Coding Report Outline

Due April 20, 2017

Team 3 (Jack Taylor, Mitchel Smith, Elan Kainen, and Jason Wen)

1. First, create a list of all of the code files and their names that have been written by the team (do not include files that may have been generated by a tool or the infrastructure). For each file, identify the hardware device on which it runs (e.g., handheld device, server, PC). This list should represent your project as of sometime on 4/20/2017 before class.
   The tree structure of our current application Xcode project. They key files are **bolded.**

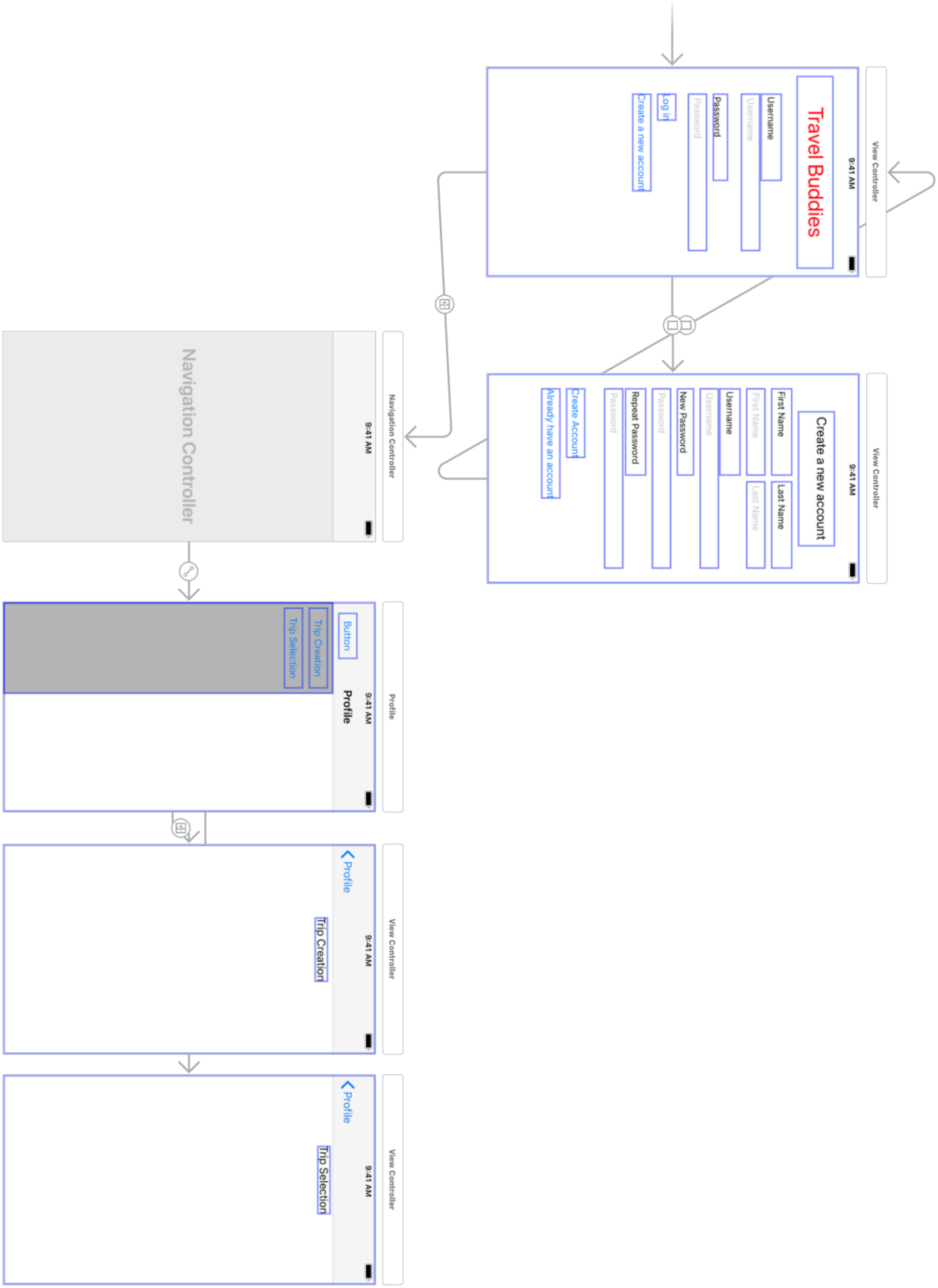
   ```
   ├── LoginWindow
   |   |   ├── AppDelegate.swift
   |   |   ├── Assets.xcassets
   |   |   |   └── AppIcon.appiconset
   |   |   |       └── Contents.json
   |   |   ├── Base.lproj
   |   |   |   ├── LaunchScreen.storyboard
   |   |   |   └── Main.storyboard
   |   |   ├── Info.plist
   |   |   ├── LoginWindow-Bridging-Header.h
   |   |   └── ViewController.swift
   |   ├── LoginWindow.xcodeproj
   |   |   ├── project.pbxproj
   |   |   ├── project.xcworkspace
   |   |   |   ├── contents.xcworkspacedata
   |   |   |   ├── xcshareddata
   |   |   |   |   └── IDEWorkspaceChecks.plist
   |   |   |   └── xcuserdata
   |   |   └── xcuserdata
   |   ├── LoginWindowTests
   |   |   ├── Info.plist
   |   |   └── LoginWindowTests.swift
   |   └── LoginWindowUITests
   |       ├── Info.plist
   |       └── LoginWindowUITests.swift
   ```

