iOS Interview Exercise

The exercise should result in a project, using data from the Marvel API, that meets the acceptance criteria of the following user stories:

1.

As a user

I want to see a list of Marvel Super Heroes

So I can pick the one I like

Acceptance criteria:

- Each element of the list should have the image of its super hero.
- This list should leverage the use of pagination (20 heroes per page).
- At each page load, show the user some feedback by displaying a loader.

2.

As a user

I want to navigate to a Hero Detail View

So I can take a look at the details of any Super Hero

Acceptance criteria:

- You must include a custom view controller transition.
- User should be able to see the following details of the following entities:
  - The details of 3 **comics** (if any) that this super hero takes part in. These details must at least include **name** and **description** of the comic.
  - The details of 3 **events** (if any) that this super hero takes part in. These details must at least include **name** and **description** of the event.
  - The details of 3 **stories** (if any) that this super hero takes part in. These details must at least include **name** and **description** of the story.
  - The details of 3 **series** (if any) that this super hero takes part in. These details must at least include **name** and **description** of the series.

3.

As a user

I want to easily find any Super Hero by their name

So I can avoid scrolling through lists of Super Heroes

Acceptance criteria:

- Search query should be done by the Super Hero's name.
- Search query should be done using the search REST endpoint, NOT a local search/filter.

4.

As a user

I want to favourite a Super Hero

So I can see all favourite Super Heroes while quickly glancing through Listing and Hero Detail Views

Acceptance criteria:

- Favourite UI components should be included in both Listing and Hero Detail Views.
- Favourite Heroes should persist between app sessions.

Notes

- Please send a link to a zipped project in dropbox (or similar) or a git repo.

- You CANNOT use third party libraries. Don't get us wrong, we love open source software but for the purpose of this exercise we believe any iOS engineer should be implement it using Apple's APIs directly.
- Feel free to use open-source code snippets, as long as you properly refer their origin.
- Performance should be an important feature of the app.
- Unit tests and UI Tests are welcome :).
- A readme file explaining the architecture or anything that you think it’s relevant is welcomed.
- Last, but not the least this exercise is not about problem solving and see if you can implement the user stories. We assume every candidate should be able to implement every single user story. This exercise is about showing how you can handle real life day to day problems in a way that is performant, readable and maintainable and this can only be done with good architecture. Although we like a beautiful app, for this exercise we vastly prefer beautiful code over a beautiful UI.