

АРХИТЕКТУРА МОБИЛЬНОГО МЕССЕНДЖЕРА НА ПЛАТФОРМЕ ANDROID

Белко А.Л.

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

Гусев А.П. – д-р техн. наук

Целью работы является эргономическое проектирование мобильного мессенджера на платформе Android. Мессенджеры стали популярным средством межличностной коммуникации – это обусловлено ростом числа пользователей смартфонов, мобильного Интернета и мобильных приложений в целом.

Существует достаточно много подходов для построения сложных систем с хорошей архитектурой. Несмотря на небольшие различия этих подходов, у них много общего. Разумеется, они все задают способы разбиения приложения на отдельные модули. При этом в каждой системе как минимум есть модули, содержащие бизнес-логику приложения, и модули для отображения данных.

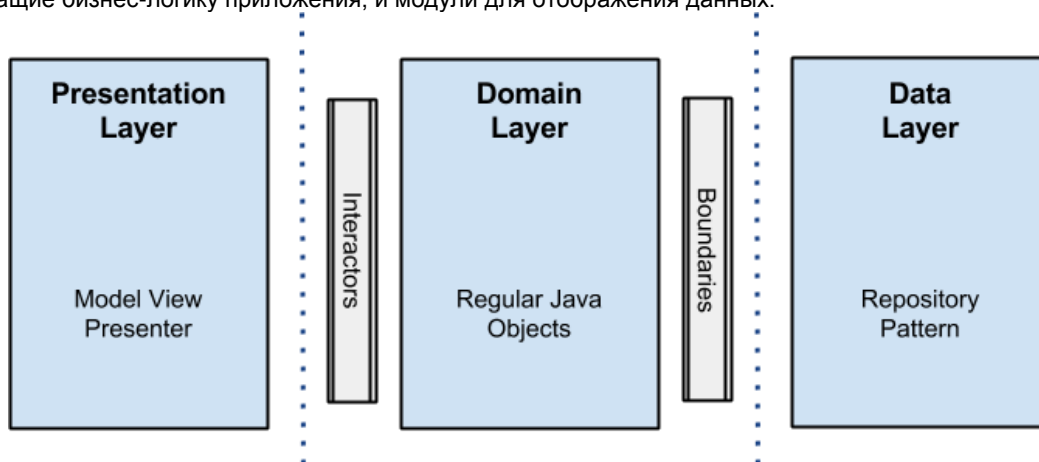


Рисунок 1 – Схема архитектуры мобильного приложения на платформе Android

Уровень представления (Presentation Layer) — это пользовательский уровень, графический интерфейс, который фиксирует события пользователя и показывает ему результаты. Он также выполняет проверку того, что во введенных пользователем данных нет ошибок форматирования, а отображаемые данные отображаются корректно. Эти операции разделены между уровнем пользовательского интерфейса и уровнем ViewModel:

- уровень пользовательского интерфейса содержит Activity и фрагменты, фиксирующие пользовательские события и отображающие данные;
- уровень ViewModel форматирует данные так, что пользовательский интерфейс показывает их определенным образом.

Уровень бизнес-логики (Domain Layer) — на этом уровне находятся все бизнес-требования, которым должно соответствовать приложение. Для этого здесь выполняются необходимые операции.

Уровень данных (Data Layer) — на этом уровне находятся данные и способ доступа к ним. Эти операции разделены между уровнем репозитория и уровнем источника данных:

- уровень репозитория реализует логику доступа к данным. Его ответственность заключается в том, чтобы получить данные. Необходимо проверить, где искать их в определенный момент. Например, вы можете сначала проверить локальную базу данных и, если там данных нет, сделать запрос к API и сохранить данные в базу данных. То есть он определяет способ доступа к данным;
- уровень источника данных отвечает непосредственно за получение данных.

В процессе реализации проекта было выполнено эргономическое проектирование и разработка мобильного мессенджера на платформе Android с использованием Java и Kotlin в качестве языков программирования.

Список использованных источников:

1. *Architecting Android. The clean way?* [Электронный ресурс]. – Режим доступа: <https://fernandocejas.com/2014/09/03/architecting-android-the-clean-way>
2. *Android-CleanArchitecture* [Электронный ресурс]. – Режим доступа: <https://github.com/android10/Android-CleanArchitecture/>