

Assignment

Evaluation Criteria for this Assignment

1. Class and method names should clearly show their intent and responsibility.
2. 'SOLID' Principles.
3. Latest App architecture and navigational pattern (if possible) should be used.
4. UIViewControllers should not know too much explicit knowledge about non UI Classes. - The need for domain objects.
5. We do not like long methods or classes. Let alone huge UIViewController classes.
6. Your software should easily accommodate possible future requirement changes.
7. We will assess a number of things including the design aspect of your solution and your object oriented programming and protocol oriented programming skills.
8. Tests are great, but testability is more important. Please Unit Test all non-UI classes. Mocking, Stubbing, TDD if possible. Please handle all types of error, which could occur.

Instructions:

1. The code must be available via GitHub in order to be downloaded, built and executed. Having expressive commit messages will add value.
2. Having a ReadMe (in the GitHub) will add value. The file can have information like
 1. Used design pattern and App architecture
 2. Code Structure
 3. User guide to use the app
 4. Notes that the developer find to be important
 5. Third parties used, If any
 6. Unit test info, If any
3. Document your code.
4. Code should not contain any warnings, or errors.
5. Downloaded Data should be Cache
6. Image loading should be smooth
7. App should support all iPhone devices (both orientation will be plus point)
8. You may use external libraries or tools for building or testing purposes. If you're using anything that's not written by you, mention it.

Requirements:

1. As a user at the Movie List Screen,
 1. I should see a popular movies list view(in grid) with the following cell
 - i. Movie Poster
 - ii. Movie name
 - iii. Release date
 - iv. Favorite icon
 2. If user tap on favorite icon it will save this movie record in local DB and mark favorite.
 3. If there are no internet or loading new list, it will show the cache data and update list on loading new movies data from the server.
 4. When I tap on any cell it will open Movie Detail Screen

2. As a user at the search screen,
 1. When I enter a name of a movie (e.g. "Avengers") in the search box and tap on "search button"
 2. Then I should see a new list view with the following rows
 - i. Movie Poster
 - ii. Movie name
 - iii. Release date
 - iv. Favorite icon
 3. When I tap on any cell it will open Movie Detail Screen
3. As a user at the Movie List Screen,
 1. When I scroll to the bottom of list
 2. Then next page should load movies if available
4. As a user at the search screen,
 1. When I enter a name of a movie that doesn't exist in the search box and tap on "search button",
 2. Then, an alert box should appear and display an error message.
 3. All types of error should be handled
5. As a user at the search screen,
 1. When I tap and focus into the search box
 2. Then an auto suggest list view will display below the search box showing my last 10 successful queries (exclude suggestions that return errors)
 3. Suggestions should be persisted.
6. As a user at the search screen with the auto suggest list view shown,
 1. When I select a suggestion
 2. Then the search results of the suggestion will be shown.
7. As a user at the Movie Detail Screen
 1. Then I should see a Detail view with the following things:
 - i. Movie Banner in header
 - ii. Movie Poster
 - iii. Movie name
 - iv. Release date
 - v. Favorite icon
 - vi. Full Movie Overview
 2. Detail Page supports different layout for portrait and landscape orientation for better readability
8. As a user at the favorite screen,
 1. Then I should see a saved (in DB) favorite list view with the following rows
 - i. Movie Poster
 - ii. Movie name
 - iii. Release date
 - iv. Favorite icon
 2. If user tap on favorite icon it will mark it unfavorite and removed from DB
 3. When I tap on any cell it will open Movie Detail Screen
9. Internet connection should be checked along with other data handling
10. You can add more features, if you want

APIs to use:

1. Movie List:
https://api.themoviedb.org/3/movie/popular?api_key=e5ea3092880f4f3c25fbc537e9b37dc1

2. Search Movie: [https://api.themoviedb.org/3/search/movie?api_key=e5ea3092880f4f3c25fbc537e9b37dc1&query=\[Movie_name\]&page=1](https://api.themoviedb.org/3/search/movie?api_key=e5ea3092880f4f3c25fbc537e9b37dc1&query=[Movie_name]&page=1)
3. Poster image: [https://image.tmdb.org/t/p/w92/\[poster_path\]](https://image.tmdb.org/t/p/w92/[poster_path])