

React Native Code Assignment

Position: React Native Developer

Introduction

This document describes the design and functionality of a test project as code assignment for the position of React Native Developer. This test is designed to have a better overview of your programming skills.

Good luck!!

Description

The test project will be a React Native application. The result of your work should be a working application for the latest Android and iOS.

Instructions

Create a native application for browsing the popular movies, display a short detail page of a selected movie and play a short movie trailer.

For this purpose you have to use **The Movie Database** API. We will use the **v3**.

API documentation:

Overview: [<https://www.themoviedb.org/documentation/api>]

Getting started: [<https://developers.themoviedb.org/3/getting-started/introduction>]

For that, you will need to register yourself and get an **api_key** [<https://www.themoviedb.org/account/signup>]

1. The first screen of your application should display a group of carousels with the most **popular movies**, the most **popular tv series**, and the genres **family** and **documentary** [see wireframe 1]. From the response use **poster_path** value to get the poster image, and the **title** value to display the title. An example of a valid image url path
[<http://image.tmdb.org/t/p/w342/rqAHkvXldb9tHlnbQDwOzRi0yVD.jpg>]
2. After selecting one of the assets in one of the carousels, the application should navigate to the detail page of the selected movie or series [see wireframe 2].
3. After pressing the “Watch Movie” button the application should display a full screen movie player and should automatically start the playback. Instead of using **The Movie Database** trailer results for that,

you will need to use a hardcoded hls video stream

[https://rawgit.com/mediaelement/mediaelement-files/master/big_buck_bunny.mp4] together with **react-native-video** [<https://github.com/react-native-community/react-native-video>]. Integration with **react-native-video** is required for this test.

4. *OPTIONAL TASKS:*

- (I) Make the app responsive
- (II) Implement search functionality
- (III) Create at least one unit test!
- (IV) Make it work on tvOS and/or AndroidTV and/or web

Hints

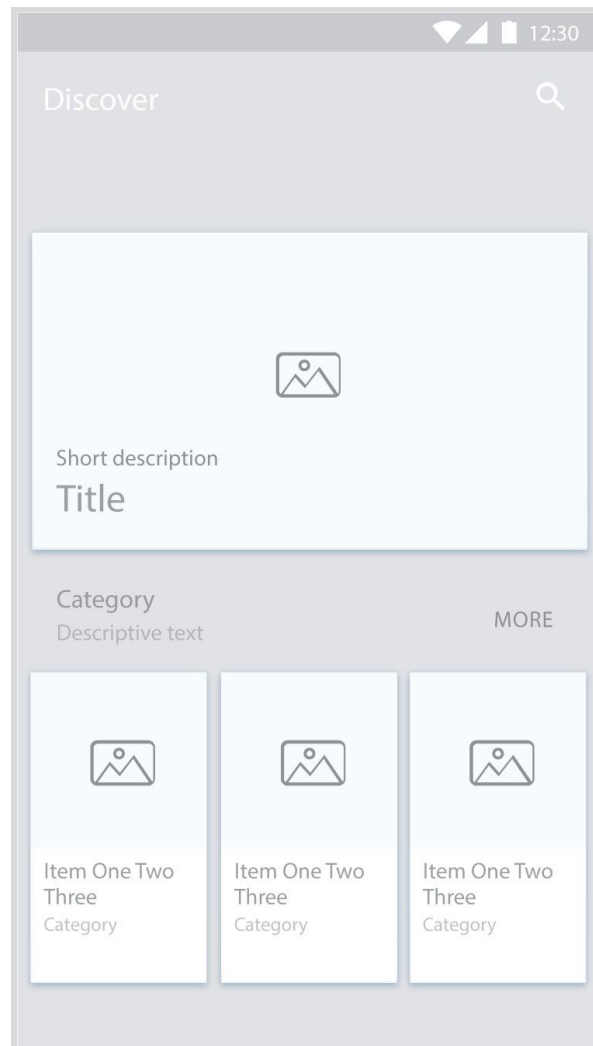
- Use the attached wireframes as a style guide & possible solution. If you think some other flow will improve the ux, go for it.
- Handle as many error cases as you can.
- A proper folder scaffolding will be a plus.
- Documentation.
- Optional tasks 1 and 2 are strongly recommended!
- Comment your code
- It's recommended to have zero warnings in the project.
- Write your code as production grade.
- Use as many libraries or utils you need. But you **MUST** use React Native.

Results

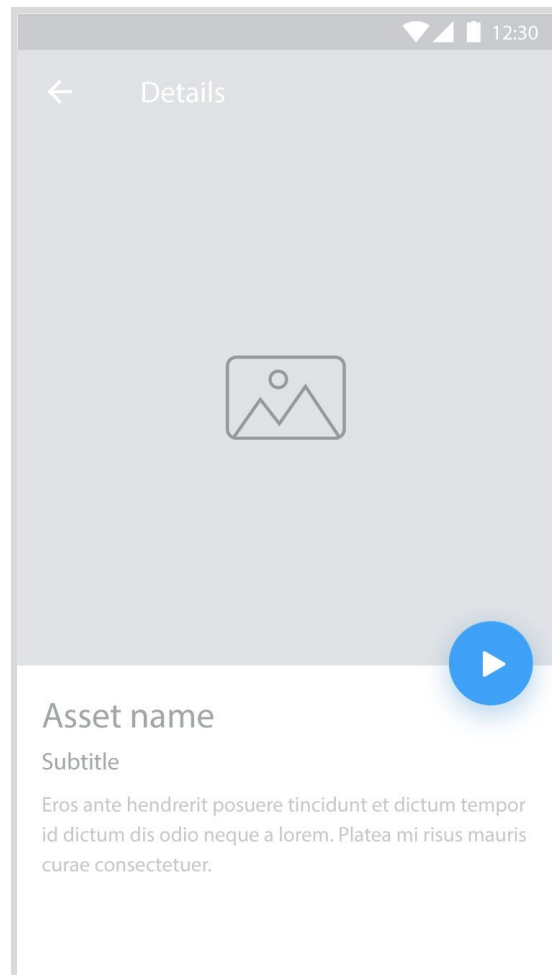
The results of your work should be a working application in iOS & Android. We strongly recommend that you create a private Git repository for your project, and share the repository URL with us.

