В качестве тестового задания необходимо создать приложение «rick and morty» используя предложенное API и макеты в Figma.

API: <https://rickandmortyapi.com/documentation>

Figma: <https://www.figma.com/file/J6IImxJS0u2R41lYt3TlWG/Rick-and-Morty-(-test-work)?type=design&mode=design&t=jvLErL7nnkfFGSXB-1>

В предложенном проекте Вам необходимо реализовать следующие экраны:

1. Загрузочный экран приложения (Launch Screen)
2. Основной экран приложения (Episodes)
3. Детальная информация о герое из выбранного эпизода (Character details)
4. Любимые эпизоды (Favourites)

**Ограничения:**

**В тестовом проекте запрещено использовать сторонние библиотеки (pods, spm, carthage).**

**Используем для верстки UIKIT**

**Вся верстка кодом**

**iOS Deployment Target 14.0**

***Время на выполнение: 10 дней с момента получения тестового задания.***

***Для каждой предлагаемой фичи будет указан бал за сложность. По общей сумме балов будет производится оценка задания.***

**Минимальный проходной бал: 15**

**Количество мест ограничено, чем больше балов наберете, тем выше шанс получить место в группе.**

**Фичи экранов:**

1. **Launch Screen**

****

|  |  |
| --- | --- |
| **Фича** | **Бал** |
| При запуске приложения отображать данный экран 3 секунды | **1** |
| На время его отображения анимировано крутить «портал» (зеленый круг). | **2** |

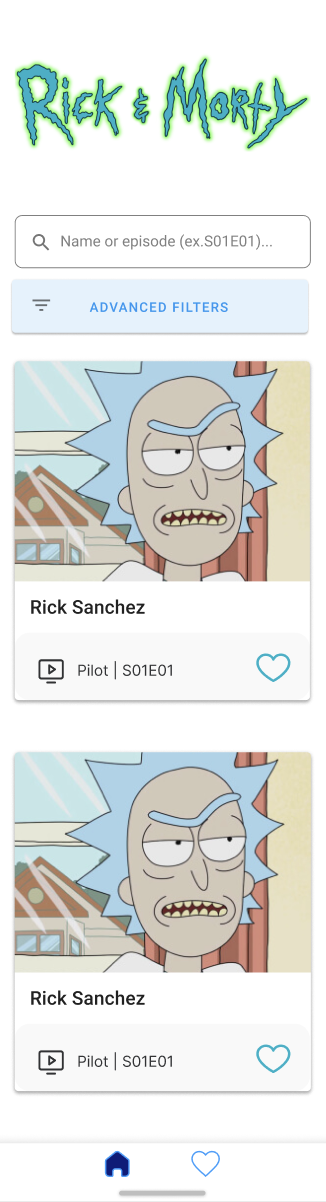
После отобразить основной экран приложения (Episodes)

1. **Episodes**

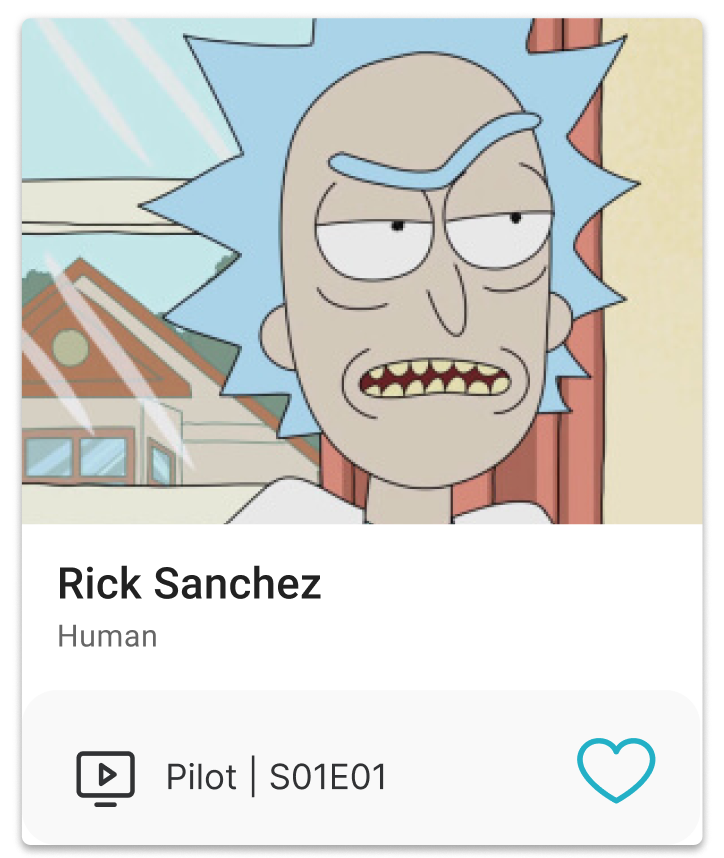
При загрузке экрана выполняем запрос:

[*https://rickandmortyapi.com/api/episode*](https://rickandmortyapi.com/api/episode)

В качестве коллекции используем **CollectionView**

****

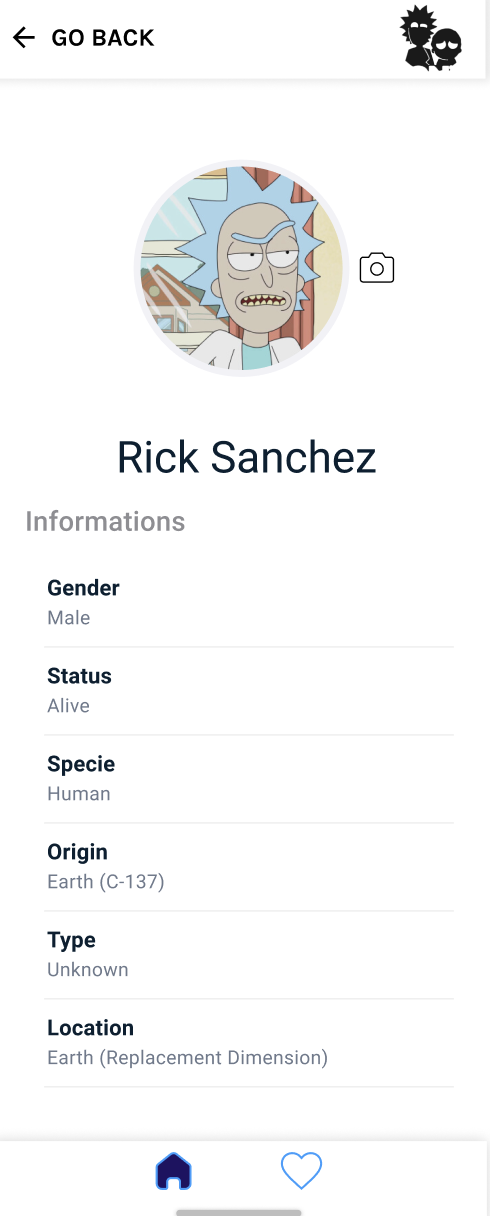
Заполняем ячейки данными



|  |  |  |
| --- | --- | --- |
| **Фича** | **Описание** | **Бал** |
| *Таб-бар* | Добавить таб-бар с двумя вкладками;   * экран Episodes * экран Favourites | *1* |
| *Навигейшн контроллер* | На экраны Favourites и Episodes добавить навигейшн контроллер | *1* |
| *Реализация  CollectionView* | Коллекцию реализовать используя UICollectionView | *1* |
| *Номер эпизода*  *(в ячейке коллекции)* | В ответе на запрос <https://rickandmortyapi.com/api/episode> есть свойство **Episode**, которое содержит массив ссылок. Выбрать рандомный и указать его номер | *2* |
| *Как получить изображение* | В ответе на запрос <https://rickandmortyapi.com/api/episode> есть свойство  **Characters**, которое содержит массив ссылок. Выбрать рандомную ссылку и выполнить запрос, например <https://rickandmortyapi.com/api/character/2> в ответе будет свойство с ссылкой на картинку | *3* |
| *Переход по тапу на картинку в ячейке* | Переход на экран «Character details» - детальная информация о персонаже, которого вы отображаете в этой ячейке. | *1* |
| *Тап по иконке сердца в ячейке* | Иконка окрашивается в красный цвет | *1* |
| Выбранный эпизод добавляется в раздел «Favourites».  Повторный тап возвращает иконку к первоначальному виду и удаляет эпизод из раздела «Favourites» | *3* |
| Добавить анимацию для нажатия (сжатие, анимация смены цвета и т.д.) Выбор анимации оставляем за Вами. | *2* |
| *Поиск* | Поиск по номеру серии (Пользователь вводит номер серии, коллекция фильтруется и отображается только та серия, которую ввел пользователь) | *2* |
| *Фильтр* | Возможность выбора поля для поиска (по тапу на кнопку отобразить список (по имени персонажа, названии локации, номеру серии эпизода) на свое усмотрение). | *3* |
| *Пагинация* | При скролле экрана подгружаются остальные эпизоды про пагинацию в API написано тут https://rickandmortyapi.com/documentation/#info-and-pagination | *3* |
| *Diffable Data Source* | Реализовать Diffable Data Source | *3* |
| *Кеширование* | Реализуете кеширование изображений | *3* |
| *Свайп по ячейке* | При свайпе ячейке влево – ячейка удаляется из коллекции | *2* |
| *Продвинутый фильтр* | При вводе номера в окно фильтра отображаются только те серии, в которых присутствует данный номер пример (пользователь вводит 2, в ответ получаем серии 2, 20, 21, 22, 23, 24, 25, 26,27,28,29,32,42 и т.д) | *3* |

