

README

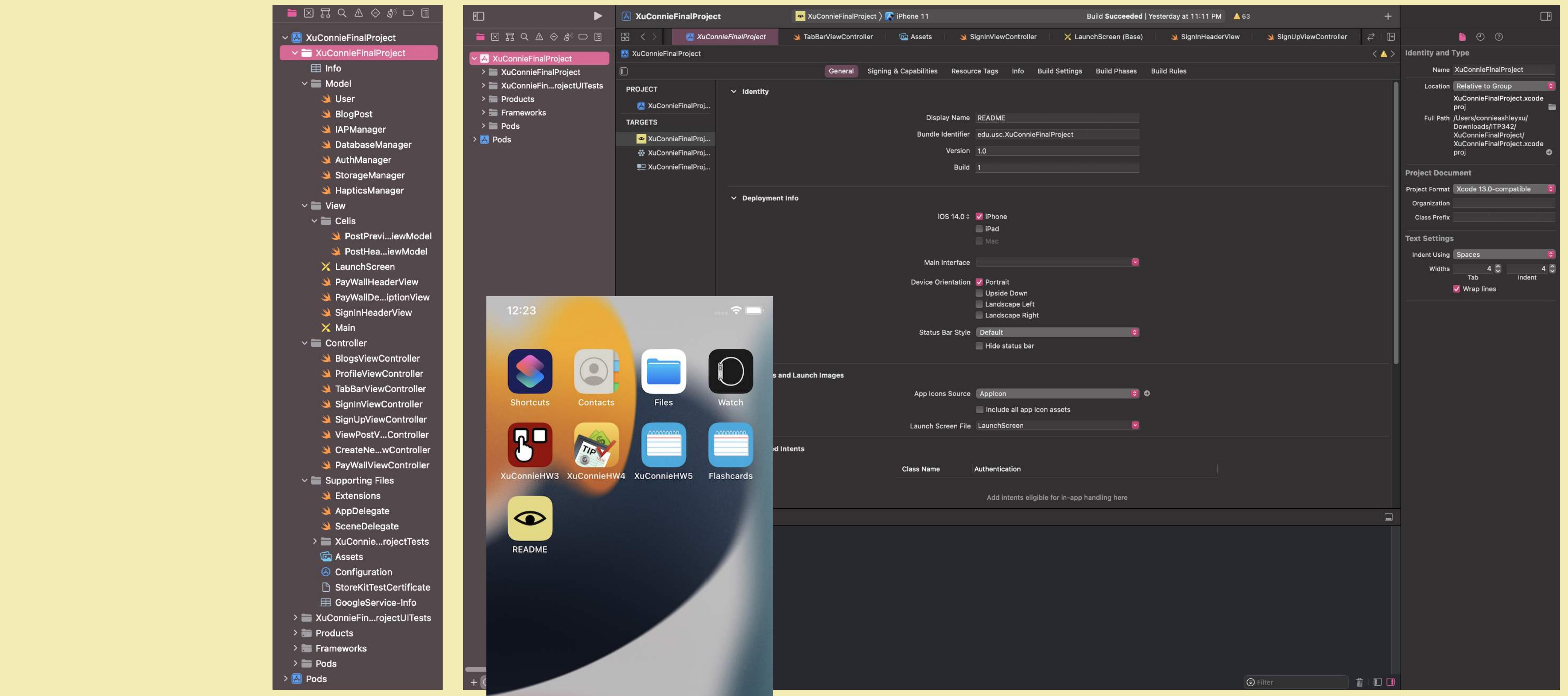
subscription-based blogging app

link to video + oral presentation [here](#)

