Test Overview

There should be two tabs in the app, prefereably put in a form of a bottom navigation. The first one should present a list of events and the second one a schedule (EPG).

Remark: some parts of the test are marked as optional. These will give you extra points when we’ll be checking your solution. But don’t stress if you don’t want to do any of these tasks- after all, we have an interview where we can just have a conversation about the missing things.

Events Screen

|  |  |
| --- | --- |
| Diagram  Description automatically generated | Events screen should present list of events available to be watched. When any item is tapped, the apprioriate video is played on Playback Screen.    Use the standard APIs to present a list of items provided by our API. Assume that the list can get fairly long (e.g up to 100 items).    Events should be ordered by date in ascending order. Extra points will be awarded for date formatting similar to the one presented in the mockups.    The data can be fetched from:  <https://us-central1-dazn-sandbox.cloudfunctions.net/getEvents> |

Schedule Screen

|  |  |
| --- | --- |
| Diagram  Description automatically generated | Schedule screen, with a schedule for tomorrow only**.** It should auto-refresh every 30 seconds.    Assume that the list can get fairly long (e.g up to 100 items). Extra points will be awarded for updating the state of the list without loosing its scroll position and without flickering.    Events should be ordered by date in ascending order.    Thumbnails should be retrieved from the listings feed.    The data can be fetched from:  <https://us-central1-dazn-sandbox.cloudfunctions.net/getSchedule> |

Playback Screen (extra points)

|  |  |
| --- | --- |
| Shape, rectangle  Description automatically generated | Additionally, there should be a Playback Screen, playing a video in the default player. Use wchichever player you prefer. The aim is to be able to play the video that is provided in selected event. |

Unit Tests

Consider writing at least a few Unit Tests for your implementation. These tests could cover any relevant part of your application.

What are we looking for in the solution (don’t stress if you feel you didn’t get all the below points right- we have an interview where we can just talk about the stuff you didn’t have time to address):

* + Clean code
  + Approach to architecture in a small app
  + Object-oriented programming
  + SOLID
  + Language knowledge and experience
  + Basic memory management
  + Unit tests
  + Testing practices
  + QA practices
  + REST APIs usage practices
  + Handling Activity lifecycle

Worth mentioning:

* + You can choose any libraries you like, the only requirement is that you’d use this library/framework in production.
  + The designs are up to you, feel free to change them. Apply the common design patterns popular on your platform. Please make sure the app is usable.
  + Assume that there would be some other functionalities added to the app in the future
  + Ideally we should be able to checkout and compile the project out of box both on mac or windows. These machines will be setup with the relevant default IDEs.