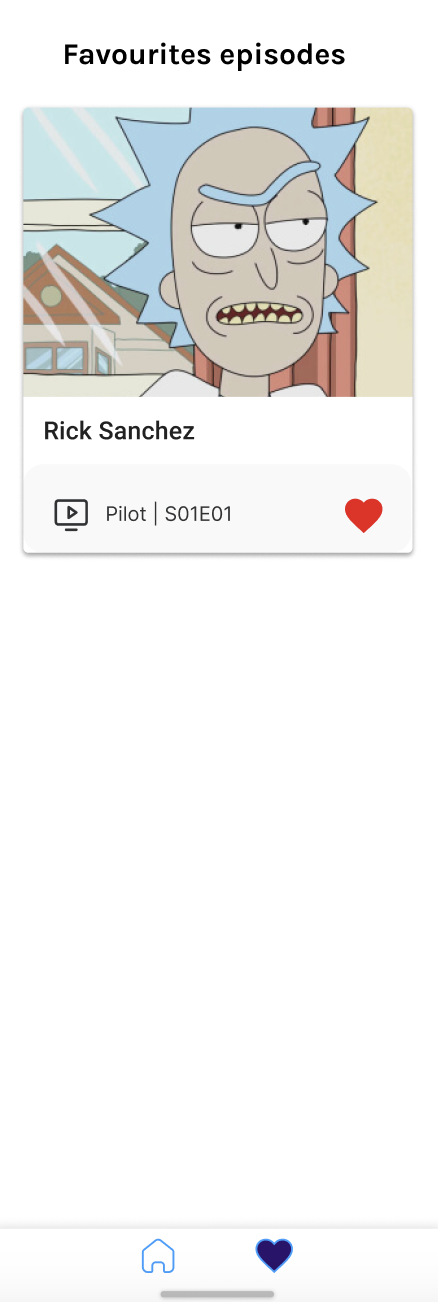
1. **Character details**

Где взять данные для наполнения экрана можно посмотреть здесь: <https://rickandmortyapi.com/documentation/#get-a-single-character>

****

|  |  |  |
| --- | --- | --- |
| **Фича** | **Описание** | **Бал** |
| *Экран деталей* | Реализовать экран деталей | **1** |
| *TableView* | Экран detail – реализовать с использованием TableView | **2** |
| *Добавить иконку в нав-бар* | В навигейшн баре в правом углу отобразить иконку как на макете. | **2** |
| *Смена и загрузка фото аватарки* | Реализуете механизм замены фото героя, используя макеты фигма (На detail экране возле аватарки персонажа добавить кнопку (иконка фото), по нажатию на которую всплывает Action Sheet (сделать фото, выбрать из галереи) | **3** |
| При первом обращении запрашивать у пользователя разрешение на использование, текст для разрешений также берем из фигмы. | **2** |
| Если пользователь не дал разрешение при повторном нажатия на иконку «фото» переводим его в настройки телефона для разрешения. | **2** |

1. **Favourites**

****

|  |  |  |
| --- | --- | --- |
| **Фича** | **Описание** | **Бал** |
| *Экран “Favourites”* | Реализовать экран | **1** |
| *Тап по иконке сердца в ячейке* | Удаляет эпизод из раздела «Favourites» | **1** |
| *Кеширование данных* | Реализоватькеширование данных | **3** |
| *Избранные серии не должны очищаться* | При закрытии приложения и его повторном открытии коллекция не должна очищаться. | **3** |
| *Тап по картинке в ячейке* | Переход на экран Character details - детальная информация о персонаже, которого вы отображаете в этой ячейке. | **1** |

***По итогу присылаем:***

1. ***Ссылку на Git с проектом***
2. ***Список выполненных фич***
3. ***Итоговый бал по количеству выполненных фич***
4. ***Видео демонстрацию выполненных фич***