PROJECT SPECS

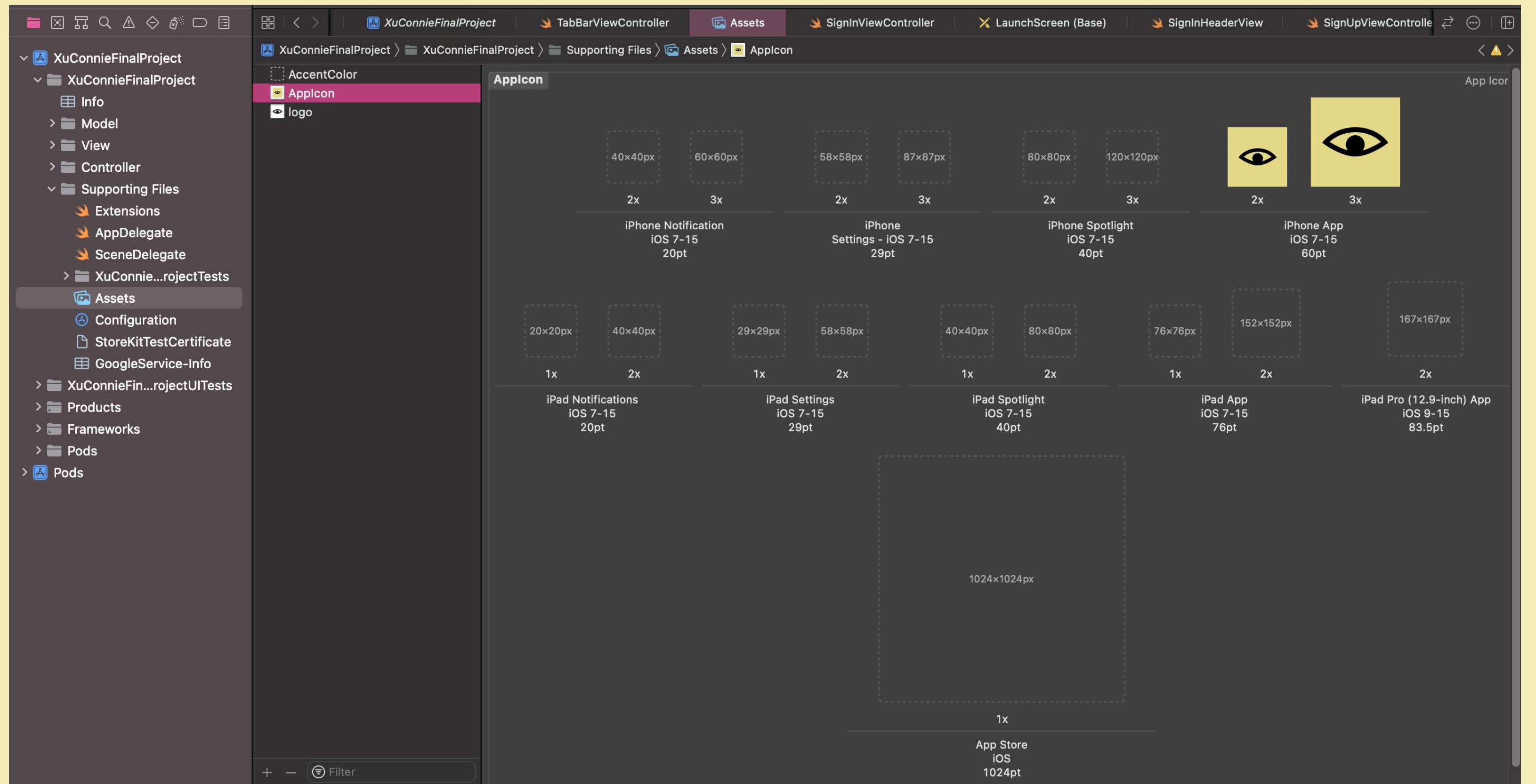
Coded in Swift

proof of file organization + home screen app icon/app name



PROJECT SPECS

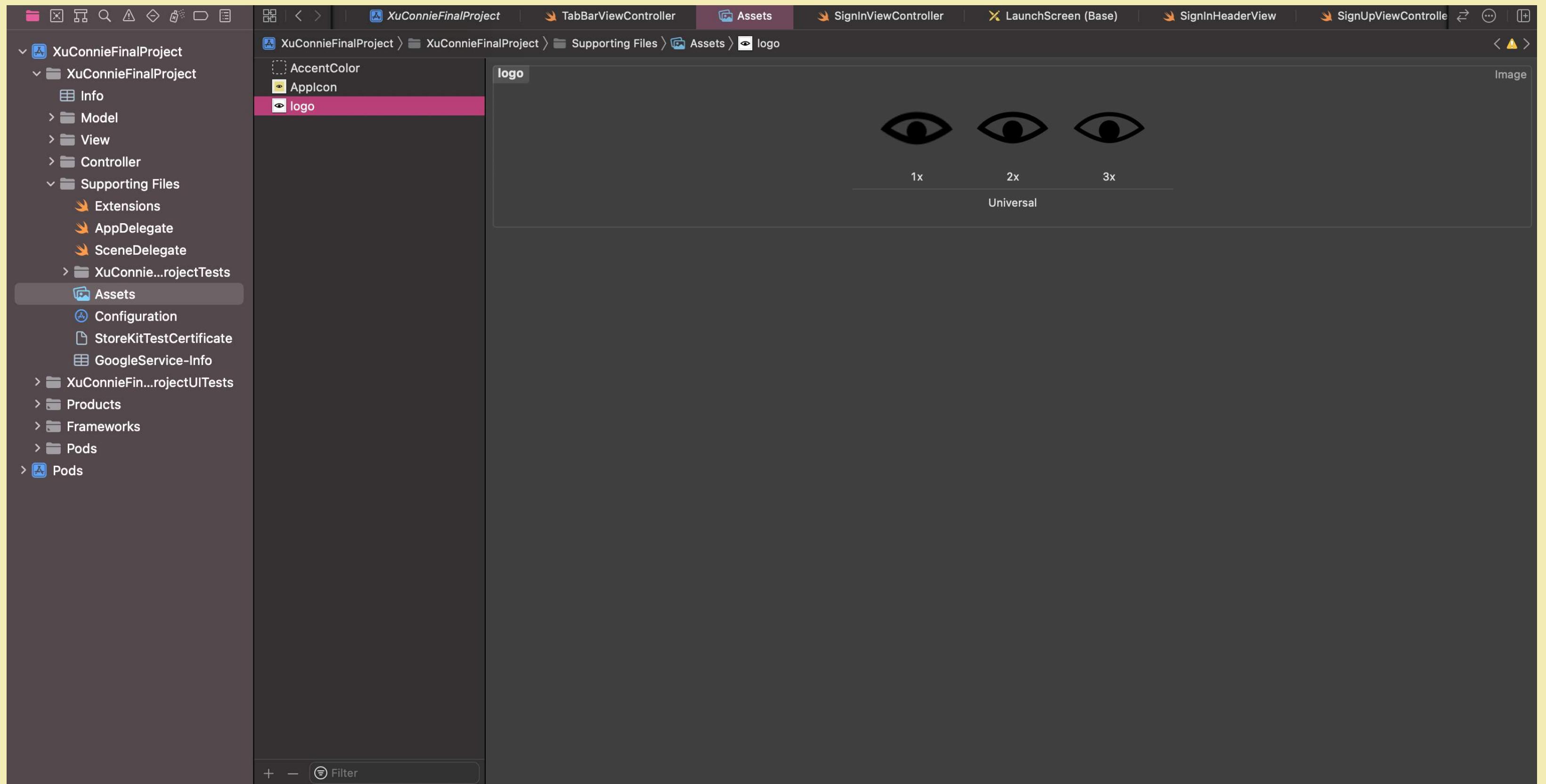
@2x & @3x App Icons



> XuConnieFinalProject > Supporting Files > Assets.xcassets

PROJECT SPECS

Minimum @2x & @3x Assets



> XuConnieFinalProject > Supporting Files > Assets.xcassets

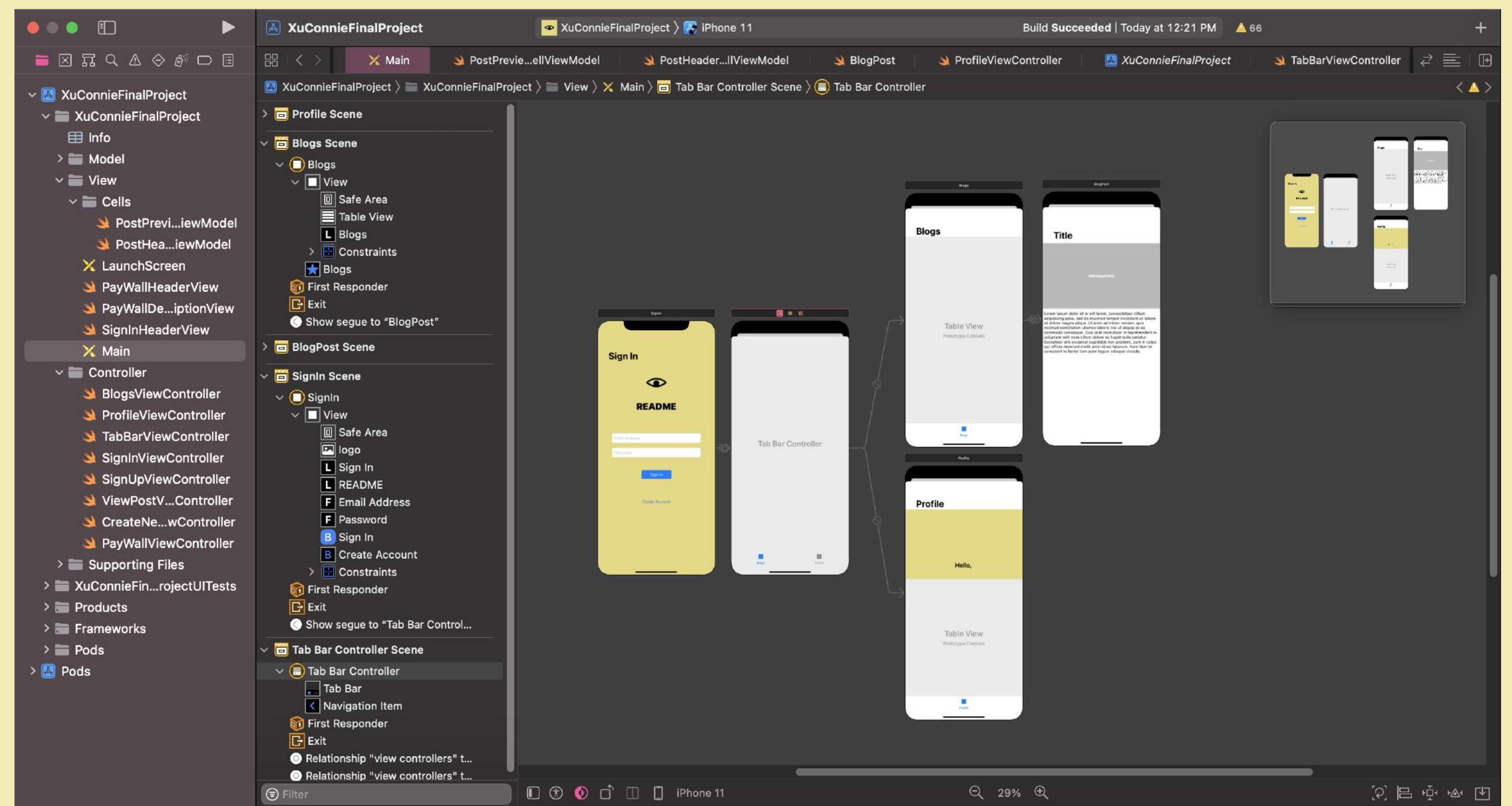
PROJECT SPECS

Storyboard

*See extra notes pivot

proof of understanding on how to use .storyboard

- auto-layout
- view controller class names
- segue connections



> XuConnieFinalProject > View > Main.storyboard

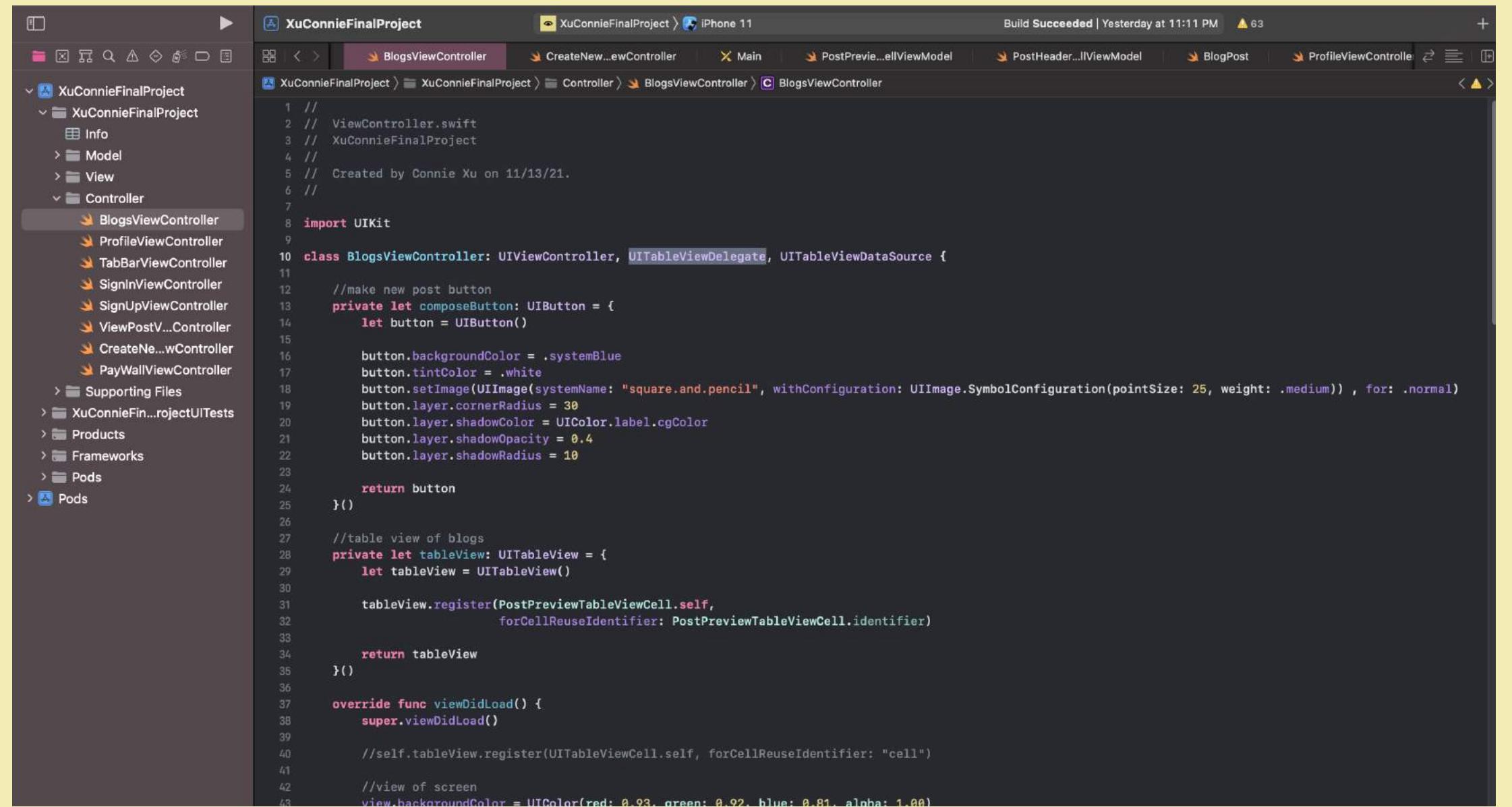
PROJECT SPECS

Delegation Design Pattern

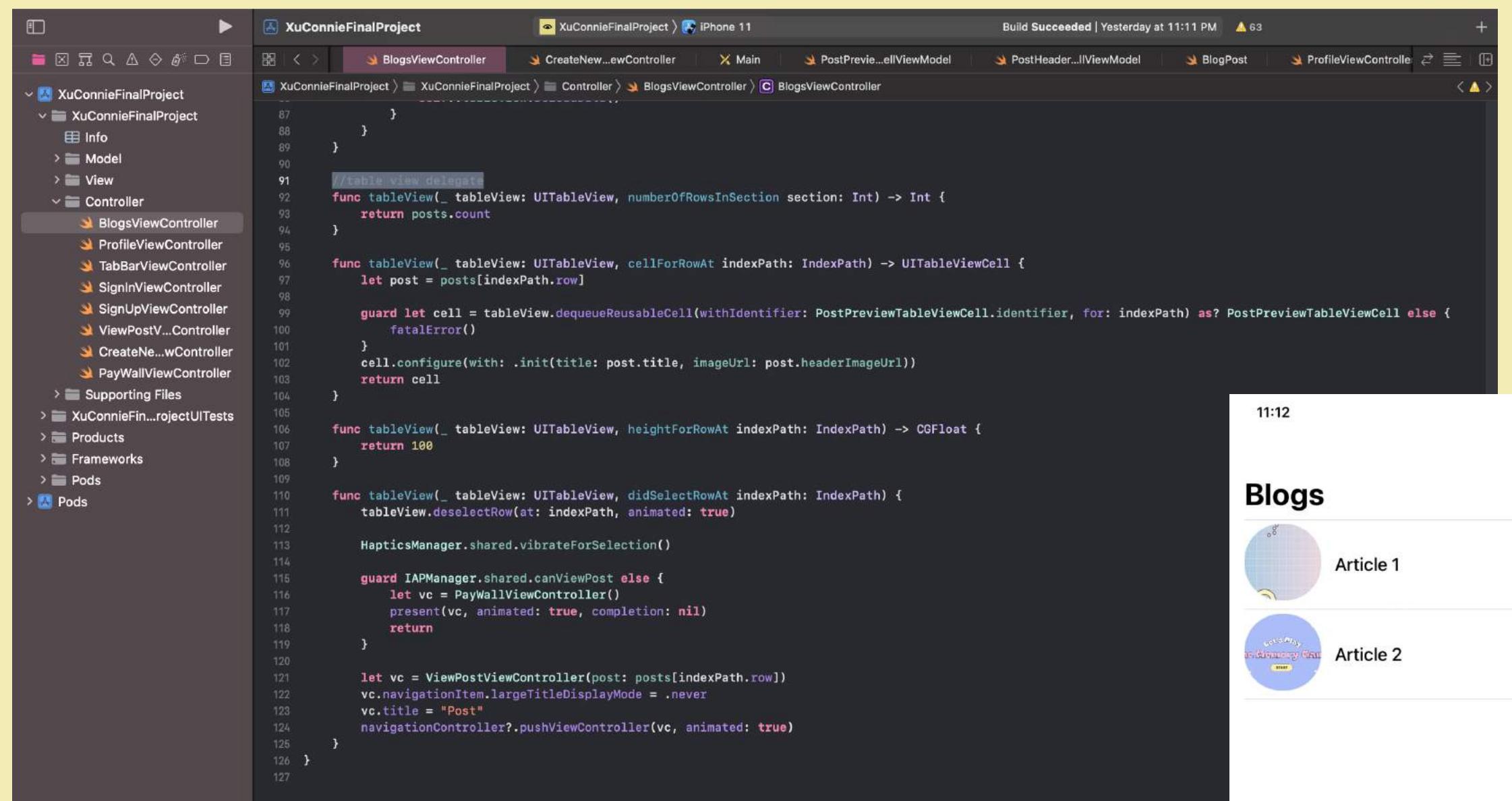
Table View

Delegate: UITableViewDelegate
Delegator: BlogsViewController

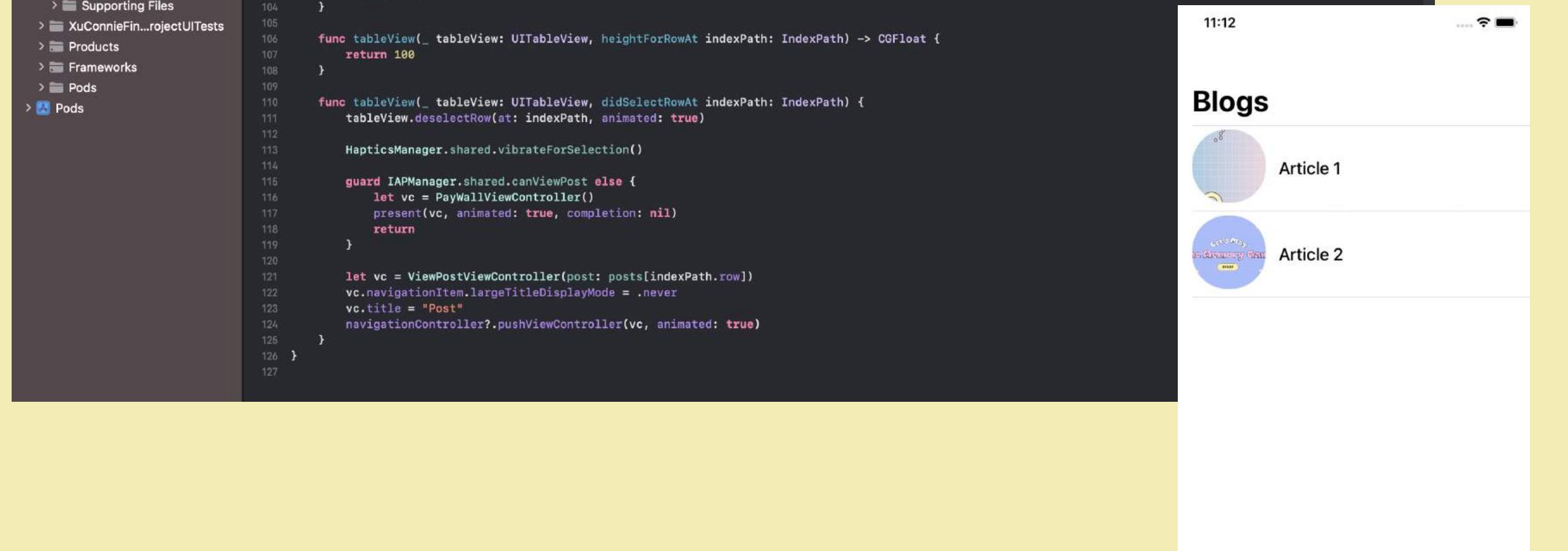
> XuConnieFinalProject > Controller > BlogsViewController.swift
> ProfileViewController.swift
> ViewPostViewController.swift



```
1 // ViewController.swift
2 // Info
3 // XuConnieFinalProject
4 // Created by Connie Xu on 11/13/21.
5 //
6 import UIKit
7
8 class BlogsViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
9
10    //make new post button
