

# Задание для квалификационного отбора на региональный чемпионат города Москва 2021

## MAD Meditation

### Введение

Вашей задачей является разработать приложение для смартфона. Дизайн приложения доступен по адресу: <https://www.figma.com/file/gAonPFbfolLOllyNHqwy41/MAD-Meditation-App?node-id=0%3A1>

Вам необходимо следовать дизайну, а так же логике переходов, которая реализована в прототипе (её можно посмотреть нажав в левом side-блоке на "prototype"). То есть все экраны должны соответствовать макету (за исключением экранов-заглушек). Текст должен быть вставлен в виде текстовых блоков, поля для ввода - полями для ввода. В прототипе присутствуют экраны-заглушки. Переход на них тоже нужно реализовать. Добавьте иконку к приложению и используйте предоставленный шрифт.

**Шрифт находится по адресу: [mskko2021.mad.hakta.pro/Alegreya.zip](https://www.figma.com/file/gAonPFbfolLOllyNHqwy41/MAD-Meditation-App?node-id=0%3A1)**

**Начальный URL к API: [mskko2021.mad.hakta.pro/api](https://www.figma.com/file/gAonPFbfolLOllyNHqwy41/MAD-Meditation-App?node-id=0%3A1)**

### Splash Screen

Экран загрузки должен отображаться при каждом запуске приложения.

### Onboarding

У пользователя должна быть возможность перейти на окно входа и экран-заглушку - регистрацию.

### Login

На данном экране пользователь должен иметь возможность ввести email и пароль к своей учетной записи и получить токен авторизации.

Перед тем, как отправить данные на сервер для получения токена, необходимо проверить заполненность полей, а также наличие символа @ в поле для Email.

Данные авторизации должны сохраняться на устройстве и после перезагрузки приложения, то есть пользователь при повторном запуске будет пропускать экраны onboarding и login.

### Main

На главном экране необходимо выводить полученную информацию о текущем пользователе (изображение профиля и имя).

На экране должен выводиться список ощущений, получаемый с сервера. Список должен листаться по-горизонтали. Список ощущений должен быть отсортирован по параметру "position" по возрастанию.

На экране должны выводиться блоки с короткими цитатами, получаемые с сервера. В случае, если список цитат не помещается на экран, то необходимо позволить пользователю листать экран полностью (за исключением нижнего таббара).

### Profile

На экране профиля добавьте кнопку "выход", после которой приложение открывает окно авторизации.

В случае, если пользователь вышел из профиля, то в окне Login в поле Email должен отображаться Email пользователя, который был последний раз авторизован в системе. Данный Email должен сохраняться и после перезагрузки приложения.

На экране должны быть отображены: имя и фотография пользователя.

В нижней части экрана должна быть показана галерея фотографий, которые были загружены из стандартной галереи приложения. На каждой фотографии должно отображаться время её загрузки. Самым последним элементом всегда должна быть кнопка добавления фотографии, при нажатии на которую открывается выбор фотографий из стандартной галереи. Фотографии должны сохраняться в памяти приложения, чтобы они остались в приложении в случае удаления их из стандартной галереи. Фотографии должны сохраняться и после перезагрузки приложения.

После нажатия на фотографию, должен открываться экран Photo.

## Photo

Фотография должна вписываться в экран по большей стороне.

При нажатии на кнопку “удалить”, фотография должна быть удалена из приложения.

При нажатии на кнопку “закрыть”, пользователь должен вернуться на экран Profile.

Реализуйте управление экраном с помощью жестов. Если пользователь проводит по экрану слева направо, то он закрывает окно просмотра, а если справа налево, то удаляет фотографию.

Добавьте возможность приближать фотографию с помощью двойного нажатия на неё.

После двойного нажатия фотография должна увеличиться в 2 раза. В случае повторного двойного нажатия, фотография должна вернуться в исходное состояние.

### Методы к API

Название метода	Описание	Принимаемые значения	Ответ
GET /quotes	Возвращает список цитат	-	<pre>{   "success": true,   "data": [     {       "id": 0,       "title": "",       "image": "",       "description": ""     },   ], }</pre>
GET /feelings	Возвращает список ощущений	-	<pre>{   "success": true,   "data": [     {       "id": 0,       "title": "",       "image": "",       "position": 0     },   ], }</pre>

<b>POST /user/login</b>	<p>Позволяет получить данные пользователя и token авторизации. Данные отправляются в POST Body. Content-Type: application/json</p> <p><b>Для тестирования авторизации используйте одну из учетных записей:</b>  <b>Login:</b> <a href="#">junior@wsr.ru</a>  <b>Password:</b> junior  <b>Login:</b> <a href="#">general@wsr.ru</a>  <b>Password:</b> general  <b>Login:</b> wsr  <b>Password:</b> wsr</p>	<pre>{   "email": "",   "password": "" }</pre>	<pre>{   "id": "",   "email": "",   "nickName": "",   "avatar": "",   "token": "" }</pre>
-------------------------	---	--	---