**Міністерство освіти і науки України**

**Харківський національний економічний університет ім. С.Кузнеця**

**Кафедра інформаційних систем**

**Звіт**

з лабораторної роботи №5

з навчальної дисципліни «**Програмування для мобільних пристроїв**»

Виконала:

студентка 4 курсу групи 6.04.122.010.21.2

факультету ІТ

Бабачанах К.С.

Перевірив:

доц. Поляков А.О.

м. Харків, 2024

**Посилання на проект -** <https://github.com/babachanakh-kateryna/Android-App-Development-with-Kotlin-Beginner-to-Advanced-Lab05.git>

# **Section 17: Publishing Your App on Google**

У цій секції я ознайомилася з теорією та корисною інформацією про критичні аспекти розробки та розгортання Android-додатків. Було розглянуто важливість версії релізу APK, процес побудови APK файлів, створення облікового запису розробника Google Play та процес публікації додатка в Google Play Store.

1. **Version Code і Version Name**: Вивчення версійності APK допомогло мені зрозуміти, що кожен реліз додатка має мати унікальний код версії (Version Code), який має бути цілочисельним значенням та збільшуватися з кожним оновленням. Версія назви (Version Name) відображається користувачам та допомагає їм ідентифікувати версію додатка.
2. **Building APK Files:** Мені було показано, як згенерувати підписаний APK файл через Android Studio, що є кроком перед публікацією додатка.
3. **Google Play Developer Account:** Я навчилася створювати обліковий запис розробника в Google Play, що є необхідністю для публікації додатків на цій платформі. Були обговорені важливі кроки та вимоги для активації облікового запису та управління публікаціями.
4. **Release Your App:** Завершальною частиною було вивчення процесу публікації додатка в Google Play, включаючи налаштування політики конфіденційності, вибір країн для розгортання, оцінку контенту додатка та завантаження необхідних активів (іконки, знімки екрану).

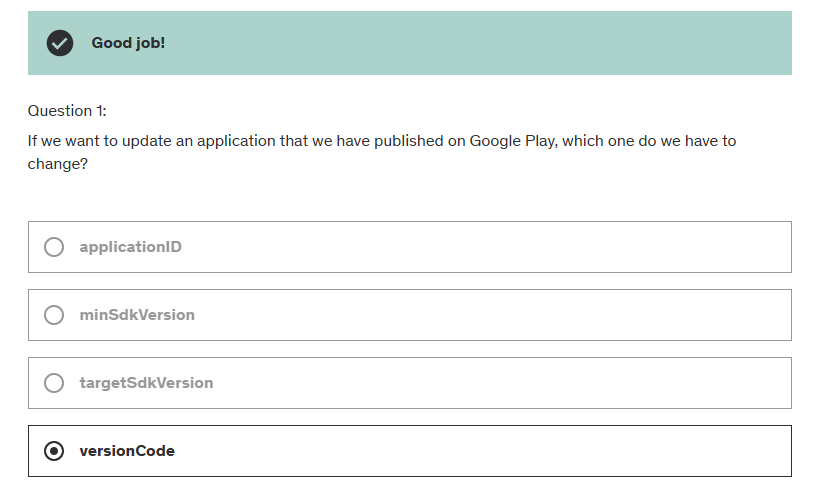


Рис. 1 **Quiz 17: quiz (q1)**

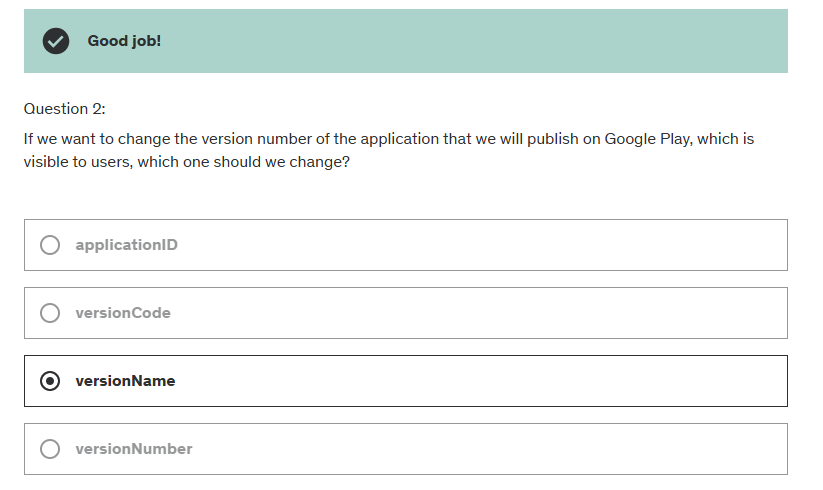


Рис. 2 **Quiz 17: quiz (q2)**

# **Section 18: Android Project 2 – Kotlin**

У цій секції потрібно було створити математичну гру.

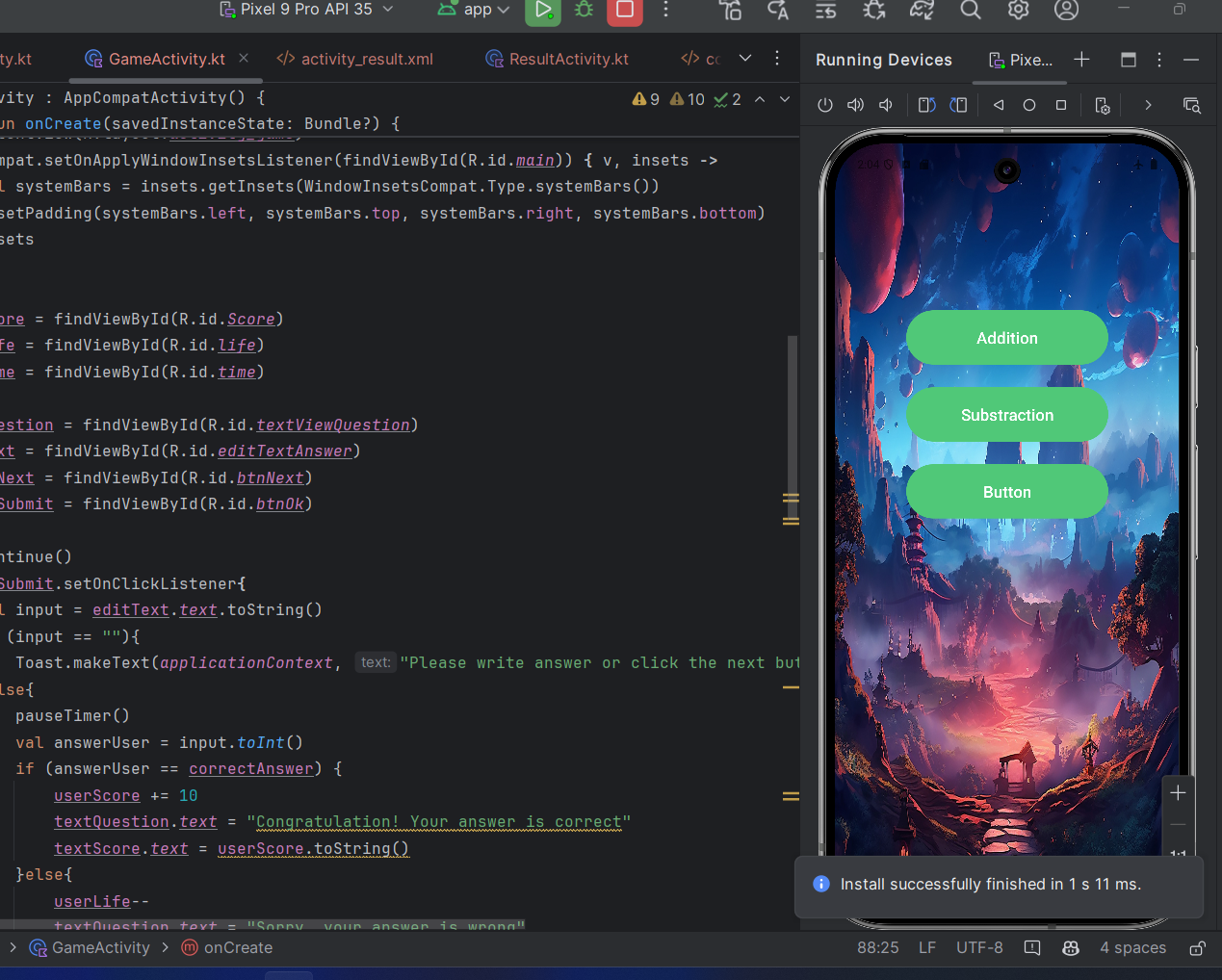


Рис. 3 Запуск застосунку

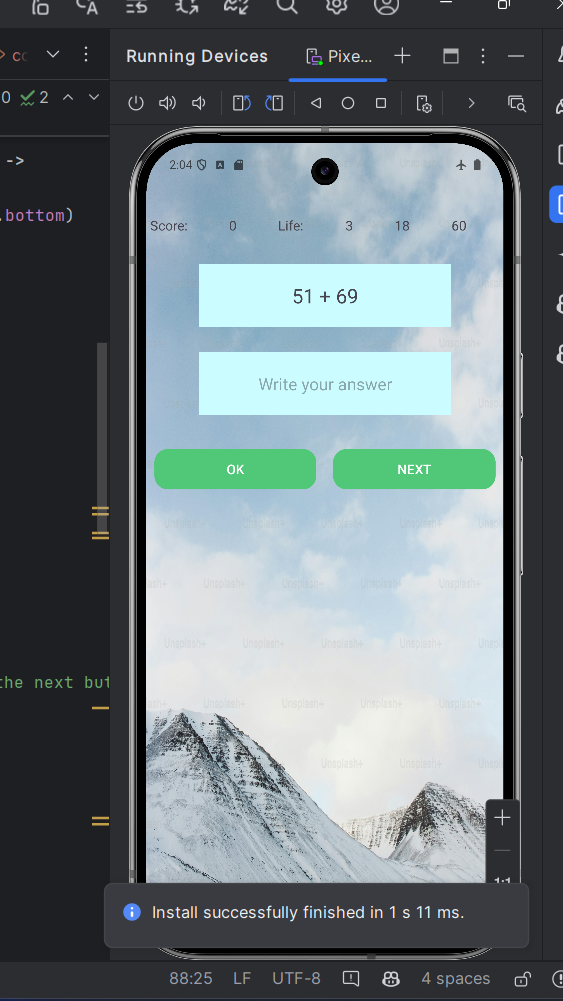


Рис. 4 **Обираємо першу кнопку**

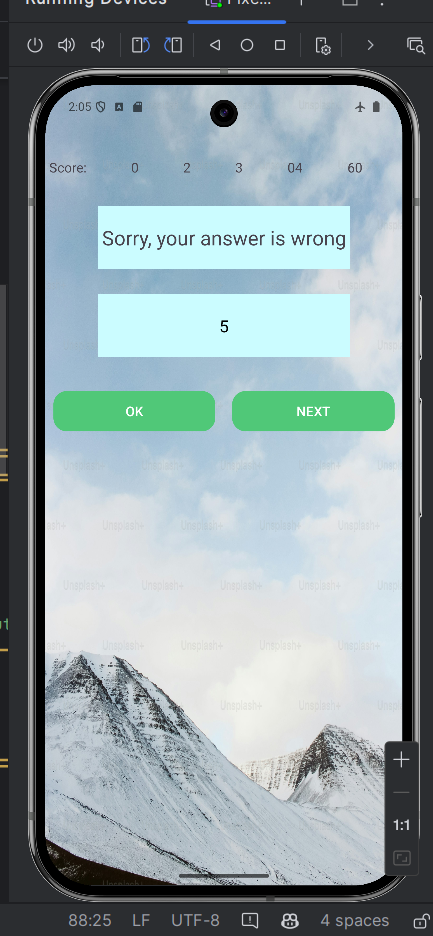


Рис. 5 **Граємо**

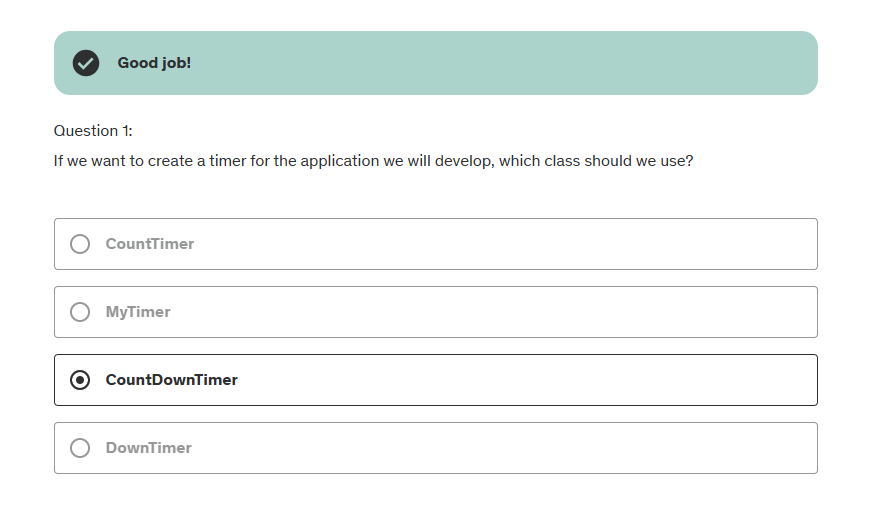
****

Рис. 6 **Quiz 18: quiz (q1)**

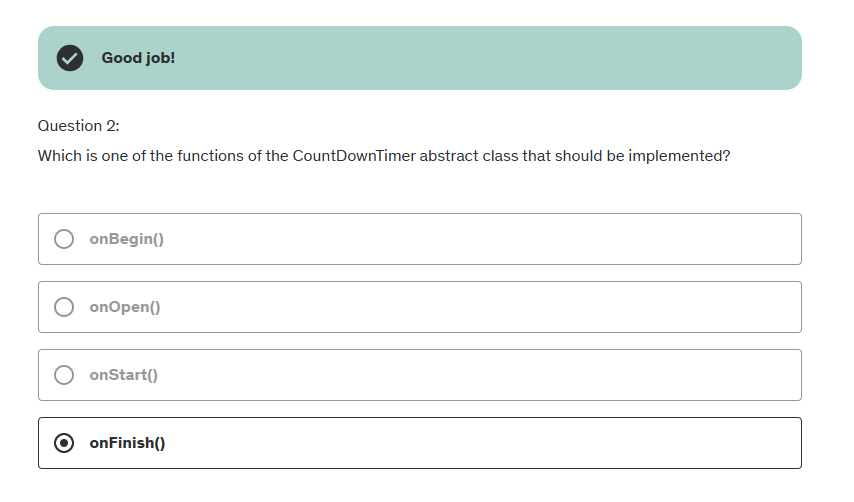
****

Рис. 7 **Quiz 18: quiz (q2)**

# **Section 19: Calculator Application**

У цій секції потрібно було створити калькулятор.



