

Техническое задание

Задача:

1. Сверстать приложение и прикрутить логику работы приложения
<https://www.figma.com/file/6XDtYTyIqXeNDJ8eyO0Qq0/date-and-flirt>
2. Внедрить WebView в приложение, которое будет показываться определенным пользователям

Логика работы приложения:

1. Пользователь скачивает и устанавливает приложение
2. Пользователь первый раз запускает приложение
3. Скачивается JSON файл с определенными данными
4. Если параметры устройства соответствуют данным из JSON, то показываем такому пользователю WebView с ссылкой из JSON
5. Если пользователь не подходит под эти параметры, то показываем ему приложение
6. При повторных запусках ориентируемся на первый запуск. То есть при первом запуске показали WebView, значит только его и показываем при следующих запусках.

Первый запуск приложения пользователя

При первом запуске приложения необходимо скачать JSON файл и посмотреть текущие параметры, по которым решаем, что показывать пользователю (WebView или Приложение).

Параметры отображения WebView:

1. Без браузерной строки и других элементов дающих пользователю понять, что это не нативное приложение, а вебвью
2. Строго зафиксировано по вертикали

JSON: https://website-tracker.com/params_dating.json

```
{
  "stopping": "true",
  "stop_brand": [
    "Google"
  ],
  "stop_isp": [
    "Tele2",
    "Beeline"
  ],
  "target_lang": [
    "rus",
    "eng"
  ],
  "target_country": [
    "RU"
  ],
  "url": "https://mail.ru/"
}
```

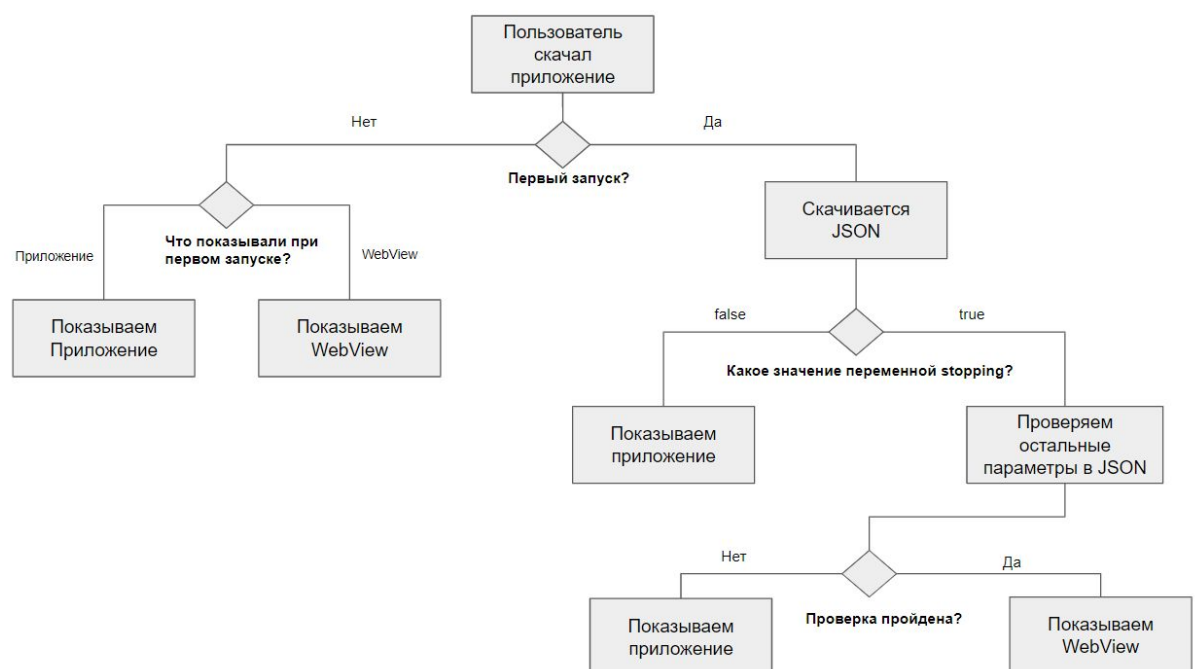
Рассмотрим все параметры JSON:

Параметр	Описание параметра	
Stopping	true - делаем проверку параметров, переходим к следующему пункту	false - проверку параметров не проводим, показываем пользователю сразу приложение из гитхаба
stop_brand	Если содержащийся в параметре бренд устройства совпадает с брендом пользователя, то прекращаем проверку и показываем приложение.	

