

Тестовое задание “Movies”

Разработать мобильное приложение, которое будет отображать список фильмов, и давать возможность просмотреть детали выбранного элемента. Так же, должна быть возможность поиска, сортировки и фильтрации фильмов по разным критериям.

Источником данных будет выступать **The Movie Database API** (<https://developers.themoviedb.org/3/>).

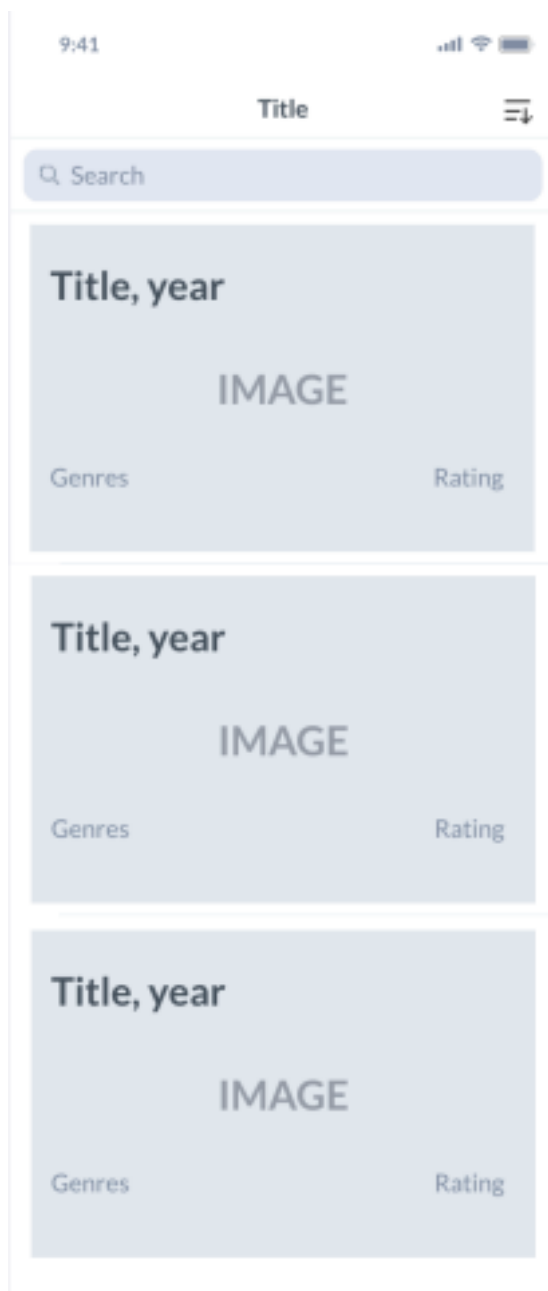
Требования к приложению:

- Можно использовать **любые** библиотеки, которые могут помочь решить данную задачу
- минимальная версия поддерживаемой ОС - **iOS 14.0**
- Язык программирования **Swift**
- Для работы с сетью использовать **Alamofire** или нативный **URLSession** (приоритетно - **Alamofire**)
- Использовать архитектуру **MVP/MVVM/VIPER/Clean Swift**.
- Для вёрстки использовать **UIKit + Autolayout**. Верстать можно как программно, так и с помощью **Storyboard, XIB**
- при любом сетевом запросе отображать какой-нибудь spinner/loader/activity indicator, чтоб дать пользователю понять, что загрузка чего-либо в процессе -

Экран 1:

- отображать список популярных фильмов, по умолчанию отсортирован по убыванию (от самых популярных, к самым непопулярным), с постепенной подгрузкой результатов (пагинация). Размер одного сегмента (страницы) составляет **20** элементов. В идеале, пагинация должна быть незаметна для пользователя.
- В ячейке должны отображаться: 1) постер (фото). 2) название фильма. 3) жанры, к котором относится фильм. 4) средний рейтинг фильма. 5) год выпуска фильма. - Ячейки должны быть с тенью (параметры тени - на своё усмотрение, в зависимости от общего интерфейса приложения)
- Сверху экрана зафиксирован **UISearchBar**, с помощью которого будет осуществляться поиск по названию фильма (только по названию). - В **UINavigationBar** должен отображаться тайтл **“Popular Movies”**
- Справа от тайтла отображать кнопку сортировки , нажатие на которую будет отображать нативный **ActionSheet/UIMenu** , с возможными опциями сортировки, и кнопкой Cancel (в случае action sheet). Выбранная опция сортировки должна быть отмечена галочкой (это тоже нативно, доступно из коробки). Если не удастся реализовать данную опцию, реализовать альтернативную, на своё усмотрение.
- При нажатии на элемент списка - открывать экран деталей выбранного элемента. Для этого выполнять соответствующий сетевой запрос.

