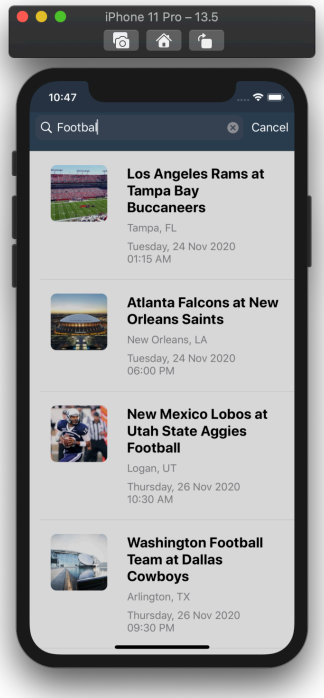
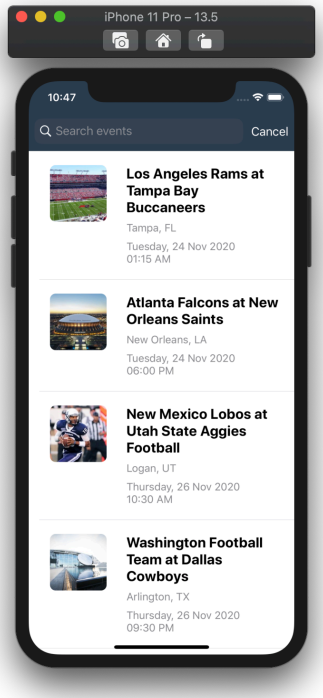
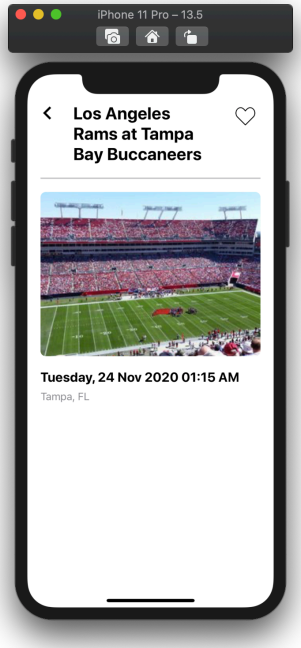
Seat Geek Coding Exercise

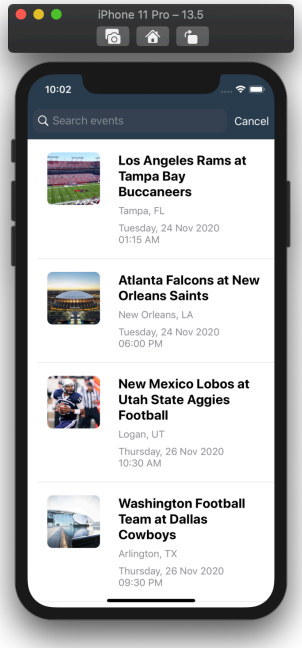
Create an iOS application that would consume the open-source **SeatGeek API** and display events in a UITableView as shown in the below screenshots. To search an event UISearchBar should be used and it should be placed on the top of the UITableView. **The application needs to fetch relevant events from SeatGeek API while user is typing in the search bar.**