Рис. 8 **Калькулятор**

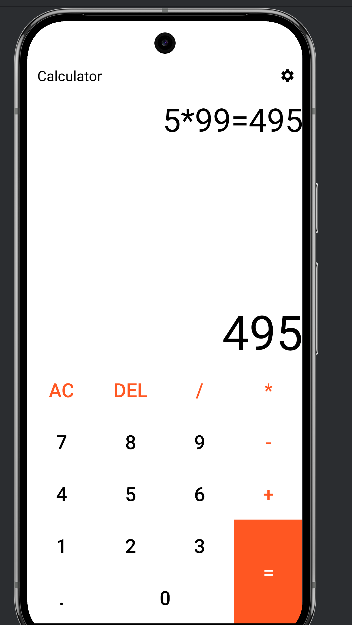


Рис. 9 **множення**

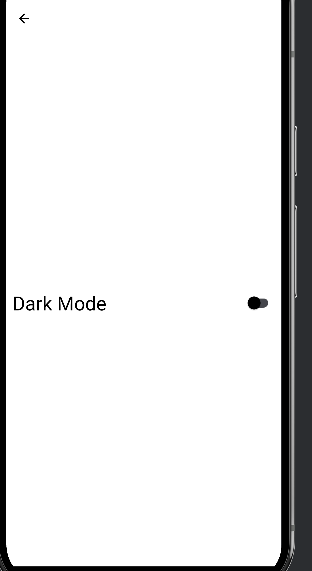


Рис. 10 **налаштування**

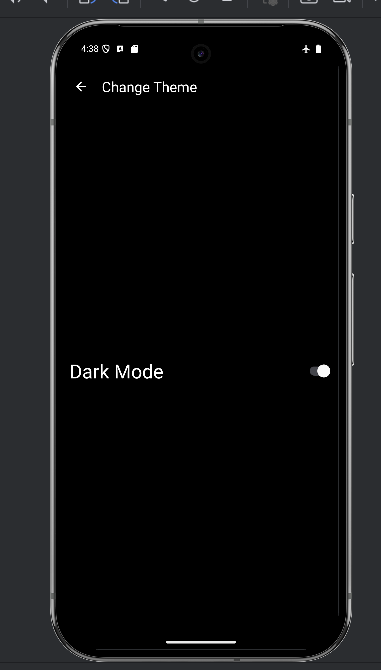


Рис. 11 **чорна тема**

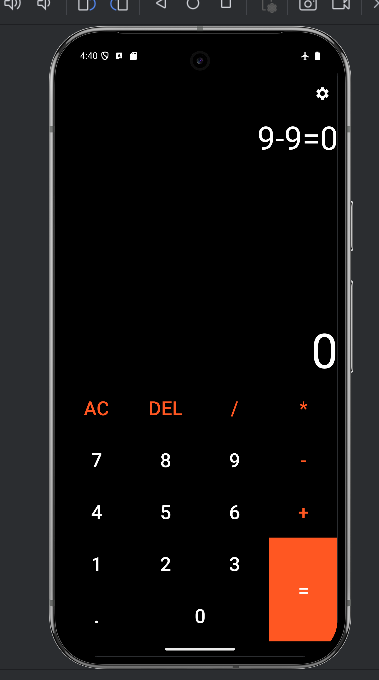


Рис. 12 **чорна тема**

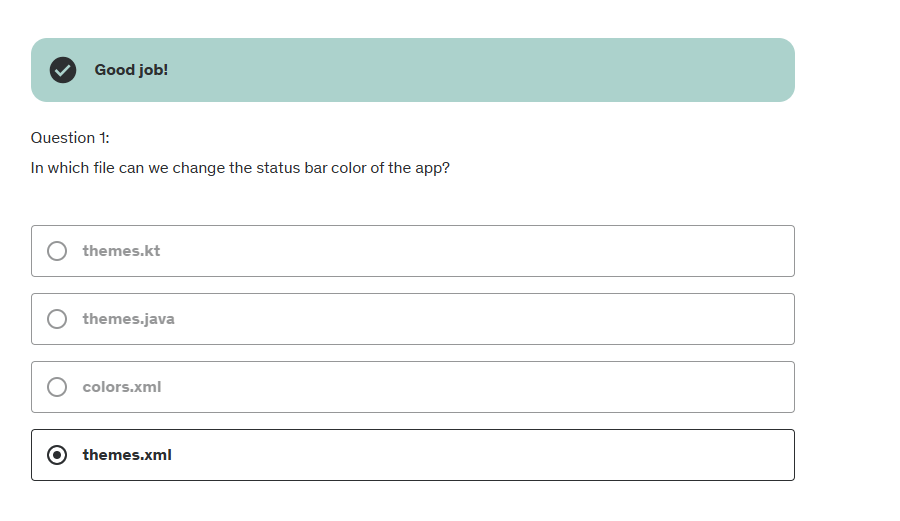


Рис. 13 **Quiz 19: Quiz**

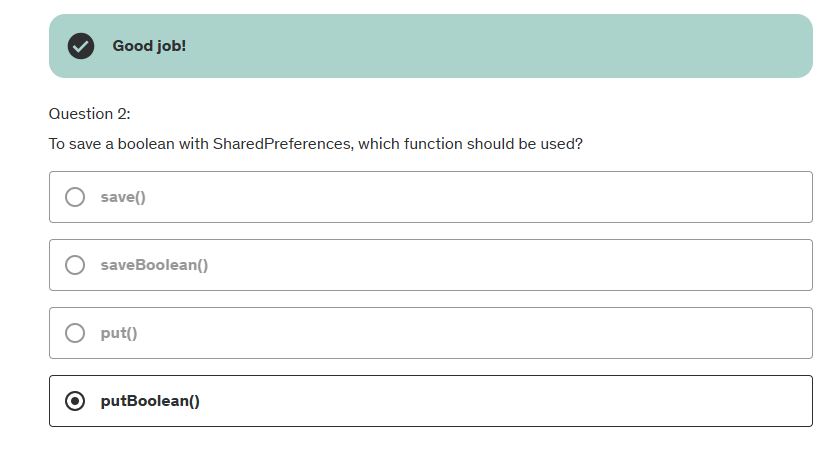


Рис. 14 **Quiz 19: Quiz**

# **Section 20: Fragment Operations in Android App Development, Kotlin**

На цьому розділі я вивчила фрагменти у розробці мобільних додатків. Вони мають кілька ключових аспектів:

1. Введення в тему фрагментів:

Фрагменти є частиною користувацького інтерфейсу в Android-додатках, які можуть бути використані повторно в різних активностях. Вони допомагають управляти більш складними інтерфейсами.

1. Загальна інформація про фрагменти:

Фрагменти мають власний життєвий цикл, можуть обробляти власні події вводу та можуть бути додані або видалені з активності динамічно.

1. Створення фрагменту:

Обговорюються основи створення фрагментів, включаючи декларування в XML та управління через Kotlin код.

1. Заміна фрагменту в активності на інший фрагмент:

Розглядається процес динамічної заміни фрагментів в рамках активності, з використанням транзакцій.

1. Фрагмент Backstack:

Пояснюється як управляти стеком навігації між фрагментами, забезпечуючи зворотній перехід між фрагментами.

1. List Fragment:

Розглядаються особливості роботи з ListFragment для створення списків в межах фрагменту, управління даними та адаптерами.

1. Dialog Fragment:

Обговорюються специфіки створення діалогових вікон як фрагментів, включаючи кастомізацію вигляду та поведінки.

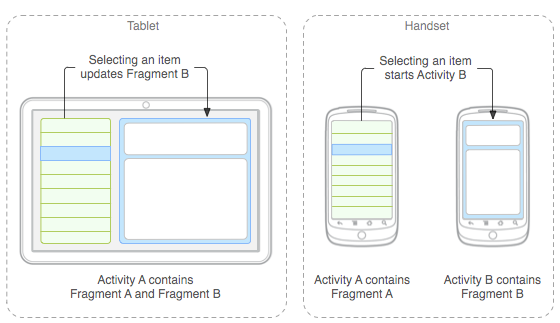


Рис. 15 **Creating and Using Fragments**

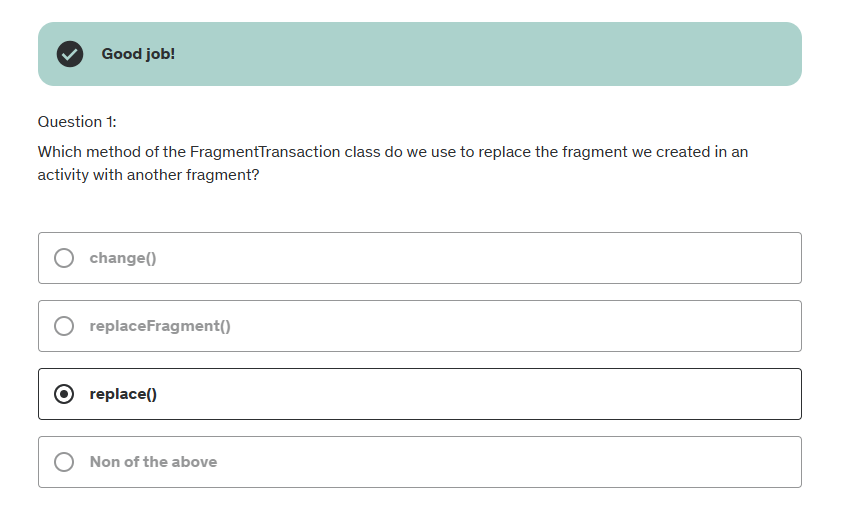


Рис. 16 **Quiz 20: quiz (q1)**

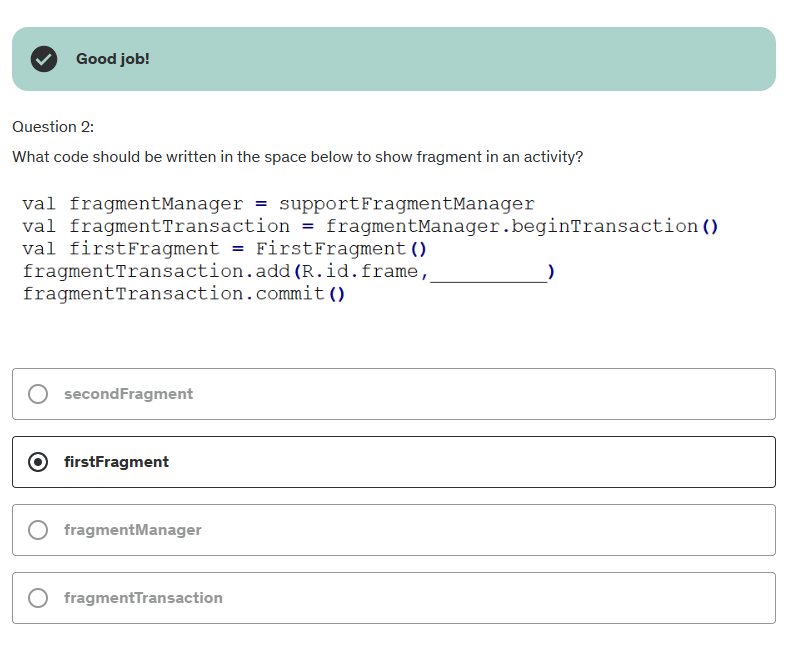


Рис. 17 **Quiz 20: quiz (q2)**

# **Section 21: Sending Data Between Screens in Mobile App Development and Android.**

Отже, у цьому розділі я розглянула та спробувала на практиці передачу даних між компонентами в Android додатках на Kotlin:

1. Передача даних між активностями:

Обговорюється процес передачі даних між активностями через Intent. Розглядаються різні методи внесення даних до Intent і відправки їх до наступної активності.

Також пояснюються підходи до отримання цих даних у цільовій активності, включаючи використання екстра (Extras) із Intent.

1. Передача даних від активності до фрагменту:

Вивчається як передавати дані від активності до фрагменту, використовуючи Bundle для капсулювання даних і метод setArguments() для передачі цього пакета фрагменту.

Розглядаються приклади як передати дані при створенні фрагмента і як ці дані можна отримати всередині фрагмента.

1. Передача даних від фрагменту до активності :

Пояснюється, як фрагмент може взаємодіяти з активністю, використовуючи різні механізми, такі як виклик методів активності напряму чи через інтерфейси.

Демонструється процес виклику методів активності з фрагмента для оновлення даних або управління активністю.

1. Передача даних між фрагментами:

Розкривається механізм комунікації між фрагментами, зазвичай через спільну активність, яка діє як посередник.

Обговорюються способи передачі даних, включно з використанням спільного ViewModel для обміну даними між фрагментами без залучення активності.

