

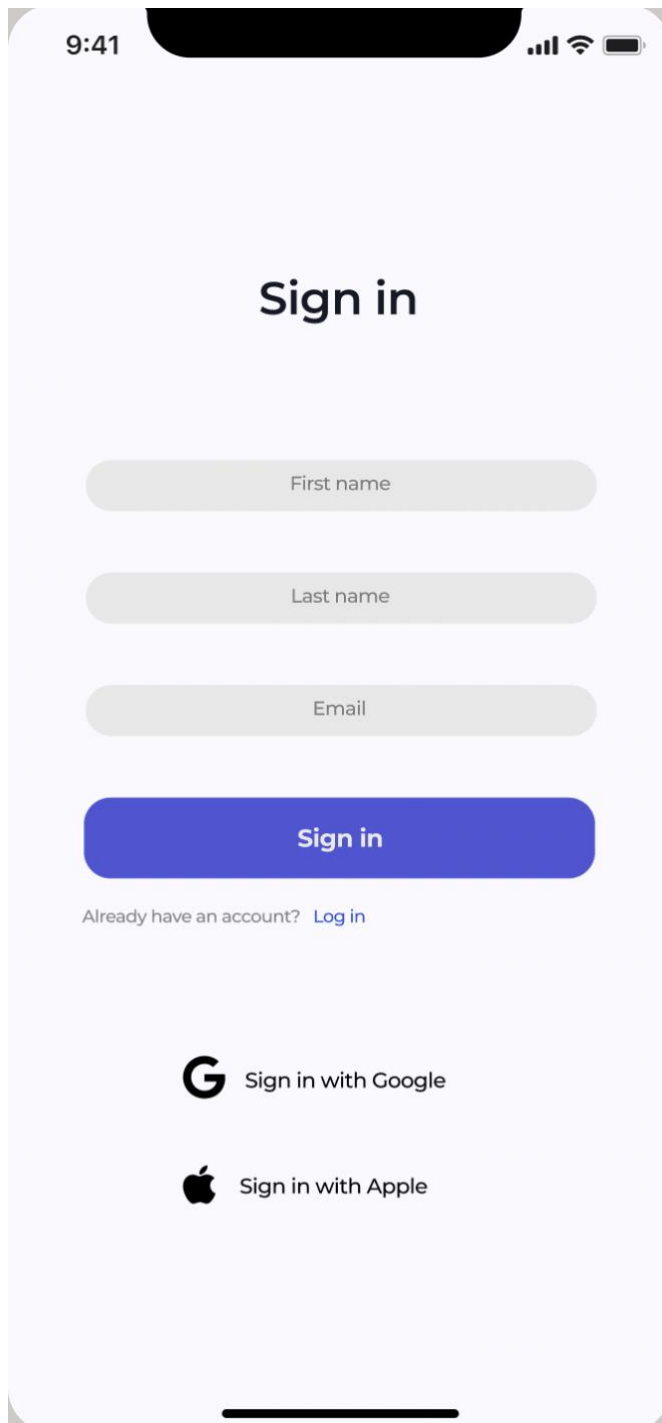
Тестовое задание

Общая информация

- Прототип в Figma (дизайн всех элементов берем оттуда) находится на диске в файле «Прототипы». Экраны нужно самостоятельно перенести в Figma
- Названия экранов в техническом задании соответствуют названию экранов в Figma
- Все шрифты, цвета и иконки можно брать из Figma. Шрифт Montserrat
- **Важно!** Функционал, который помечен «*» обязателен к реализации, а где цвет текста отмечен серым, то реализация по возможности
- Для проверки задания нужно прислать видео работающего приложения, всех его экранов, а также ссылку на GIT
- Понимаем, что тестовое задание объемное. Сделано это для оптимизации процесса найма, так как в прошлый раз было очень много кандидатов
- Было много вопросов "Не просите ли вы меня под видом ТЗ написать вам реальное приложение?". Так как ТЗ объемное, это резонный вопрос. Ответ - нет. Так как тут самые элементарные части, их любой мидл сделает быстрее, чем мы искать будем кого-то. В реальности такие экраны на порядок сложнее.
- На выполнение ТЗ дается 2 недели с момента получения этого файла
- В гите не указывайте название компании, чтобы другие участники не смогли скопировать вашу работу
- Желаем вам успехов!