2. Next, please create a snapshot of the code listed under question 1 above. This should consist of a printed copy of each code file written by the team. I would prefer that the code be printed with line numbers, if possible. You should submit a single document containing your responses to these questions.

```
//
//  AppDelegate.swift
//  LoginWindow
//
//  Created by Wen, Jason on 4/4/18.
//  Copyright © 2018 Wen, Jason. All rights reserved.
//

import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

var window: UIWindow?
//Test comment to show how git works

func application(_ application: UIApplication,
didFinishLaunchingWithOptions launchOptions:
[UIApplicationLaunchOptionsKey: Any]?) -> Bool {
// Override point for customization after application launch.
return true
}

func applicationWillResignActive(_ application: UIApplication) {
// Sent when the application is about to move from active to
inactive state. This can occur for certain types of temporary
interruptions (such as an incoming phone call or SMS message) or
when the user quits the application and it begins the transition to
the background state.
// Use this method to pause ongoing tasks, disable timers, and
invalidate graphics rendering callbacks. Games should use this
method to pause the game.
}

func applicationDidEnterBackground(_ application: UIApplication) {
// Use this method to release shared resources, save user data,
invalidate timers, and store enough application state information
to restore your application to its current state in case it is
terminated later.
// If your application supports background execution, this method
is called instead of applicationWillTerminate: when the user quits.
}

func applicationWillEnterForeground(_ application: UIApplication) {
// Called as part of the transition from the background to the
active state; here you can undo many of the changes made on
entering the background.
}
func applicationDidBecomeActive(_ application: UIApplication) {
// Restart any tasks that were paused (or not yet started) while
the application was inactive. If the application was previously in
the background, optionally refresh the user interface.
}

func applicationWillTerminate(_ application: UIApplication) {
// Called when the application is about to terminate. Save data if
appropriate. See also applicationDidEnterBackground:.
}


}
```

```
//
//  ViewController.swift
//  LoginWindow
//
//  Created by Wen, Jason on 4/4/18.
//  Copyright © 2018 Wen, Jason. All rights reserved.
//

import UIKit

class ViewController: UIViewController {



    @IBOutlet weak var leadingConstraint: NSLayoutConstraint!


    var menuShowing = false

    override func viewDidLoad() {
        super.viewDidLoad()
    }


    @IBAction func openMenu(_ sender: Any) {

        if (menuShowing){
            leadingConstraint.constant = -140
        }
        else{
            leadingConstraint.constant = 0
        }

        menuShowing = !menuShowing
    }

}
```

Main.storyboard



```
//
//  LoginWindowTests.swift
//  LoginWindowTests
//
//  Created by Wen, Jason on 4/4/18.
//  Copyright © 2018 Wen, Jason. All rights reserved.
//

import XCTest
@testable import LoginWindow

class LoginWindowTests: XCTestCase {

    override func setUp() {
        super.setUp()
        // Put setup code here. This method is called before the
invocation of each test method in the class.
    }

    override func tearDown() {
        // Put teardown code here. This method is called after the
invocation of each test method in the class.
        super.tearDown()
    }

    func testExample() {
        // This is an example of a functional test case.
        // Use XCTAssert and related functions to verify your tests
produce the correct results.
    }

    func testPerformanceExample() {
        // This is an example of a performance test case.
        self.measure {
            // Put the code you want to measure the time of here.
        }
    }

}
```

```
//
//  LoginWindowUITests.swift
//  LoginWindowUITests
//
//  Created by Wen, Jason on 4/4/18.
//  Copyright © 2018 Wen, Jason. All rights reserved.
//

import XCTest

class LoginWindowUITests: XCTestCase {

    override func setUp() {
        super.setUp()

        // Put setup code here. This method is called before the
invocation of each test method in the class.

        // In UI tests it is usually best to stop immediately when a
failure occurs.
        continueAfterFailure = false
        // UI tests must launch the application that they test. Doing
this in setup will make sure it happens for each test method.
        XCUIApplication().launch()

        // In UI tests it’s important to set the initial state - such
as interface orientation - required for your tests before they run.
The setUp method is a good place to do this.
    }

    override func tearDown() {
        // Put teardown code here. This method is called after the
invocation of each test method in the class.
        super.tearDown()
    }

    func testExample() {
        // Use recording to get started writing UI tests.
        // Use XCTAssert and related functions to verify your tests
produce the correct results.
    }

}
```