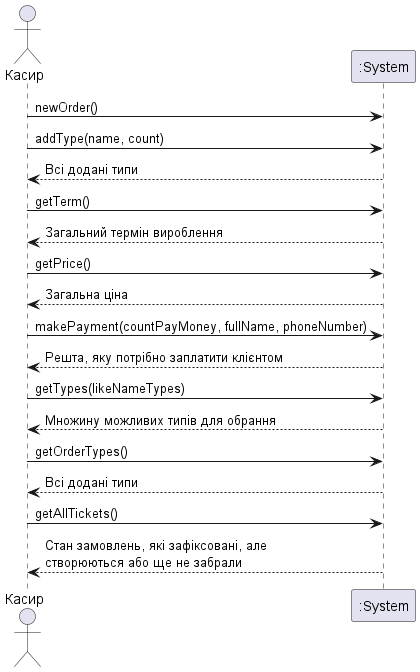


Рисунок 3.1  - Системні операції на підставі ВВ «Замовлення послуги»

4.2 обґрунтування вибору класу – контролер.

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

4.3 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 4.1, 4.2, 4.3, 4.4 та 4.5.

4.3.1 Проектне рішення newOrder

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

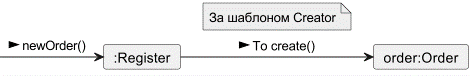


Рисунок 4.1  - Проектне рішення newOrder

4.3.2 Проектне рішення addType

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

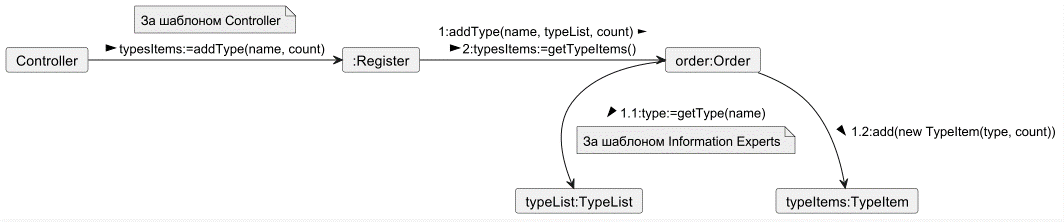


Рисунок 4.2  - Проектне рішення addType

4.3.3 Проектне рішення getTerm

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

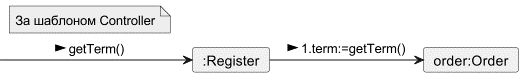


Рисунок 4.3  - Проектне рішення getTerm

4.3.4 Проектне рішення getPrice

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



Рисунок 4.4 - Проектне рішення getPrice

4.3.5 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.

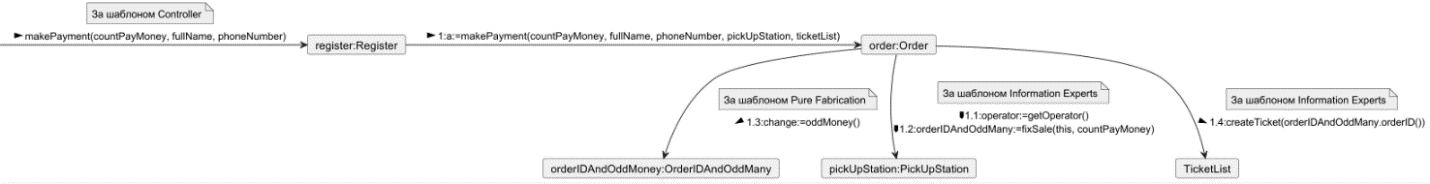


Рисунок 4.5 - Проектне рішення makePayment

4.4 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 getType для додавання до екземпляру Order типу послуги. У третьому також використовується контролер. getTerm відповідає за розрахунок терміну. У четвертому рахується загальна ціна, також використовується контролер Register. Для п’ятого використовується шаблон information experts через PriceList та PickUpStation для фіксування замовлення, отримання здачі за послугу та його розрахунок відповідно замовлення, OrderIDAndOddMany використовується за шаблоном Pure Fabrication для повертання даних з PickUpStation fixSale(), також використовується Register з шаблоном конструктора для взаємодії з зовнішній системи, а саме ГКІ.