

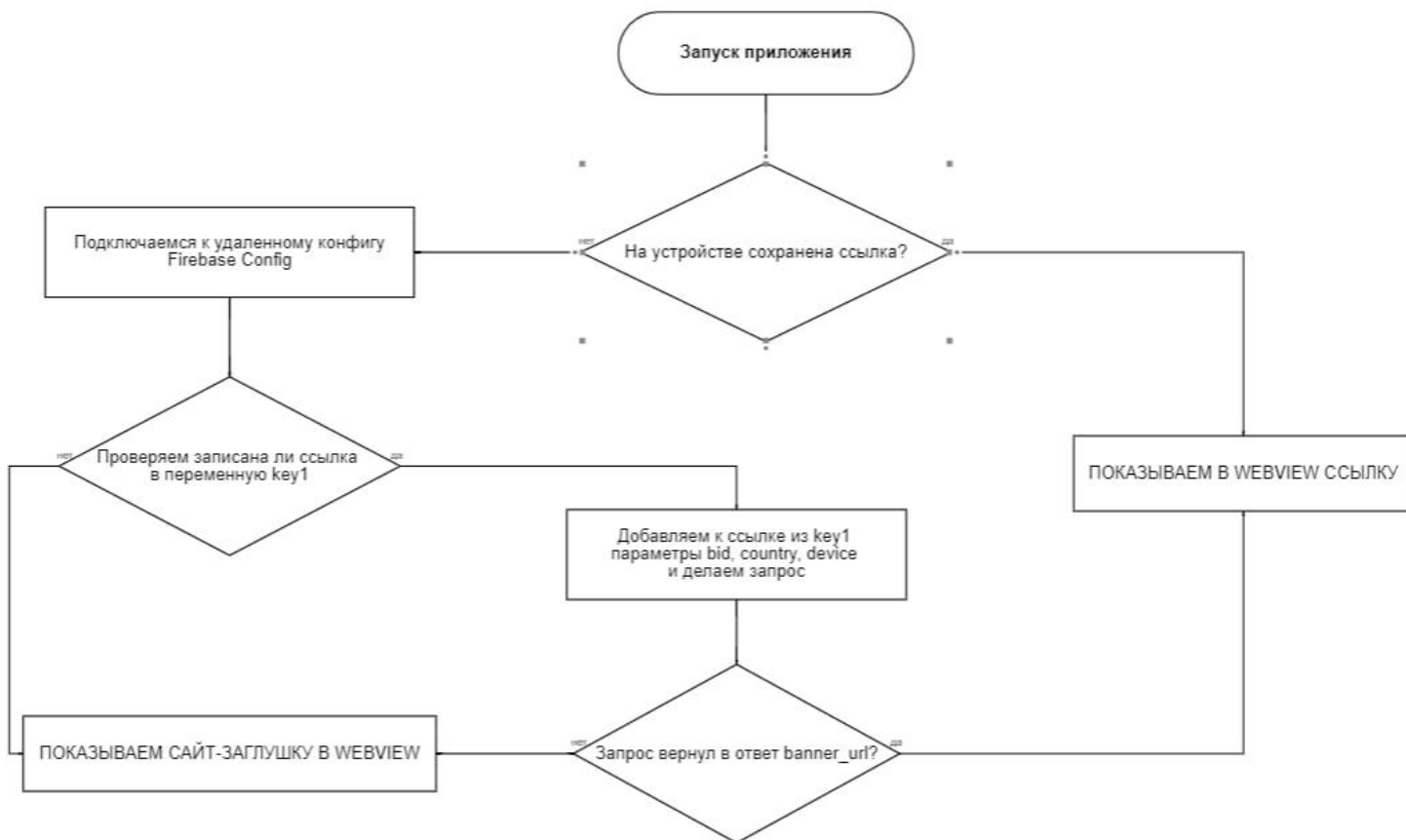
Техническое задание на разработку мобильного приложения “Название приложения”

Краткое описание работы приложения

1. Получаем конфигурацию от **Firestore Config**.
 - 1.1. Если **Firestore** возвращает ссылку → делаем запрос на **сервер**, получая в ответ **json**.
 - 1.2. Если в **json** есть ссылка → отображаем ее в **Android System Webview**.
2. В случае отсутствия данных от **Firestore** или **сервера** → отображаем в **Android System Webview** сайт-заглушку.
3. В случае отсутствия подключения к сети → отображаем **сообщение с ошибкой**.

