Senior iOS Developer

Develop an iPad application that conforms with the requirements below, given the provided endpoints. Provide a public github repo with the code and instructions, if needed.

Requirements

1. Set the navigation bar title to “Challenge Accepted!”
2. Use Swift
3. Fetch the data every time the app becomes active:
   Posts: http://jsonplaceholder.typicode.com/posts/
   Users: http://jsonplaceholder.typicode.com/users
   Photos: http://jsonplaceholder.typicode.com/albums
   Photos: http://jsonplaceholder.typicode.com/photos
4. Persist the data in Core Data with relationships
5. Merge fetched data with persisted data, even though the returned data of the API currently never changes. See (6).
6. Our UI has a master/detail view. The master view should always be visible and has a fixed width. The detail view adapts to the space available depending on orientation.
7. Display a table of all the posts in the master view. For each post display the entire post title and the users email address below the title. The cell will have a variable height because of that
8. Implement swipe to delete on a post cell. Because of (4) the post will appear again after the next fetch of the data which is expected.
9. Selecting a post (master) displays the detail to this post. The detail view consists of the post title, post body and related albums.
10. The creator of this post has some favorite albums that we want to display along the post. An album consists of the album title and a collection of photos that belong to the album.
11. The photos should be lazy-ly loaded when needed and cached. If the photo is in the cache it should take that, otherwise load from web and update the photo cell.
12. In general, provide UI feedback where you see fit.

Bonus points

- It would be nice to be able to show/hide the photos of an album. All albums are collapsed in the default state.
- Because the collection of photos can get quite long, we would like the headers to stick to the top.
- Include a search bar to search in the master view and provide live feedback to the user while searching.