11    private let composeButton: UIButton = {
12        let button = UIButton()
13
14        button.backgroundColor = .systemBlue
15        button.tintColor = .white
16        button.setImage(UIImage(systemName: "square.and.pencil", withConfiguration: UIImage.SymbolConfiguration(pointSize: 25, weight: .medium)), for: .normal)
17        button.layer.cornerRadius = 30
18        button.layer.shadowColor = UIColor.label.cgColor
19        button.layer.shadowOpacity = 0.4
20        button.layer.shadowRadius = 10
21
22        return button
23    }()
24
25
26    //table view of blogs
27    private let tableView: UITableView = {
28        let tableView = UITableView()
29
30        tableView.register(PostPreviewTableViewCell.self,
31                           forCellReuseIdentifier: PostPreviewTableViewCell.identifier)
32
33        return tableView
34    }()
35
36    override func viewDidLoad() {
37        super.viewDidLoad()
38
39        //self.tableView.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
40
41        //view of screen
42        view.backgroundColor = UIColor(red: 0.93, green: 0.92, blue: 0.81, alpha: 1.00)
43    }
44}
```



```
87
88
89
90
91    //table view delegate
92    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
93        return posts.count
94    }
95
96    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
97        let post = posts[indexPath.row]
98
99        guard let cell = tableView.dequeueReusableCell(withIdentifier: PostPreviewTableViewCell.identifier, for: indexPath) as? PostPreviewTableViewCell else {
100            fatalError()
101        }
102        cell.configure(with: .init(title: post.title, imageUrl: post.headerImageUrl))
103        return cell
104    }
105
106    func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
107        return 180
108    }
109
110    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
111        tableView.deselectRow(at: indexPath, animated: true)
112
113        HapticsManager.shared.vibrateForSelection()
114
115        guard IAPManager.shared.canViewPost else {
116            let vc = PayWallviewController()
117            present(vc, animated: true, completion: nil)
118            return
119        }
120
121        let vc = ViewPostViewController(post: posts[indexPath.row])
122        vc.navigationItem.largeTitleDisplayStyle = .never
123        vc.title = "Post"
124        navigationController?.pushViewController(vc, animated: true)
125    }
126}
```