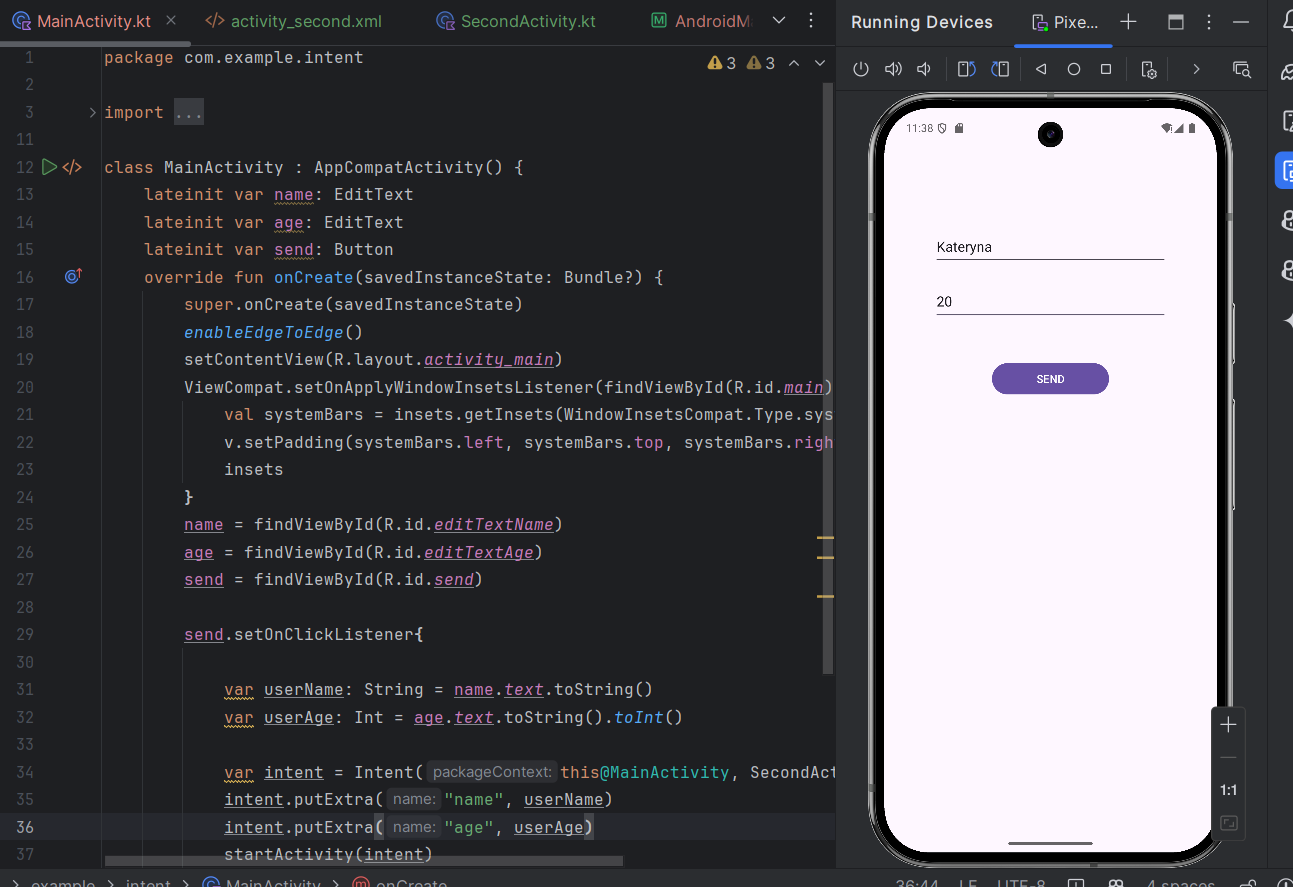


Рис. 18 **Вводжу данні та відправляю**

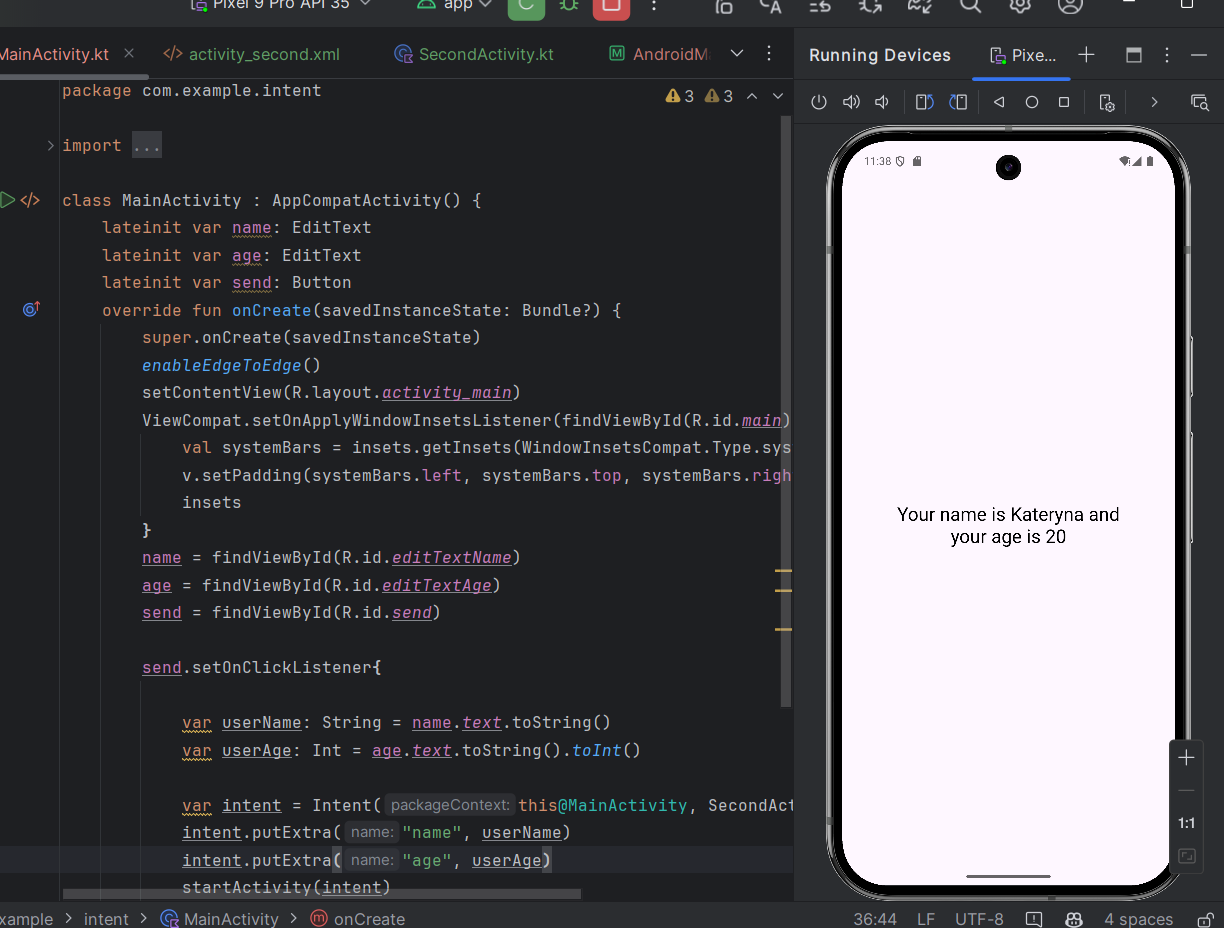


Рис. 19 **Отримаю**

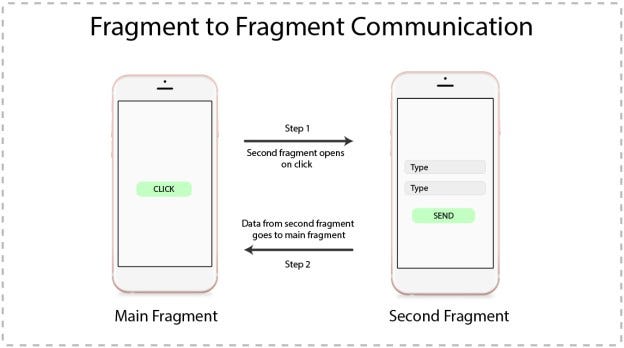


Рис. 20 **Communication between Activity and Fragments**

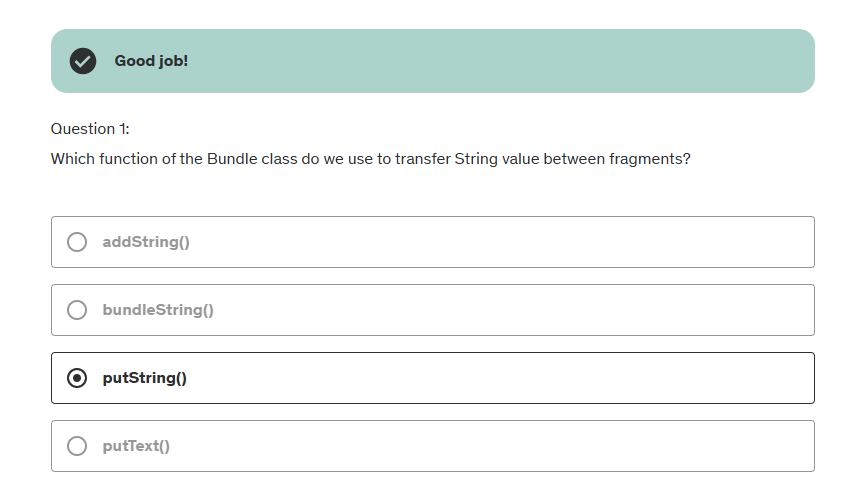


Рис. 21 **Quiz 21: quiz**

# **Section 22: Flag Quiz App**

У цій секції потрібно було створити квіз.

