Mobile Engineer Take Home Evaluation

Imagine that you are building a native app for film enthusiasts to explore the top-rated movies of the year. By integrating with our movie information API, your app should support the following use-cases:

- Discovery - users of your app can view the top 10 movies of the year at a glance
- Research - users can inspect movie metadata, such as ratings, actors, etc. in order to select a film to screen. Users must be able to view a movie poster.
- Booking - users are able to proceed to purchase tickets to the desired movie via a button (the button's implementation is not required, the button can lead them to a webview of Zocdoc's homepage)

From a functional and UX perspective, how the movies or their details interact and are displayed is entirely up to you. As long as the above requirements are met, feel free to focus on areas you're most interested in.

Considerations

- This challenge should take approximately 3 hours to complete, however, you should complete this app as if you were planning on publishing and releasing it. This is an opportunity for you to put your best foot forward and showcase your talent!
- We collaborate closely with a great design team so there is no need to wow us with your design skills. Our engineers care a lot more about code quality and design patterns so you should put less of an emphasis on UI slickness.
- We know that some engineers prefer using the native visual editor for creating and designing their application UI. While we'll accept submissions that were created using the visual editor, our engineers at Zocdoc prefer to create the UI programmatically in code. You should expect to talk through how you would create your UI programmatically even if you use the visual editor.
- The solution must be a native app using Swift for iOS, Kotlin for Android.
- Feel free to use 3rd party libraries - unless otherwise specified in the requirements section.
- The movie poster images must be fetched from an external server. You're welcome to use a service such as https://placehold.it/ to generate an image with the movie ID or other information as a stand-in for an actual poster.
- To submit your solution: send a zip file consisting of all files necessary to run and view your code.

Your authentication token: 3b502b3f-b1ff-4128-bd99-626e74836d9c

APIs

1. **Retrieve the movie set**

```
GET https://interview.zocdoc.com/api/1/FEE/AllMovies
```

Parameters

Name	Type	Description
authToken	string	**Required.** A unique key that allows you to retrieve data.

Sample Output


```
[
   {
      "Id":381,
      "Rank":1,
      "Name":"The Lives of Others",
      "Duration":"137 min",
      "Description":"In 1984 East Berlin, an agent of ...",
      "Director":"Florian Henckel von Donnersmarck",
      "Genres":[
         "Drama",
         "Thriller"
      ],
      "Actors":[
         "Martina Gedeck"
      ]
   },
   {
      "Id":262,
      "Rank":75,
      "Name":"The Intouchables",
      "Duration":"112 min",
      "Description":"After he becomes a quadriplegic ...",
      "Director":"Olivier Nakache",
      "Genres":[
         "Biography"      ],
      "Actors":[
         "François Cluzet",
         "Omar Sy"
      ]
   },
   ...
]
```


2. **Retrieve the basic movie information based on rank**

   `GET https://interview.zocdoc.com/api/1/FEE/MoviesByRank`

Parameters

Name	Type	Description
authToken	string	**Required.** A unique key that allows you to retrieve data.

startRankIndex	int	**Required.** The rank of the first movie you're interested in retrieving. The movies retrieved will start at this rank. The first possible index is 1 (ie. the top movie is assigned rank 1).
numMovies	int	**Required.** The total number of movies to be retrieved. Movies ranked from *startRankIndex* to *startRankIndex* + *numMovies* will be retrieved.

Sample Output


```
[
   {
      "Id":34,
      "Rank":5,
      "Name":"The Shawshank Redemption"
   },
   {
      "Id":100,
      "Rank":6,
      "Name":"The Intouchables"
   },
   {
      "Id":55,
      "Rank":7,
      "Name":"Schindler's List"
   }
]
```


3. Retrieve detailed movie information

`GET https://interview.zocdoc.com/api/1/FEE/MovieDetails`

Parameters

Name	Type	Description

authToken	string	**Required.** A unique key that allows you to retrieve data.
movieIds	Int[]	**Required.** The ids of the movies to be retrieved.

Sample Output


```
[
   {
      "Id":332,
      "Name":"Back to the Future",
      "Duration":"116 min",
      "Description":"A young man is accidentally ...",
      "Director":"Robert Zemeckis",
      "Genres":[
         "Adventure"
      ],
      "Actors":[
         "Michael J. Fox",
         "Christopher Lloyd"
      ]
   },
   {
      "Id":290,
      "Name":"The Green Mile",
      "Duration":"189 min",
      "Description":"The lives of guards on Death ...",
      "Director":"Frank Darabont",
      "Genres":[
         "Crime"
      ],
      "Actors":[
         "Tom Hanks",
         "David Morse",
         "Michael Clarke Duncan"
      ]
   }
]
```