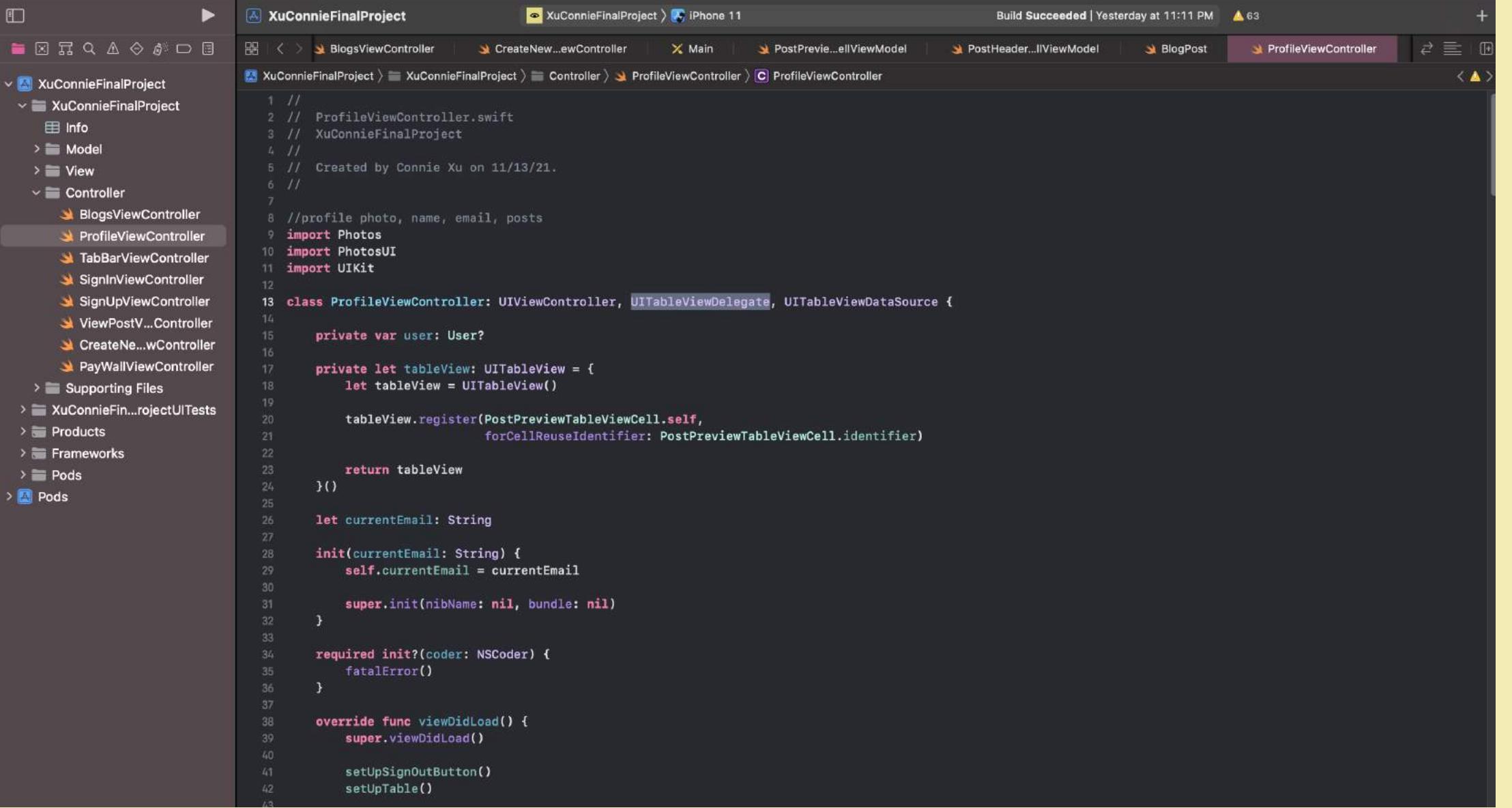
PROJECT SPECS

Delegation Design Pattern

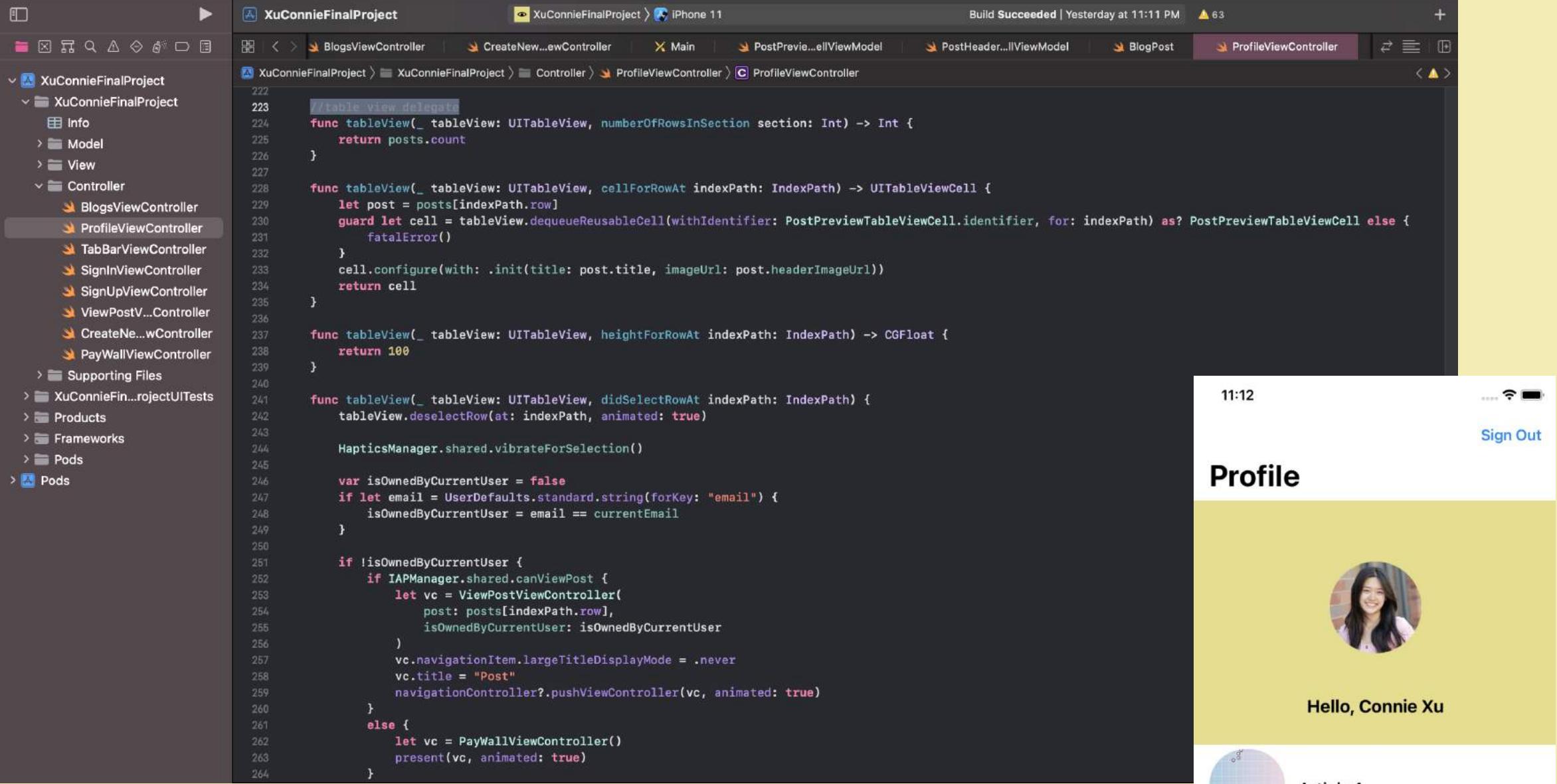
Table View

Delegate: UITableViewDelegate
Delegator: ProfileViewController

> XuConnieFinalProject > Controller > BlogsViewController.swift
> ProfileViewController.swift
> ViewPostViewController.swift



```
1 // ProfileViewController.swift
2 // XuConnieFinalProject
3 // Created by Connie Xu on 11/13/21.
4 //
5 // profile photo, name, email, posts
6 import Photos
7 import PhotosUI
8 import UIKit
9
10 class ProfileViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
11
12     private var user: User?
13
14     private let tableView: UITableView = {
15         let tableView = UITableView()
16
17         tableView.register(PostPreviewTableViewCell.self,
18                           forCellReuseIdentifier: PostPreviewTableViewCell.identifier)
19
20         return tableView
21     }()
22
23     let currentEmail: String
24
25     init(currentEmail: String) {
26         self.currentEmail = currentEmail
27
28         super.init(nibName: nil, bundle: nil)
29     }
30
31     required init?(coder: NSCoder) {
32         fatalError()
33     }
34
35     override func viewDidLoad() {
36         super.viewDidLoad()
37
38         setUpSignOutButton()
39
40         setUpTable()
41     }
42 }
```



```
1 // UITableViewDelegate
2 func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
3     return posts.count
4 }
5
6 func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
7     let post = posts[indexPath.row]
8     guard let cell = tableView.dequeueReusableCell(withIdentifier: PostPreviewTableViewCell.identifier, for: indexPath) as? PostPreviewTableViewCell else {
9         fatalError()
10    }
11    cell.configure(with: .init(title: post.title, imageUrl: post.headerImageUrl))
12    return cell
13 }
14
15 func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
16     return 100
17 }
18
19 func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
20     tableView.deselectRow(at: indexPath, animated: true)
21
22     HapticsManager.shared.vibrateForSelection()
23
24     var isOwnedByCurrentUser = false
25     if let email = UserDefaults.standard.string(forKey: "email") {
26         isOwnedByCurrentUser = email == currentUserEmail
27     }
28
29     if !isOwnedByCurrentUser {
30         if IAPManager.shared.canViewPost {
31             let vc = ViewPostViewController(
32                 post: posts[indexPath.row],
33                 isOwnedByCurrentUser: isOwnedByCurrentUser
34             )
35             vc.navigationItem.largeTitleDisplayStyle = .never
36             vc.title = "Post"
37             navigationController?.pushViewController(vc, animated: true)
38         } else {
39             let vc = PayWallViewController()
40             present(vc, animated: true)
41         }
42     }
43 }
```



