

**ЛАБОРАТОРНА РОБОТА № 1**  
**ДОСЛІДЖЕННЯ РОБОТИ З ЕЛЕМЕНТАМИ КЕРУВАННЯ**

**Виконав:** Група ІІІ-21 Гарасимів Віктор Андрійович

**Мета роботи:** дослідити створення простого застосунку під платформу Андроїд та набути практичні навички з використання елементів керування інтерфейсу, мов програмування Java чи Kotlin.

**ЗАВДАННЯ**

Написати програму під платформу Андроїд, яка має інтерфейс для введення або/та вибору даних згідно варіанту (таблиця) і відображає результат взаємодії з цим інтерфейсом у деяке текстове поле цього інтерфейсу. Передбачити наступне: якщо не всі дані введені або обрані, а користувач натискає кнопку для отримання результату, то відобразити вікно, що спливає, з повідомленням завершити введення всіх даних.

*Примітки:* варіант завдання обирається за списком слухачів дисципліни (загальним, де 169 студентів, 18-й за списком обирає 1-й варіант, 19-й - друге завдання і т.д.).

Також можна спробувати сформулювати своє подібне завдання для якогось практичнішого застосування (але тоді ознайомтесь з завданнями на 2 та 3 роботи).

Варіант	<i>Вікно інтерфейсу</i>
2.	Вікно містить текстове поле для введення деякого питання, дві опції (радіо-батони) для відповіді (так або ні) та кнопку «ОК». Вивести введене питання і обрану відповідь у інше текстове поле.

# ХІД ВИКОНАННЯ

## Проектування інтерфейсу

Розроблено макет додатку з використанням файлу `activity_main.xml`. Макет включає текстове поле для введення питання, групу радіо-кнопок для вибору відповіді, кнопку надсилення даних та текстове поле для виведення результату. Дизайн інтерфейсу забезпечує інтуїтивну взаємодію користувача з додатком. (рис. 1).

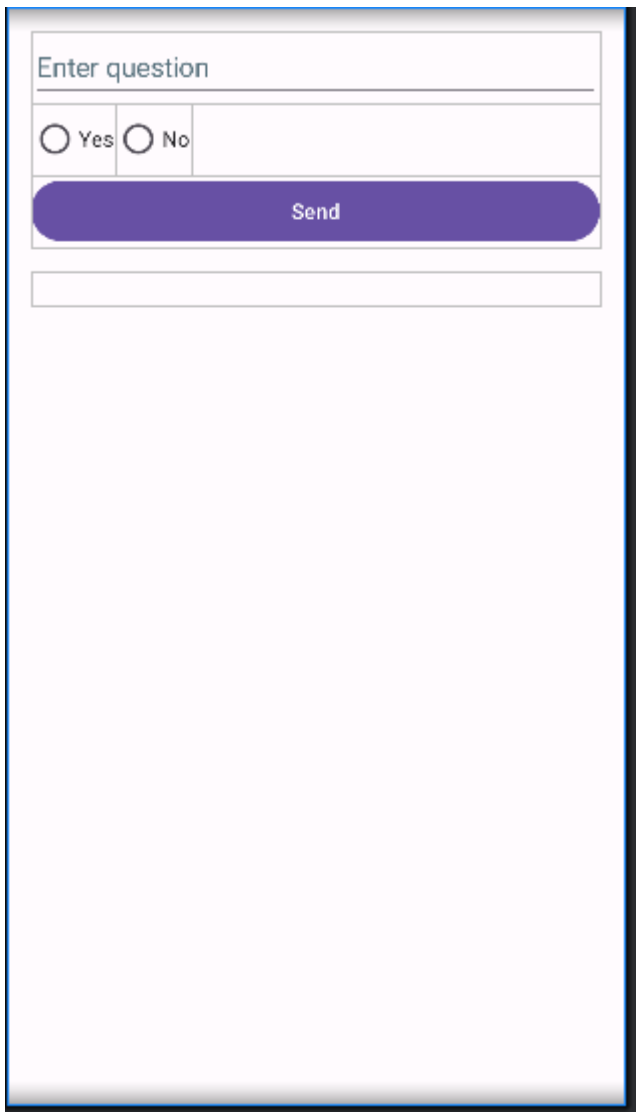


Рисунок 1 – Дизайн інтерфейсу

## Реалізація логіки додатку

У класі `MainActivity` реалізовано основну логіку роботи додатку. Для ініціалізації компонентів використано метод `findViewById()`. Розроблено обробник події натискання кнопки, який виконує валідацію введених даних, формує та виводить результат опитування. (рис. 2)

```
val submitButton = findViewById<Button>(R.id.submitButton)
val questionEditText = findViewById<EditText>(R.id.questionEditText)
val answerRadioGroup = findViewById<RadioGroup>(R.id.answerRadioGroup)
val resultTextView = findViewById<TextView>(R.id.resultTextView)
```