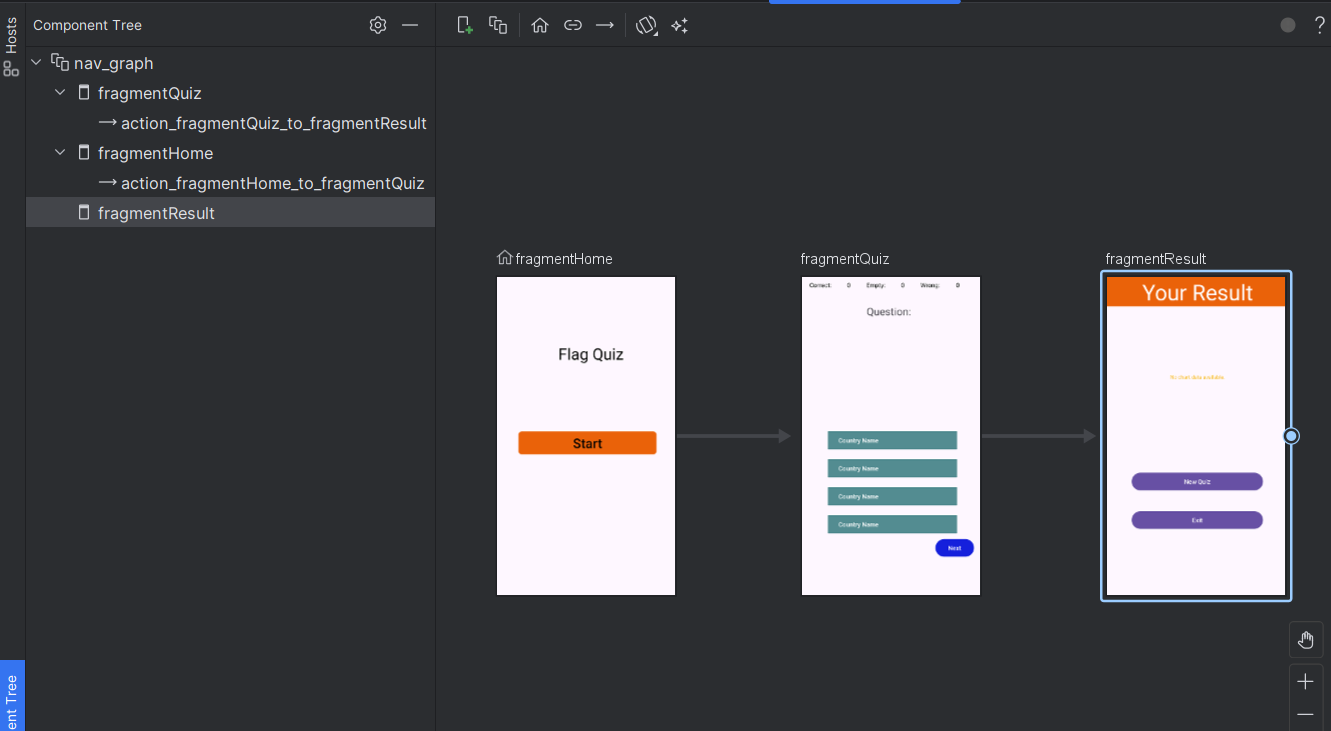


Рис. 22 **Навигация**

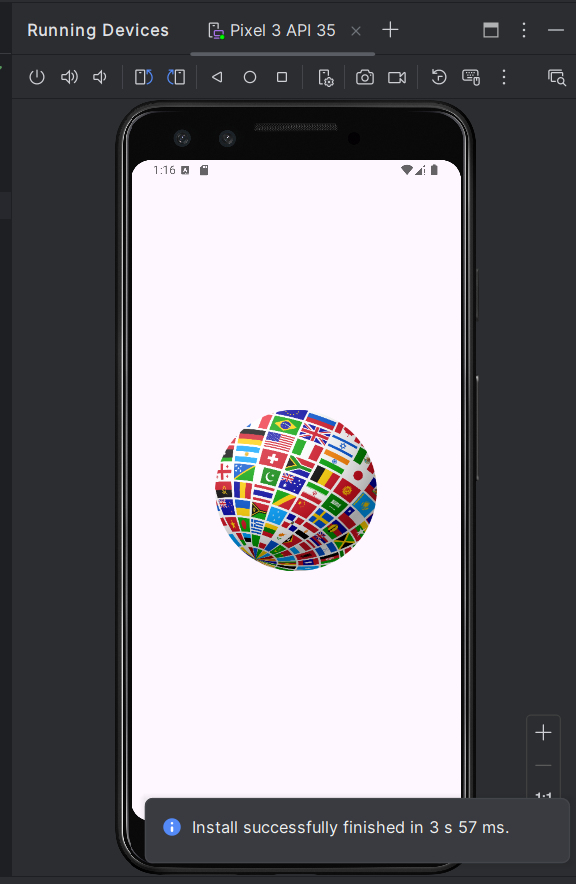


Рис. 23 **Завантаження застосунку**

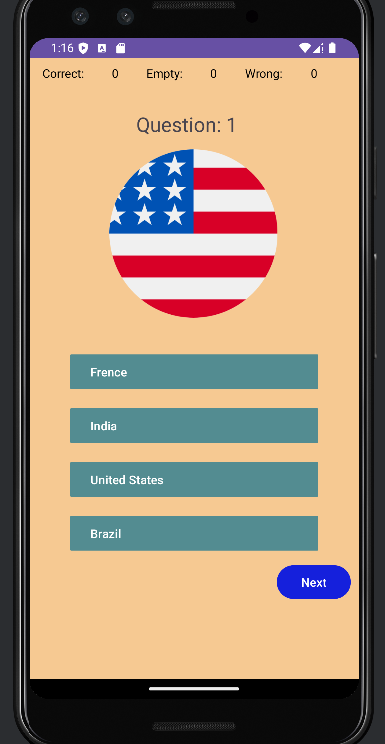


Рис. 24 **Починаємо грати**



Рис. 25 **обираю не правильну відповідь**

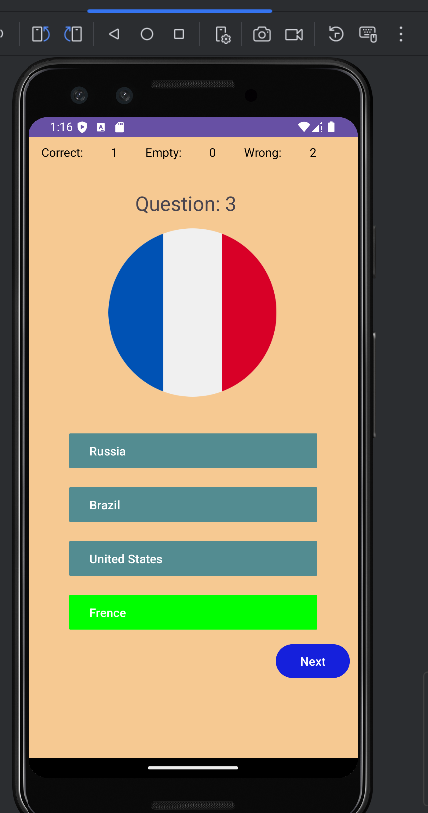


Рис. 26 **обираю правильну відповідь**

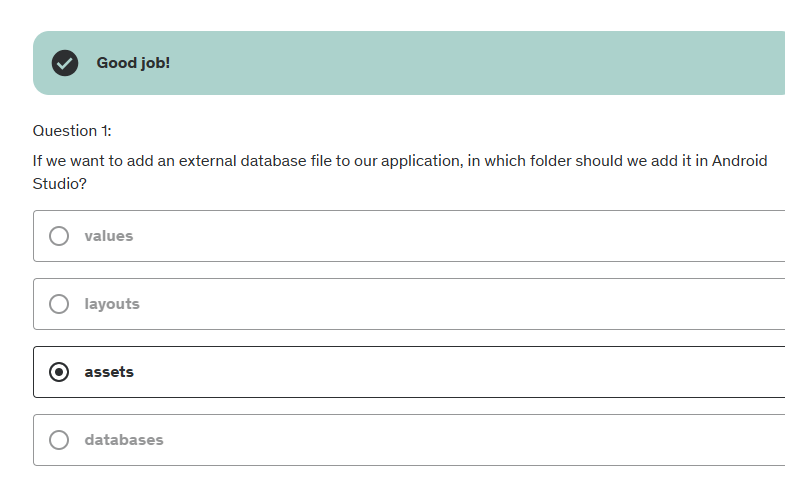


Рис. 27 **Quiz 22: Quiz (q1)**

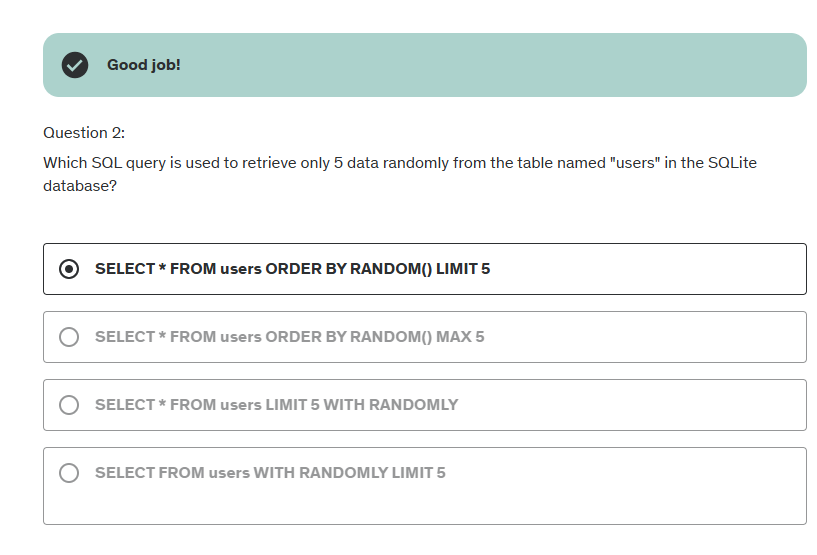


Рис. 28 **Quiz 22: Quiz(q2)**

# **Section 23: Accessing the Features of Android Device in android app.**

Цій розділ був зосереджений на доступі до різних функцій Android пристроїв, таких як відправлення SMS, електронних листів, здійснення телефонних дзвінків та перетворення мови в текст. Кожна з цих тем охоплює важливі функціональні можливості, які можуть підвищити інтерактивність вашого додатку Android. Отже, ось що я дізналась:

1. Відправлення SMS:

* Використання SmsManager для відправки текстових повідомлень.
* Необхідність дозволу SEND\_SMS.
* Обробка статусу доставки за допомогою PendingIntent.

1. Відправлення електронної пошти:

* Запуск клієнта електронної пошти з допомогою Intent.ACTION\_SEND.
* Автоматичне заповнення полів адреси, теми та тексту листа.

1. Здійснення дзвінків:

* Використання Intent.ACTION\_CALL для ініціації дзвінків.
* Потреба в дозволі CALL\_PHONE.

1. Перетворення мови в текст:

* Використання SpeechRecognizer або RecognizerIntent для розпізнавання мови.
* Управління мікрофоном і дозволом RECORD\_AUDIO.

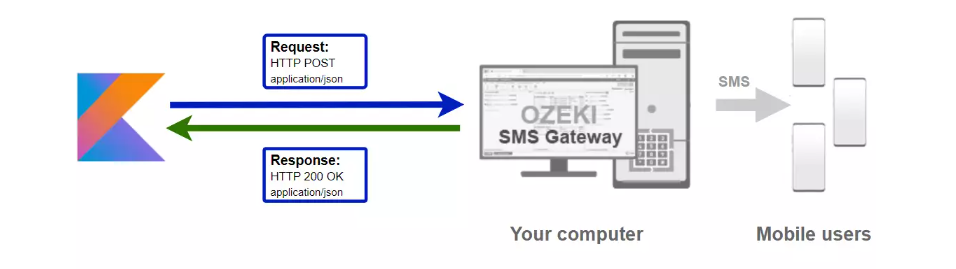


Рис. 29 **Як відправити SMS з Kotlin**

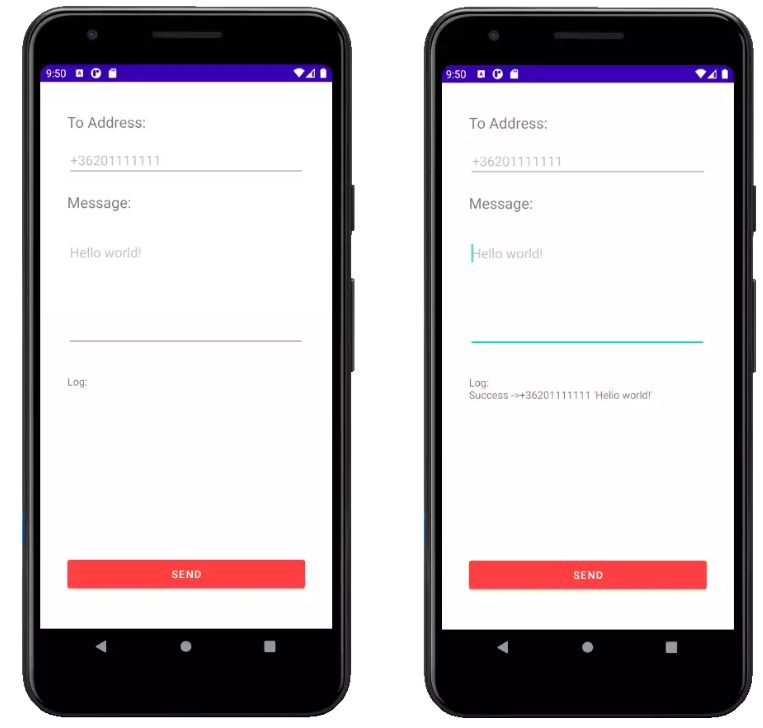


Рис. 30 **Приклад програми до та після надсилання повідомлення**

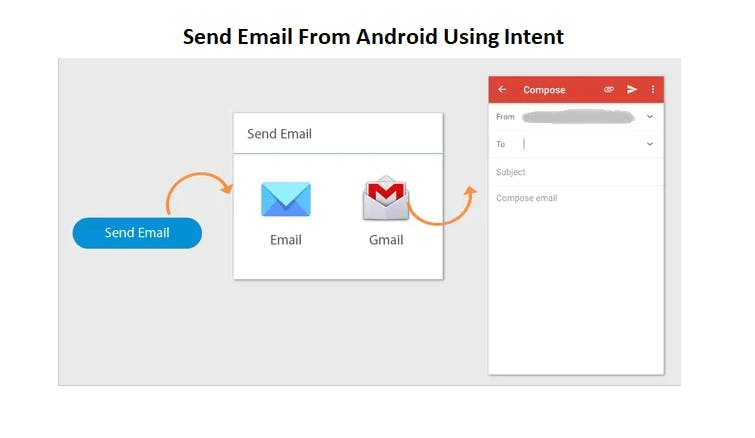


Рис. 31 **Як надіслати електронну пошту в Android за допомогою інтернету**

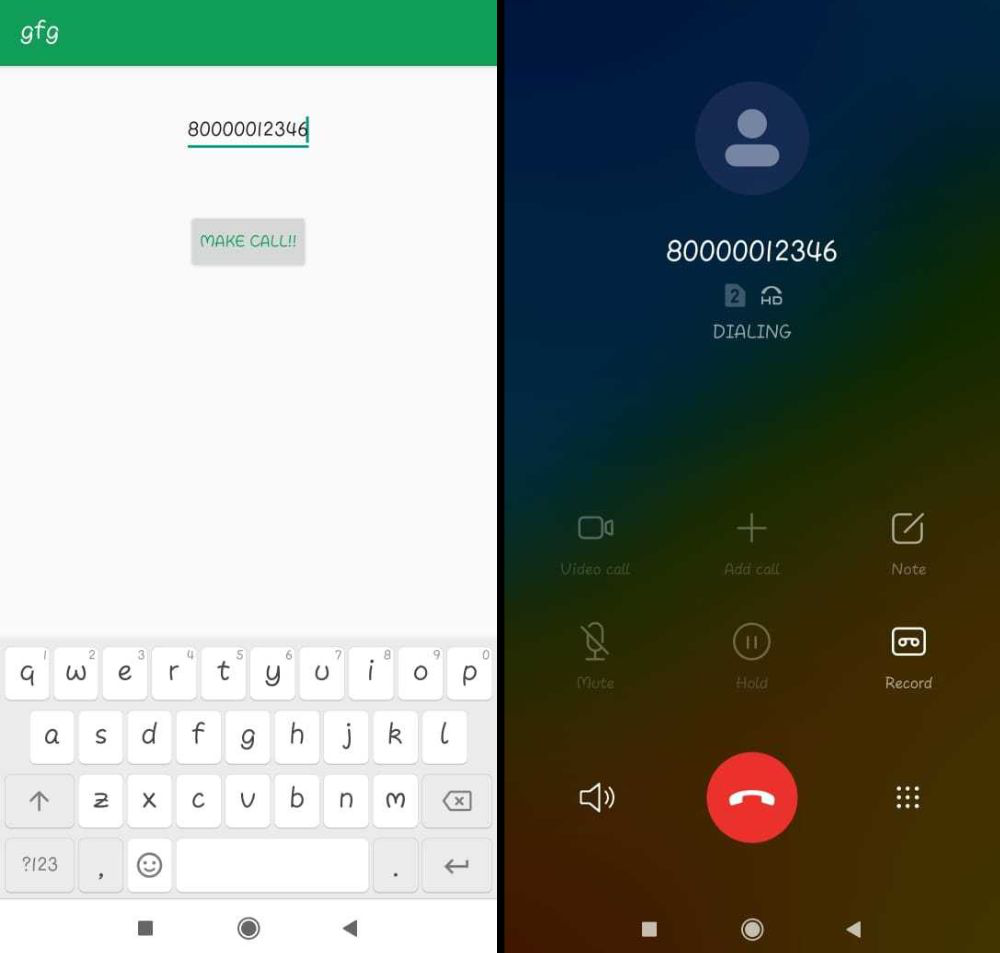


Рис. 32 **Здійснення дзвінка в Android, Kotlin, Java**



Рис. 33 **Quiz 23: quiz (q1)**



Рис. 34 **Quiz 23: quiz (q2)**

# **Section 24: Notification**

У цьому розділі було охоплено ключові аспекти роботи з системою повідомлень, включаючи створення та відправку локальних повідомлень, налаштування повторюваних повідомлень, детальні процедури створення багатофункціональних повідомлень з використанням додаткових кнопок та графіки, а також використання push-повідомлень для ефективної взаємодії з користувачами в реальному часі.