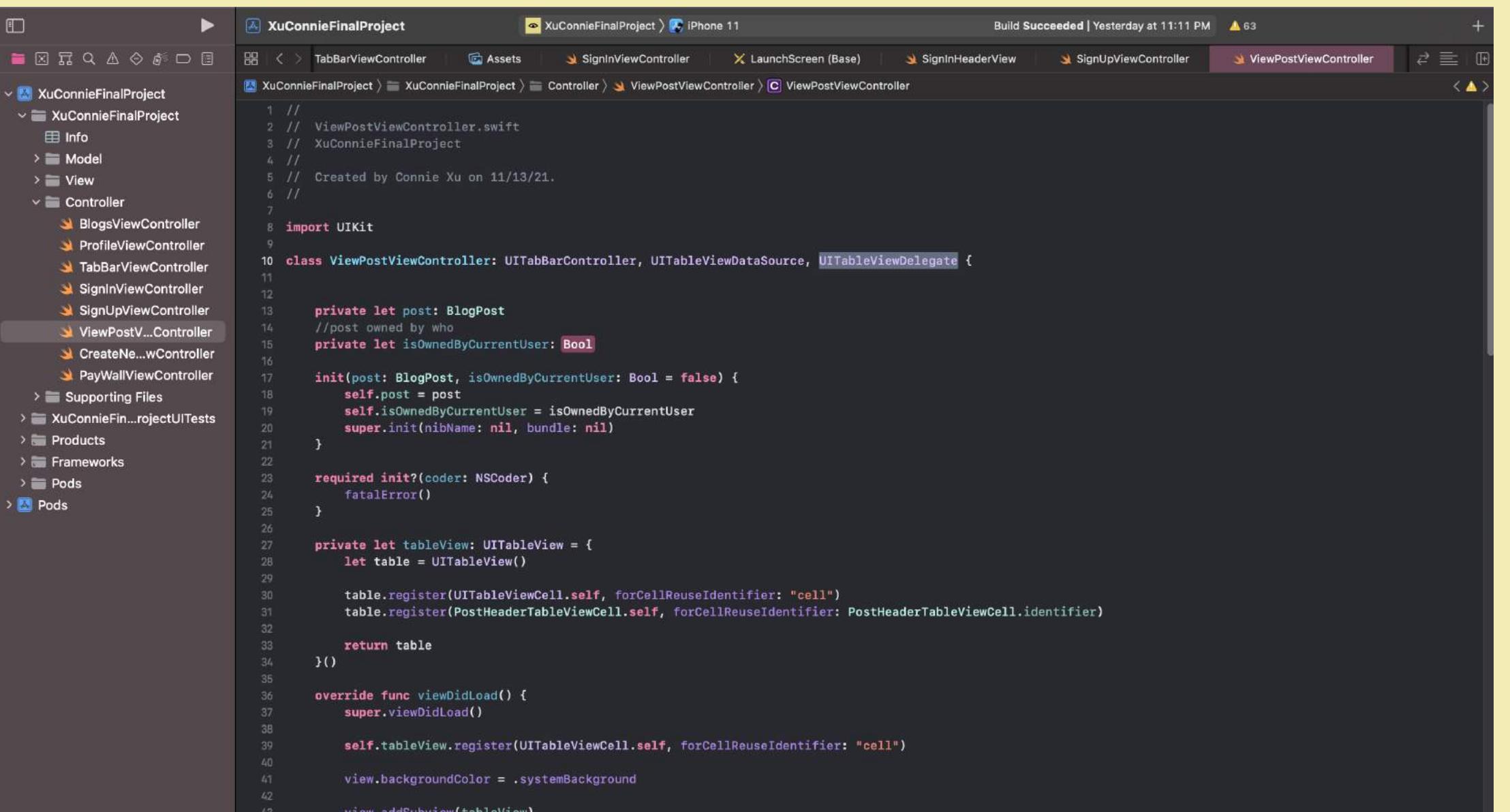
PROJECT SPECS

Delegation Design Pattern

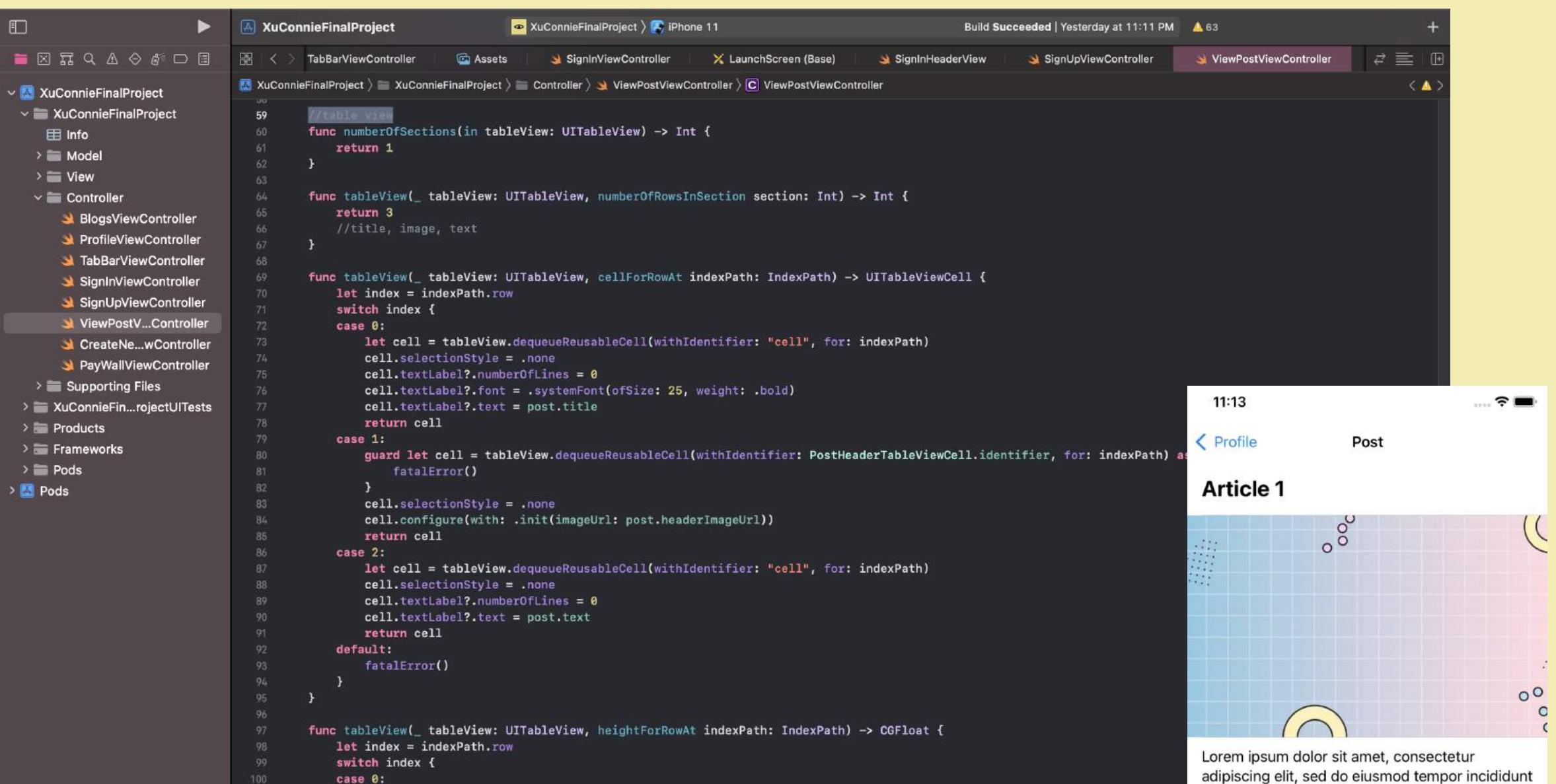
Table View

Delegate: UITableViewDelegate

Delegator: ViewPostViewController



```
1 // ViewPostViewController.swift
2 // XuConnieFinalProject
3 // Created by Connie Xu on 11/13/21.
4 //
5 import UIKit
6
7 class ViewPostViewController: UITabBarController, UITableViewDataSource, UITableViewDelegate {
8
9     private let post: BlogPost
10    private let isOwnedByCurrentUser: Bool
11
12    init(post: BlogPost, isOwnedByCurrentUser: Bool = false) {
13        self.post = post
14        self.isOwnedByCurrentUser = isOwnedByCurrentUser
15        super.init(nibName: nil, bundle: nil)
16    }
17
18    required init?(coder: NSCoder) {
19        fatalError()
20    }
21
22    private let tableView: UITableView = {
23        let table = UITableView()
24
25        table.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
26        table.register(PostHeaderTableViewCell.self, forCellReuseIdentifier: PostHeaderTableViewCell.Identifier)
27
28        return table
29    }()
30
31    override func viewDidLoad() {
32        super.viewDidLoad()
33
34        self.tableView.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
35
36        view.backgroundColor = .systemBackground
37
38        view.addSubview(tableView)
39    }
40
41    override func viewDidLayoutSubviews() {
42        super.viewDidLayoutSubviews()
43
44        tableView.frame = view.bounds
45    }
46
47    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
48        return 1
49    }
50
51    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
52        let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)
53
54        cell.selectionStyle = .none
55        cell.textLabel?.numberOfLines = 0
56        cell.textLabel?.font = .systemFont(ofSize: 25, weight: .bold)
57        cell.textLabel?.text = post.title
58
59        return cell
60    }
61
62    func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
63        let index = indexPath.row
64
65        switch index {
66            case 0:
67                let cell = tableView.dequeueReusableCell(withIdentifier: PostHeaderTableViewCell.Identifier, for: indexPath)
68
69                cell.selectionStyle = .none
70                cell.configure(with: .init(imageUrl: post.headerImageUrl))
71
72                return cell.bounds.height
73            default:
74                let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)
75
76                cell.selectionStyle = .none
77                cell.textLabel?.numberOfLines = 0
78                cell.textLabel?.font = .systemFont(ofSize: 18, weight: .normal)
79                cell.textLabel?.text = post.text
80
81                return cell.bounds.height
82        }
83    }
84
85    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
86        let index = indexPath.row
87
88        switch index {
89            case 0:
90                let storyboard = UIStoryboard(name: "Main", bundle: nil)
91                let controller = storyboard.instantiateViewController(withIdentifier: "PostDetailController") as! PostDetailController
92
93                controller.post = post
94
95                self.navigationController?.pushViewController(controller, animated: true)
96            default:
97                break
98        }
99    }
100}
```



```
1 // ViewPostViewController.swift
2 // XuConnieFinalProject
3 // Created by Connie Xu on 11/13/21.
4 //
5 import UIKit
6
7 class ViewPostViewController: UITabBarController, UITableViewDataSource, UITableViewDelegate {
