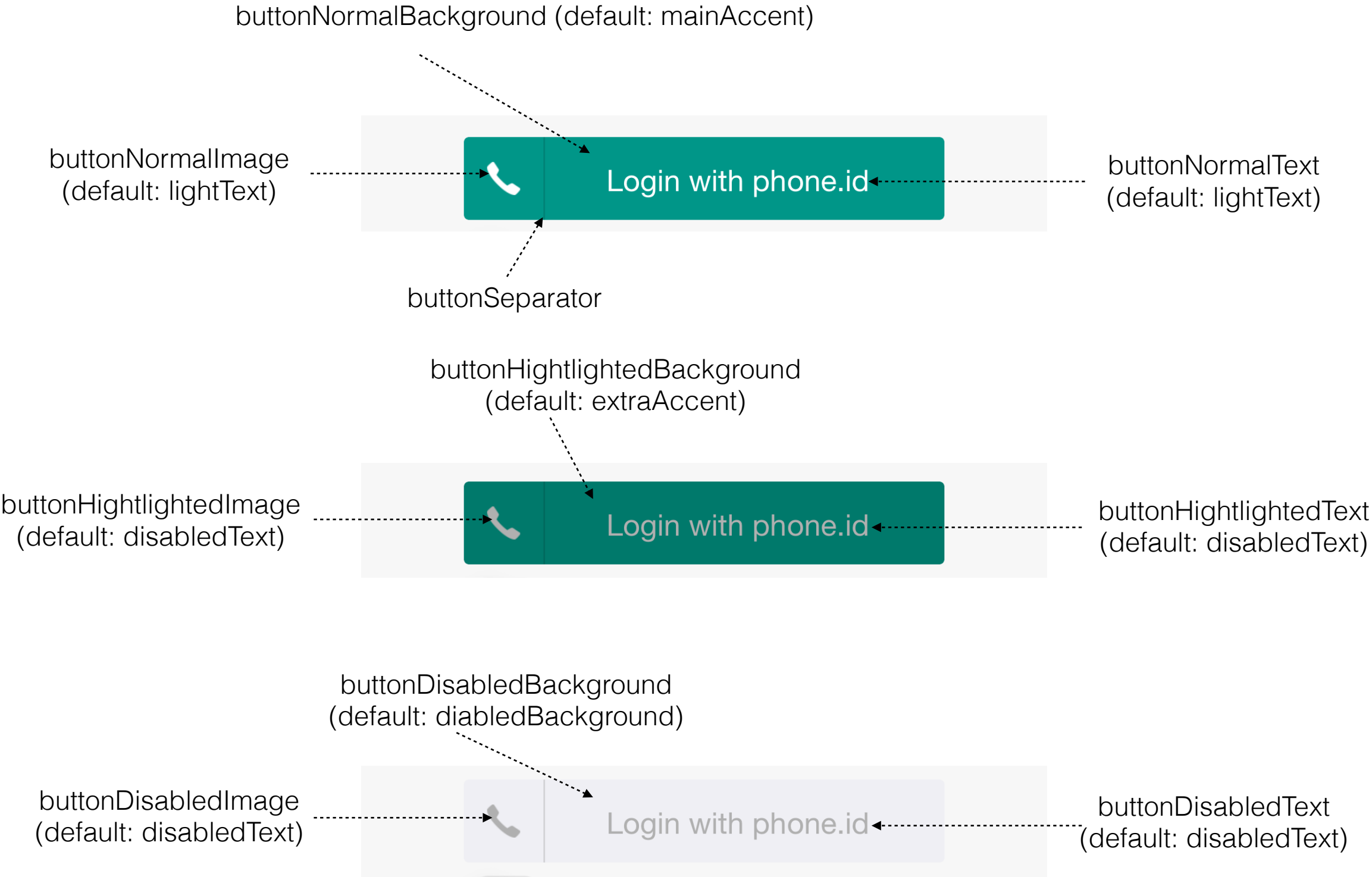
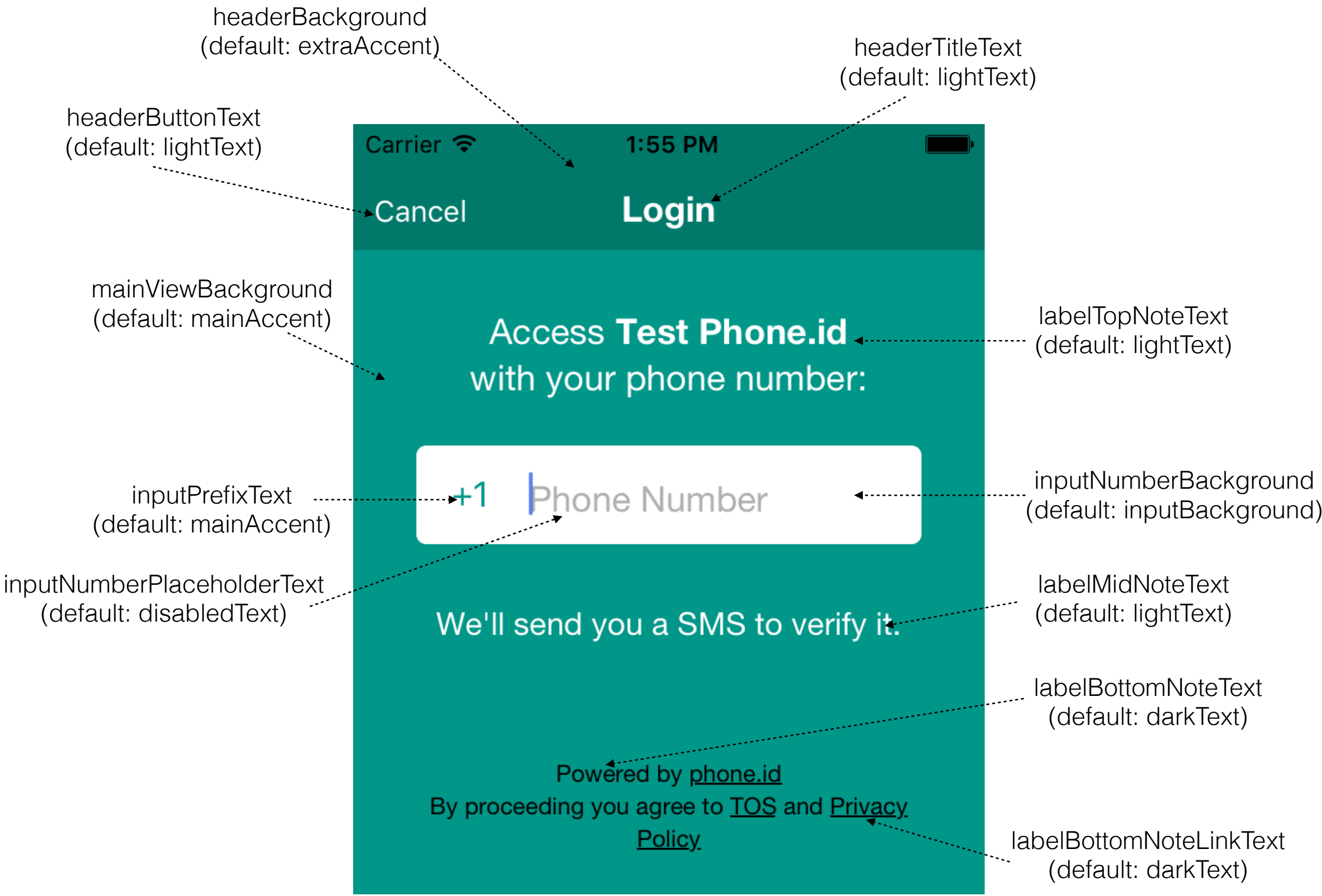


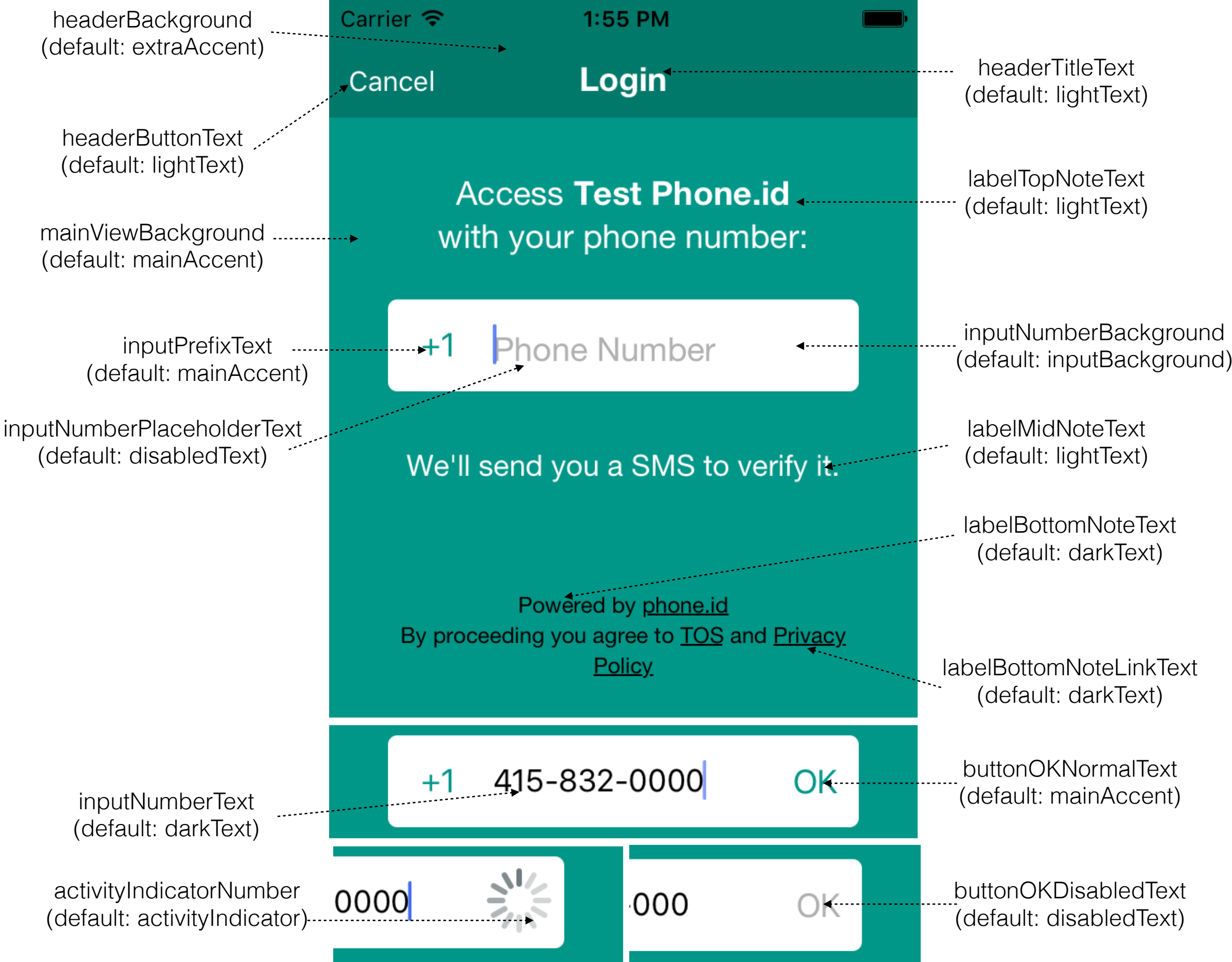
mapping of phoneid_iOS.ColorScheme fields to UI elements: **PhoneIdButton**



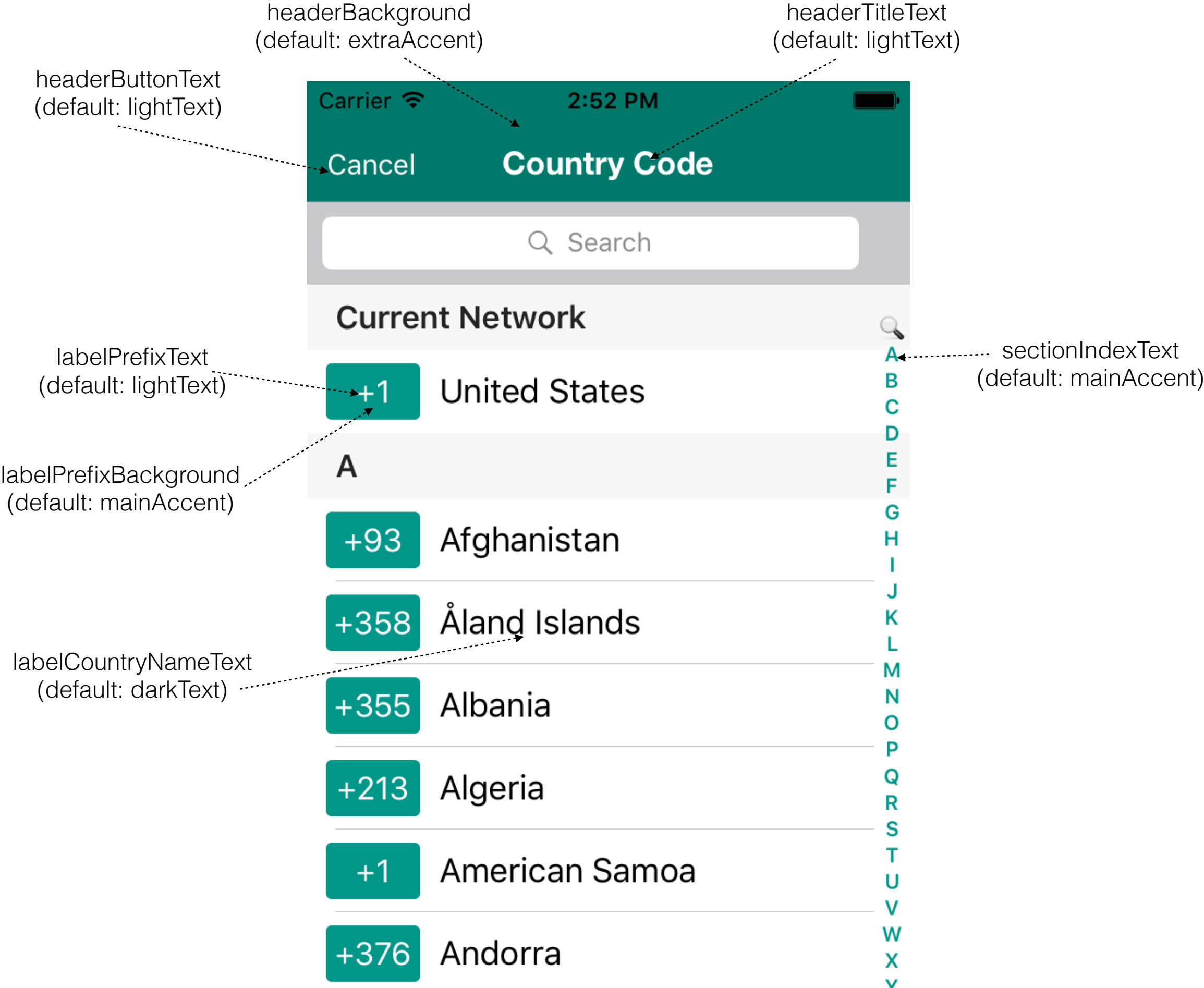
mapping of phoneid_iOS.ColorScheme fields to UI elements: **Number input**



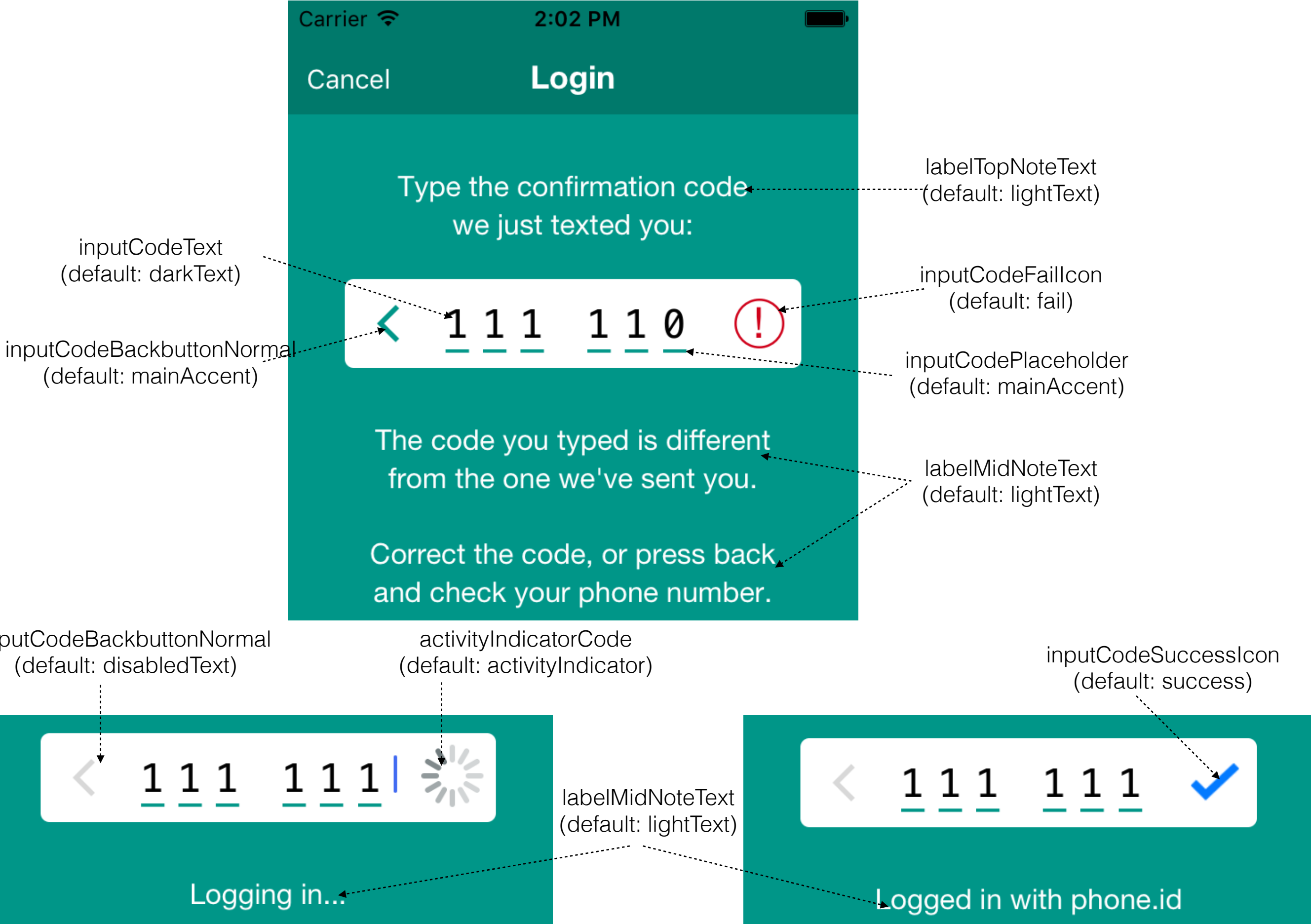
mapping of phoneid_iOS.ColorScheme fields to UI elements: **Number input**