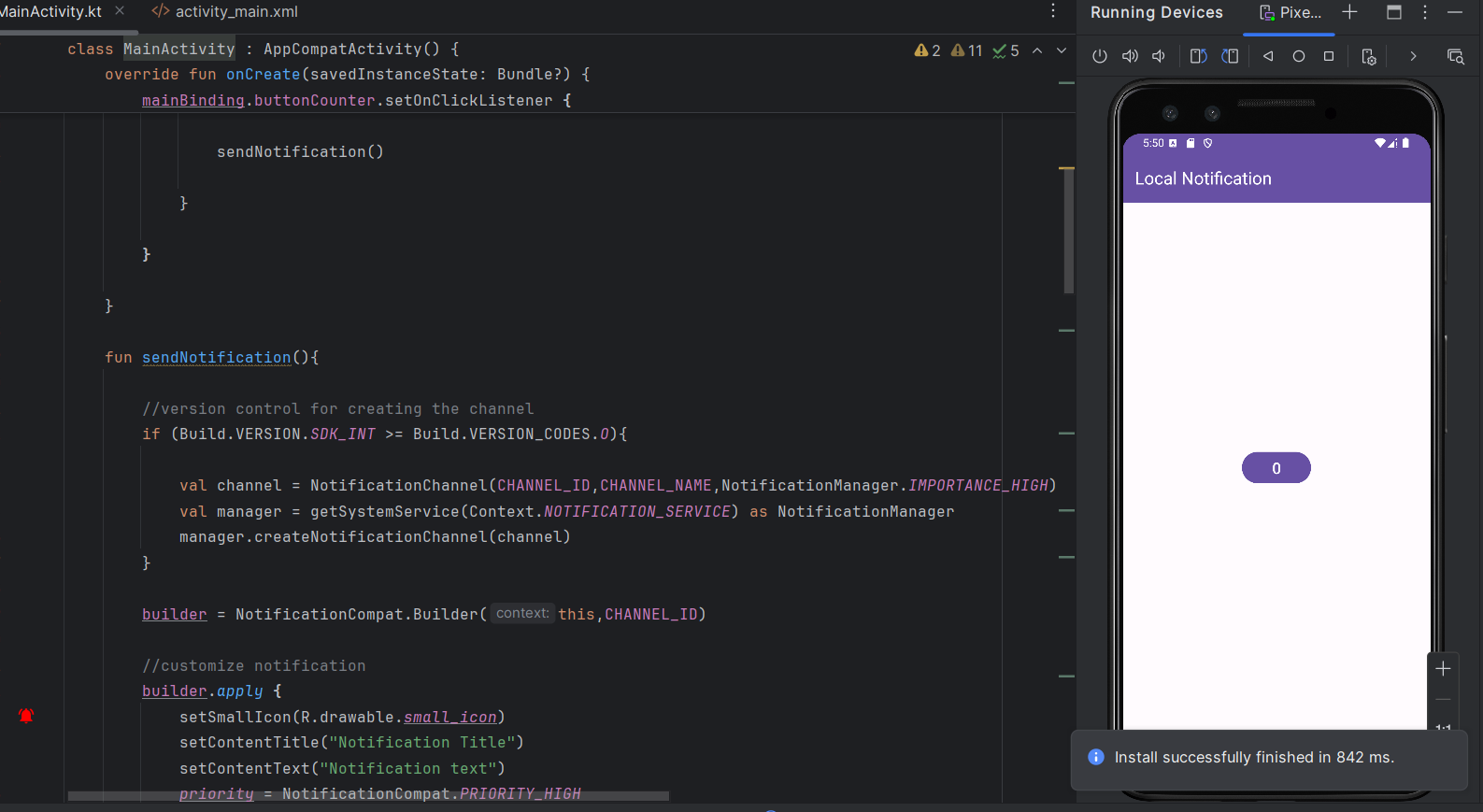


Рис. 35 **запуск застосунку**

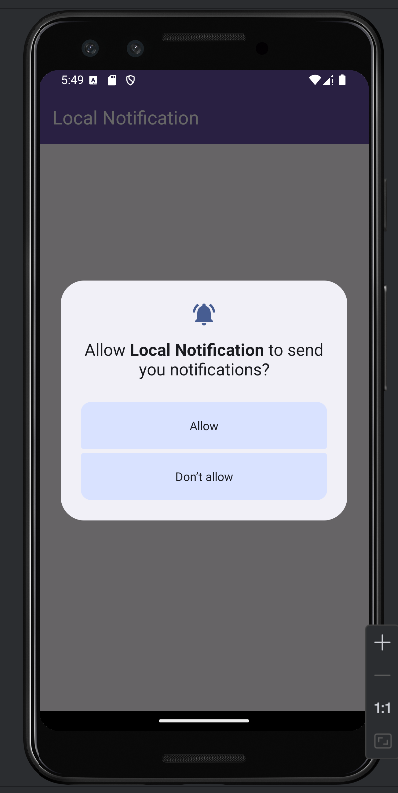


Рис. 36 **дозвіл на сповіщення**

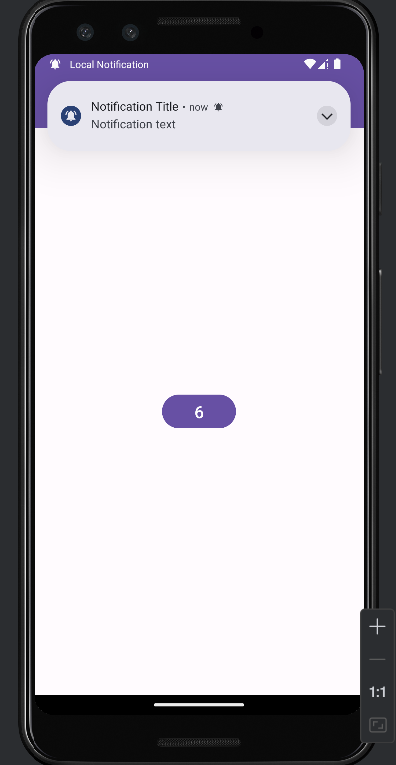


Рис. 37 **Повідомлення**



Рис. 38 **Повторювані повідомлення - запуск застосунку**

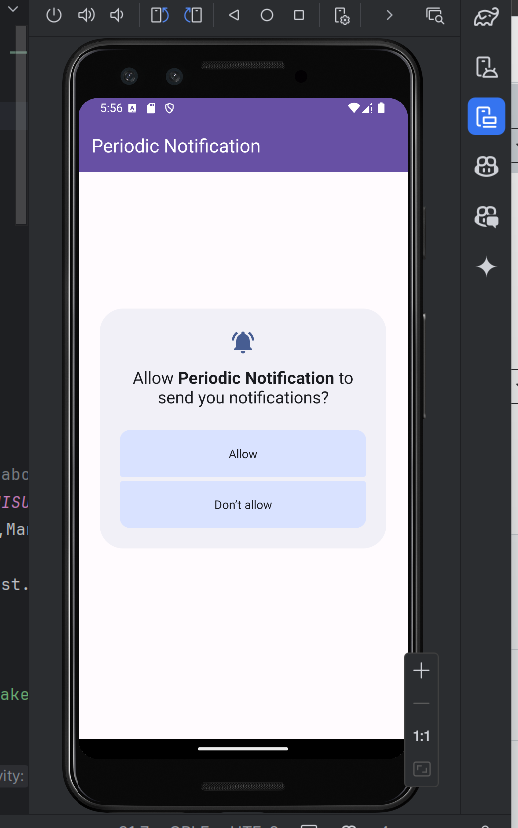


Рис. 39 **дозвіл на повторювані сповіщення**

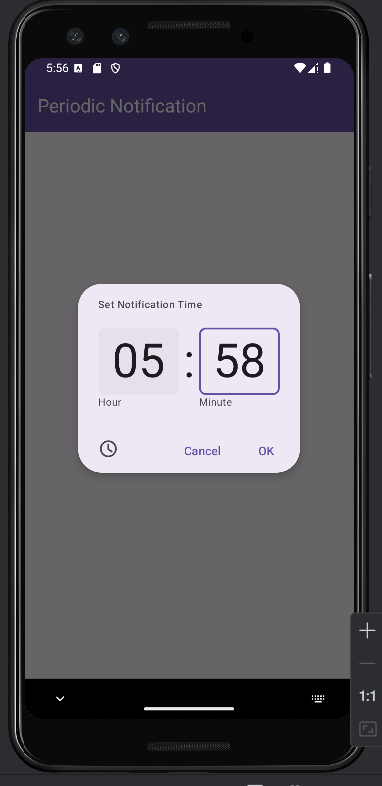


Рис. 40 **Обраємо час**

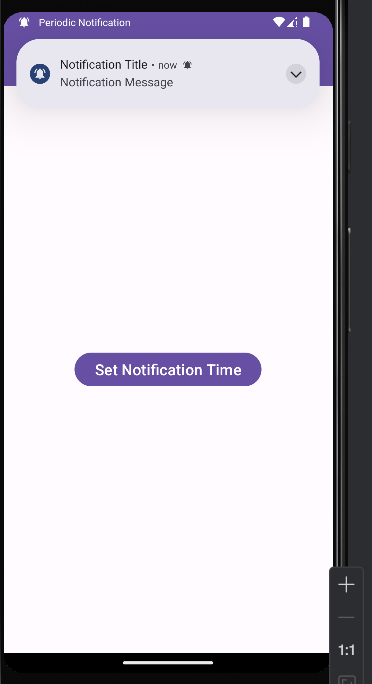


Рис. 41 **Отримаємо повідомлення**

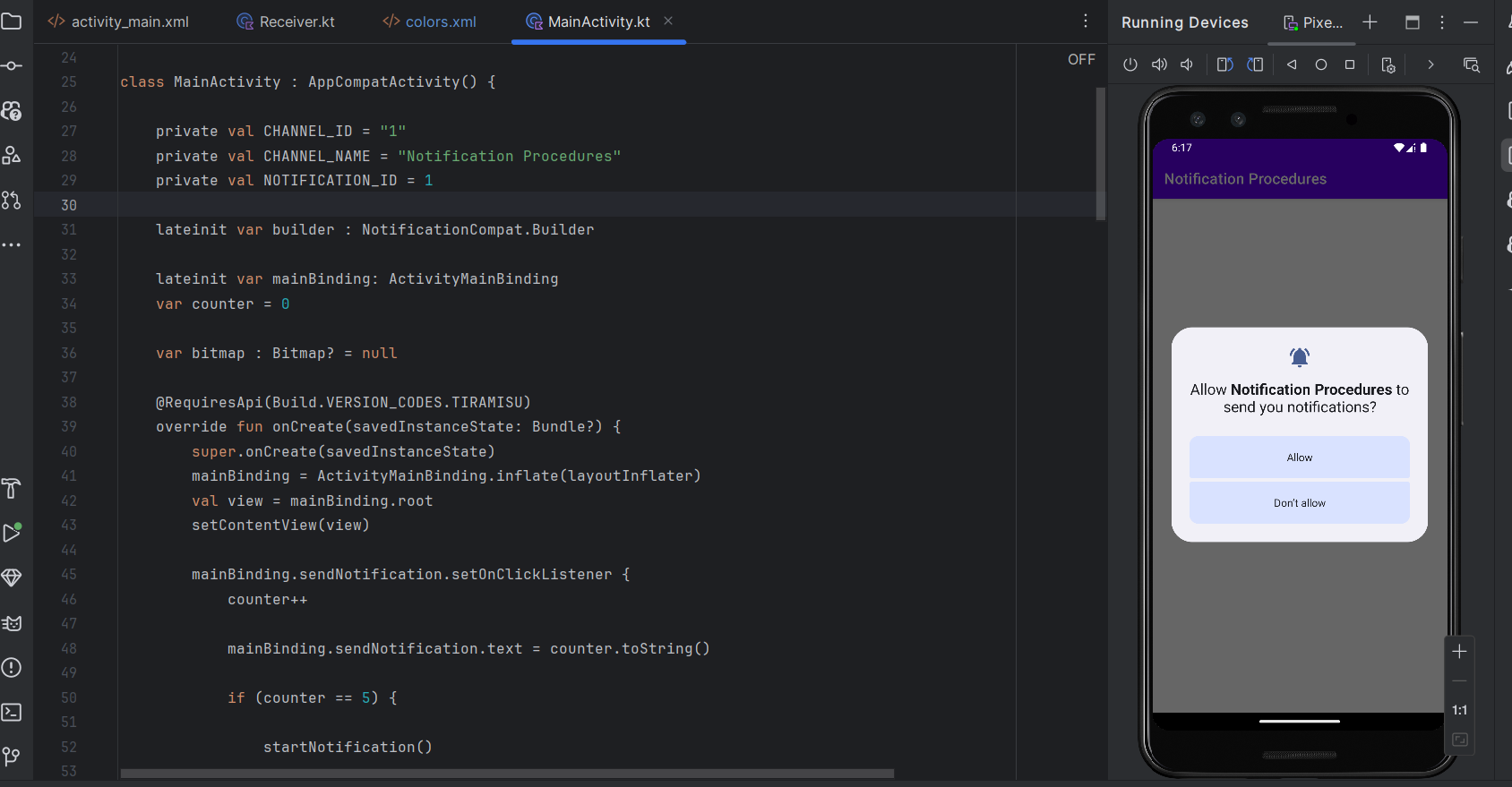


Рис. 42 **Процедури повідомлень – даємо дозвіл**

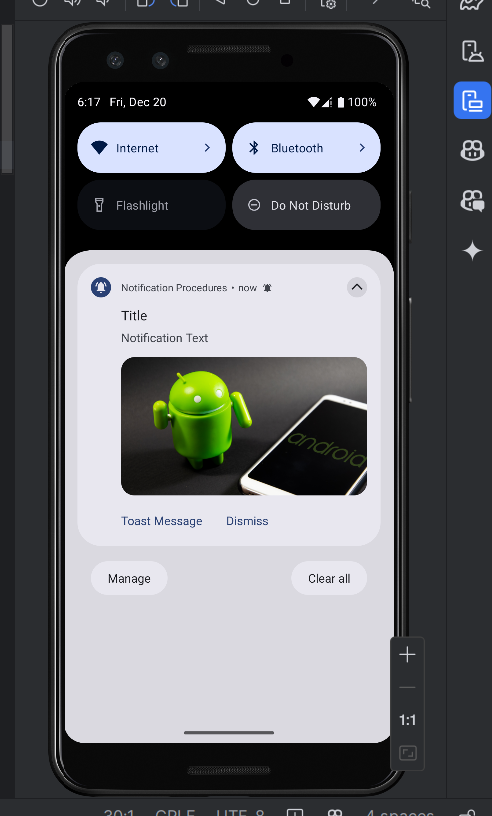


Рис. 43 **Процедури повідомлень - отримаємо повідомлення**

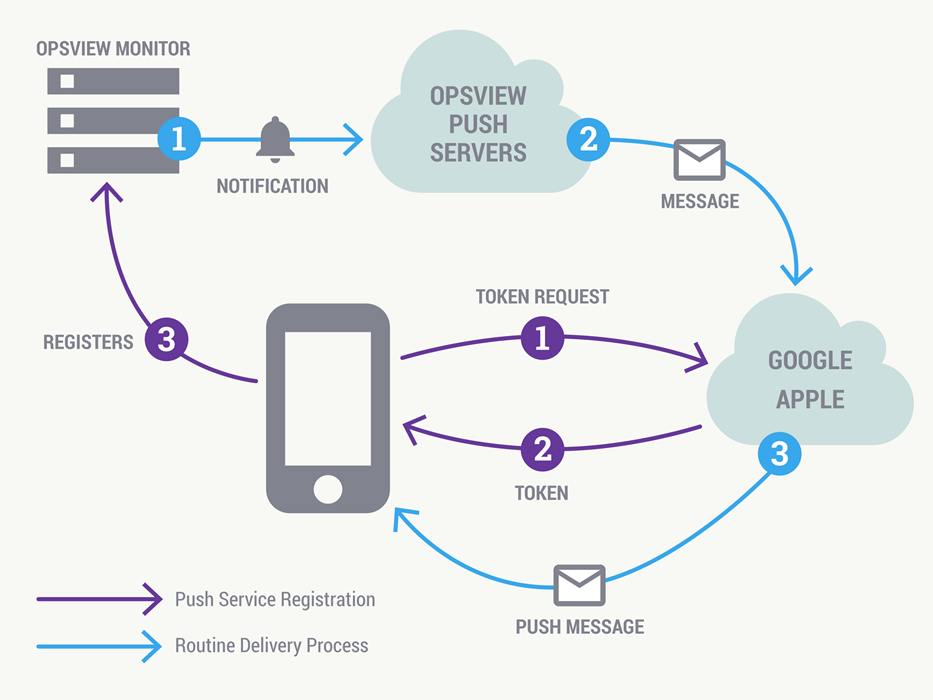


Рис. 44 **Push Notifications**

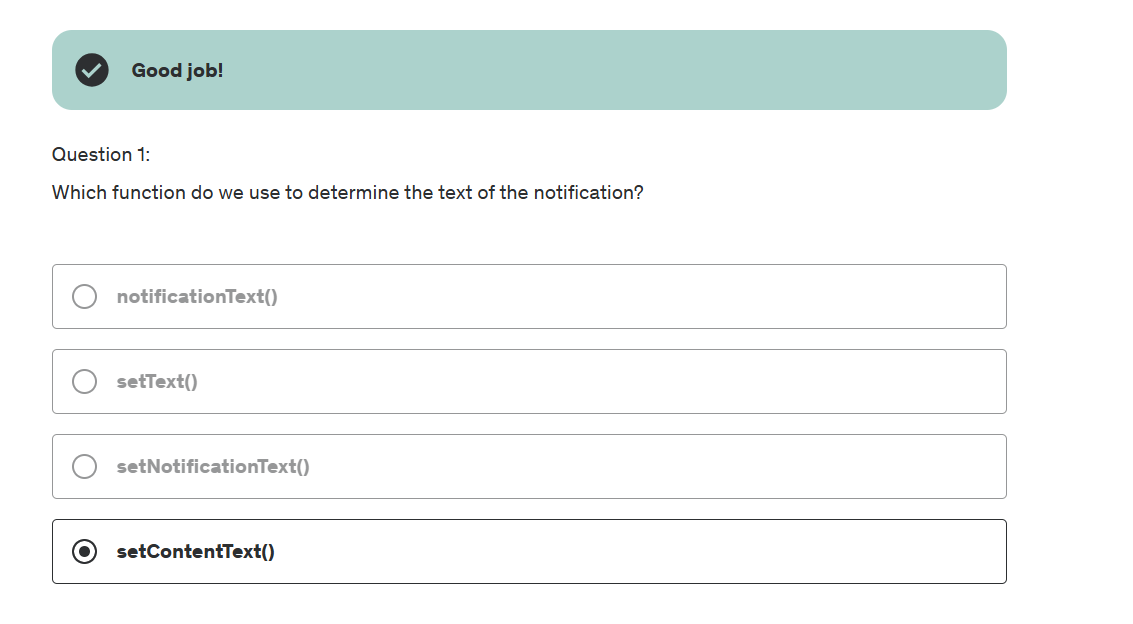


Рис. 45 **Quiz 24: quiz**

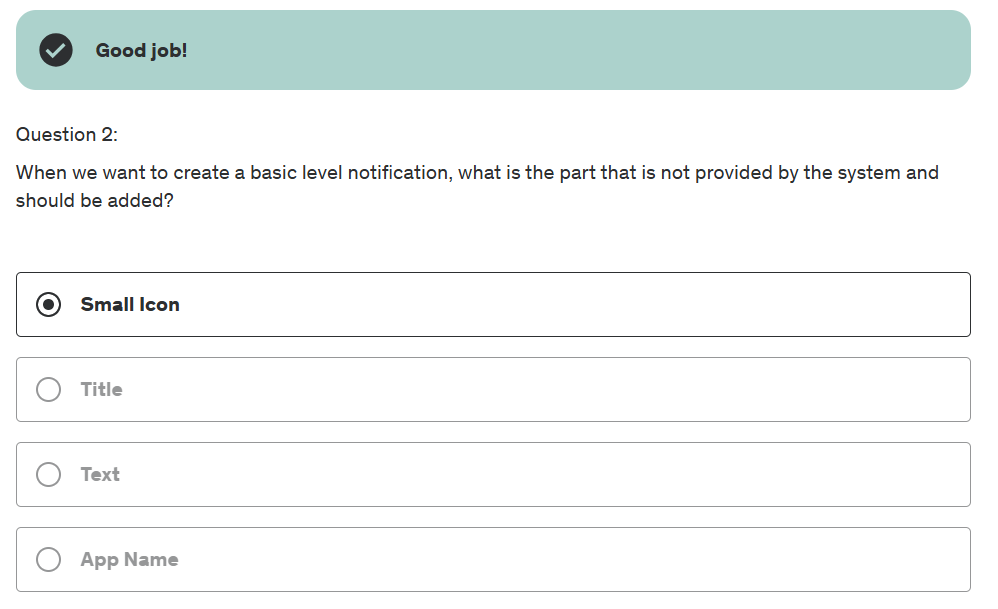


Рис. 46 **Quiz 24: quiz**

# **Section 25: Architectural Room Database in Android App Development with Kotlin**

У цьому розділі розглядається робота з базами даних, включаючи використання SQLite і Room Database, а також створення повноцінного додатку для ведення записів. Ось краткий огляд основних тем:

1. **Основи баз даних** – введення у поняття баз даних та їх важливість у програмуванні.
2. **SQLite Database** – основи роботи з SQLite, легковага транзакційна база даних, що вбудована в Android.
3. **Room Database** – вищий рівень абстракції для SQLite, що забезпечує більш зручну роботу з базами даних через анотації та DAO (Data Access Objects).

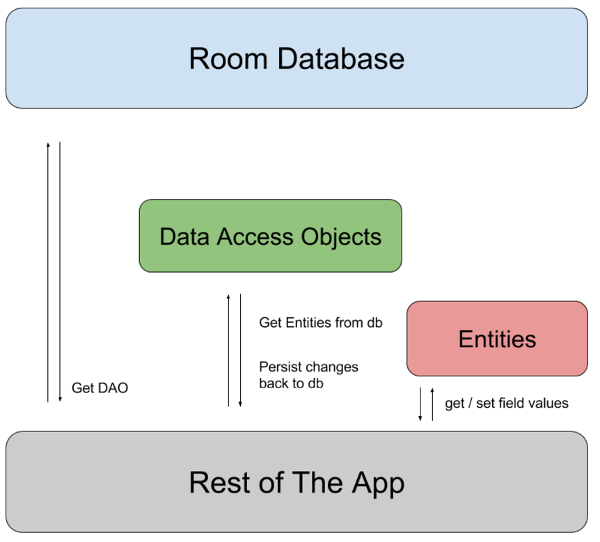


Рис. 47 **Room Database**

1. **Архітектурні компоненти** – вивчення архітектурних компонентів, які використовуються для побудови структурованого та стабільного додатку.

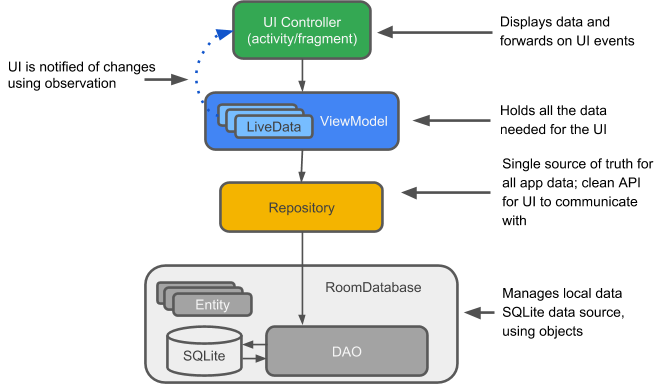


Рис. 48 **Архітектурні компоненти**

1. **Створення додатку для ведення записів** – розробка базової структури додатку з Kotlin.
2. **Створення елементів бази даних:**

* Entity: Оголошення моделей даних.



Рис. 49 **Entity**

* DAO: Визначення інтерфейсів для доступу до даних.

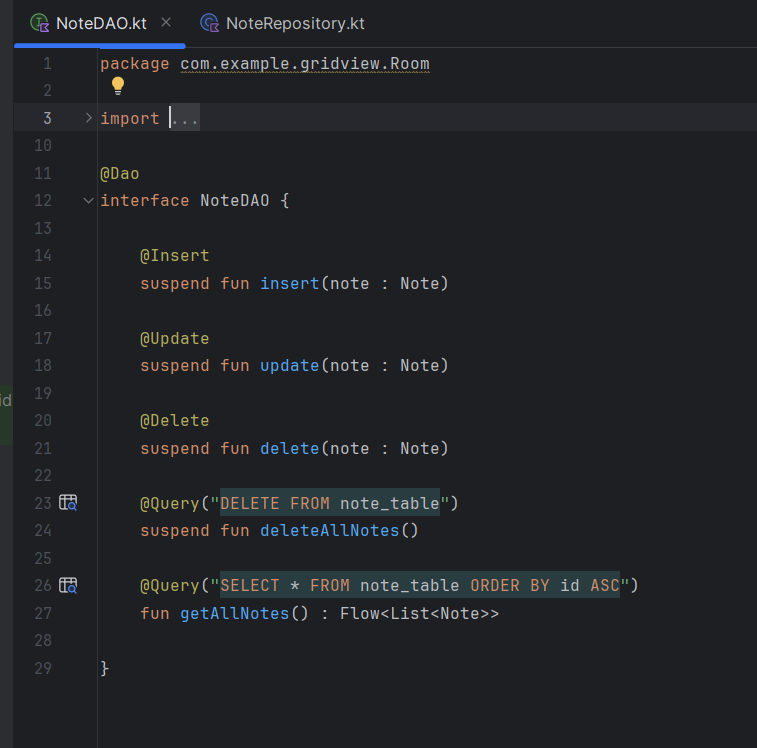


Рис. 50 **Створення інтерфейсу NoteDAO**

* Database: Створення самої бази даних.

