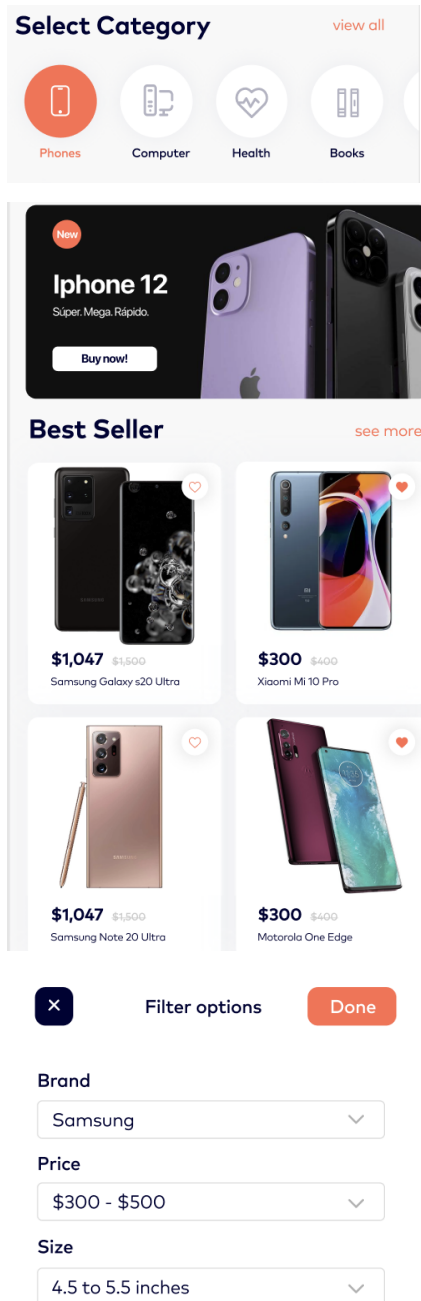


Тестовое задание.

1. Экран «Main»



API:

GET - <https://run.mocky.io/v3/654bd15e-b121-49ba-a588-960956b15175>

Общие требования:

Select Category - При нажатии на иконку она меняет цвет (как по дизайну)
Выбрана и выделенная иконка может быть только одна.

Hot sales - карусель при свайпе вправо меняет элемент

- Изображение;
- Метка New, появляется только на новых товарах;
- Бренд товара;
- Краткое описание;
- Кнопка купить;

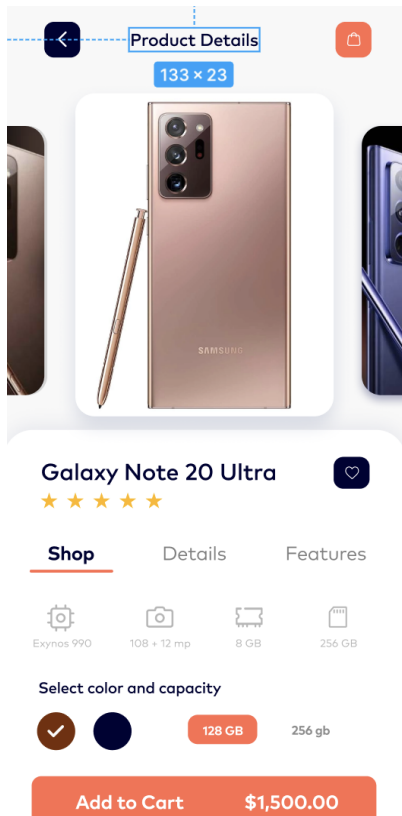
Best Seller - Список товара имеет следующие параметры:

- Название;
- Изображение;
- Цену за шт.: цена с учетом скидки ;
- Цену за шт.: цена без учета скидки ;
- Добавить в избранное ;

Filter options - раскрывающийся список бренда и размера, диапазон цен от 0 до 10000\$
Фильтр открывается по нажатии на кнопку в правом верхнем углу

- Brand
- Price
- Size - только отрисовать как в Figma

2. Экран «Product Details»



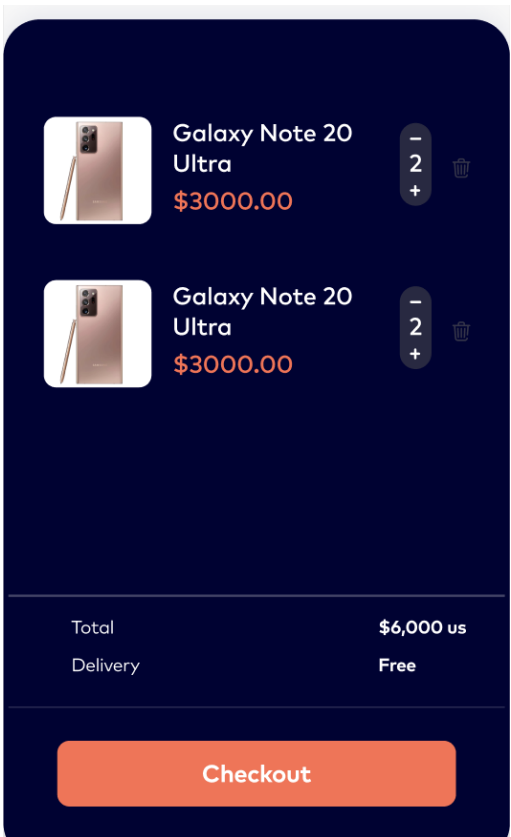
API: (пример только по одной модели телефона, то есть при нажатии на любой телефон, будет всегда открываться Samsung)

GET - <https://run.mocky.io/v3/6c14c560-15c6-4248-b9d2-b4508df7d4f5>

Общие требования:

- Изображение - карусель при свайпе вправо меняет элемент;
- Название;
- Метка favorites;
- Рейтинг товара;
- Цвет товара;
- Детали товара;
- Характеристика;
- Добавить в корзину;

3. Экран «My Cart»



API:

GET - <https://run.mocky.io/v3/53539a72-3c5f-4f30-bbb1-6ca10d42c149>

Функционал
Пользователь видит в тапбаре иконку корзины.
Пользователь видит количество добавленных товаров на иконке в тапбаре при условии, что в корзину добавлен один или более товаров.
Пользователь может перейти в корзину при нажатии на иконку Корзины в тапбаре.

Пользователь видит список добавленных товаров в корзине.

Каждый товар в списке имеет следующие параметры:

- Название;
- Изображение;
- Цену за шт.: цена с учетом скидки ;
- Итоговую цену с учетом количества –данного товара, добавленного в корзину;
- Стоимость доставки;

Рекомендуемый стек технологий:

Android

- Kotlin
- Корутины (но если на RX сможете, тоже отлично)
- Flow или LiveData или RX
- Dagger или Koin
- MVVM
- AdapterDelegates
- Верстка обычная на XML
- Clean Architecture
- Многомодульность (обязательно)

iOS

- Swift
- MVVM + Coordinator
- SwiftUI
- Combine

Flutter

- Dart
- Block (Или любой стейт менеджер)
- Clean Architecture
- Retrofit

Для проверки задания присылайте ссылку на GIT и видео с записью работы приложения

Критерии приемки

- 1) Насколько визуал соответствует фигме
- 2) Насколько чисто и расширяемо написан код, и соответствует SOLID
- 3) Какое архитектурное решение реализовано
- 4) Умение пользоваться основными библиотеками