Описание экранов

Экран «Sign in page» и экран «Login»

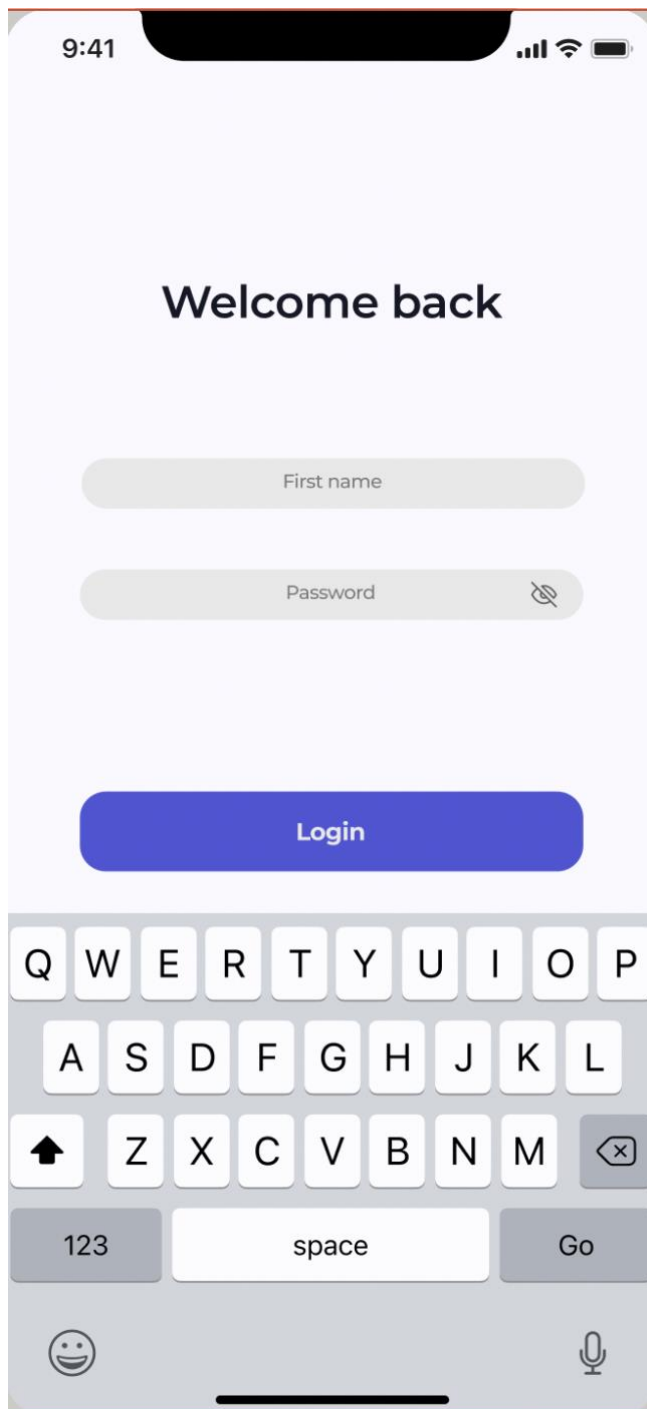


Список элементов:

- Заголовок экрана – брать со скрина
- Поле ввода имени – содержит placeholder «First name».
- Поле ввода фамилии – содержит placeholder «Last name».
- Поле ввода почты – содержит placeholder «Email».
- Кнопка для регистрации аккаунта – при нажатии на кнопку открывается Tab Bar и экран «page 1»

- Кнопка для входа в существующий аккаунт – перед кнопкой должен быть написан текст, уточняющий есть ли у пользователя аккаунт, далее сама кнопка. При ее нажатие осуществляется переход на экран «Login»
- Кнопка для авторизации через Google – слева иконка компании, справа текст, действие при нажатии на кнопку можно не прописывать
- Кнопка для авторизации через Apple – слева иконка компании, справа текст, действие при нажатии на кнопку можно не прописывать

Экран «Login»



Список элементов:

