Задача:

Реализовать функционал для просмотра картинок.

Функциональные требования

- На экране будет только 2 кнопки: “Показать котика” и “Показать утку”.
- После нажатия на кнопку будет отображаться соответствующая картинка.
- Во время первого нажатия кнопки сдвигаются, и на этом месте отображается фото.

Технические требования по заданию

- Для получения фото нужно производить запросы на один из двух API.
- Для получения фото котика воспользоваться документацией API котиков.
- Для получения фото уток воспользоваться документацией для API уточек и делать запросы на endpoint **/random**.
- После получения json от API взять поле с картинкой и загрузить через предложенную ниже библиотеку.

Общие технические требования

- Использовать только стек, изложенный в соответствующем разделе.
- Придерживаться одного стиля при именовании переменных, методов и классов.
- Давать смысловые названия переменным, методам и классам.
- Разрешается пользоваться Android Architecture Components.

Что надо знать и что стоит использовать, чтобы справиться с заданием?

- Язык программирования: Kotlin.
- DI (опционально): Koin (Dagger2)*.
- Архитектура слоя представления (опционально): MVVM (MVP)*.
- Многопоточная работа: Coroutines (RxJava)*.
- Работа с сетью: Retrofit.
- Загрузка фото: Picasso / Glide / Coil

*В скобках указана альтернатива.

Задача для тех, кто выполнил первое задание и готов повысить уровень сложности:

Теперь картинки котиков и уточек можно лайкать и сохранять для будущего просмотра (как сохраненные посты в инстаграме).

Для реализации предлагается два варианта на выбор:

1. В любом углу картинки расположить кнопку лайка (не выходя за границы самой картинки).
2. Лайк будет ставиться после двойного нажатия по картинке.

После лайка будет отображаться Toast с сообщением о сохраненной картинке.

Картинку можно будет посмотреть на другом экране.
Переход на экран может осуществляться любым способом:

- через Bottom navigation:
- через Tabs;
- через App bars: bottom;
- и т.п.

Экран с понравившимися картинками должен представлять собой список изображений (по желанию можно добавить дату лайка).

По желанию на экране списка картинок можно добавить:

- Кнопку для очистки списка.
- Возможность снимать лайк (соответственно, удалять из списка в этот же момент).

Технические требования по заданию

- После того, как мы лайкнули картинку, она должна сохраняться в базу данных.
- Если будет выполняться функционал “по желанию”, должны производиться запросы в базу данных на удаление. Фильтрации списков не принимаются.
- Можно использовать хранилище данных, реализующее паттерн Observer (LiveData, варианты хранилищ из RxJava и т.п.).

Дополнительный стек технологий:

- База данных: Room.
- Можно получить дополнительные данные, если реализовать в проекте Single Activity.