

Re: Android development mentorship program - Progress

From Mohamed Elsdody <mohamed.elsdody@itworx.com>

Date Thu 4/10/2025 9:17 AM

To Bassel Samir < Bassel.Samir@itworx.com>

Cc Ayman Roshdy < Ayman.Roshdy@itworx.com>

Hello Basel.

Good Job Ya Basel, FYI our program will be ended soon so you need to complete the below task to complete the program and that will be the last task:

As part of your technical development in modern Android architecture and tools, I'm assigning you the following task. This task is designed to test your ability to build scalable, testable, and modern Android features using the following:

- MVI (Model-View-Intent) architecture
- · Kotlin Coroutines & Flow
- · Dagger & Hilt for Dependency Injection
- Jetpack Compose for UI

Task Overview: Build a Feature-Rich Note Taking App

of Functional Requirements:

- · Users can:
 - · Add new notes
 - · Edit existing notes
 - Delete notes
 - View note details
- Each note should include:
 - Title
 - Description
 - Tags (optional)
 - · Timestamps (Created, Last Modified)

Ul Requirements (Jetpack Compose):

- Use Material 3 design principles
- Main screen shows a list of notes:
- Each note displayed in a Card with title, short description preview, and last modified timestamp
- · Floating Action Button (FAB) to add a new note
- Add/Edit Note screen:
 - o Inputs for title, description (multi-line), and tags
 - Use OutlinedTextField and validation messages where necessary
- Compose animation on note item deletion (e.g., fade out or slide)
- · Optional:
 - Add filter/search bar at the top
 - Tag chips or dropdown for filtering
 - Support light and dark themes

Architecture & Technical Requirements:

- Follow MVI with unidirectional data flow:
 - Intent (user actions)
 - Model (business logic)
 - ViewState (UI rendering)
- Use ViewModel + StateFlow for exposing state
- Repository pattern with Coroutine + Flow for data access
- Use Hilt for DI across all layers (Repository, ViewModel, UseCases, etc.)
- Use **Room** for local persistence

Bonus Points:

· Add unit tests for ViewModel and UseCases

- Use sealed classes for Intents and ViewStates
- Follow Clean Architecture layering: data, domain, presentation

Project Structure Recommendation:

markdown CopyEdit

- data/
 - local/
 - repository/
- domain/
 - model/
 - usecase/
- presentation/
 - viewmodel/
 - ui/
- di/

O Deadline:

Please complete and share the GitHub repository in 2 weeks. Ensure clean code, commit history, and a well-written README that explains architecture decisions.



Mohamed Elsdody

Engineering Manager (Mobile Team)

General

From: Bassel Samir < Bassel.Samir@itworx.com>

Date: Sunday, 6 April 2025 at 7:32 PM

To: Mohamed Elsdody <mohamed.elsdody@itworx.com>

Cc: Ayman Roshdy < Ayman.Roshdy@itworx.com>

Subject: Android development mentorship program - Progress

Dear Sdody,

I hope this email finds you well.

I have great news! I am sending to inform you that I have finally completed the Kotlin Coroutines and Flow course.

I have attached the certificate of completion.

Looking forward to your feedback and next steps!

Here is the updated timeline of our progress.

Topic	Source
Kotlin Programming language	https://www.udemy.com/course/kotlinmasterclass/?couponCode=KEEPLEARNING
Master Android Development	https://www.udemy.com/course/android-kotlin-developer/?couponCode=KEEPLEARNING https://developer.android.com/courses/pathways/android-architecture https://www.youtube.com/watch?v=kQllplh3flw
The SOLID Principles for Android Developers	https://medium.com/kayvan-kaseb/the-solid-principles-for-android-developers-75fd4ca3ef84 https://codersee.com/solid-principles-with-kotlin-examples/ https://www.youtube.com/watch?v=EDRpKp6UcCg
Android Dependency injection with Dagger	https://www.udemy.com/course/dependency-injection-in-android-with-dagger/?couponCode=KEEPLEARNING
Android Dependency injection with Koin	https://blog.mindorks.com/kotlin-koin-tutorial/
Android Kotlin Coroutine & flow	https://www.udemy.com/course/coroutines-on-android/?couponCode=KEEPLEARNING