- Заголовок экрана – брать со скрина
- Поле ввода имени – содержит placeholder «First name».
- Поле ввода пароля и кнопка видимости - содержит placeholder «Password». Кнопка видимости располагается справа в поле и по умолчанию имеет состояние «невидимый».
- Кнопка для входа в аккаунт – при нажатии на кнопку открывается Tab Bar и экран «page 1»

ПРИМЕЧАНИЕ:

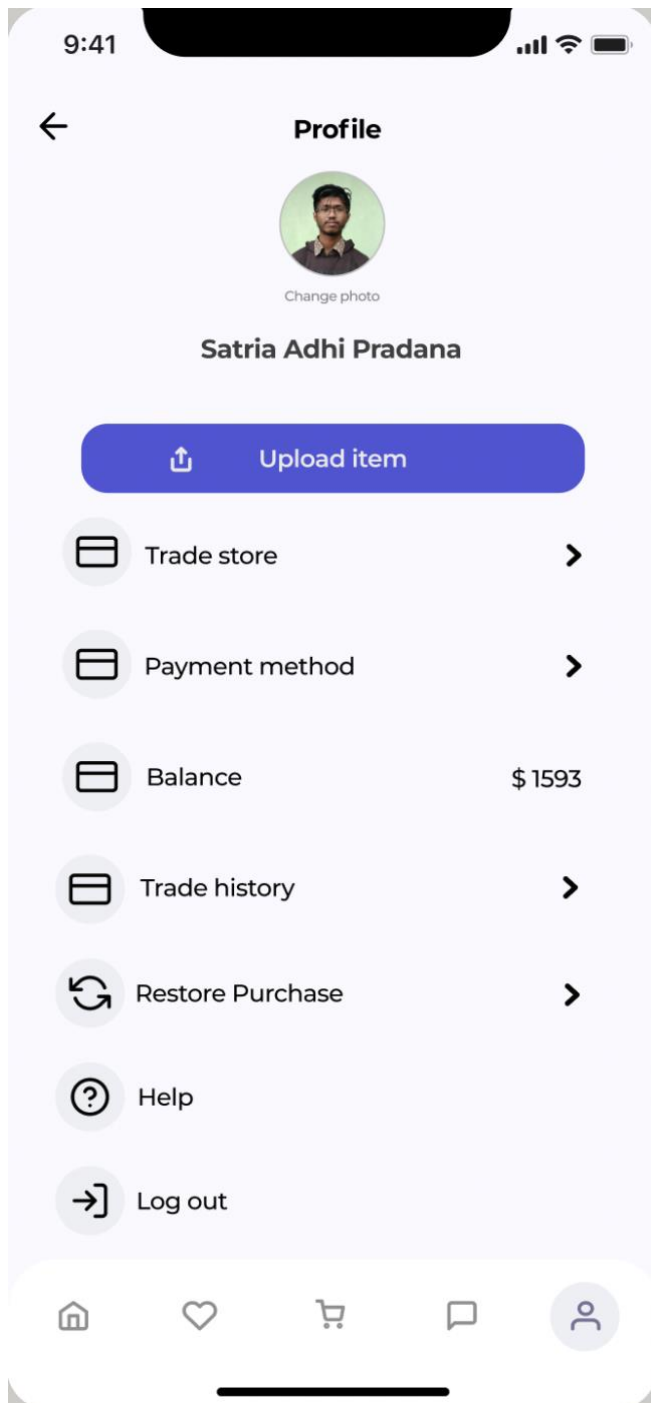
При вводе почты нужно проверить валидность введенных данных. То есть нужно, чтобы написанное пользователем соответствовало некому шаблону, чтобы точно убедиться, что введенные данные это Email. В случае если написанное это не почта, то выдать ошибку валидации. Например: jack-hofman10 – неверно, jack-hofman10@mail.ru – верно.

