

Результатом выполненного задания должно быть приложение, отображающее список результатов поиска по введенному пользователем текстовому запросу.

Поиск должен производиться сразу в двух сервисах:

- 1) Apple Search API (результаты поиска по iTunes)
- 2) GitHub Search API (Users - результаты поиска среди пользователей)

Внешний вид:

В `NavigationBar` должен быть `SegmentControl`, имеющий 2 закладки: *iTunes* & *GitHub*.

Под `NavigationBar` должен быть `SearchBar` для ввода тестового запроса, которой не должен перемещаться при скроллинге списка. `SearchBar` должен быть общим для обеих закладок.

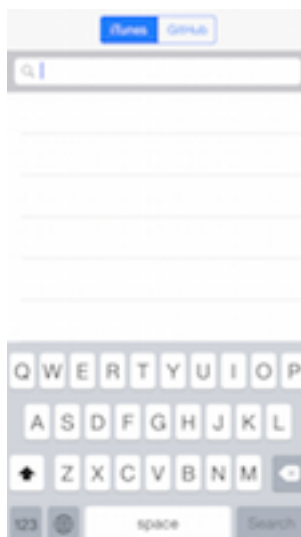
В закладке *iTunes*:

1. Отображаются результаты поиска в iTunes. Ячейка отображающая результат поиска в iTunes должна иметь иконку объекта, название и автора.
2. В нечетной ячейке иконка должна быть справа, а название и автор слева от иконки, а в каждой четной иконка должна быть слева, а название и автор справа от иконки.

В закладке *GitHub*:

1. Отображаются результаты поиска в GitHub. Ячейка отображающая результат поиска в GitHub должна иметь аватарку, логин и ссылку на аккаунт пользователя.
2. В нечетной ячейке иконка должна быть слева, а логин и ссылка справа от иконки, а в каждой четной иконка должна быть справа, а логин и ссылка на аккаунт слева от иконки.

(четность/нечетность определяется относительно отображаемой позиции, где элемент на индексе 0 является первой ячейкой, то есть нечетной)



Функциональные/Технические требования:

1. Картинка должна загружаться асинхронно.
2. При нажатии на иконку она должна анимировано расшириться и отображаться на весь экран. При повторном нажатии иконка должна возвращаться на место.
3. Верстку можно выполнить как полностью в коде, так и с использованием xib и storyboards (будьте готовы обосновать решение)
4. Нельзя использовать сторонние библиотеки. (исключение для библиотек верстки)
5. Если поиск не дал результатов, то мы должны уведомить пользователя о причине.
6. Не нужно использовать RAC/rx.
7. Предпочтительно выполнять задание на Swift, но в виде исключения можно и на Objective-C.
8. В первую очередь интересуют Unit-тесты, UI-тесты приветствуются.