8
9     private let post: BlogPost
10    private let isOwnedByCurrentUser: Bool
11
12    init(post: BlogPost, isOwnedByCurrentUser: Bool = false) {
13        self.post = post
14        self.isOwnedByCurrentUser = isOwnedByCurrentUser
15        super.init(nibName: nil, bundle: nil)
16    }
17
18    required init?(coder: NSCoder) {
19        fatalError()
20    }
21
22    private let tableView: UITableView = {
23        let table = UITableView()
24
25        table.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
26        table.register(PostHeaderTableViewCell.self, forCellReuseIdentifier: PostHeaderTableViewCell.Identifier)
27
28        return table
29    }()
30
31    override func viewDidLoad() {
32        super.viewDidLoad()
33
34        self.tableView.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
35
36        view.backgroundColor = .systemBackground
37
38        view.addSubview(tableView)
39    }
40
41    override func viewDidLayoutSubviews() {
42        super.viewDidLayoutSubviews()
43
44        tableView.frame = view.bounds
45    }
46
47    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
48        return 1
49    }
50
51    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
52        let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)
53
54        cell.selectionStyle = .none
55        cell.textLabel?.numberOfLines = 0
56        cell.textLabel?.font = .systemFont(ofSize: 25, weight: .bold)
57        cell.textLabel?.text = post.title
58
59        return cell
60    }
61
62    func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
63        let index = indexPath.row
64
65        switch index {
66            case 0:
67                let cell = tableView.dequeueReusableCell(withIdentifier: PostHeaderTableViewCell.Identifier, for: indexPath)
68
69                cell.selectionStyle = .none
70                cell.configure(with: .init(imageUrl: post.headerImageUrl))
71
72                return cell.bounds.height
73            default:
74                let cell = tableView.dequeueReusableCell(withIdentifier: "cell", for: indexPath)
75
76                cell.selectionStyle = .none
77                cell.textLabel?.numberOfLines = 0
78                cell.textLabel?.font = .systemFont(ofSize: 18, weight: .normal)
79                cell.textLabel?.text = post.text
80
81                return cell.bounds.height
82        }
83    }
84
85    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
86        let index = indexPath.row
87
88        switch index {
89            case 0:
90                let storyboard = UIStoryboard(name: "Main", bundle: nil)
91                let controller = storyboard.instantiateViewController(withIdentifier: "PostDetailController") as! PostDetailController
92
93                controller.post = post
94
95                self.navigationController?.pushViewController(controller, animated: true)
96            default:
97                break
98        }
99    }
100}
```

> XuConnieFinalProject > Controller > BlogsViewController.swift
> ProfileViewController.swift
> ViewPostViewController.swift

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Porta non pulvinar neque laoreet. Ipsa a arcu cursus vitae congue. Nunc lobortis mattis aliquam faucibus purus in massa tempor. Eget arcu dictum varius duis. Magna fermentum iaculis eu non diam phasellus vestibulum. Adipiscing elit at aliquam purus sit amet luctus venenatis. Mi bibendum neque egestas congue quisque egestas diam in. Nisi vel pretium lectus quam id leo in vitae.

