CoinWallet

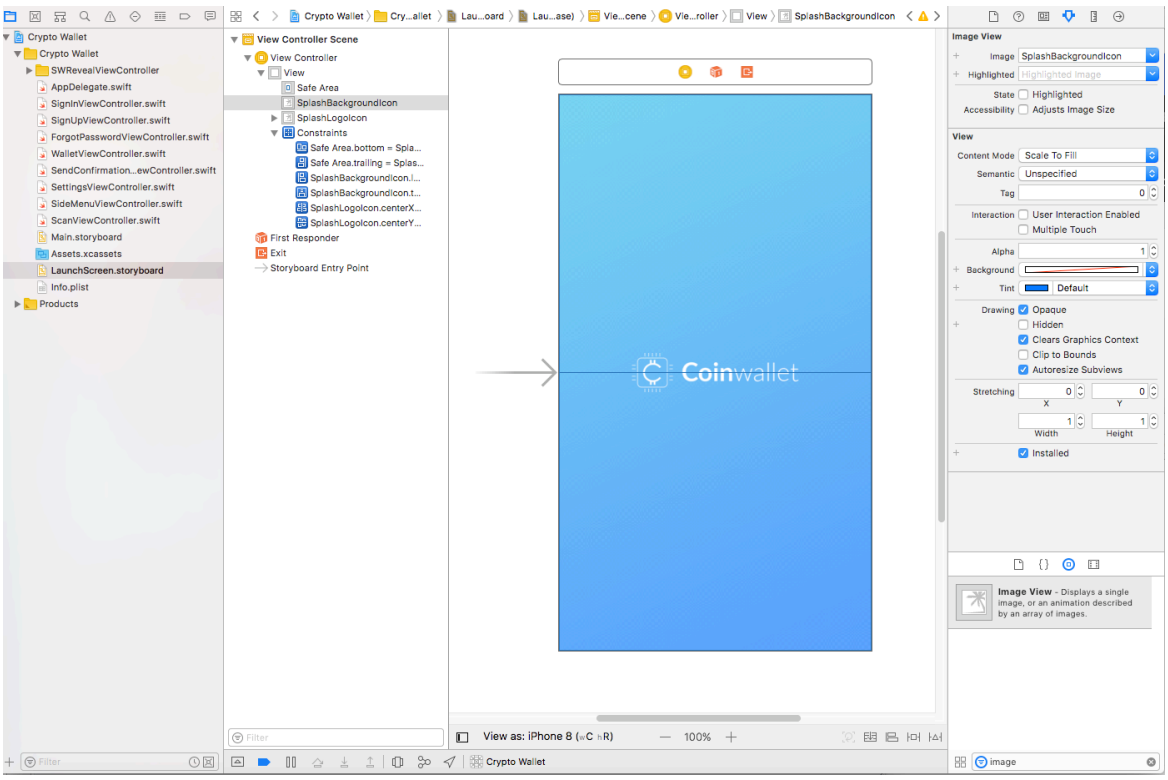
Created: 21/03/2018
By: iTechTheme
Email: itechtheme@gmail.com

Thank you for purchasing my app. If you have any questions that are beyond the scope of this help file, please feel free to email via my user page contact form (**https://codecanyon.net/user/itechtheme**)
Thanks so much!