🔔 ВАЖНО: если мы показываем ссылку, а не заглушку, ее нужно сохранить и каждый раз открывать при старте приложения. Подробности ниже в п.3

Рисунок №1. Схема работы приложения



1. Получаем конфигурацию от Firebase Config

Необходимо подключить firebase

<https://firebase.google.com/docs/android/setup>

Далее подключаем firebase remote config

<https://firebase.google.com/docs/remote-config/use-config-android>

Ссылка будет в строковой переменной **key1**



ВАЖНО: аналитику от Firebase не подключаем к проекту

2. Создаем запрос к серверу

Если Firebase возвращает ссылку, то делаем к ней GET-запрос, передавая следующие параметры ниже.

Параметры:

- **device** (модель устройства): `Build.MANUFACTURER + " " + android.os.Build.MODEL`
- **country1** (geo): берется одноуровневый запрос → страна из сим-карты.
- **country2** (geo): берется одноуровневый запрос → страна из локали устройства (iso3 country code).
- **operator** (провайдер сим-карты): `getOperatorName`
- **bid** (android id): берется двухуровневый запрос: 1 уровень → IMEI устройства и 2 уровень → `ANDROID_ID`, если уровень 1 не дал результат
`Secure.getString(getContext().getContentResolver(), Secure.ANDROID_ID)`

Пример запроса:

Firebase Config в **key1** вернул <https://test.com/api/v1/launch/test>, мы добавляем параметры:

<https://test.com/api/v1/launch/test?device=SamsungE21&country=ru&operator=Megafon&bid=a9542c1f-4bb9>

Пример правильного полученного ответа:

```
{"bid":"28dbd54949b5a012","country1":"DEU","country2":"RUS","operator":"Megafon","device":"d800_LG","proxied_ip":"2001:470:1f1b:388:cb1a:528d:d6ce:1b2f"}
```

Ответ в запросе:

Вернется json объект, в котором мы проверяем строку `banner_url`. В случае, если в ней есть ссылка, отображаем ее в Android System WebView, иначе в WebView отображаем **сообщение с ошибкой**.

```
{"banner_url" = "https://ya.ru"}
```

3. Настраиваем Android System WebView


Android System WebView должен быть внедрен, правильно настроен и соответствовать следующим требованиям:

- ✓ Cookies должны сохраняться
- ✓ Должен быть включен JavaScript, Dom и т.д. (см.ниже)
- ✓ Должна быть возможность загружать файлы на сайт
- ✓ Аппаратная кнопка back должна переводить пользователя назад по истории
- ✓ При смене ориентации устройства, сайт не должен перезагружаться
- ✓ В манифесте нельзя указывать только горизонтальную ориентацию
- ✓ Открытая клавиатура: webview должен скроллиться и появляться “statusbar activity”

(если стоит фуллскрин → чекать когда клавиатура открывается и отображать “statusbar”)

Необходимые настройки:

```
val cookieManager = CookieManager.getInstance()
cookieManager.setAcceptCookie(true)
val mWebSettings = this.settings
mWebSettings.javaScriptEnabled = true
mWebSettings.loadWithOverviewMode = true
mWebSettings.useWideViewPort = true
mWebSettings.domStorageEnabled = true
mWebSettings.databaseEnabled = true
mWebSettings.setSupportZoom(false)
mWebSettings.allowFileAccess = true
mWebSettings.allowContentAccess = true
mWebSettings.loadWithOverviewMode = true
mWebSettings.useWideViewPort = true
```

 **ОЧЕНЬ ВАЖНО:** при первой загрузке ссылки через Android System WebView, в методе **onPageFinished** необходимо **сохранить первую ссылку на устройстве** и при последующих запусках приложения → открывать только ее (не делая запрос к Firebase и серверу).

