Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра ІПІ

ЗВІТ

Лабораторної роботи № 2

з дисципліни: «Розробка мобільних застосувань під Android»

“ ДОСЛІДЖЕННЯ РОБОТИ З КОМПОНЕНТОМ FRAGMENT”

Варіант 5

Виконала *ІП-32 Зелюк Ю.М.*
(шифр, прізвище, ім'я, по батькові)

Перевірив *Орленко С. П.*
(прізвище, ім'я, по батькові)

Київ 2026

1 Мета

Мета роботи: дослідити створення та взаємодію з компонентом Фрагмент (Fragment) компоненту Діяльність та набути практичні навички з використання фрагментів для інтерфейсу користувача.

2 Завдання

Написати програму під платформу Андроїд, яка має інтерфейс, побудований з декількох фрагментів згідно варіанту. Перший фрагмент представляє з себе форму для введення даних та кнопку підтвердження («OK»), а інший фрагмент відображає результат взаємодії. Тобто другий фрагмент містить тестове поле з результатом та кнопкою «Cancel» (якщо згідно варіанту така існує, якщо ж за варіантом її немає – можете додати за власним бажанням), яка очищає або приховує (або видаляє) другий фрагмент та очищає форму введення з першого фрагменту. Зверніть увагу, що робота з фрагментами відбувається в рамках однієї Діяльності.

Примітка 1: завдання відповідає варіанту лабораторної роботи No1.

3 Виконання

1. Створення спільного сховища даних (ViewModel)

Для передачі стану замовлення між фрагментами було створено Java-клас FlowerViewModel, що наслідує ViewModel. Це дозволяє зберігати дані замовлення (назву, колір, ціну) незалежно від життєвого циклу окремих фрагментів та забезпечує доступ до них з обох екранів через MutableLiveData.

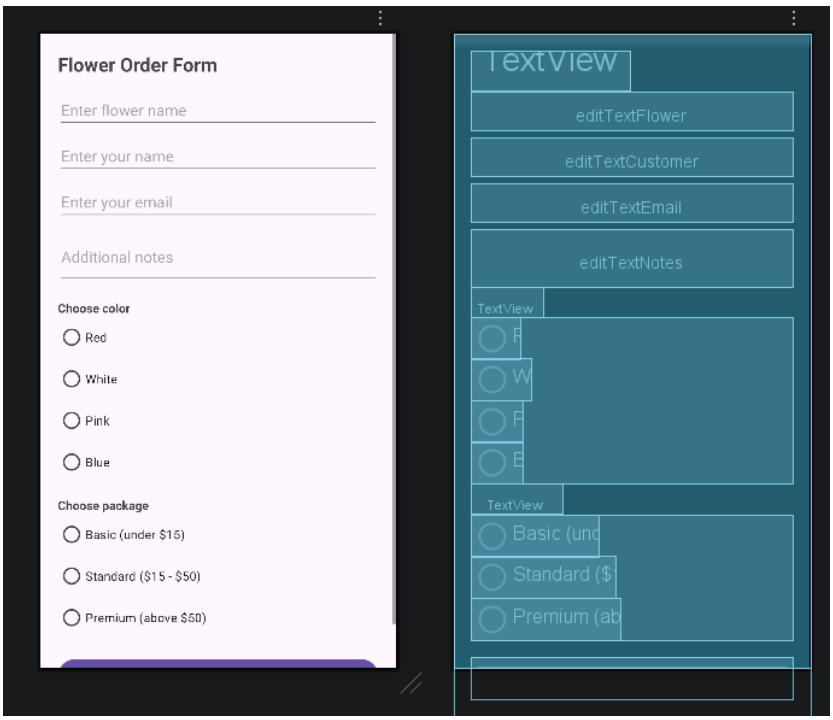
2. Розробка інтерфейсу фрагментів (Layout)

Було спроєктовано два окремих макети XML:

fragment_first.xml — містить форму замовлення з ЛР№1 (поле введення та радіо-кнопки для вибору кольору/ціни).

fragment_second.xml — містить поле для відображення результату та кнопку «Cancel» для очищення та повернення назад.

Головний файл activity_main.xml було змінено на контейнер FragmentContainerView для динамічної заміни фрагментів.



3. Реалізація логіки фрагментів та навігації

У класі FirstFragment.java реалізовано зчитування даних та їх валідацію. При натисканні кнопки «OK» дані записуються у ViewModel, а FragmentManager здійснює транзакцію replace() для переходу до другого екрана . У класі SecondFragment.java реалізовано підписку (observe) на дані моделі та логіку кнопки «Cancel», яка очищує ViewModel та повертає користувача до попереднього фрагмента через popBackStack().