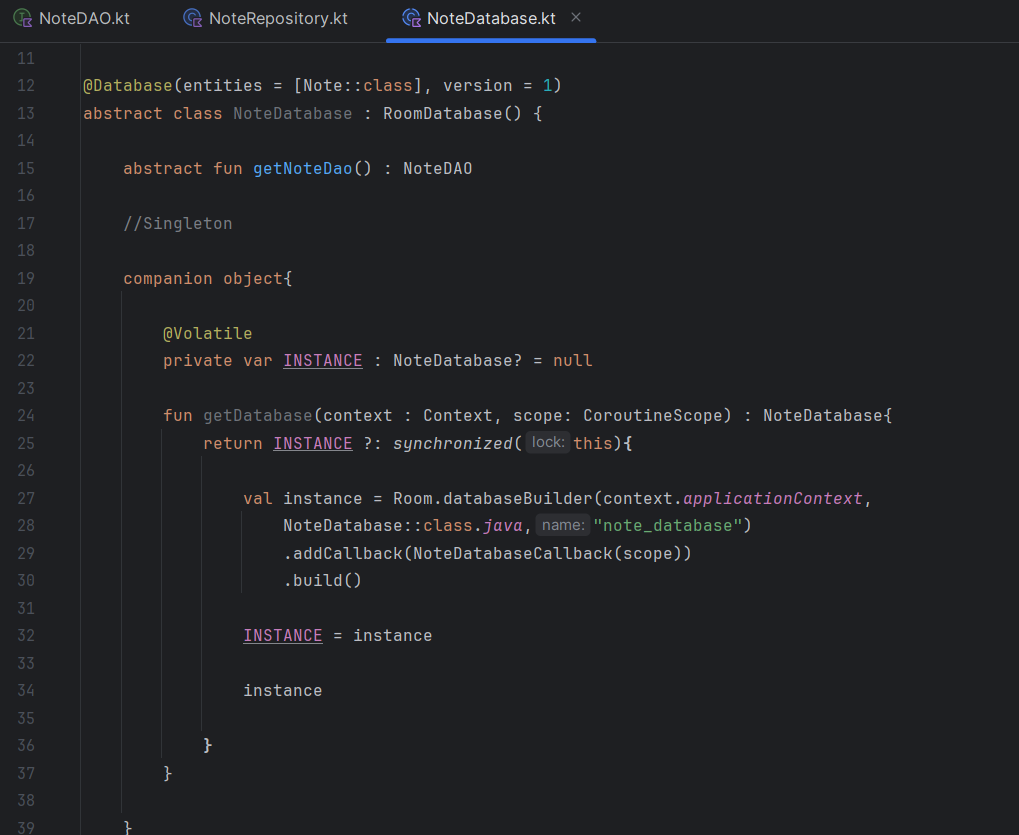


Рис. 51 **Database**

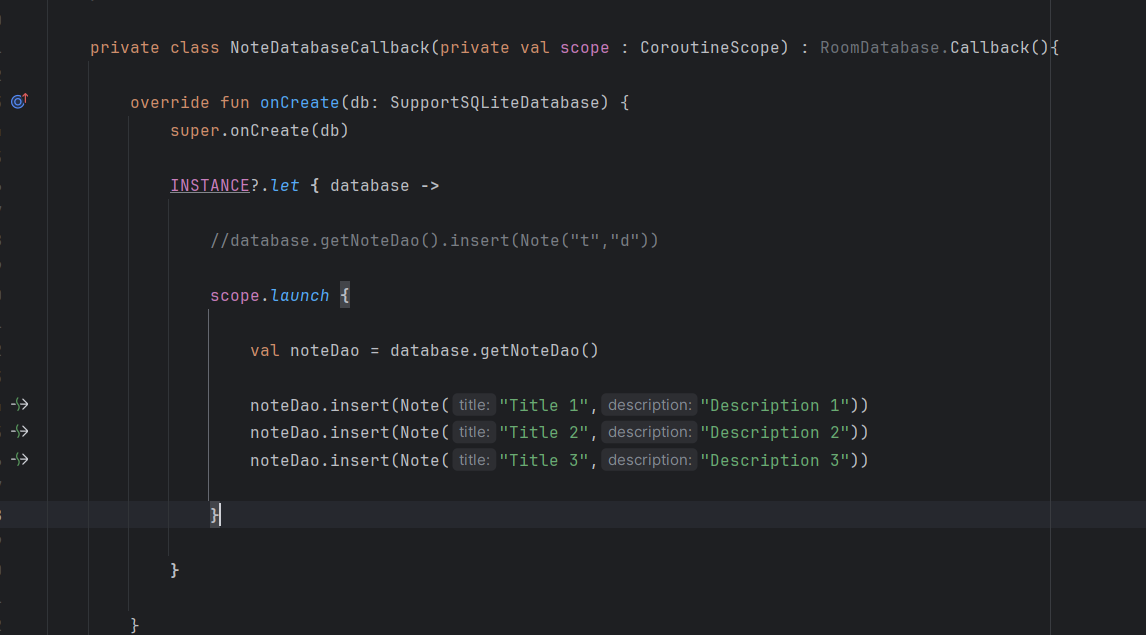


Рис. 52 **Database**

* Repository: Логіка взаємодії між базою даних і UI.

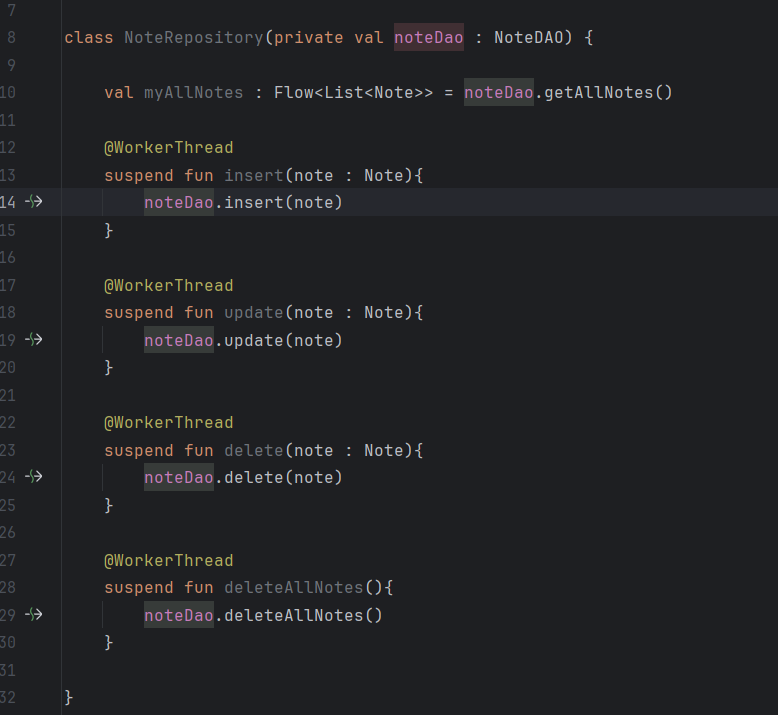


Рис. 53 **Repository**

* View Model: Управління даними UI.

1. **Дизайн користувацького інтерфейсу** – створення інтерфейсу для додатку.
2. **Робота з RecyclerView через Adapter** – показ даних з бази в списковому вигляді.
3. **Додавання, видалення та оновлення даних** – функціонал для зміни даних користувачем у базі.

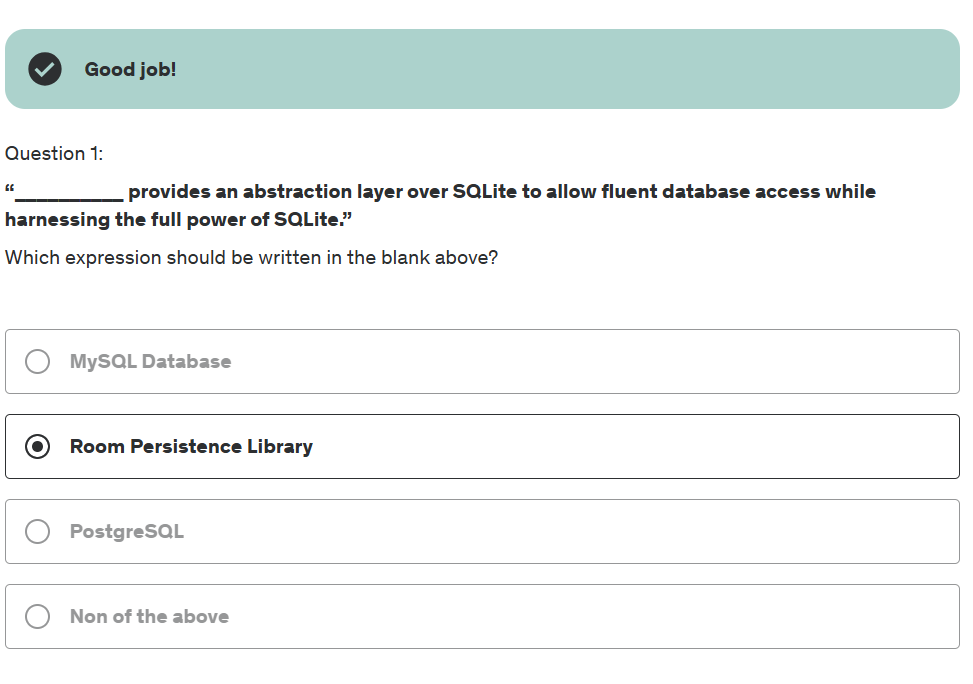


Рис. 54 **Quiz 25: quiz (q1)**

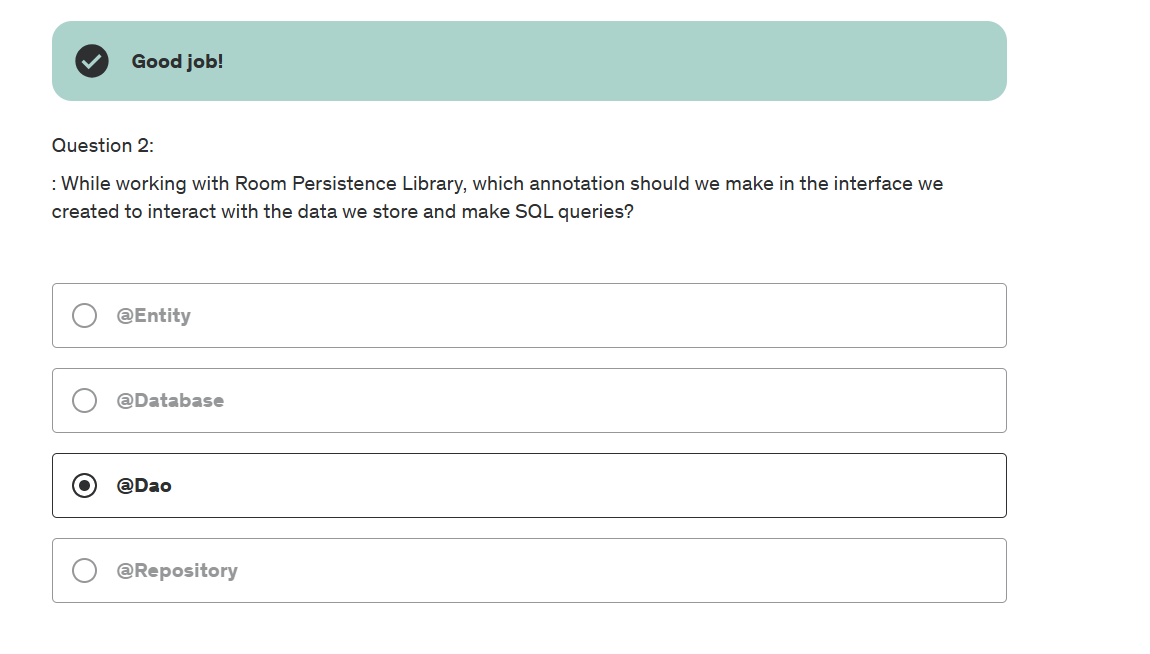


Рис. 55 **Quiz 25: quiz (q2)**

# **Section 26: Photo Album Application**

У цій секції потрібно було створити Photo Album Application.