Рисунок 2 – метод `findViewById()`

### Механізм перевірки даних

Впроваджено систему перевірки повноти введених даних. Перевірка здійснюється шляхом аналізу наявності тексту питання та вибору радіо-кнопки. У разі неповного заповнення форми виводиться діалогове вікно з попередженням. (рис. 3)

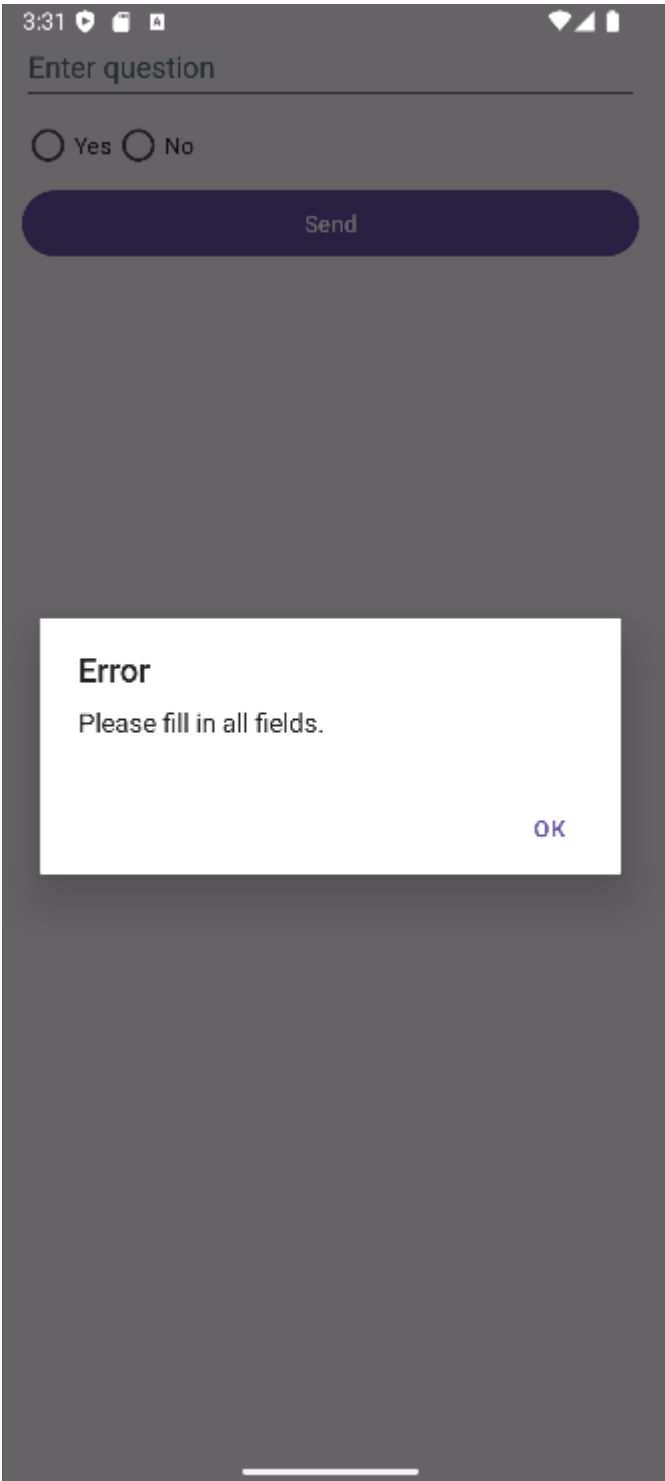


Рисунок 3 – метод findViewById()

### Результати роботи

Розроблено мобільний додаток для Android, який забезпечує повний цикл обробки даних з валідацією введення. Створено зручний та інтуїтивно зрозумілий користувацький інтерфейс. (рис.4)