Сценарии использования:

Пользователь хочет авторизоваться

- Пользователь запускает мобильное приложение. Открывается начальный экран авторизации «sign in page» *
- а. Если у пользователя уже есть аккаунт, то он нажимает кнопку «log in», которая находится под кнопкой «sign in». Открывается экран «Login», где пользователь вводит свои данные и нажимает кнопку «Login». Проверяется есть ли действительно такой аккаунт в базе данных. Если аккаунт есть в базе, то открывается Tab Bar и экран «Page 1», если такого пользователя не существует, то показывается соответствующая ошибка *
- б. Если у пользователя нет аккаунта, то он вводит в соответствующие поля Имя, Фамилию, Email и нажимает кнопку «sign in». Введенные данные сохраняются локально в базу данных. Пользователь также может авторизоваться с помощью аккаунта Google или Apple (сделать только визуально). Открывается Tab Bar и экран «Page 1» *
- с. Если пользователь не помнит, что уже регистрировался и вводит данные для регистрации, которые есть в базе данных, то показывается соответствующая ошибка об уже существующем пользователе с такими данными *

Экран «Profile»



Список элементов:

- Выполнить верстку всех элементов как показано на скриншоте. Имя пользователя, баланс и остальные элементы берем со скриншота.
- Кнопка для загрузки фотографии пользователя – при нажатии на кнопку можно загрузить фотографию пользователя
- Кнопка выхода из аккаунта пользователя – при нажатии выходит из аккаунта (открывается начальный экран «sign in page»)

Сценарии использования:

Пользователь хочет посмотреть информацию в своем профиле

- Пользователь нажимает на иконку с контроллером «Profile» в Tab Bar, чтобы перейти на соответствующий экран *
- На экране выводится вся доступная информация о профиле, управлении им и его настройках *

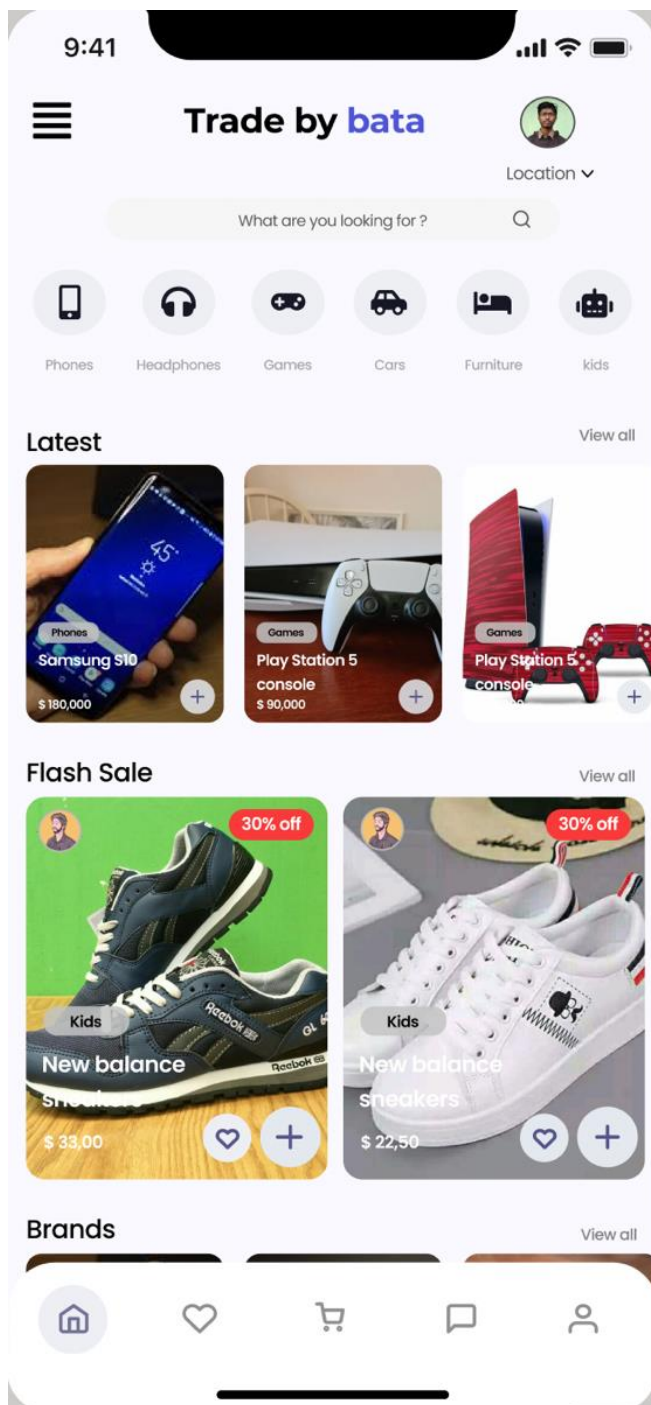
Пользователь хочет выйти из профиля

- Пользователь находится на экране «Profile» *
- Пользователь хочет выйти из профиля и нажимает на кнопку «Log out». Происходит лог аут и открывается начальный экран «sign in page» *

