



Задание

- 1) Разработать мобильные приложения для смартфона и для умных часов для управления коллекцией фильмов.
- 2) Подготовить приложения для публикации (к релизу)

Общие требования к приложениям (для смартфона и для умных часов):

1. Необходимо строго следовать предложенному дизайну.
2. Макеты обоих приложений доступны по ссылке:

<https://www.figma.com/file/yLAIsuVrxWDIWgCXzdyD0j/KOD1.4-Variant1?node-id=0%3A1>

3. Проект каждого приложения должен быть структурирован по экранам, то есть исходные файлы конкретного экрана должны быть в соответствующей папке.
4. Общие для нескольких экранов классы необходимо поместить в папку common.
5. Описание протокола API доступно по ссылке:

<https://app.swaggerhub.com/apis-docs/WorldSkills-MAD/WorldCinema/1.0.0>

Необходимый функционал приложения для смартфона:

1. иконка приложения должна соответствовать макету.
2. экран **Launch Screen** должен соответствовать макету. Текст на экране должен быть отдельным элементом. Логотип приложения должен быть расположен по центру экрана.
3. экран **SignUp Screen** должен соответствовать макету:
 - при нажатии на кнопку "Зарегистрироваться" проверяются поля для ввода на пустоту, а также email на корректность (требования к email описаны в документации к API). При некорректном заполнении ошибка отображается с помощью диалогового окна. Так же проверяется равенство пароля и его повтор.
 - при корректном заполнении формы регистрации запрос регистрации отправляется на сервер. При получении ошибки от сервера она отображается с помощью диалогового окна. При успешной регистрации автоматически осуществляется авторизация и переход на **Main Screen**.
 - при нажатии на кнопку "У меня уже есть аккаунт" осуществляется переход на **SignIn Screen**.

- при первом запуске приложения после **Launch Screen** отображается **SignUp Screen**. При последующих - **SignIn Screen**.
4. экран **SignIn Screen** должен соответствовать макету:
- при нажатии на кнопку "**Войти**" проверяются поля для ввода на пустоту, а также email на корректность (требования к email описаны в документации к API). При некорректном заполнении ошибка отображается с помощью диалогового окна. При корректном заполнении формы на сервер отправляется соответствующий запрос.
 - при нажатии на кнопку "**Регистрация**" осуществляется переход на **SignUp Screen**.
 - при успешной авторизации осуществляется переход на экран **Main Screen**. При получении ошибки от сервера она отображается с помощью диалогового окна.
5. экран **Main Screen** должен соответствовать макету:
- на экране отображаются обложки фильмов из подборки «new» (информация о фильмах запрашивается с сервера). Обложки отображаются в виде карусели, они листаются с помощью жеста *swipe*.
 - при пролистывании обложек название фильма в верхней части экрана меняется.
 - при нажатии на обложку происходит переход на **Chat Screen** для соответствующего фильма.
6. экран **Profile Screen** должен соответствовать макету:
- данные о пользователе запрашиваются с сервера.
 - при нажатии на кнопку "**Изменить**" реализовано изменение аватара пользователя:
 - 1) пользователь выбирает источник фотографии (камера или Галерея), выбор источника реализован с помощью диалогового окна. При выборе *Галереи* открывается *Галерея*. При выборе камеры переход на экран *Камеры* не нужен.
 - 2) пользователь выбирает фотографию.
 - 3) фотография отправляется на сервер.
 - 4) в случае успеха аватар пользователя заменяется на новый, в случае ошибки она отображается с помощью диалогового окна.
 - при нажатии на кнопку "**Выход**" осуществляется переход на экран **авторизации**.
 - при нажатии на кнопку «**Обсуждения**» осуществляется переход на соответствующий экран.
7. экран **Chat List Screen** должен соответствовать макету:
- информация запрашивается с сервера (запрос чатов пользователя). Если информация с сервера содержит дуближи – они удаляются. Если пользователь еще не имеет чатов,

сервер присылает пустой список, от текущего пользователя в чате отправляется сообщение с $id = 1$ с помощью *Swagger* или *Postman*.

- в подзаголовке ячейки отображается последнее сообщение в соответствующем чате + имя его автора. Текст сообщения обрезается до двух строк.
- реализовано отображение постеров к фильмам. Для получения постеров используется подходящий запрос из API. Если постер для данного фильма нельзя получить из API, генерируется аббревиатура по следующему правилу: если название фильма состоит из одного слова - берутся первые две буквы слова; иначе - первые буквы первого и второго слова.
- при нажатии на ячейку осуществляется переход на ***Chat Screen*** для выбранного фильма.

8. экран ***Chat Screen*** должен соответствовать макету:

- сообщения упорядочены от старых к новым сверху вниз. Для сегодняшних сообщений отображается заголовок "Сегодня".
- "Облако" сообщения растянуто по содержимому.
- последовательно идущие сообщения одного автора сгруппированы (расстояния между сообщениями как на макете).
- реализован блок отправки сообщения как на макете. При вводе сообщения поле для ввода растягивается по вертикали.
- при нажатии на кнопку "*Отправить*" сообщение отправляется на сервер. При позитивном ответе от сервера отображается сообщение в чате. При возникновении ошибки – отображается ошибка с помощью диалогового окна.
- поле для ввода валидируется на пустоту. При отсутствии текста сообщения отображается ошибка с помощью диалогового окна.

9. экран ***Collections Screen*** должен соответствовать макету:

- при нажатии на иконку в правом верхнем углу осуществляется переход на экран ***Create Collection Screen***.
- на экране отображаются созданные коллекции. Информация о коллекциях хранится в памяти устройства. Хранится название коллекции и иконка.
- для удаления коллекции реализован *Swipe-to-delete*, в том числе из памяти устройства.

10. экран ***Create Collection Screen*** должен соответствовать макету:

- при открытии экрана в качестве иконки выбирается случайное изображение из коллекции иконок.
- при нажатии на кнопку "*Выбрать иконку*" осуществляется переход на экран *Icon Selection*, который реализован в соответствии с макетом.
- при нажатии на кнопку "*Сохранить*" новая коллекция сохраняется в памяти устройства и экран закрывается.
- поле для ввода названия коллекции проверяется на пустоту. При отсутствии значения отображается сообщение об ошибке.

Необходимый функционал приложения для умных часов:

1. иконка приложения настроена согласно макету.
2. экран *SignIn Screen* должен соответствовать макету:
 - при нажатии на кнопку "*Войти*" проверяются поля для ввода на пустоту, а также email на корректность (требования к email описаны в документации к API). При некорректном заполнении отображается ошибка с помощью диалогового окна. При корректном заполнении формы на сервер отправляется соответствующий запрос.
 - при успешной авторизации осуществляется переход на экран *Main Screen*. При получении ошибки от сервера она отображается с помощью диалогового окна.
3. экран *Main Screen* согласно макету.
 - при нажатии на иконку осуществляется переход на соответствующий экран. Если экран не описан в Задании - экран отображается с надписью "В разработке".
 - при нажатии на кнопку *Cancel* (для Android - при свайпе вправо) происходит возврат на *SignIn Screen*.
4. экран *Chat List Screen* должен соответствовать макету:
 - данные запрашиваются с сервера.
 - при нажатии на кнопку *Cancel* (для Android - при свайпе вправо) происходит возврат на *Main Screen*.
5. экран *Movies Screen* должен соответствовать макету:
 - на данном экране отображаются фильмы из подборки «new» в виде списка с вертикальной прокруткой. Данные берутся с сервера.