

РАЗРАБОТКА АРХИТЕКТУРЫ МОБИЛЬНОГО ПРИЛОЖЕНИЯ НА ПЛАТФОРМЕ ANDROID

Турцевич А.М.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Андруш В.Г. – к.т.н., доцент

В работе производится проектирование архитектуры мобильного приложения на основе принципов «Clean Architecture».

Для достижения надежности, удобства в сопровождении, легкости тестирования и гибкости для добавления новых функций в приложение, были поставлены следующие требования к архитектуре приложения:

- отсутствие зависимости от фреймворков;
- тестируемость приложения;
- независимость работы приложения от UI;
- независимость от БД;
- независимость от внешних служб.

Цель построения архитектуры состоит в том, чтобы разделить задачи таким образом, что бизнес-логика приложения ничего не знала о внешнем мире и можно было бы ее тестировать без каких-либо внешних зависимостей.

Было решено разбить проект на три слоя, каждый из которых имеет свою задачу и независим от других.

Каждый слой имеет свою модель данных, что позволяет достичь необходимой независимости. В качестве архитектуры приложения была выбрана «Clean Architecture» [1, 2].

Схема архитектуры, используемой в приложении изображена на рисунке 1.

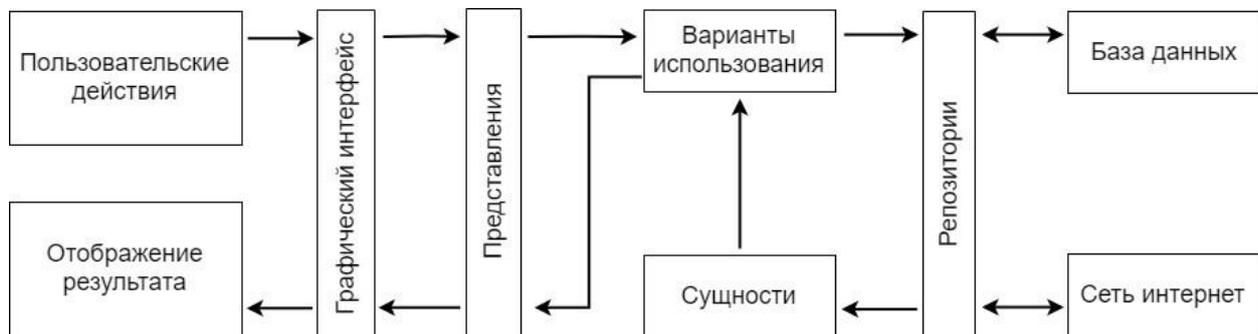


Рисунок 1 – Схема архитектуры приложения

Слой представления

Данный слой представляет собой реализацию шаблона модель вид представитель [3, 4] (далее по тексту MVP).

MVP - шаблон проектирования, производный от шаблона модель представление контроллер (далее по тексту MVC), который используется в основном для построения пользовательского интерфейса.

MVC - схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер – таким образом, что модификация каждого компонента может осуществляться независимо.

Здесь логика связывается с UI приложения и происходит отображение данных на экран. Presenter'ы связываются с «посредниками», происходит работа не в UI-потоке, и получают информацию о том, что отображать в момент времени на экране.

Слой бизнес логики

В данном слое реализованы посредники, с помощью которых обновляется графический интерфейс приложения. Вся логика приложения находится в этом слое.

В этом слое нет никаких Android-зависимостей, вся логика реализована на чистой Java. Все внешние компоненты для связи с бизнес- объектами используют интерфейсы. Схема данного слоя представлена на рисунке 2.



Рисунок 2 – Схема слоя бизнес логики

Слой данных

Все данные, которые необходимы для работы приложения поставляются из слоя данных. Для этого реализован интерфейс UserRepository, который находится в слое бизнес-логики. Данный интерфейс использует шаблон Repository [5]. Источником данных выбирается в зависимости от текущих условий. Источником данных может быть: память устройства, БД, либо сервер. Схема слоя данных представлена на рисунке 3.

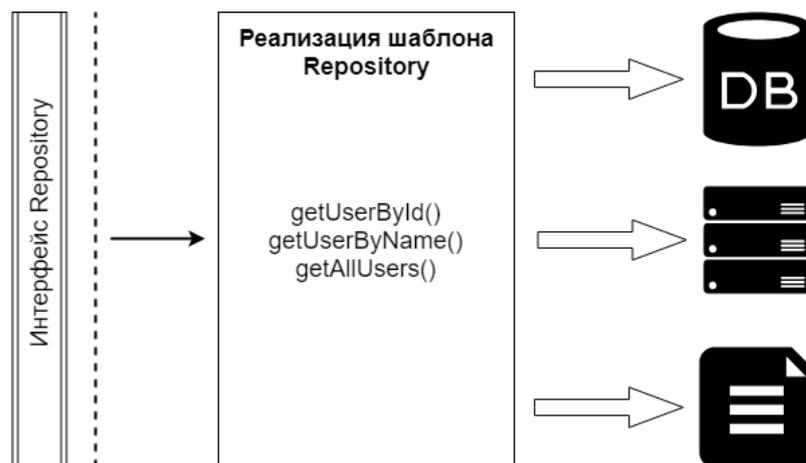


Рисунок 3 – Схема слоя данных

Список использованных источников:

1. Clean architecture of Android apps with practical examples [Электронный ресурс] // RubyGarage. – Режим доступа: <https://rubygarage.org/blog/clean-android-architecture>. – Дата доступа: 01.04.2020.
2. Detailed Guide on Android Clean Architecture [Электронный ресурс] // Medium. – Режим доступа: <https://medium.com/@pavan.careers5208/detailed-guide-on-android-clean-architecture-9eab262a9011>. – Дата доступа: 01.04.2020.
3. Model View Presenter (MVP) in Android with a simple demo project [Электронный ресурс] // Medium. – Режим доступа: prelink.ru/XsGnA. – Дата доступа: 01.04.2020.
4. MVP и Dagger 2 – скелет Android-приложения – часть 1 [Электронный ресурс] // Хабр – Режим доступа: <https://habr.com/ru/post/434498/>. – Дата доступа: 01.04.2020.
5. The «Real» Repository Pattern in Android [Электронный ресурс] // Medium. – Режим доступа: <https://proandroiddev.com/the-real-repository-pattern-in-android-efba8662b754>. – Дата доступа: 01.04.2020.