Your source code will be evaluated with a possibility of personal discussion with the development team.

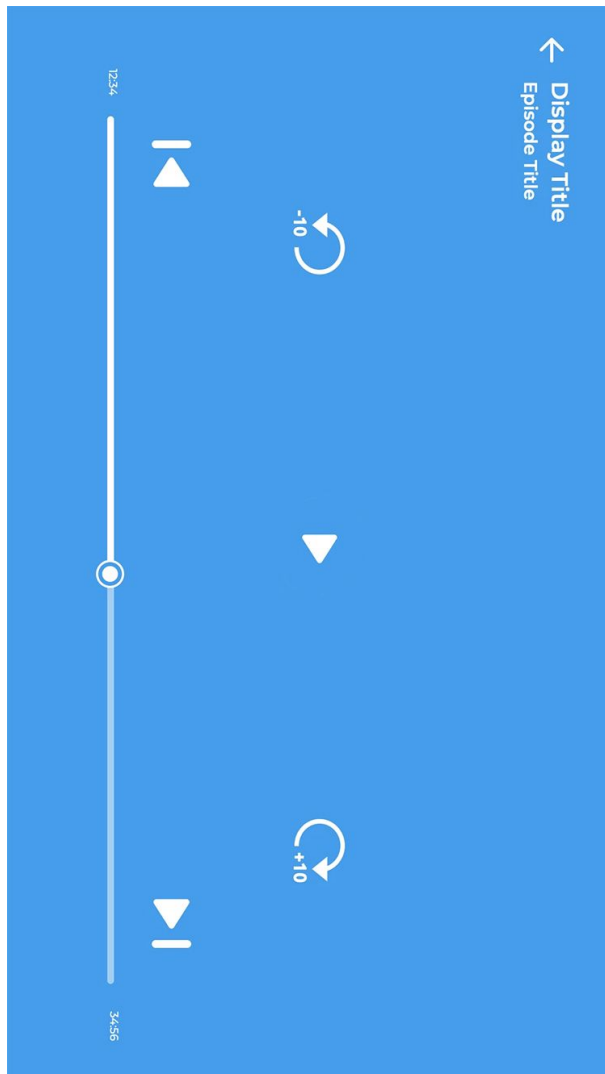
Wireframes



Wireframe 1 (Dashboard view)

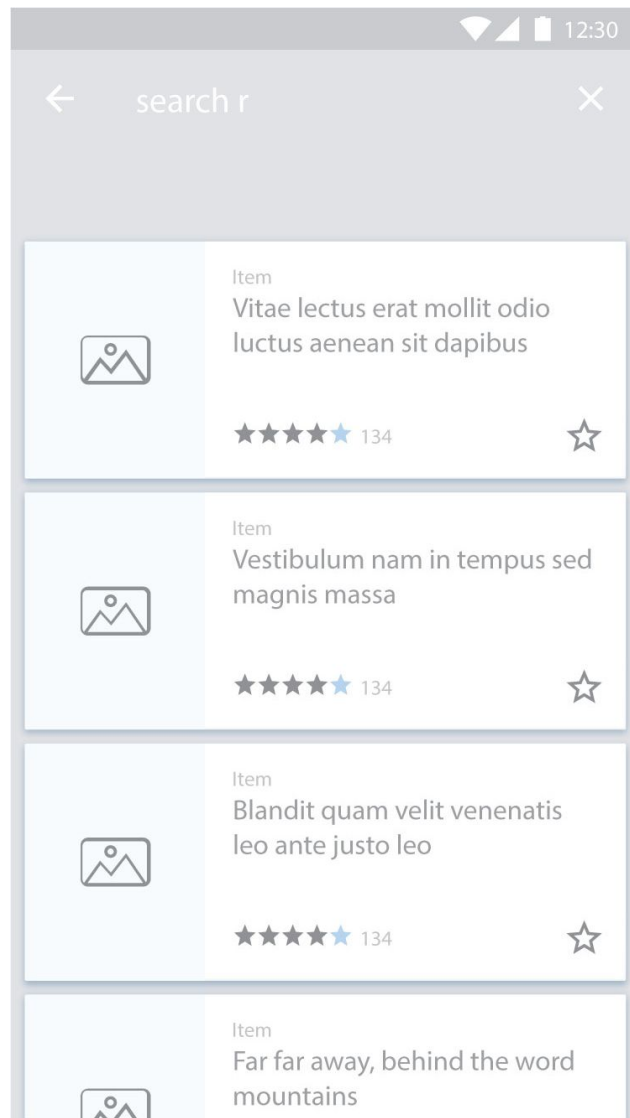


Wireframe 2 (Detail view)



Wireframe 3 (Player view)

Optional Wireframes



Wireframe 4 (Search view)