Technical Task

Thank you for your application for an engineer position at Entain. As part of our interview process, we have prepared the following technical task to get a better understanding of your skills, thought process and methodology.

There is no fixed time limit for this task. What matters is that you showcase your abilities to the fullest. Once you have completed the task, please upload your solution to a public repository and share the link with us, along with any necessary testing instructions.  
  
**Task**

Create an iOS app that displays ‘Next to Go’ races using our API.

A user should always see 5 races, and they should be sorted by time ascending. Race should disappear from the list after 1 minute past the start time, if you are looking for inspiration look at <https://www.neds.com.au/next-to-go>

**Requirements**

* As a user, I should be able to see a time-ordered list of races ordered by advertised start ascending.
* As a user, I should not see races that are one minute past the advertised start.
* As a user, I should be able to filter my list of races by the following categories: Horse, Harness & Greyhound racing.
* As a user, I can deselect all filters to show the next 5 of all racing categories.
* As a user, I should see the meeting name, race number and advertised start as a countdown for each race.
* As a user, I should always see 5 races and data should automatically refresh.

**Technical Requirements**

* Use the latest stable Xcode and a deployment target of the latest stable iOS
* Use Swift 6 Language Mode
* Use SwiftUI and SwiftUI App life cycle, avoid UIKit
* Use Structured Concurrency (e.g. async/await), avoid Combine
* Use Swift Testing not XCTest where possible. Full coverage is not necessary, but there should be at least some testing for key areas/files.
* Consider the user experience with accessibility features such as:
  1. Dynamic type, e.g. use scalable/adaptive layouts
  2. VoiceOver and Voice Control, e.g. accessibility elements/labels
* Use custom Decodable implementations where beneficial
* Use SF Symbols for icons, you may import custom symbols
* Documentation e.g. outline requirements you didn’t have time to implement, technical architecture, decisions etc.

Categories are defined by IDs and are the following:

* Greyhound racing: category\_id: 9daef0d7-bf3c-4f50-921d-8e818c60fe61
* Harness racing: category\_id: 161d9be2-e909-4326-8c2c-35ed71fb460b
* Horse racing: category\_id: 4a2788f8-e825-4d36-9894-efd4baf1cfae

GET <https://api.neds.com.au/rest/v1/racing/?method=nextraces&count=10>

Content-type: application/json

Remember to take into consideration:

* Unit Tests
* Documentation
* Styling
* Linting
* Best practices

This test is crucial for your advancement to the next stages and for receiving an offer with us. We recommend giving it even more care and diligence than you might initially think. In addition to your code, elements like tests, README, and documentation are essential aspects the team looks for in a strong candidate.   
  
We're excited to see your technical abilities in action and look forward to your submission. If you have any further inquiries or need any assistance along the way, please don't hesitate to reach out.