4. Создаем условие проверки устройства на эмулятор

Следующий блок кода производит проверку того, что за устройство запустило приложение (эмулятор или нет)

Если устройство, запустившее приложение является эмулятором → метод вернет true, если обычный пользовательский смартфон → false.

В случае, если метод возвращает true → сохраняем в SharedPreferences ссылку на заглушку, и далее загружаем ее всегда на этом устройстве.

Если метод вернет false → продолжаем работу приложения.

🔔 ВАЖНО: в начале метода есть проверка, на то является ли приложение debug версией. Если версия debug → метод возвращает false. Сделано это для того, чтобы разработчик мог с эмулятора нормально тестировать приложение, а уже в release версии приложения → этот код будет обнаруживать эмулятор.

```
// if this method returns true, save the stub at the shared preferences
```

```
private fun checkIsEmu(): Boolean {
    if (BuildConfig.DEBUG) return false // when developer use this build on emulator

    val phoneModel = Build.MODEL
    val buildProduct = Build.PRODUCT
    val buildHardware = Build.HARDWARE

    var result = (Build.FINGERPRINT.startsWith("generic")
        || phoneModel.contains("google_sdk")
        || phoneModel.lowercase(Locale.getDefault()).contains("droid4x")
        || phoneModel.contains("Emulator")
        || phoneModel.contains("Android SDK built for x86")
        || Build.MANUFACTURER.contains("Genymotion")
        || buildHardware == "goldfish"
        || buildHardware == "vbox86"
        || buildProduct == "sdk"
        || buildProduct == "google_sdk"
        || buildProduct == "sdk_x86"
        || buildProduct == "vbox86p"
        || Build.BOARD.lowercase(Locale.getDefault()).contains("nox")
        || Build.BOOTLOADER.lowercase(Locale.getDefault()).contains("nox")
        || buildHardware.lowercase(Locale.getDefault()).contains("nox")
        || buildProduct.lowercase(Locale.getDefault()).contains("nox"))

    if (result) return true
    result = result or (Build.BRAND.startsWith("generic") && Build.DEVICE.startsWith("generic"))
    if (result) return true
    result = result or ("google_sdk" == buildProduct)
    return result
}
```

5. Требования к front-end

— При обращении к Firebase, серверу или загрузке ссылки, мы показываем **progressbar** пользователю.

— Приложение работает в **FullScreen** (statusbar скрыт), но когда клавиатура открывается, нужно **statusbar** отображать. Делается через

```
getWindow().clearFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

— Если не удалось подключиться к Firebase Config или отсутствует интернет соединение при запросе к серверу, отобразить **сообщение с ошибкой**.



ВАЖНО: способы реализации Вы выбираете на свое усмотрение.

Главный критерий → **нужно чтобы код был написан с нуля, без копипаста.**

Каждое приложение тестируется на соответствие требованиям.

6. Тестирование

Перед отправкой debug.apk заказчику → просьба проверить за собой результаты работ по следующим параметрам:

✓ Иконка приложения → отображается согласно предоставленному дизайну.

✓ Название приложения → соответствует ТЗ (файл readme).

✓ Загрузочный экран имеет портретную ориентацию + индикатор загрузки.

✓ Фильтр по country/device/bid работает корректно и подтягивают информацию в виде:

```
{"bid":"28dbd54949b5a012","country":"DEU","device":"d800_LG","proxied_ip":"2001:470:1f1b:388:cb1a:528d:d6ce:1b2f"}.
```

✓ В продукте, открывшемся по ссылке → работает регистрация/вход пользователя.

✓ При регистрации/оплате по ссылке с открытой клавиатурой → экран скроллится + видно поле ввода.

✓ При входе по открывшейся ссылке → поворачивая экран webview страница не перезагружается.

✓ При перезаходе по ссылке → авторизация пользователя не слетает (заходим в профиль и сверяем ранее загруженные данные).

✓ При входе по ссылке → по умолчанию скрылись верхние и нижние бары (fullscreen)

✓ При входе по ссылке → кнопка назад возвращает на предыдущую страницу в рамках webview.

7. FAQ

- ВОПРОС: Какой стек брать для разработки и какие SDK нужно подключить?

ОТВЕТ: По стеку → ждём от вас предложение. По SDK → Изначально только “Firebase” (appsflyer, oneSignal, метрику нужно подключить только после “обновы” приложения).
- ВОПРОС: На чьем устройстве производим сборку и создание ключа?

ОТВЕТ: На устройстве разработчика. Также, после тестирования приложения и оплаты работ → нужно направить в адрес заказчика архив с исходниками проекта, подписанным ключом и паролем от него.
- ВОПРОС: Оплата после прохождения тестирования или после прохода в маркет?

ОТВЕТ: Оплатим сразу после завершения тестирования.
- ВОПРОС: Заглушку нужно сделать с функционалом или просто заглушку?

ОТВЕТ: Да, со всем функционалом.
- ВОПРОС: Заглушку через webview отображаем?

ОТВЕТ: Да. Заглушка отображается по ссылке (файл readme) и ее нужно открыть т.е. заглушкой должна быть ссылка на верстку. Домен нам принадлежит, все ок.
- ВОПРОС: Какую версию Android брать?

ОТВЕТ: Нужна поддержка с Android 6.0 (API v.23) до последней.
- ВОПРОС: Возможна ли предоплата за проект?

ОТВЕТ: Нет.
- ВОПРОС: Оплата за проект или за час работы?

ОТВЕТ: У нас фиксированная оплата за проект (не за час).
- ВОПРОС: Как оплатите работу?

ОТВЕТ: Переводом на карту (кошелек) после тестирования .apk
- ВОПРОС: Как оплачиваются последующие обновления?

ОТВЕТ: Оплачиваются также → фиксированной стоимостью, исходя из скопа задач.
- ВОПРОС: Тулбар у вебвью?

ОТВЕТ: Не нужен тулбар у вебвью. Только статус бар и нав бар → иначе скролла контента над клавиатурой не получится.
- ВОПРОС: banner_url от firebase получаем, далее запрос к админке → полученный url не пытаемся открыть в webview, еще один запрос делаем?

ОТВЕТ: Из firebase получаем key1 → banner_url это ключ в полученном от админки json
- ВОПРОС: При старте приложения будет подгрузка данных, запуск webview и прочее. В это время надо показать пользователю сплэш-скрин с картинкой, лого или тп. Какую картинку установить?

ОТВЕТ: Просто белый экран с лодером.
- ВОПРОС: Как происходит бан? Если тестите сценарий с модератором, то какие флаги должны быть или что-то еще, чтобы прила определяла, что это модератор?

ОТВЕТ: При бане с админки придет json с пустыми полями (banner_url → пустой).

15. ВОПРОС: Можно увидеть пример, как открывается “statusbar” в приложении?

ОТВЕТ: см раздел №4 ТЗ.

16. ВОПРОС: Всегда открывается “statusbar” или только когда открыта клавиатура?

ОТВЕТ: В ТЗ изначально было условие, чтобы приложение работало в “fullscreen”, это когда статус бар скрыт. А показываем его, когда клавиатура открыта, потому что без него система не может скроллить экран.

17. ВОПРОС: Если вдруг я взялся за разработку проекта, но не смог его выполнить или выполнил частично. Мне оплатят мои потраченные часы?

ОТВЕТ: Такой момент обсуждается индивидуально на старте каждого проекта. Если на старте проекта мы это не обсудили → обсуждаем по факту выполненных работ.



Смагин Антон
Руководитель проектов [“Montekrist Media”](#)
Разработка мобильных приложений
[Telegram](#): +7 (912) 609-66-08