Table of contents

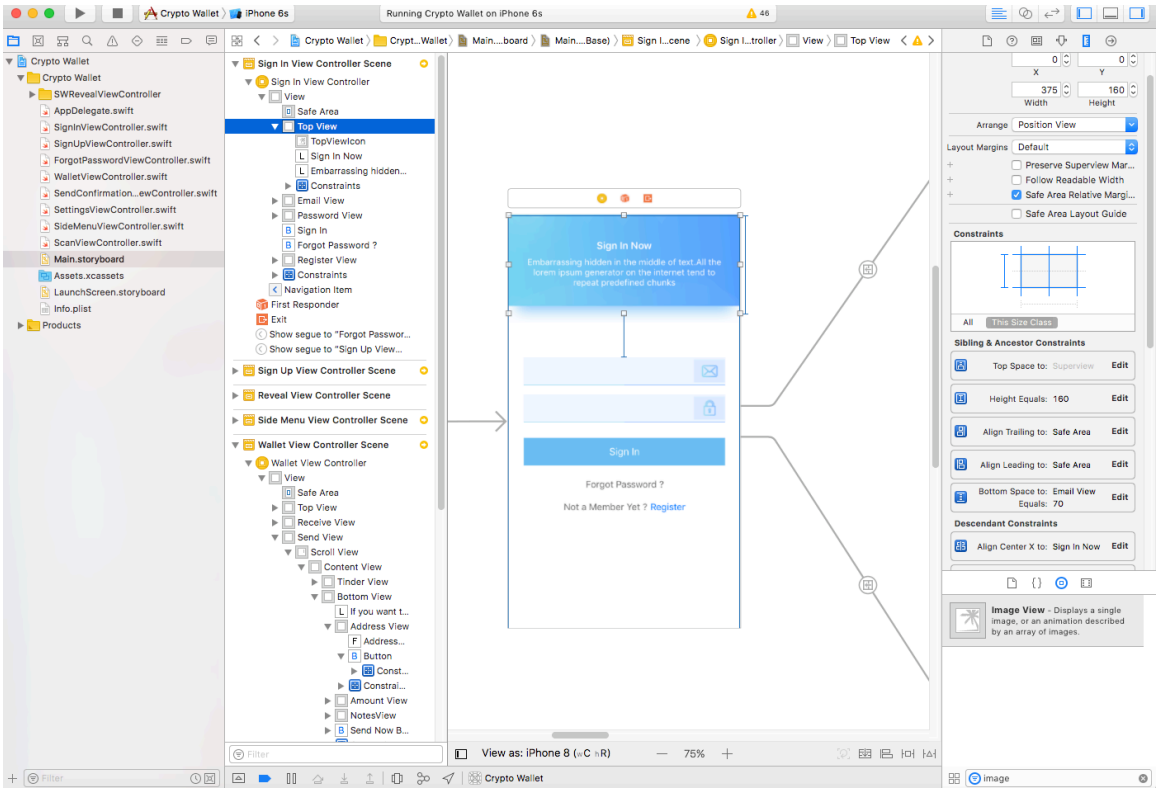
A. LaunchScreen (LaunchScreen.storyboard)

The launch Screen appears instantly when the app starts up. Here to replace or change the launch screen just change “SplashBackgroundIcon” and “SplashLogoIcon“ images from SplashIcons Group in Assets and update two Imageviews from attribute inspector.



B. Storyboard Structure and Constraint

This app has been designed using auto layout and constraint so as to fit in different screens size. Drag and drop a control from object library to a scene and set constraint for it to fit in various device screens.



Here at first we've put a UIView (Top View) from Object Library to the Scene and set all constraints (**Leading Space 0, Trailing Space 0, Top Space 0 and Height 160**) to fit in different screens. Then drag a UIImageView from Object Library to UIView and set all constraints (**Leading Space 0, Trailing Space 0, Top Space 0 and Bottom Space 0**) to fit whole view. Then select the desired background image from attribute inspector for ImageView. You can change the background image by changing the image from Attribute Inspector Image property.

Now let's think about "Email View" UIView to change the left and right alignment you've to change the Top Space, Leading Space and Trailing Space (In right side bar's size inspector) respectively. You can also set height constraint to set fixed height. Inside UIView We've put another two child's view UITextField and UIImageView too and set constraints now respect to "Email View". To change the UIView's background color just click attribute inspector and Change background color.

All other controls and Scenes have been also set with Auto layout and constraint in this way and if you want to edit the specific part of app then find out screen and select the control from storyboard.

C. Swift Code Explanation

The app includes 8 UIViewController Classes to customize the app:

1. SignInViewController.swift
2. SignUpViewController.swift
3. ForgotPasswordViewController.swift
4. WalletViewController.swift
5. SendConfirmationViewController.swift
6. SettingsViewController.swift
7. SideMenuViewController.swift
8. ScanViewController.swift

1. SignInViewController.swift

This ViewController is defined for SignIn Screen and it has following methods and actions:

Use this function to call API for verifying the existing users.

```
//SignIn action

   @IBAction func SignInAction(_ sender: Any) {

       // Code goes here to call API

// Navigate to the Dashboard
if  let swvc=self.storyboard?.instantiateViewController(withIdentifier:"SWReveal")
 {
self.navigationController?.pushViewController(swvc, animated: true)

 }
}
```

Following Instance Method is called after the controller's view is loaded into memory and use to perform additional initialization on views that were loaded from nib files

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

```
 // additional settings will go here after loading view

}
```

The instance property is used to set preferred status bar style for the view controller.

```
  // Change Status Bar style
    override var preferredStatusBarStyle: UIStatusBarStyle {

        return .lightContent    // White Status Bar

    }
```

3. ForgotPasswordViewController.swift

This Viewcontroller is used to reset new password and it includes following actions and methods-

Use this function to go back previous screen (SignIn Screen)

```
// Back to previous controller
    @IBAction func backAction(_ sender: Any) {

        self.navigationController?.popViewController(animated: true)
    }
```

Following function will be used to call an API to send code or reset-link to the email.

```
// Action to send code into Email

    @IBAction func sendNowAction(_ sender: Any) {
        // Code goes here
    }
```

