

- 1 *What have you learned recently about iOS development? How did you learn it? Has it changed your approach to building apps?*

I learned how to use Google Cloud functions to perform operations in the cloud instead of the devices handling all with the application. This feature has me very impressed because one can perform computationally expensive operations in the cloud, that otherwise would make your app slower. It is also an excellent way to add security because the code is not in your application, so one could use for example a token hard coded in the cloud without compromising the security.

- 2 *Can you talk about a framework that you've used recently (Apple or third-party)? What did you like/dislike about the framework?*

I recently used CoreLocation, Apple's framework to point your location. I liked it has some functionality built in it, like finding the distance from a location object to another location object. Or that is easy to obtain latitude and longitude from a location object, it is also neat that it comes with a timestamp that allows you to know how old is the location information. I do not like that it is not very reliable when used with devices that only use WiFi and do not have GPS integrated, like the iPad.

- 3 *Describe how you would construct a Twitter feed application (here is an example of Udacity's Twitter feed) that at minimum can display a company's Twitter page. Please include information about any classes/structs that you would use in the app. Which classes/structs would be the model(s), the controller(s), and the view(s)?*

I will create a struct

```
user{
    name: String
    urlString: String//the path to an image URL
}
```

I will create a struct that contains all the information of the Tweet, that would be

```
struct tweet {
    author: user
    text: String
    timeStamp: Date
    stars: Int
    hashtags: [Strings]?
}
```

Those would be part of my model. Then I would have a subclass of UITableViewController called TweetViewController. The TweetViewController class would contain an array of type [tweet] to be the data source. Then I would have a subclass of UITableViewCell, that would contain a UIImageView for the picture a UILabel for the text and four buttons at the bottom. The buttons would have a property of delegate, and the delegate protocol must be satisfied by the tweetViewController. For example, when a star button is pressed it would be let by delegation to the tweetViewController to increase the number of stars in the corresponding tweet object, that way is consistent with MVC. Or when the button with the three dots is pressed we update our views displaying the summary, again consistent with MVC.

- 4 *Describe some techniques that can be used to ensure that a UITableView containing many UITableViewCell is displayed at 60 frames per second.*

I am not sure I know the answer to this. When using a UITableViewCell y implement it the "standard way." You use the datasource delegate and the table view delegate. Then return the number of cells that you have on your table for the number of rows and dequeue a cell from the table so you can reuse it. You can usually display the cells fairly quick.

- 5 *Imagine that you have been given a project that has this ActorViewController. The ActorViewController should be used to display information about an actor. However, to send information to other ViewControllers, it uses UserDefaults. Does this make sense to you? How would you send information from one ViewController to another one?*

It does not make sense to me, perhaps I could send information to another View controller if it is going to be pushed into, you instantiate the viewController inside the ActorViewController and assign to the viewController properties with the information I wanted to send.

- 6 *Imagine that you have been given a project that has this `GithubProjectViewController`. The `GithubProjectViewController` should be used to display high-level information about a `GitHub` project. However, it's also responsible for finding out if there's network connectivity, connecting to `GitHub`, parsing the responses and persisting information to disk. It is also one of the biggest classes in the project. Follow-up question:: How might you improve the design of this view controller?*

It does not make sense to me to send information to another `ViewController` using `NSUserDefaults`, at least not in general. If the another `ViewController` is going to be pushed into `ActorViewController`, you might instantiate the `viewController` inside the `ActorViewController` and assign to the `viewController` properties with the information I wanted to send. If your `viewController` is somehow unrelated to the `ActorViewController` and you want to send information, you would need to create a model with the information you want to send and save it like in `coreData` or `FirebaseDatabase`. Then retrieve the necessary information on the another `ViewController` when it loads or when it is going to appear.

- 7 *If you were to start your `iOS` developer position today, what would be your goals a year from now?*

I would love to have a more of a leadership position within the company, where I could influence `Kuri`'s behavior and capabilities, not just implement the code. Another big goal would be to learn as much as I can of `React native` and anyother thing that would be need it to make `Kuri` the best robot it can be.