UDC 004.41

CLEAN ARCHITECTURE FOR ANDROID APPLICATIONS

R. KRASKO, A. OSKIN
Polotsk State University, Belarus

The article describes the features of developing a mobile Android application using a clean architecture. A set of tools for software implementation of this approach was proposed.

Introduction. The architecture of the application is the most important aspect in software development - it must be reliable, stable, flexible in testing, easy to expand and change. At the same time, the architecture should be understandable for developers with different levels of knowledge and experience to ensure ease of maintenance.

Clean acchitecture. When using pure architecture the entire application code is divided into levels that adhere to one rule: the internal level does not need to know anything about the external. The internal level contains business logic, and its external implementation, depending on the platform. This approach should also meet other requirements:

– independence on tools. Architecture should not rely on the existence of any library. This allows you to use with minimal cost to change the implementation of business logic [1];

– database independence, business logic should not be tied to specific databases [1];

– independent of any external agent, business logic should not know anything about the interaction in external data sources [1].

The figure shows the various levels and components of the architecture and their interaction.

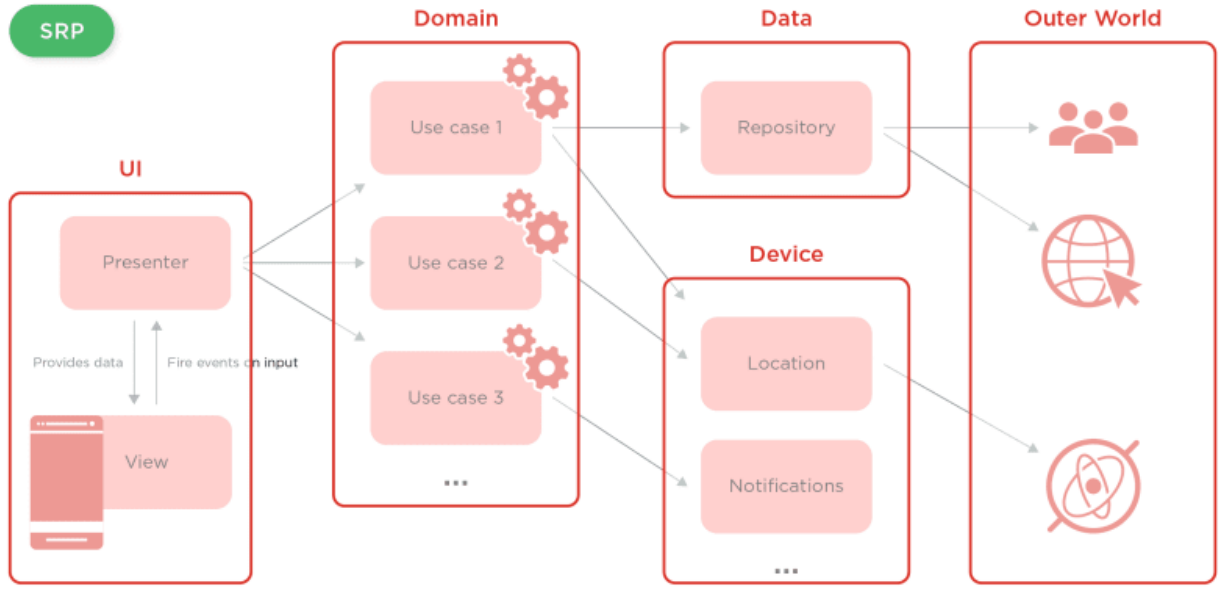

Figure. – Components of clean architecture

Presentation level:

– for this layer MVP (Model VIew Presenter) pattern is selected, using of MVVM is also possible;

– view is an implementation of the passive view pattern and provides a set of interfaces that must be implemented by components from the Android SDK (Fragment, Activity, Adapter, Custom View) [2]

– presenter is the intermediary between the user interface and business logic. It must be platform independent to improve code testability [2];

– frameworks JUnit and Mockito for testing. Esspresso is also used for integration tests.

Domain level:

– contains use cases, which are the most primitive parts of the business logic of the application for maximum simplicity and the possibility of their further reuse;

– frameworks JUnit and Mockito for testing.

Data level:

– the repository pattern is responsible for the interface on which use cases from the domain level can receive the necessary data [3];

– implementation of caching is also possible at this level;

– frameworks JUnit, Mockito and Robolectrick for testing.

Device level:

– Contains implementations of platform-dependent functionality: notifications, sensors, various managers, and so on.

Communication between layers, navigation:

– dependency injection using Dagger 2 framework;

– data flow between layers using RxJava framework;

– Cicerone framwork for navigation;

Conclusion. The features of using clean architecture when developing applications for Android mobile devices were described. A set of tools for implementing this approach is proposed. This architecture is flexible in support, easy to test and very convenient when working in large teams with specialists of various levels.

REFERENCES

1. Clean architecture for Android and iOS [Electronic resource] / Apptractor © 2020 – Mode of access: https://apptractor.ru/develop/chistaya-arhitektura-na-android-i-ios.html – Date of access: 27.02.2020.
2. Model–view–presenter [Electronic resource] / Wikipedia © 2020. – Mode of access: https://en.wikipedia.org/wiki/Model–view–presenter – Date of access: 27.02.2020.
3. Repository [Electronic resource] / Martinfowle © 2020. – Mode of access: https://martinfowler.com/eaaCatalog/repository.html – Date of access: 27.02.2020.