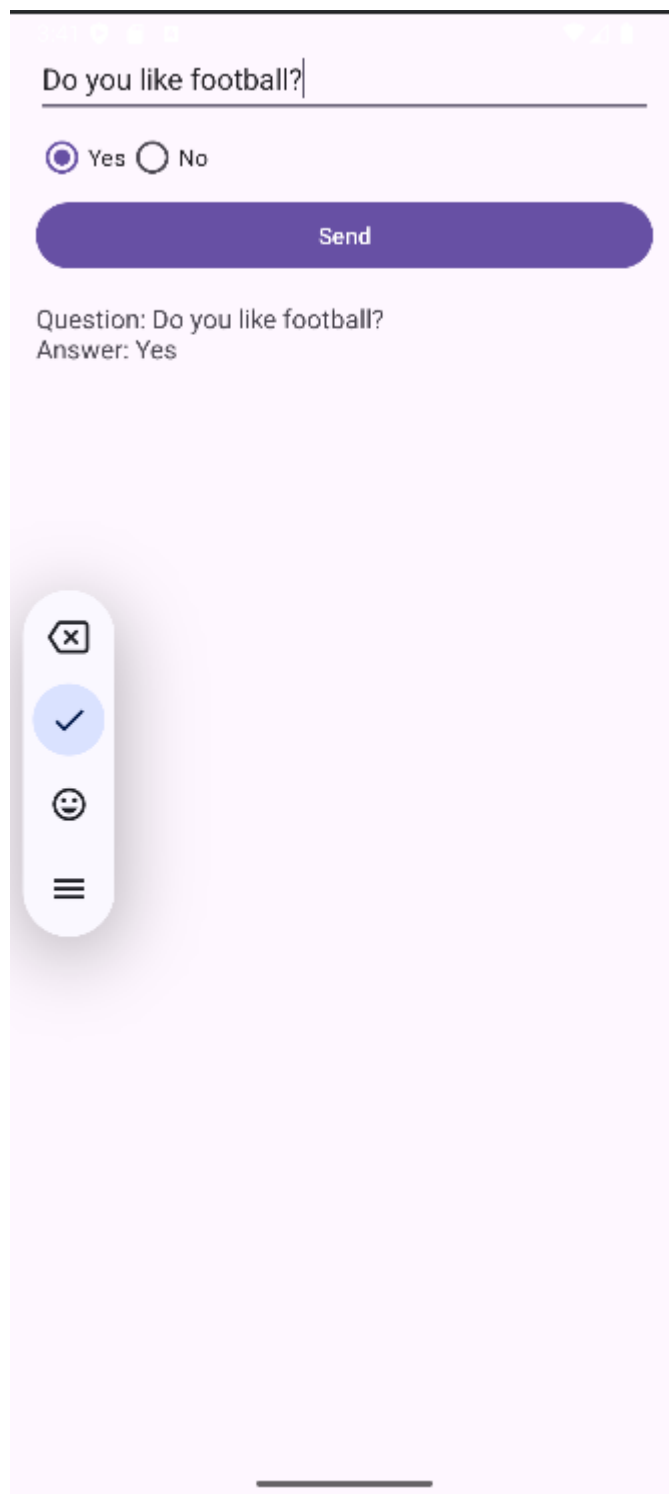


Рисунок 4 – Розроблений застосунок

### КОНТРОЛЬНІ ПИТАННЯ

1. Архітектура Android-застосунку базується на багат шаровій структурі, де ядром є Linux Kernel, а далі йдуть бібліотеки, середовище виконання Android Runtime (ART), фреймворк застосунків та власне застосунки. Вона розподіляє функціонал між різними компонентами для зручності розробки та ефективності роботи.
2. Компоненти Android-застосунку включають діяльності (Activity), служби (Service), приймачі мовлення (Broadcast Receiver) та постачальники контенту (Content Provider).

3. Життєвий цикл діяльності включає такі стани: створення (onCreate), старт (onStart), відновлення (onResume), пауза (onPause), зупинка (onStop) та знищення (onDestroy). Також є метод onRestart, який викликається при повторному запуску.
4. Життєвий цикл служби включає методи onCreate, onStartCommand (або onBind, якщо це зв'язана служба), onDestroy. Вона може працювати у фоновому режимі, виконуючи завдання без взаємодії з інтерфейсом.
5. Процеси в Android керуються системою, і кожен застосунок запускається у своєму процесі за замовчуванням. Існують також пріоритети процесів: активні, видимі, фонові діяльність, закешовані процеси.
6. Компоненти активуються за допомогою інтентів (Intent). Вони можуть бути явними, коли вказується конкретний компонент, або неявними, коли система підбирає відповідний обробник.
7. Файл маніфесту AndroidManifest.xml містить важливу інформацію про застосунок, таку як оголошення компонентів, необхідні дозволи, мінімальну версію API, теми оформлення та інші налаштування.
8. Ресурсами є зовнішні дані застосунку, такі як рядки, зображення, макети, стилі, звуки. Вони зберігаються у каталозі res і визначаються через XML-файли або у коді через R-клас.

## **ВИСНОВОК**

Під час виконання лабораторної роботи досягнуто наступних цілей: вивчено принципи розробки мобільних додатків під Android, набуто практичних навичок роботи з мовою Kotlin, опановано механізм ініціалізації компонентів через findViewById(), реалізовано логіку обробки користувацького введення.