Upišite se

Android app architecture

Android Dev Academy 2024
David Takač

Content

- Why we need architecture
- Presentation and business Logic
- Model-View-ViewModel
- Live demo
- Homework
- Further reading
- Questions

Why we need architecture

Counter Example

- Put everything in onCreate
- Try adding functionality
- Hard to understand later
- Especially for colleagues
- Minimize time spent understanding someone’s mess

The Need For Structure

Organize code so that it is easy to:

- Understand
- Upgrade and fix
- Test

Software Architecture

- Organized way to split code into decoupled and cohesive units
- Useful ways become de facto standards
- Recognized by developers within the domain
- Code is split by “concerns”

Presentation and Business Logic

Presentation Logic

- Displays app data
- Allows interaction with app data
- Should not handle data processing or storage

Business Logic

- Provides app data
- Modifies app data
- Should not care about how data is displayed

Why Mixing These Is Bad

- Android owns the presentation layer
- You own the business logic/state layer
- Depending on presentation to store state results in data loss
- UI changes can accidentally break logic, and vice versa
- Way harder (if not impossible) to test

Solution: Clear Separation

- Instead, clearly separate these two layers
- With architectural patterns like MVVM

Model-View-ViewModel (MVVM)

Model-View-ViewModel

Architectural pattern that enforces this separation

Divides app into

- Model: Business logic
- ViewModel: Presentation logic
- View: Presentation logic

Fits perfectly with Compose and state-driven UI

Model

Data layer

- Encapsulates business logic and app data
- Retrieves data from data source
- Modifies data in data source
- Hides its data sources

View

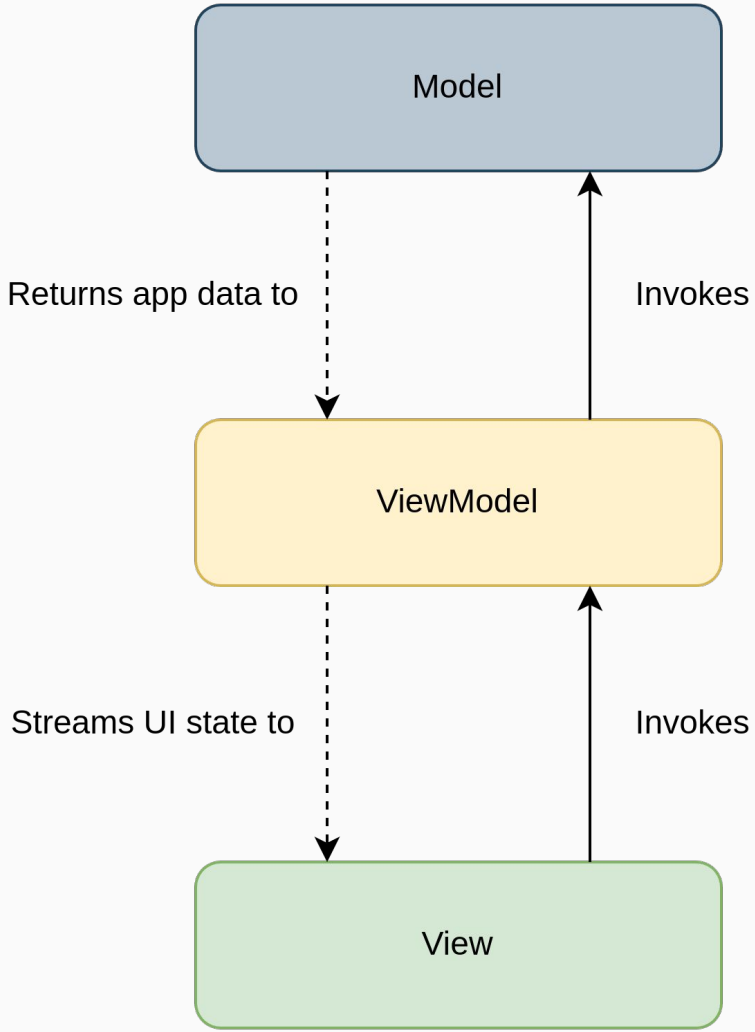
UI layer

- Displays data from ViewModel
- Forwards user actions to ViewModel
- Pure Android/Compose

ViewModel

Bridge between Model and View

- State holder
- Outlives View
- Transforms Model data into View data (UI state)
- Exposes UI state reactively
- Decides how to transform Model data based on user input


Model
Returns app data to
Invokes
ViewModel
Streams UI state to
Invokes
View

Live Demo

Quotes App

- Home screen that presents greeting and some quotes
- Settings screen that presents username and some app settings
- Opens edit screen when username clicked
- Change has to be reflected immediately on home and settings
- Implement using MVVM

Homework

Notes app

- Make an app that shows a list of notes
- Note click takes you to screen where you can edit it
- Saving changes returns to the list
- Edited note is updated in the list reactively
- Edited note stays in the same place in the list

Details

- Save notes in a regular list for now; nothing fancy
- Use MVVM, of course

Further reading

Further reading

- Guide to app architecture: https://developer.android.com/topic/architecture
  - Visit every page in this subsection (“About app architecture”, “UI layer”, “Domain layer”, etc.) They are in the sidebar under “Guide to app architecture”)
- Clean Architecture: A Craftsman's Guide to Software Structure and Design by Robert C. Martin
  - Very useful for getting into the proper mindset
  - **Don’t get carried away** and start adding use cases everywhere! Be frugal. You will just need repositories 99% of the time. You will know when use cases are required.

Further reading

- Navigation with Compose
  https://developer.android.com/develop/ui/compose/navigation
  - Visit links from Get Started until Deep Links
- Kotlin coroutines on Android
  https://developer.android.com/kotlin/coroutines
  - Visit “Introduction to coroutines” and “Advanced coroutines concepts”

Questions?

Thanks!