Requirements:

Modify the provided Assignment6 project so that it implements at least the tests described below.

Unit Tests:

A unit test should be added that tests the `CatService` `categories()` method. The test should retrieve the `CatService` singleton instance and call its `categories()` function. At a minimum the test should assert that the resulting `NSFetchedResultsController` instance contains more than 0 sections and that each section contains more than 0 items. You can use the various XCT prefixed functions that are available (`XCTAssertGreaterThan`, `XCTFail`, etc) to make these assertions. Calls to the XCT functions should provide a reasonable message that indicates what the assert was testing when it failed.
Hints:

- It is appropriate to call `XCTFail` in `catch` or `else` blocks to fail the test when expected conditions are not met
- If a test passes it can be useful to confirm the validity of the test by reversing/negating the assert statements and making sure the test fails. If they do not, your assert statements are potentially not testing anything important.

An additional test should be added that tests the `CatService` `images(for:)` method. The test should retrieve the `CatService` singleton instance and call its `images(for:)` method. At a minimum the test should assert that the resulting `NSFetchedResultsController` instance contains more than 0 sections and that each section contains more than 0 items.
Hints:

- Previous hints apply
- You will need to call the `categories()` method to retrieve a `Category` instance to pass to `images(for:)`
- Depending on how you want to retrieve a `Category` from the `NSFetchedResultsController`, you may need to construct an `IndexPath` instance for section 0 row 0.

UI Tests:

A test should be added that tests the basic functionality of the application. You can utilize UI recording as shown in class to create the basis for this test or write the code by hand. At a minimum the test should find the table element on the categories list and assert that it contains more than 0 cells. The test should then tap the first cell and wait for the cat images screen to appear. It should then find the collection view element on the cat images screen and assert that it contains more than 0 cells.

Hints:

- One good strategy to wait for a screen to appear is to setup an expectation for the navigation bar with the title of the expected screen to exist. This can be done with the code below:

```
let navBar = app.navigationBars["<#Navigation Item Title#>"]
let existsPredicate = NSPredicate(format: "exists == TRUE")
expectation(for: existsPredicate, evaluatedWith: navBar,
handler: nil)
waitForExpectations(timeout: 5.0, handler: nil)
```