Грубый макет экрана:

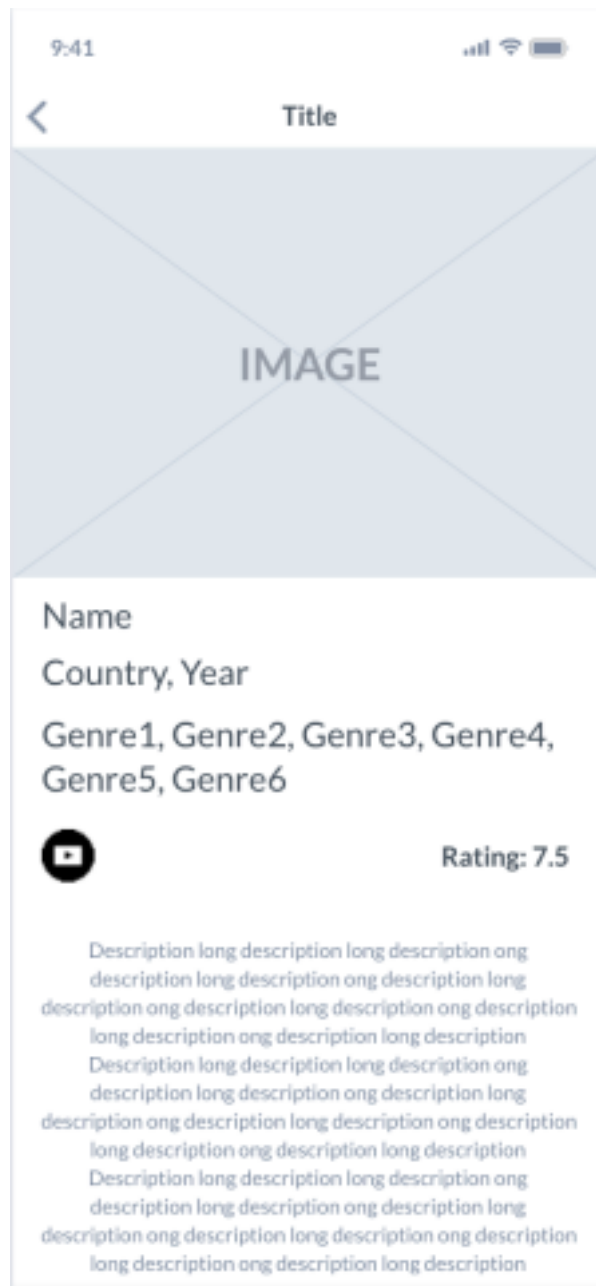


Экран 2:

- Отображать детали выбранного фильма, а именно:
 - 1) постер в фиксированном размере.
 - 2) Название фильма.
 - 3) Год и страну выпуска.
 - 4) жанры, к которым относится фильм.
 - 5) Описание фильма.
 - 6) средняя оценка (рейтинг) фильма.
 - 7) кнопка **“Trailer”** - отображать, только если трейлер доступен! По нажатию на кнопку **“Trailer”** открывать системный экран плеера, и показывать трейлер. По нажатию на постер (картинку) - модально открывается **Экран 3**, в котором постер открывается в **полном размере**. Картинку можно приближать и скроллить жестами. Экран должен закрываться по кнопке **“Close”** или по **Swipe**-у вниз (Кнопки **Close** может и не быть).

- В **UINavigationController** отображать название фильма и кнопку < назад.

Грубый макет экрана:



Будет плюсом:

- При отсутствии интернета отображать список элементов, которые были загружены ранее. (При попытке войти на детали такого элемента, показывать Алерт **“You are offline. Please, enable your Wi-Fi or connect using cellular data.”**). Так же, реализовать и обработать локальный поиск фильмов по названию, если интернет отсутствует. Сортировка пусть работает только онлайн.
- Картинки должны быть закешированы, чтоб мы могли видеть их и в оффлайне. (Напоминаю, что для изображений можно тоже использовать **любые** библиотеки).

- Реализовать Dependency injection
- сделать приложение доступным на английском и украинском языках (разумеется, контент получаемый от апи мы локализовать не можем)