1. Предложить варианты архитектуры проекта для возможности командной работы и мультипроектной сборки

2. Написать основу приложения:

 - Главный экран, кнопка подключения и меню

 - Кнопка подключения - по нажатию мы имитируем установку соединения, кнопка анимирована пульсацией, время соединения - по евенту таймера.

 - При клике по элементу меню (например About) - перейти на новый экран и показать TableView/RecycleView со списком стран с отображением ячейки - флаг в круге, страна. При выборе элемента из списка - закрыть окно и отобразить его на первом окне (другими словами - идем на другой экран, выбираем страну, отображаем выбор с флагом на изначальном экране)

3. Предложить вариант логгирования ошибок в приложении, телеметрии (кроме Firebase/Crashlytics - предположим что они запрещены политикой компании)

4. Результат - на github

Тест в первую очередь на способность применить основы, реального кода почти быть не должно. Прошу честно написать там, где нет опыта и нет представления о чем речь.

Так же честно написать сколько времени ушло на задание, какие части пришлось гуглить а какие нет.

1.Предлагаю MVVM для архитектуры проекта, которая даст нам возможность отделить View от Бизнес Логики, благодаря чему командная работа будет более эффективной.

2. Создам Main Activity, на котором установлю Action bar и button

Кнопку анимированной пульсации сделаю таким образом։

Private void pulseAnimation(){

ObjectAnimator scaleDown = ObjectAnimator.ofPropertyValuesHolder(ImageView,

PropertyValuesHolder.ofFloat("scaleX", 0.5f),

PropertyValuesHolder.ofFloat("scaleY", 0.5f));

scaleDown.setDuration(300);

scaleDown.setRepeatCount(ObjectAnimator.INFINITE);

scaleDown.setRepeatMode(ObjectAnimator.REVERSE);

scaleDown.start();

}

Создам json, в которой будем иметь id (drawable) флагов стран.

В Меню, в случае нажатия элемента About, откроем FlagActivity (startActivtyForResult), в котором будем иметь LiveData<List<Country>> countries (с помощью RecyclerView покажем флаги стран). Из ViewModel-а получим countries и сделаем observable, а внутри ViewModel есть LiveData<List<Country>>, которую получим из DBModel. В DBModel, используя gson, возьмем Data из json и преобразуем в List<Country>. При выборе флага страны, с помощью setResult передадим country Main Activity и покажем флаг.

3. Кроме Firebase не использовал других sdk для получения crash.

4.

На задание ушло 50 минут. Использовал Google для кода анимированной пульсации.