Пользователь хочет обновить фотографию в профиле (выполнение по возможности)

- Пользователь находится на экране «Profile»
- Нажимает на кнопку «Change photo», которая находится под фотографией пользователя. Открывается библиотека фотографий телефона
- Пользователь выбирает фотографию и загружает ее
- Фотография пользователя обновляется

Экран «Page 1» и экран «Page 2»



[API](#) для недавно просмотренных товаров
[API](#) для распродажи товаров

ПРИМЕЧАНИЕ:

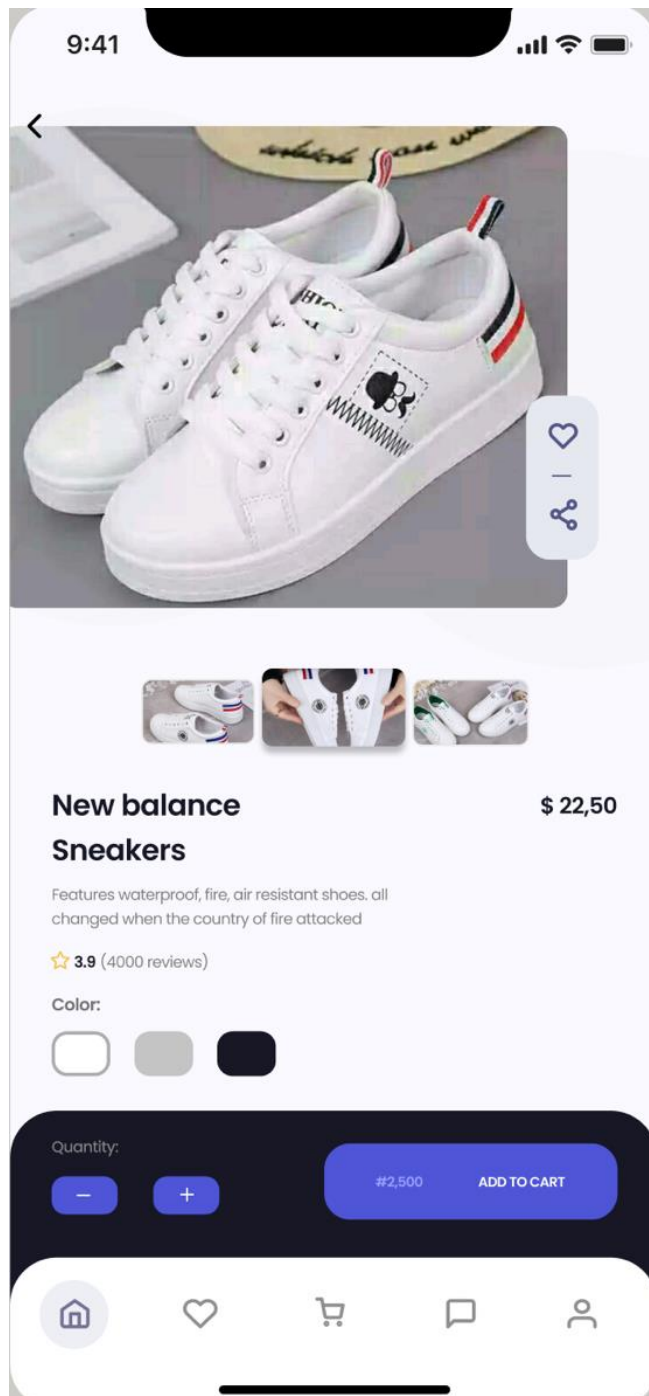
1. Экран «Page 2» реализуется по возможности

2. Данные в блоке *Latest* и *Flash Sale* можно получить с помощью двух разных API, но обработка и дальнейшее отображение данных на экране следует производить СТРОГО после того, когда были получены ВСЕ данные, другими словами, после объединения полученных данных. Например: если пришли данные для *Latest*, но не пришли для *Flash Sale*, то отображение данных в обоих блоках не происходит.

