

---

## Not A Basement Studio

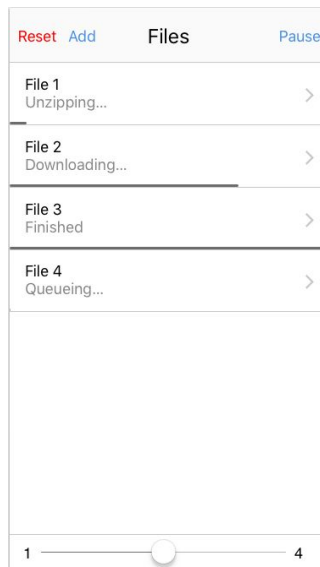
#4.13, Thien Son Plaza, 800 Nguyen Van Linh Str.  
Tan Phu Ward, District 7, Ho Chi Minh City, Vietnam

# Image Downloader

## OVERVIEW

This is the screening test for Not A Basement Studio iOS Dev interview. This test focuses on back-end using Cocoa Touch.

## SPECIFICATIONS



(figure 1)

### Screen 1

- The list will be empty on first launch.
- Tap on “Add”:
  - Download [this zip file](#) and populate its data to the app.
  - The zip file contains multiple JSON files, each file represents one File Record as you can see in *figure 1*.
  - In each File Record, there is a list of images to download. Download those images and link them to the File Record accordingly.
  - Tapping the button again will append those File Record again to the end of the list.
- Tap on “Reset”:
  - Stop all download processes and delete everything.
  - The list should be empty like on first launch.
- Tap on “Pause” / “Resume”:
  - Pause or resume all download processes.
  - This button should be disabled when there is no activity.
- File Record cell:
  - Tap on it will open Screen 2
  - Title: shows filename
  - Subtitle: shows current file state (Unzipping, Downloading, Queueing, Finished...), you are free to add more if you see fit
  - Progress bar: shows the progress of the current File Record. Feel free to break the progress proportion in whatever way you see fit
- The Slider at bottom will control the maximum concurrent image download processes.



(figure 2)

## Screen 2

- This screen shows the content of each File Record
- The content consists of images that can be either jpg, png, pdf, or zip (do unzip if you need to).
- Each item cell should somehow show the progress or activities (by showing UIActivityIndicator / error states / download progress).
- Tap on "Reload" will force redownload all the images
- Tap on each item will lead to Screen 3



(figure 3)

## Screen 3

- This screen is a simple viewer to browse all the images in each File Record.

## REQUIREMENTS

### Required

- Duration: 3 days starting from the date you received this document.
- Programming language: Objective-C and/or Swift (Swift 3.0 is preferable!)
- Code cleanliness and readability matter.
- The app should be responsive and run at 60fps (at least!).
- Use iOS Background Transfer Service to support downloading when app is suspended.  
Sample code: [Simple Background Transfer](#)
- Use NSProgress for progress reporting in Screen 1.  
Sample code: [Photo Progress](#)
- Use AAPLCollectionView to present the table in Screen 1 and the grid in Screen 2.  
Sample code: [Advanced Collection View](#)

### Optional

- You are allowed to add any external library you are familiar with.  
If you do, please use dependency managers. Don't just drag other people's code into your project!
- You are free to improve the look and feel of this app as you see fit.
- Support all device screens (with Split Screen and Slide Over) using iOS 10 adaptive layout features.