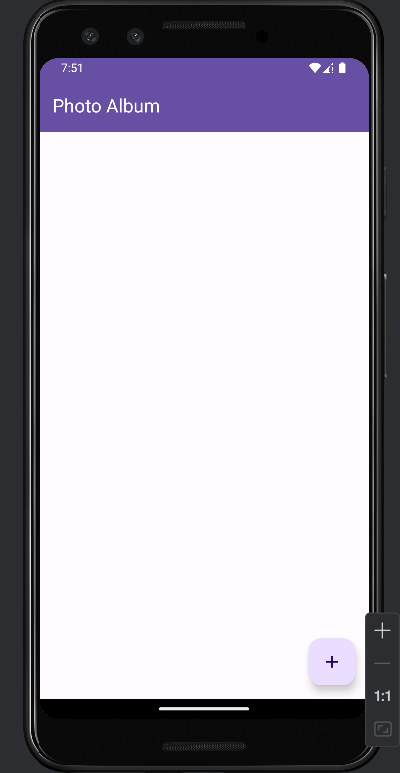


Рис. 56 **Запуск застосунку**

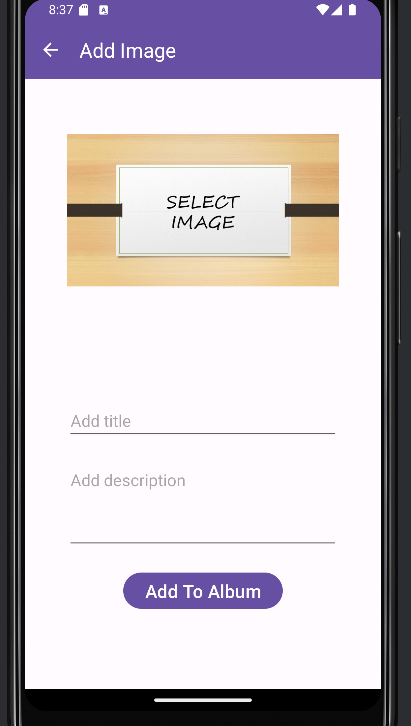


Рис. 57 **Потрібно ввести текст та обрати фото**

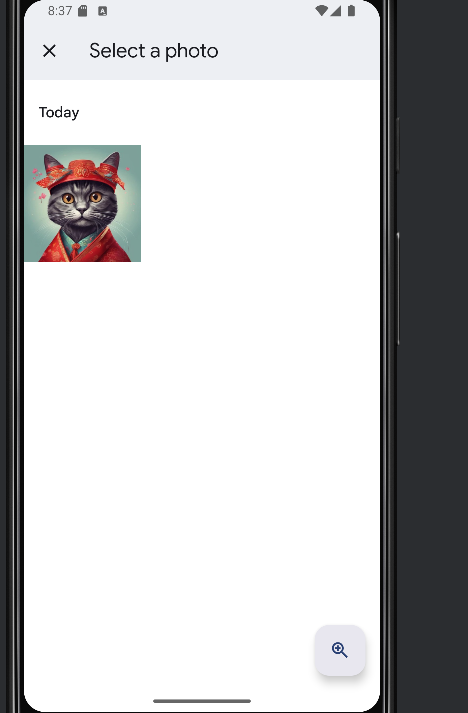


Рис. 58 **Обираю фото**

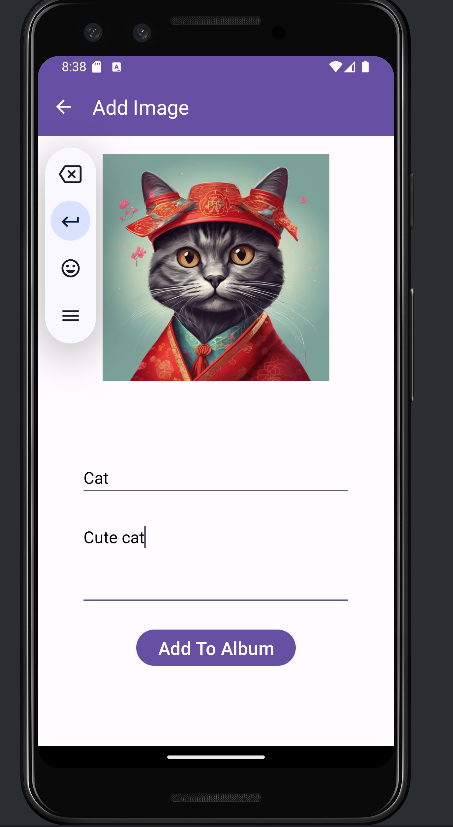


Рис. 59 **Додаю картку до альбому**

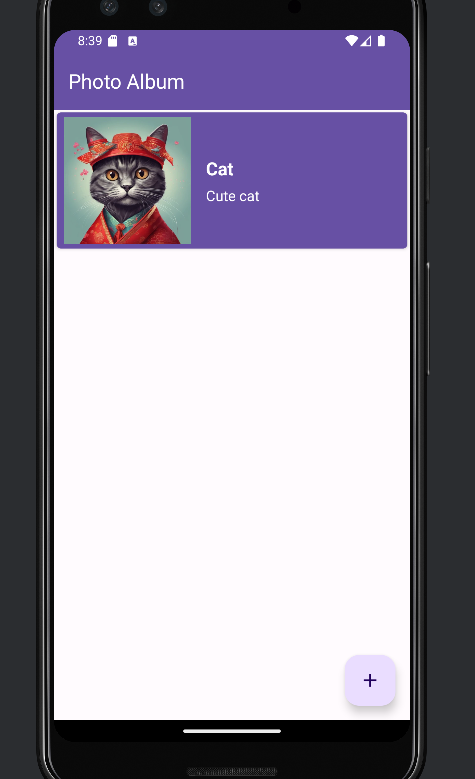


Рис. 60 **Картка додана**

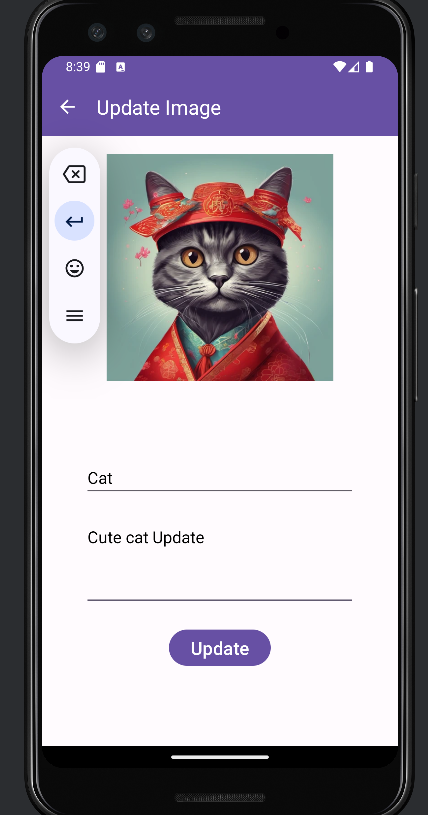


Рис. 61 **Змінюю параметри**

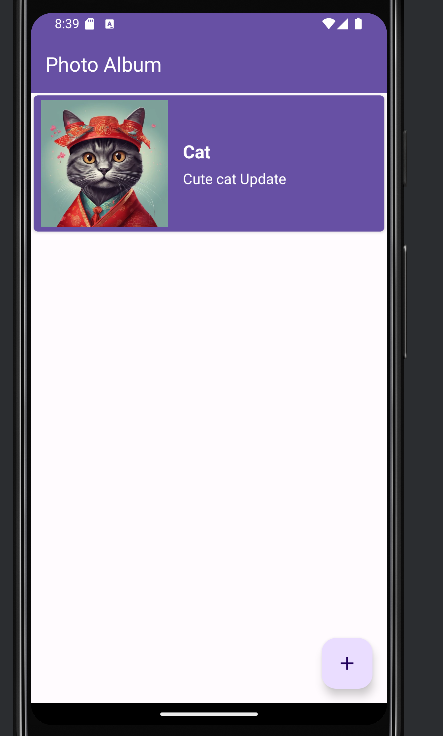


Рис. 62 **Ось новий текст обновився**

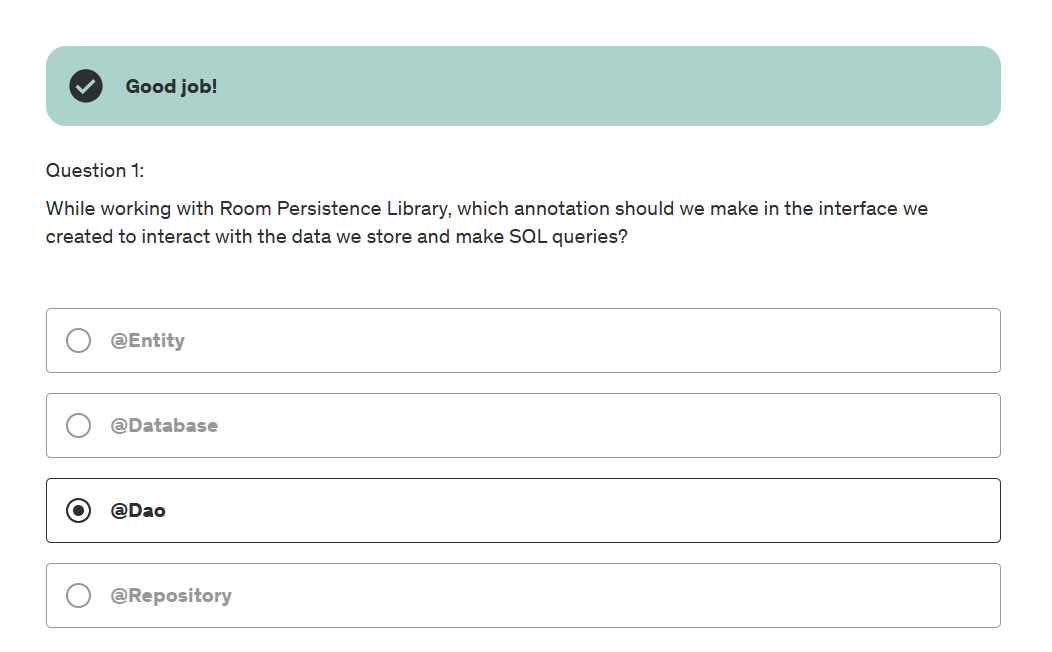


Рис. 63 **Quiz 26: Quiz (q1)**

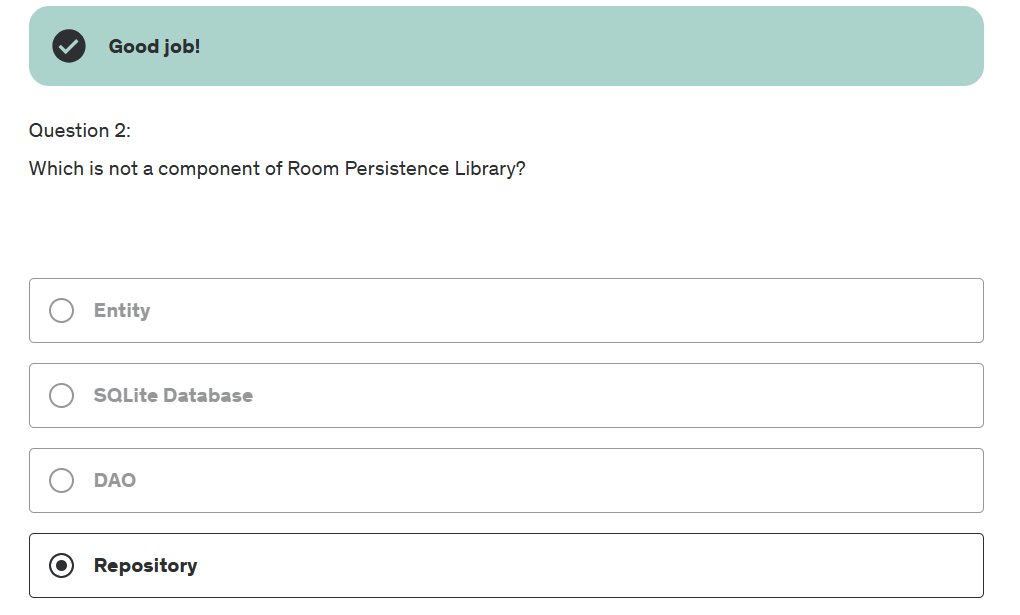


Рис. 64 **Quiz 26: Quiz (q2)**

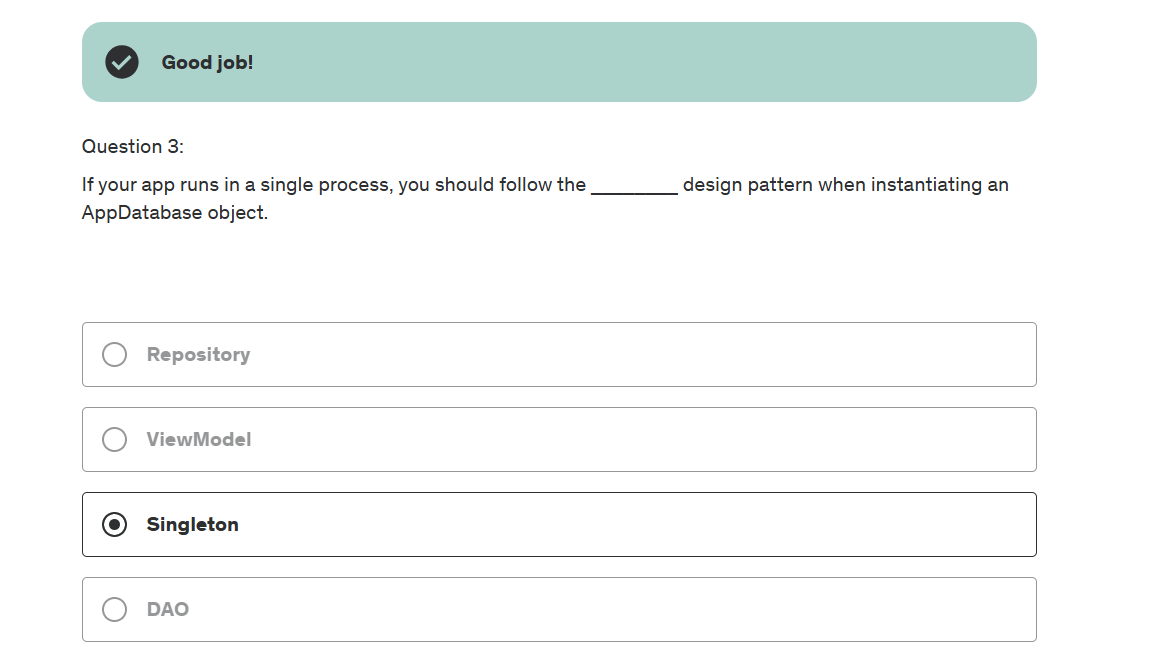
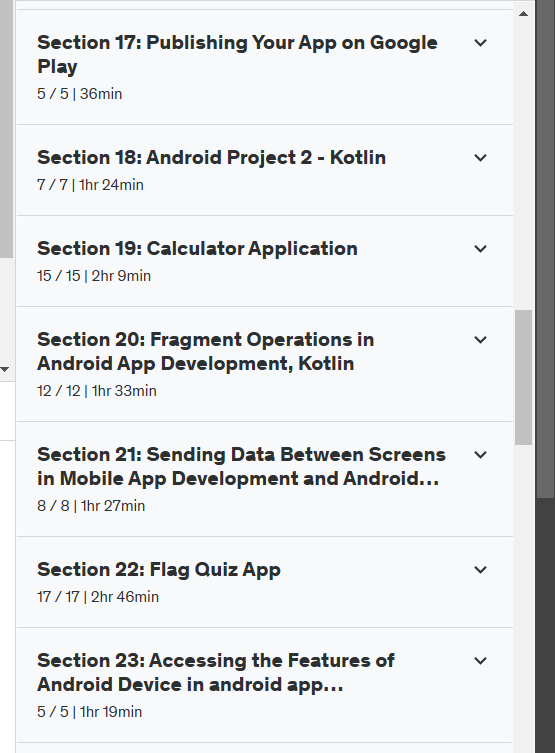


Рис. 65 **Quiz 26: Quiz (q3)**

# **Висновок**



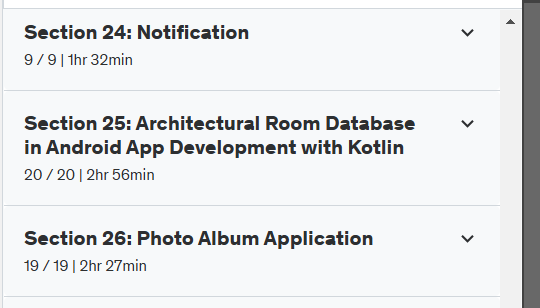


Рис. 66 **Усі секції пройдені**