mapping of phoneid_iOS.ColorScheme fields to UI elements: **Country Code**



mapping of phoneid_iOS.ColorScheme fields to UI elements: **Code verification**



Theming in action

```
import UIKit
import phoneid_iOS

class CustomComponentFactory:DefaultComponentFactory{

    override func defaultBackgroundImage()->UIImage?{
        return UIImage(named:"background")!
    }

    override func colorScheme()->ColorScheme{

        let scheme = super.colorScheme()

        // You can change main colors
        scheme.mainAccent = UIColor(hex: 0xAABB44)
        scheme.extraAccent = UIColor(hex: 0x886655)
        scheme.success = UIColor(hex: 0x91C1CC)
        scheme.fail = UIColor(hex: 0xD4556A)
        scheme.inputBackground = UIColor(hex: 0xEEEEED).colorWithAlphaComponent(0.6)

        scheme.applyCommonColors()

        // But also, if some of main colors don't fit to your color solution,
        // you can specify your own colors for certain UI element:
        scheme.buttonHighlightedImage = UIColor(hex: 0x778230)
        scheme.buttonHighlightedText = UIColor(hex: 0x778230)
        scheme.buttonHighlightedBackground = UIColor(hex: 0xBBC86A)

        return scheme
    }
}

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

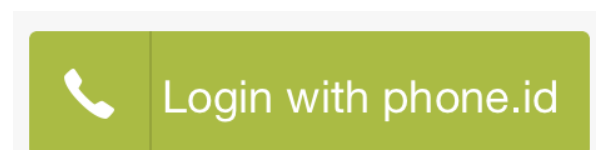
    var window: UIWindow?


    func application(application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {

        //configure phone.id
        PhoneIdService.sharedInstance.componentFactory = CustomComponentFactory()

        PhoneIdService.sharedInstance.configureClient("TestPhoneId");
        return true
    }
}
```

Normal



Highlighted
(when user touches)