Список элементов:

- Navigation Bar – состоит из кнопок меню слева, title по центру, справа фото пользователя и локацию под фото пользователя
- Поиск по интернет-магазину – это обычный поисковик по приложению для поиска товара. Содержит placeholder, как на скриншоте
- Категории интернет-магазина – это карусель, при свайпе вправо/влево происходит скролл. Все иконки и их названия можно скачать из Figma
- Недавно просмотренные товары – это карусель, при свайпе вправо/влево происходит скролл. Сам блок начинается с названия блока (“Latest”) слева и кнопки просмотреть все (“View all”) справа. Далее каждый товар содержит в себе фото товара, категорию товара, название товара, цену товара и кнопку добавления в корзину. Расположить все элементы как на скриншоте
- Распродажа товаров – это карусель, при свайпе вправо/влево происходит скролл. Сам блок начинается с названия блока (“Flash sale”) слева и кнопки просмотреть все (“View all”) справа. Далее каждый товар содержит в себе фото товара, дисконт товара, категорию товара, название товара, цену товара, кнопку добавить в избранное и кнопку добавления в корзину. Расположить все элементы как на скриншоте
- Бренды – это карусель, при свайпе вправо/влево происходит скролл.

Экран «Page 2» (выполнение по возможности)



Список элементов:

- Кнопка назад – элемент Navigation Controller, при нажатии возвращается на экран «Page 1»
- Фото товара – показывается по одной фотографии. Карусель, при свайпе вправо или влево меняется фотография и эта фотография выделяется в блоке «выбор фотографии» ниже сделать по возможности
- Выбор фотографии – возможность выбрать любую фотографию из трех, которая будет чуть-чуть больше остальных и соответственно выбранная фотография показывается в блоке «фото товара» выше. Сделать карусель по возможности
- Блок с кнопкой добавления в избранное и кнопкой поделиться – две кнопки, функционал реализовывать не нужно
- Блок с информацией о товаре – содержит в себе название товара, ниже описание, ниже рейтинг товара, ниже варианты расцветок товара, с возможностью их выбрать (для всех цветов будет отображаться один и тот же товар) и справа цена товара
- Блок с итогом – слева содержит кнопки плюс и минус, с помощью которых можно выбрать количество товара, справа кнопка добавления в корзину, но при изменении количества товара, цена, которая находится в кнопке, меняется

API для получения данных

Сценарии использования:

Пользователь хочет посмотреть детальную информацию о товаре и сделать заказ (выполнение по возможности)

- При нажатии на товар в списке товаров в разделе «Flash Sale» пользователь переходит с экрана «Page 1» на экран «Page 2» для просмотра детальной информации о товаре
- Пользователь просматривает фотографии товара
- Пользователь выбирает цвет товара
- Может добавить товар в избранное
- Может поделиться выбранным товаром
- Выбирает количество товара, от которого меняется итоговая сумма товара
- Нажимает на кнопку «Add to cart» для добавления в корзину и оплаты товара. Открывается соответствующий экран (реализовывать экран и функционал не нужно)

*Пользователь хочет найти товар через поисковик в приложении
(выполнение по возможности)*

- Пользователь находится на экране «Page 1»
- Пользователь вводит слово или часть слова, или одну букву в поле ввода поисковика. Спустя 1 секунду после прекращения ввода выполняется запрос на сервер по этому API, где находятся слова
 - а. Если введенная строка пользователем совпадает со словами или с одним словом, которые были получены из запроса, то эти слова из запроса отображаются как подсказка (продолжение слова) в виде выпадающего списка под поисковиком
 - б. Если введенная строка не совпадает ни с одним словом из запроса, то ничего не отображать

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

Критерии оценки

1. Количество сделанных требований
2. Насколько визуал соответствует Figma
3. Насколько чисто и расширяемо написан код, и соответствует SOLID
4. Какое архитектурное решение реализовано
5. Умение пользоваться основными библиотеками
6. Наличие многомодульности (только у Android)