

Техническое задание на разработку мобильного приложения IOS

Настоящее техническое задание описывает требования к разработке мобильного приложения для платформы IOS.

Описание приложения

Дизайн приложения :

[https://www.figma.com/file/MyELJL3DGnYFee5q2fmyr5/Melon-App-\(IOS\)?type=design&node-id=21-7&mode=design&t=QFVKO49ela4Dypfl-0](https://www.figma.com/file/MyELJL3DGnYFee5q2fmyr5/Melon-App-(IOS)?type=design&node-id=21-7&mode=design&t=QFVKO49ela4Dypfl-0)

Приложение представляет собой простейший клиент, который позволяет находить и скачивать нужные моды на устройство через REST API. Возможна покупка подписки.

Необходимо добавить Firebase & GA4. Реализовать отслеживание конверсий.

Интегрировать <https://qonversion.io> для аналитики подписок

Интегрировать Affise sdk + отслеживание ивентов

Экраны

1 - Экран загрузки

2 - Экран проверки подписки, показывается при необходимости (отсутствие или окончание подписки) 2 разновидность экрана, пояснение в ниже в тз

3 - Основной экран с модами

4 - Экран мода

5 - Экран меню с возможностью выбора категорий, политики конфиденциальности и условий пользования. Также активная кнопка подписки, если она не оформлена

Также необходимо реализовать вызов окна оценки приложения

Важно: при первом запуске требуется инициировать окно запроса доступа к idfa, также у каждого экрана подписки должна быть кнопка restore, а также окно оценки приложения.

Требования к технической реализации

Приложение должно быть разработано на языке программирования Swift.

Приложение должно поддерживать как смартфоны, так и планшеты.



Бизнес логика

Экран загрузки (1)

Показывается при старте приложения всегда

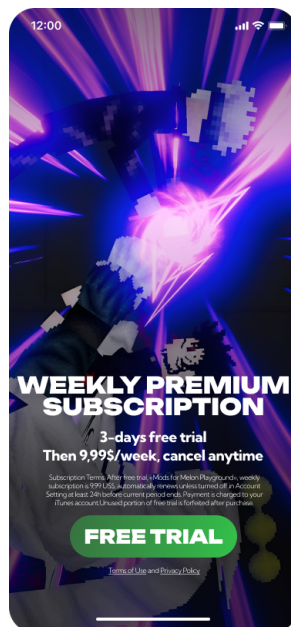
На этом экране происходит логика проверки подписки, или окончания пробного периода. Исходя из этого происходит 2 возможных перехода (должен происходить плавный fade переход):

- На основной экран
- На экран подписки (Возможны 2 варианта экрана подписки, пояснение далее в Экран подписки)

Важно чтоб экран не исчезал быстро, пользователь должен увидеть интерактивную загрузку не менее секунды

Экран подписки(2) (2 возможных вариации)

1 Вариация - Стандартная, необходима для прохождения модерации в App Store



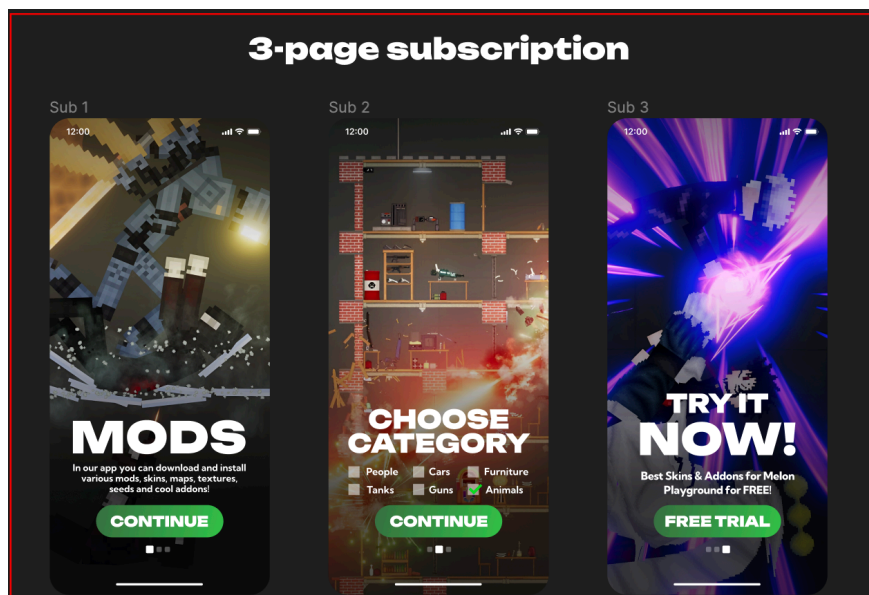
Кнопка закрыть (регулируется через фаербейс). Дает возможность пропустить экран, но в этом случае возможность скачать необходимый мод будет отсутствовать(логика будет описана на экране мода).