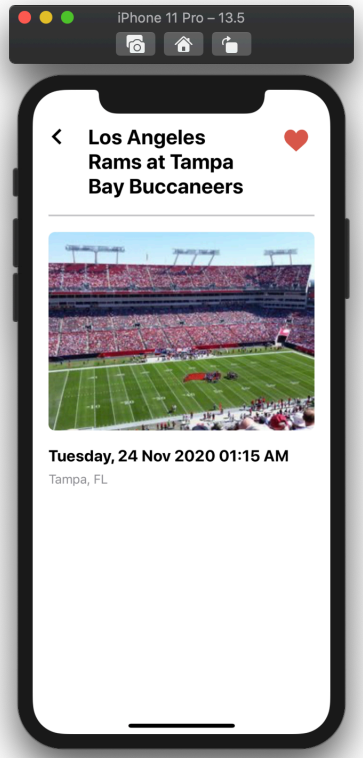


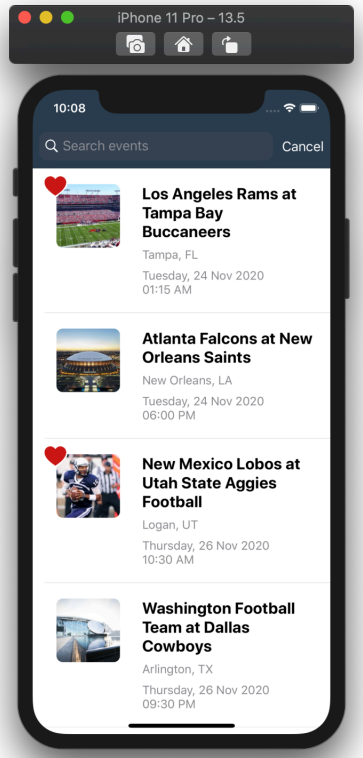
Tapping on Event UITableViewCell should display the corresponding event in a detail screen. Tapping on the back button should take user back to the events tableview.



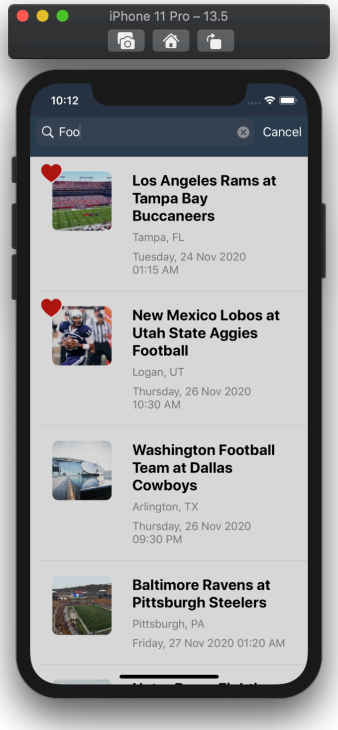


User should be able to favorite events from the detail screen by hitting the favorite button. Favorited events should be displayed on the events tableview as shown below.





Even when searching or closing the application, a favorited event should remain favorited. A user should be able to unfavorite the event as well.



Requirements

- Write your application for Native iOS Platform preferably with Swift (or Objective-C)
- Favorited events are persisted between app launches
- Events are searchable through SeatGeek API
- Unit tests are preferable.
- Third party libraries are allowed.
- *Cocoapods*, *Carthage*, *Swift Package Manager* are all allowed as long as there is a clear instructions on how to build and run the application.
- Make sure that the application supports **iOS 12** and above.
- The application must compile with ***Xcode 12.x.x***
- Please add a **README** or equivalent documentation about your project.
- The screenshots are just blueprints. UI doesn't have to follow them.

What do I need to submit?

Please write an iOS application as described above. Please see the API section below for further details regarding endpoint.

How do I submit it?

Provide a link to a public repository, such as GitHub or BitBucket, that contains your code to your recruiter.

FAQ

How will this exercise be evaluated?

An engineer will review the code you submit. At a minimum the app must provide the expected results. You should provide any necessary documentation within the repository. While your solution does not need to be fully production ready, you are being evaluated so put your best foot forward

I have questions about the problem statement

For any requirements not specified, use your best judgement to determine expected result.

Can I provide a private repository?

If at all possible, we prefer a public repository because we do not know which engineer will be evaluating your submission. Providing a public repository ensures a speedy review of your submission. If you are still uncomfortable providing a public repository, you can work with your recruiter to provide access to the reviewing engineer.

How long do I have to complete the exercise?

There is no time limit for the exercise. Please take as much time as you need to complete the work.

SeatGeek API

SeatGeek is an open source API that maintains a canonical directory of all live events in the United States. Please check the website for more information (https://platform.seatgeek.com/)

Here's the Resource endpoints

API Endpoint

```
https://api.seatgeek.com/2
```

Resource Endpoints

```
/events
/events/{EVENT_ID}
/performers
/performers/{PERFORMER_ID}
/venues
/venues/{VENUE_ID}
```

*** The application should consume the **/events** endpoint

SeatGeek API requires an authentication during API calls for that reason user of this service should create/receive a ***client_id*** and ***client_secret*** from the **SeatGeek Developers Page.**

Make sure to pass your ***client_id as your query parameters otherwise authentication would fail