	Если совпадений нет, то переходим к проверке следующего параметра.
stop_isp	Если содержащийся в параметре мобильный оператор совпадает с моб оператором пользователя, то прекращаем проверку и показываем приложение. Если совпадений нет, то переходим к проверке следующего параметра.
target_lang	Если язык устройства совпадает с параметром из JSON, то показываем Webview. Если нет, то приложение
target_country	Если страна устройства совпадает с параметром из JSON, то показываем WebView. Если нет, то приложение
url	Если все проверки успешно пройдены, то показываем WebView с указанной ссылкой.

Исключение: Если пользователь использует VPN, то показываем ему приложение, даже если он успешно прошел все проверки.

Схематичное представление работы приложения:



Ссылка на картинку: <https://prnt.sc/slsiqf>

Повторный запуск приложения

При повторном запуске приложения необходимо ориентироваться на результат при первом запуске.

Если пользователь прошел проверку при первом запуске, то при следующих запусках ему сразу необходимо показывать WebView. При этом при повторном запуске необходимо продолжать текущую сессию в браузере WebView.

То есть не нужно повторно читать JSON и брать из него URL. **Ссылка в JSON будет редиректить при каждом заходе на новый сайт. Поэтому нам важно чтобы пользователь оставался на том сайте, который посетил первый раз - конечном URL после редиректа.** Схематичное представление:

WebView -> сайт с редиректом -> сайт с редиректом №2 -> сайт с редиректом №3 -> сайт с редиректом №... -> конечный сайт.

Необходимо выделить домен последнего редиректа и запомнить его. (при этом необходимо учесть, что конечный сайт может иметь субдомен, выделять необходимо в любом случае только домен).

При повторном запуске приложения необходимо открывать этот домен.

Дополнительные функции

Необходимо интегрировать пуш уведомления для пользователей:

Отправлять сообщение каждый день по 5 в день со следующим содержанием:

1. "XX women looking for dates in 3 km area near you!" - вместо хх поставляем случайное значение от 20 до 80
2. Somebody liked you - take a look!
3. Your matches are waiting for your messages!
4. You receive a Like - check out!
5. You have received a new message!

Сообщения рассылаются в рандомной последовательности в:

- 10:XX
- 13:xx
- 18:xx
- 20:xx

- 22:xx

вместо XX подставляем случайное значение от 0 до 59

Логика работы экранов

Screen 0.

Welcome экран. Функционал отсутствует.

Screen 1.

Экран, который видит пользователь при запуске. У пользователя есть возможность зарегистрироваться и залогиниться в приложение.

Регистрацию необходимо настроить через firebase

Screen 2

Экран для регистрации пользователя.

Активация кнопки Sign up -> Введенные пароли должны совпадать. Галочка на против terms and conditions должна стоять. Email должен быть введен в правильном формате (в формате email)

Screen 3.

Узнаем пол пользователя. Данные записывать в БД

Screen 4.

Узнаем кого ищет пользователь. Данные записывать в БД

Screen 5.

Узнаем дату рождения пользователя. Данные записывать в БД

Screen 6.

Даем пользователю возможность загрузить фото к профилю. Фото загружаем в БД

Screen 7.

В зависимости от того, кого ищет пользователь показываем ему пары. Данные берутся из БД (фотографии + описание, возраст и расстояние до пользователя).

Screen 8.

Функционал отсутствует

Интегрировать myTracker SDK:

<https://tracker.my.com/docs/sdk/android/api> - задача передавать установки в трекер
SDK ключ 91295278886155800759