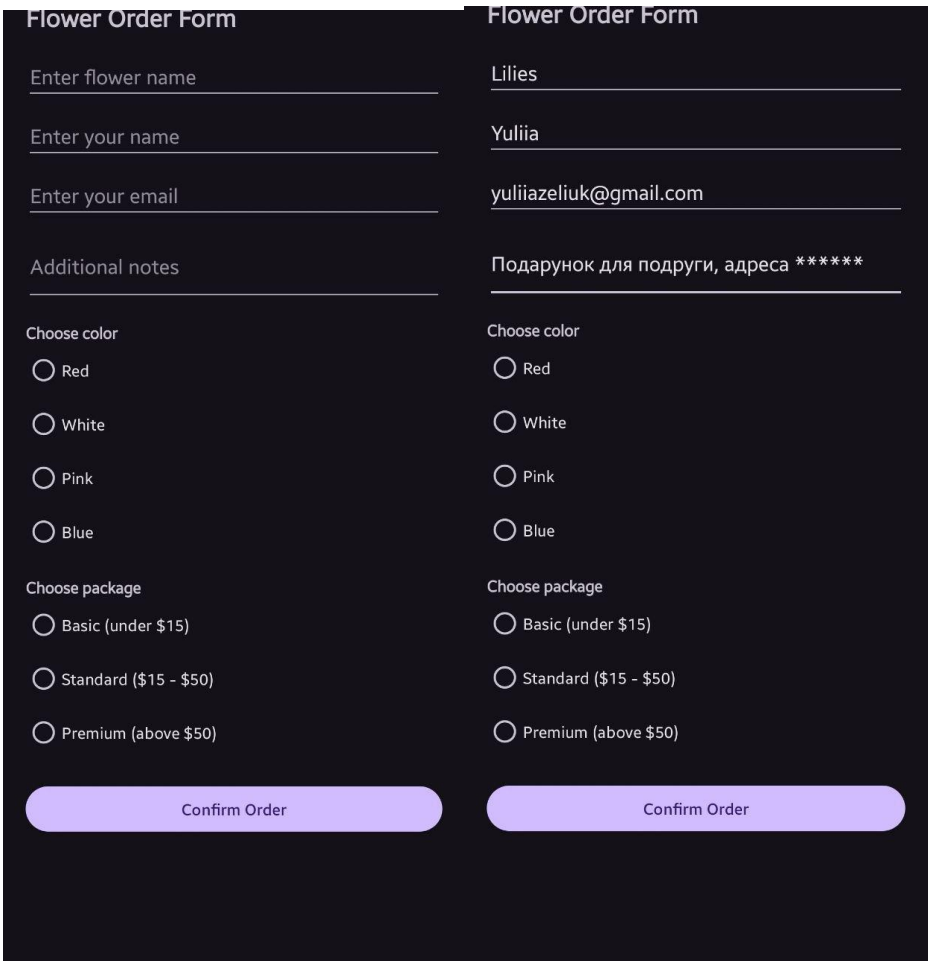
4. Тестування роботи застосунку на емуляторі

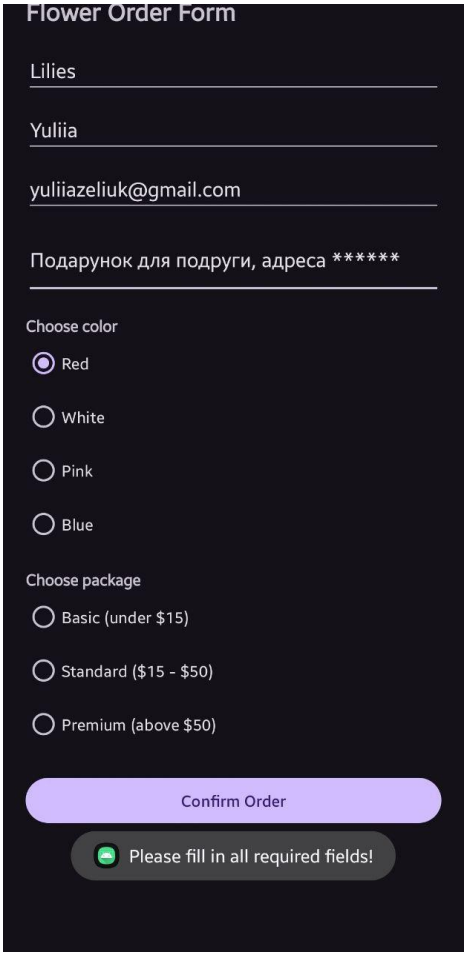
Під час тестування перевірено:

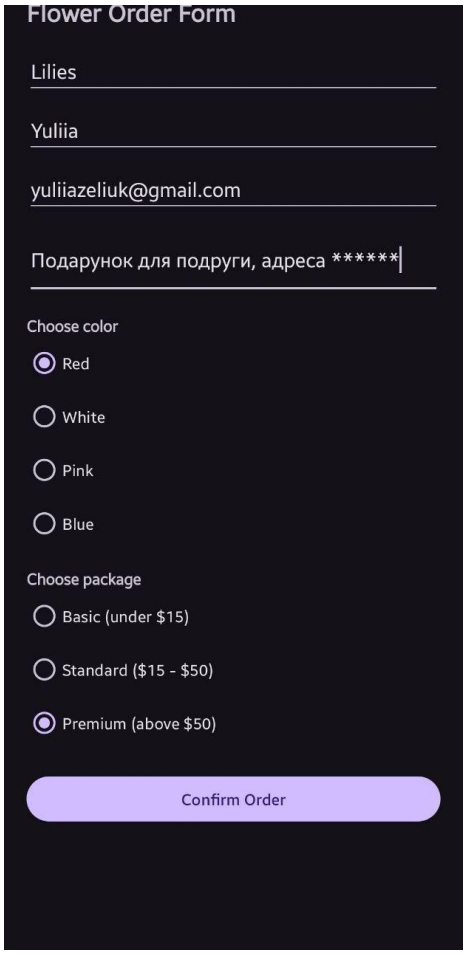
Коректність передачі введених даних між екранами.

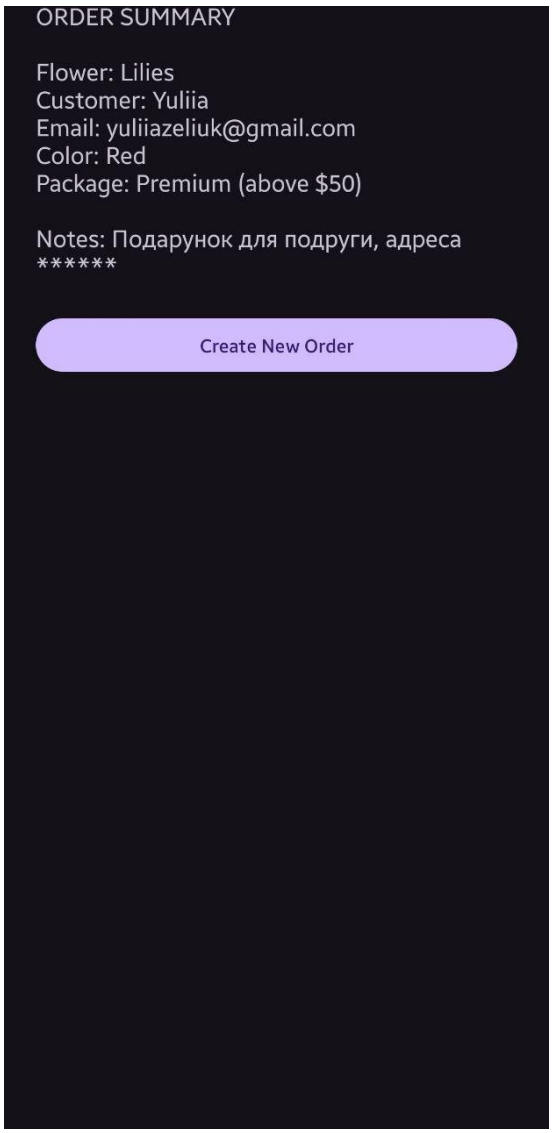
Роботукнопки «Cancel»: повне очищення форми введення після повернення з другого фрагмента.

Збереження стану даних при зміні орієнтації екрана завдяки використанню ViewModel.

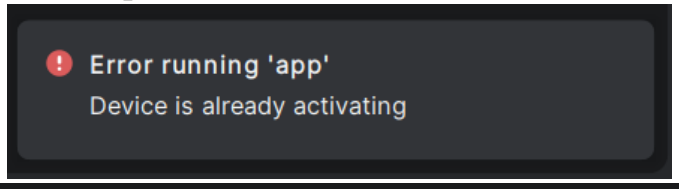

Flower Order Form
Enter flower name
Enter your name
Enter your email
Additional notes
Choose color
Red
White
Pink
Blue
Choose package
Basic (under $15)
Standard ($15 - $50)
Premium (above $50)
Confirm Order
Flower Order Form
Lilies
Yuliia
yuliiazeliuk@gmail.com
Подарунок для подруги, адреса ******
Choose color
Red
White
Pink
Blue
Choose package
Basic (under $15)
Standard ($15 - $50)
Premium (above $50)
Confirm Order

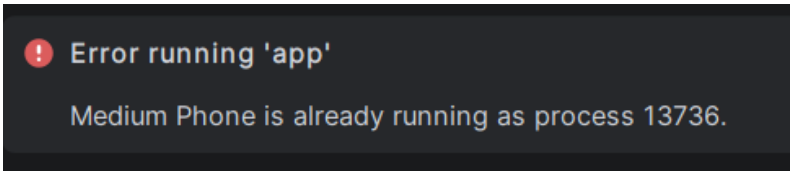

Flower Order Form
Lilies
Yuliia
yuliiazeliuk@gmail.com
Подарунок для подруги, адреса ******
Choose color
Red
White
Pink
Blue
Choose package
Basic (under $15)
Standard ($15 - $50)
Premium (above $50)
Confirm Order
Please fill in all required fields!