Кнопка индикатор прокрутки влево в этой вариации экрана отсутствует

При нажатии free trial должен оформляться демо период с переходом на основной экран

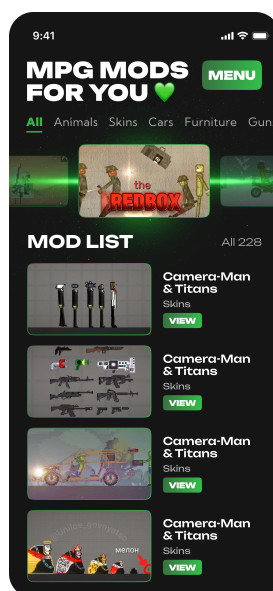
1 Расширенная - Активируется после модерации установкой нужной галочки на firebase

Представляет собой 3 однотипные страницы, переход по которым должен осуществляться либо нажатием continue, либо свайпами всего экрана



По итогу попадаем на правый экран, логика работы аналогична первой (модерационной) версии экрана

Основной экран(3)



Представляет собой экран категориями содержащими моды
Категории прокручиваются до окончания

Приложение стартует с нажатой крайней левой категорией и ее данными, по тапу других категорий происходит загрузка их данных(модов)

Загрузка данных должна имитировать бесконечную загрузку с докачкой данных через limit offset (описание ниже)

По тапу на мод будет происходить открытие экрана с детальным описанием мода(экран 4)

По тапу на кнопку информации(слева), будет происходить переход на экран с информацией по установке мода(экран 7)

По тапу на бургер меню будет происходить переход на (экран 5)

Список названий категорий (в апи):

korabli

zhivotnye

lyudi-i-personazhi
mashiny
mebel
oruzhie
paks
samolety

Список названий категорий (в приложении):

Ships
Animals
Skins
Cars
Furniture
Guns
Packs
Aircraft

Сетевая часть

Путь

{base_url}/category/{category_name}

например: category/zhivotnye

Для запроса используются названия категорий выше(без модификация в виде капса и пробелов)

Параметры запроса

limit, offset (При использовании параметров важно учитывать поле count из ответа, оно показывает общее количество модов/сейвов в категории)

Ответ сервера

```
"count": 1, "items": [
```

```
{
```

```
"path": "cornobul-mutantny-oruzhie-veshhi", "title": "Cornobul – мутанты, оружие, вещи",
```

```
"image_url": "https://skinsforbr.com/wp-content/uploads/2022/12/oplot.jpg" }
```

```
]
```

path необходимо прокидывать на экран мода для последующего сетевого запроса

Экран мода(4)

Экран мода представляет собой экран конкретного мода, с его картинкой описанием, и возможностью скачать мод (в виде zip архива)

Картинка и текст берутся из сетевого ответа(ниже)

Кнопка скачать всегда присутствует на экране, но в случае если пользователь пропустил экран 2 кнопкой закрыть, вместо скачивания должен всплывать экран 2 (при отсутствии подписки)

Кнопка назад, возвращает назад

Сетевая часть

Путь

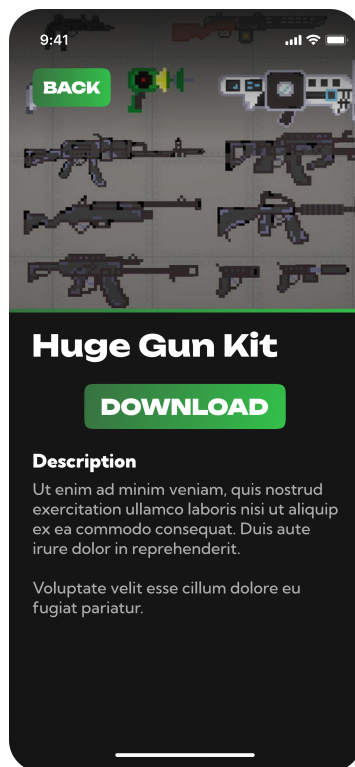
{base_url}/item/{path}

path прокидывается с прошлого экрана например: item/dinozavr-raptor

Ответ сервера

```
{
```

```
"title": "Динозавр — Раптор",
```

```
"image_url": "https://skinsforfbr.com/wp-content/uploads/2022/04/cornobul.jpg",  
"description": "Добавляет хищного динозавра – Раптора. По размерам немного  
больше арбуза. С подобными модами можно сделать классный сценарий  
"zip_url": "https://skinsforfbr.com/downloads/cornobul-mutanty-oruzhie-veshhi.zip"  
}
```

Содержит все необходимые для экрана данные (описание с переносами)
Внешний вид состояния загрузки данных

Экран меню

Возможность выбора необходимой категории. Кнопка Subscribe активна, если триал или подписка не оформлены. Возможность открыть политику или правила (открываются по ссылке в браузере).

Сторонние сервисы

1. *Firebase & GA4* (<https://github.com/firebase/firebase-android-sdk>) Необходимо встроить sdk firebase + GA4 для отслеживания ивентов.

Ивенты: first open, in_app_purchase, app_store_subscription_convert, app_store_subscription_renew, purchase

Также необходимо включить firebase crashlytics.

Необходимые конфиги в firebase:

- Выбор пейвола
- Включение/выключение окна оценки
- Таймер на появление крестика для закрытия подписки
- Прозрачность крестика
- Изменение текста на странице подписки (стоимость и триал), если изменение не происходит автоматически при изменении в appstore

2. *qonversion.io* для аналитики и управления подпиской
(<https://documentation.qonversion.io/docs/ios-sdk-setup>)

3. *Affise sdk* с отслеживанием конверсий
(<https://github.com/affise/sdk-ios>)

Отправляете событие Purchase - event name - Purchase, дополняете его данными:
affise_p_price, affise_p_revenue, affise_p_currency

Тут так же: Отправляете событие Subscribe - event name - Subscribe, дополняете его данными: affise_p_price, affise_p_revenue, affise_p_currency

affise_p_price, affise_p_revenue - могут быть одинаковыми значениями

Обязательно подключить модуль `Affise.moduleStart(.Advertising)`

Добавить костыль для исправления бага в стандартной библиотеке для реализации idfa:

<https://developer.apple.com/forums/thread/690607>

<https://stackoverflow.com/questions/69312640/attrackingmanager-stopped-working-in-ios-15>