4. WalletViewController.swift

Very time we need to set effect for button's normal and selected state. Following way we can give effect-using image as below:

```
// Set state for Receive selector button

 receiveSelectorButton .setImage(UIImage(named:"ReceiveNormalIcon"), for:
UIControlState.normal)
 receiveSelectorButton .setImage(UIImage(named:"ReceiveSelectIcon"), for:
UIControlState.selected)
```

If you want to set custom placeholder then use the following ways-

```
// set custom placeholder color for address textfield

addressTextField.attributedPlaceholder=NSAttributedString(string:"Bitcoin Address",
attributes: [NSAttributedStringKey.foregroundColor :UIColor.lightGray])
```

transactionTableView is used to presents data and it has the following instance methods to present the data and before implementing these methods set delegate and datasource in countryTableView.

```
// Set delegate and datasource

    coutryTableView.delegate = self
    coutryTableView.dataSource = self

// Return no. of sections in countryTableView

func numberOfSectionsInTableView(tableView: UITableView) -> Int {
        return 1;
    }

// Return no. of rows in countryTableView's per section

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return country.count
  }

// Return  row height of countryTableView

    func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) ->
CGFloat
    {
        return 50; // Height of Row
    }

// Ask the datasource for a cell to insert in a particular location of countryTableView

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {

       let cell = tableView.dequeueReusableCell(withIdentifier:transactionIdentifier )
        cell?.selectionStyle=UITableViewCellSelectionStyle.none
        cell?.backgroundColor = .clear
        let paymentImageView=cell?.viewWithTag(1) as! UIImageView
        let paymentLabel=cell?.viewWithTag(2) as! UILabel
        if receivedButton.isSelected {
            paymentLabel.text="Payment Received"
            paymentImageView.image=UIImage(named:"PaymentReceivedIcon")
        }else{
            paymentLabel.text="Payment Send"
            paymentImageView.image=UIImage(named:"PaymentSendIcon")
        }
        return cell!
  }
```

5. SendConfirmationViewController.swift

To verify sending coins this view controller is used and there're several functions and actions to handle this.
In viewDidLoad method some controls have been setup as:

To make button's corner curve use the following codes and increase value if you need to update curve

```
// Make round corner

confirmButton.layer.cornerRadius=2.0
confirmButton.clipsToBounds=true
```

To make only particular corner(Top-right, Bottom-left or Top-right) curve use the following codes-

```
//  Make round corner for top-left and top-right for BottomView
    let bottomViewPath = UIBezierPath(roundedRect:bottomView.bounds,
                        byRoundingCorners:[.topRight, .topLeft],
                          cornerRadii: CGSize(width: 20, height:  20))

    let maskLayer = CAShapeLayer()
    maskLayer.path = bottomViewPath.cgPath
    bottomView.layer.mask = maskLayer
```

Use the following function to confirm the transaction-

```
// Confrim action to complete the transaction
  @IBAction func confirmButtonAction(_ sender: Any) {
     // Submission code goes here
  }
```

8. ScanViewController.swif

This view controller has been used to scan QR Code using AVFoundation Framework. We've to do following things to create QR Code scanner-

At first import "AVFoundation and add "AVCaptureMetadataOutputObjectsDelegate" protocol . Declare following variables and outlet –

// Declare variables

```
 @IBOutlet var preView: UIView!
@IBOutlet var squareImageView: UIImageView!
var session: AVCaptureSession!
```

```
var previewLayer: AVCaptureVideoPreviewLayer!
```

And additional setup has been done in viewDidLoad method.

When the AVCaptureMetadataOutput object recognizes a QR code, the following delegate method of AVCaptureMetadataOutputObjectsDelegate will be called:

```
Func metadataOutput(_ output:AVCaptureMetadataOutput, didOutput
metadataObjects:[AVMetadataObject],from connection: AVCaptureConnection) {

// Decoding code will go here

}
```

You can have more helps and details from following links:

https://www.appcoda.com/qr-code-reader-swift/

****** All other Controllers contain same type of coding methods and properties that's why didn't repeat the code***

D. Assets

We've used several icons and images for the app and the icons have been grouped as listed:

- ReceiveIcons
- ScanIcons
- SendIcons
- SettingsIcons
- SideMenuIcons
- SplashIcons
- TransactionIcons

If you'd like to change the Receive, Send, Transaction or Sidemenu Screens' Icons, then replace it as same name otherwise you've to change it from attributed inspector from storyboard. Do the same things for all Icons or images too.

E. Sources and Libraries

We've used following UIViewController subclass and AVFoundation Framework that allows presenting side view controllers and scan QR Code respectively as listed-

- SWRevealViewController (***Source***: https://www.appcoda.com/sidebar-menu-swift/)
- AVFoundation Framework (***Source***: https://www.appcoda.com/qr-code-reader-swift/)

Thanks again for purchasing the app and looking forward to helping you if you've any questions regarding this app.

iTechtheme.