Flower Order Form
Lilies
Yuliia
yuliiazeliuk@gmail.com
Подарунок для подруги, адреса ******
Choose color
Red
White
Pink
Blue
Choose package
Basic (under $15)
Standard ($15 - $50)
Premium (above $50)
Confirm Order



Замість стандартного емулятора, був використаний мій телефон(Samsung A34) підключений через USB-кабель у режимі USB Debugging (Developer options). Таким чином, проблеми з віртуальним телефоном були повністю виключені, пришвидшене тестування та робота з застосунком стала зручніша. Під час тестування підтверджено, що ViewModel стабільно утримує дані про замовлення квітів навіть при зміні орієнтації екрана на реальному девайсі. Нижче наведені приклади помилок в Android Studio, були вирішені видаленням файлів .lock в файлах програми та шляхом примусового завершення процесів adb.exe і gradle-daemon через Диспетчер завдань.





ВИСНОВОК

У ході роботи було опановано використання компонентів Fragment для створення модульного інтерфейсу. Реалізовано розділення програми на InputFragment та OutputFragment в межах однієї Activity. Взаємодія між ними та збереження стану замовлення організовано через ViewModel, що забезпечує стабільність даних при зміні конфігурації пристрою. Тестування на фізичному пристрої Samsung A34 підтвердило ефективність обраної архітектури.

Відповіді на контрольні питання:

1. Призначення та можливості компоненту Фрагмент

Фрагмент - це модульна частина інтерфейсу в межах однієї Activity. Він дозволяє створювати гнучкі та адаптивні дизайни, які легко підлаштовуються під різні розміри екранів, а також забезпечує можливість повторного використання одного й того ж компонента в різних частинах застосунку.

2. Життєвий цикл компонента Фрагмент

Життєвий цикл фрагмента тісно пов'язаний з Activity, але має додаткові етапи: onAttach() (приєднання до Activity), onCreateView() (створення розмітки), onDestroyView() (знищення графічних елементів) та onDetach() (від'єднання). Основні активні стани схожі на Activity: onStart, onResume, onPause, onStop.

3. Способи створення компонента Фрагмент

Фрагмент можна створити двома способами:

Статично: прописавши його безпосередньо у файлі розмітки XML (activity_main.xml) за допомогою тегу <androidx.fragment.app.FragmentContainerView>.

Динамічно: створивши екземпляр класу фрагмента в Java-коді та додавши його в контейнер під час виконання програми.

4. Способи управління компонентом Фрагмент

Управління фрагментами здійснюється за допомогою FragmentManager та FragmentTransaction. Основні операції включають: add() (додати), replace()

(замінити), remove() (видалити). Для підтримки навігації кнопкою «Назад» використовується метод addToBackStack().

5. Способи взаємодії між Фрагментами

Основним і рекомендованим способом взаємодії є використання спільної ViewModel, де дані зберігаються в LiveData. Також можлива взаємодія через Fragment Result API, використання спільних інтерфейсів (Callback) або передача параметрів через Bundle (Arguments) при створенні фрагмента.

6. Поняття системи, малої системи та мобільної платформи

Система - сукупність взаємопов'язаних компонентів, що працюють як єдине ціле.

Мала система — це підсистема з обмеженим набором функцій.

Мобільна платформа - програмне середовище (наприклад, Android або iOS), яке надає інструменти та API для роботи апаратного забезпечення та запуску мобільних застосунків.

7. Типи мобільних застосунків

Нативні: розроблені під конкретну платформу (Java/Kotlin для Android).

Веб-застосунки: працюють у браузері і не потребують встановлення.

Гібридні: поєднують веб-технології (HTML/CSS/JS) з нативним контейнером для доступу до функцій телефона.

8. Середовища розробки мобільних застосунків

Основним середовищем для Android є Android Studio, яке базується на IntelliJ IDEA. Воно надає редактор коду, інструменти для дизайну інтерфейсу, емулятори, систему збирання проєкту Gradle та засоби налагодження (Logcat, Device Manager).

9. Класифікація мобільних платформ

Платформи класифікують за типом коду (відкриті як Android, закриті як iOS), за архітектурою ядра (Linux-подібні) та за методами розповсюдження (через офіційні магазини Google Play або App Store).