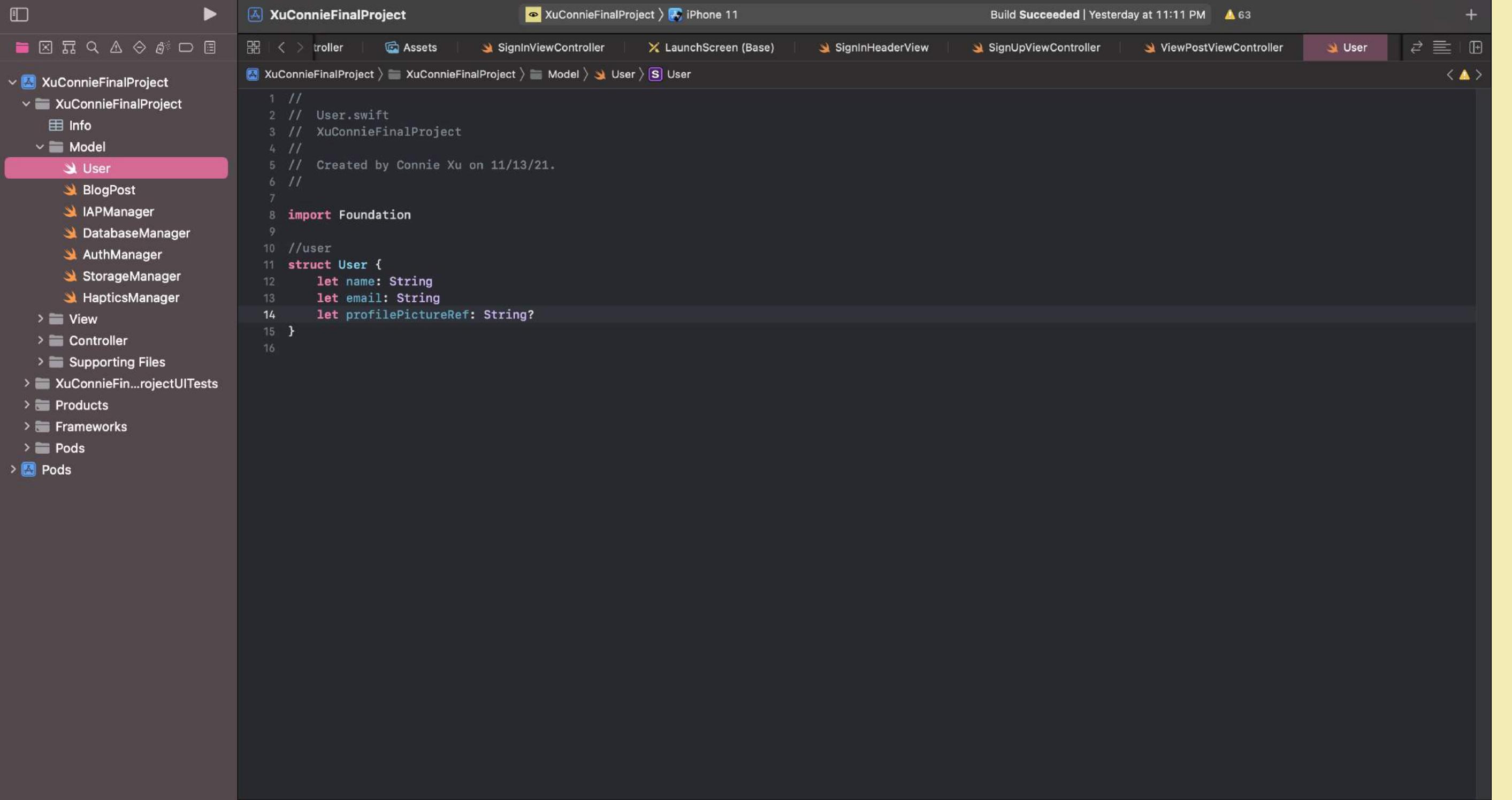
PROJECT SPECS

MVC Design Pattern

Model

User

- Requirements for user profile



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar displays the project structure, including 'Info', 'Model' (which is expanded to show 'User', 'BlogPost', 'IAPManager', 'DatabaseManager', 'AuthManager', 'StorageManager', and 'HapticsManager'), 'View', 'Controller', 'Supporting Files', and 'Pods'. The right pane shows the code editor with the 'User.swift' file selected. The code defines a struct 'User' with properties: name (String), email (String), and profilePictureRef (String?). The code is as follows:

```
1 //  
2 // User.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/13/21.  
6 //  
7  
8 import Foundation  
9  
10 //user  
11 struct User {  
12     let name: String  
13     let email: String  
14     let profilePictureRef: String?  
15 }
```

> XuConnieFinalProject > Model > User.swift
> BlogPost.swift

> IAPManager.swift
> DatabaseManager.swift
> AuthManager.swift
> StorageManager.swift
> HapticsManager.swift

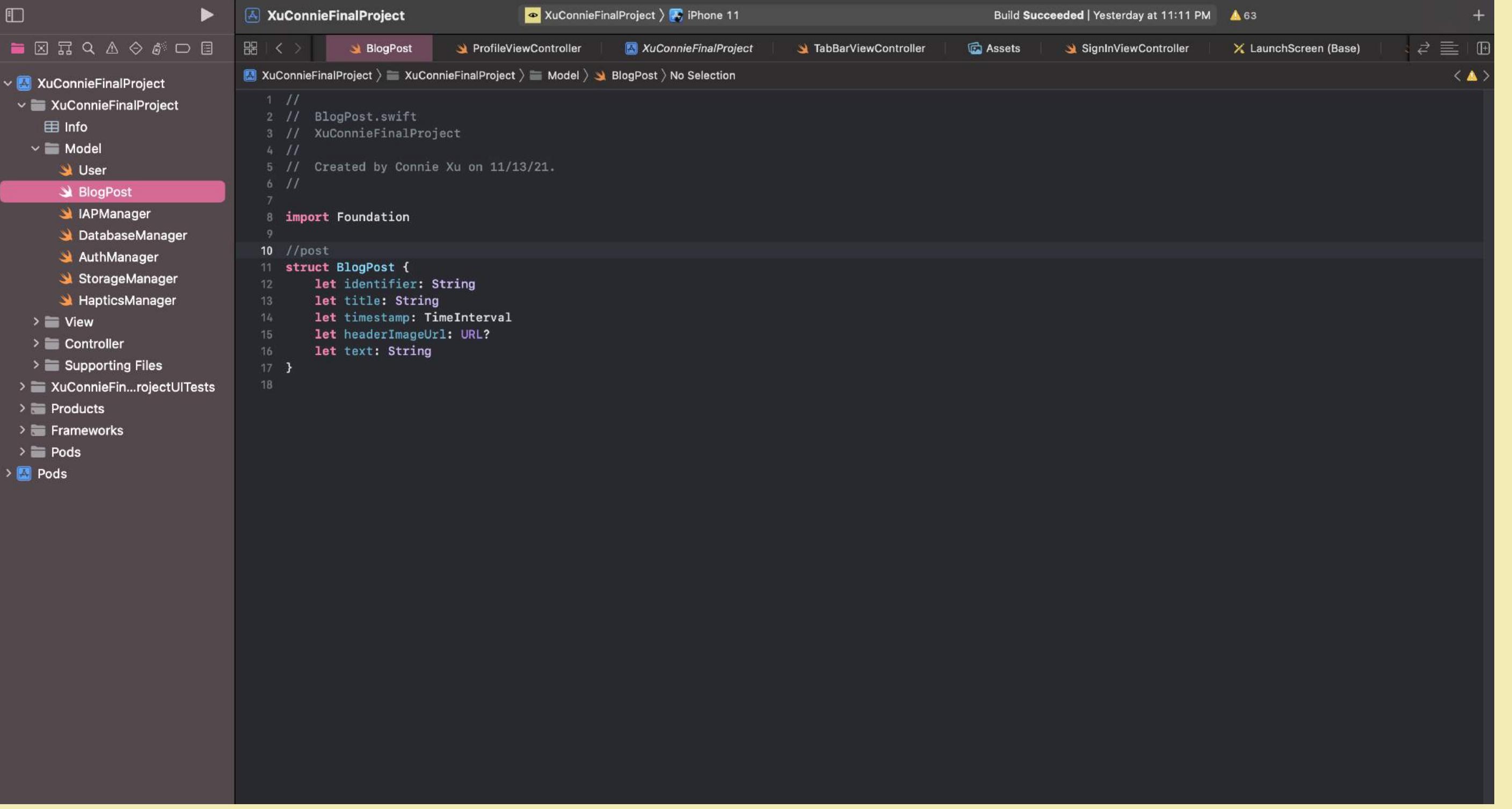
PROJECT SPECS

MVC Design Pattern

Model

BlogPost

- Requirements for blog post



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar displays the project structure:

- XuConnieFinalProject (selected)
- XuConnieFinalProject (Info)
- XuConnieFinalProject (Model)
- User
- BlogPost (highlighted)
- IAPManager
- DatabaseManager
- AuthManager
- StorageManager
- HapticsManager
- View
- Controller
- Supporting Files
- XuConnieFinalProjectUITests
- Products
- Frameworks
- Pods
- Pods

The right pane shows the content of the 'BlogPost.swift' file:

```
1 //  
2 // BlogPost.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/13/21.  
6 //  
7  
8 import Foundation  
9  
10 //post  
11 struct BlogPost {  
12     let identifier: String  
13     let title: String  
14     let timestamp: TimeInterval  
15     let headerImageUrl: URL?  
16     let text: String  
17 }  
18
```

> XuConnieFinalProject > Model > User.swift
> BlogPost.swift

> IAPManager.swift
> DatabaseManager.swift
> AuthManager.swift
> StorageManager.swift
> HapticsManager.swift

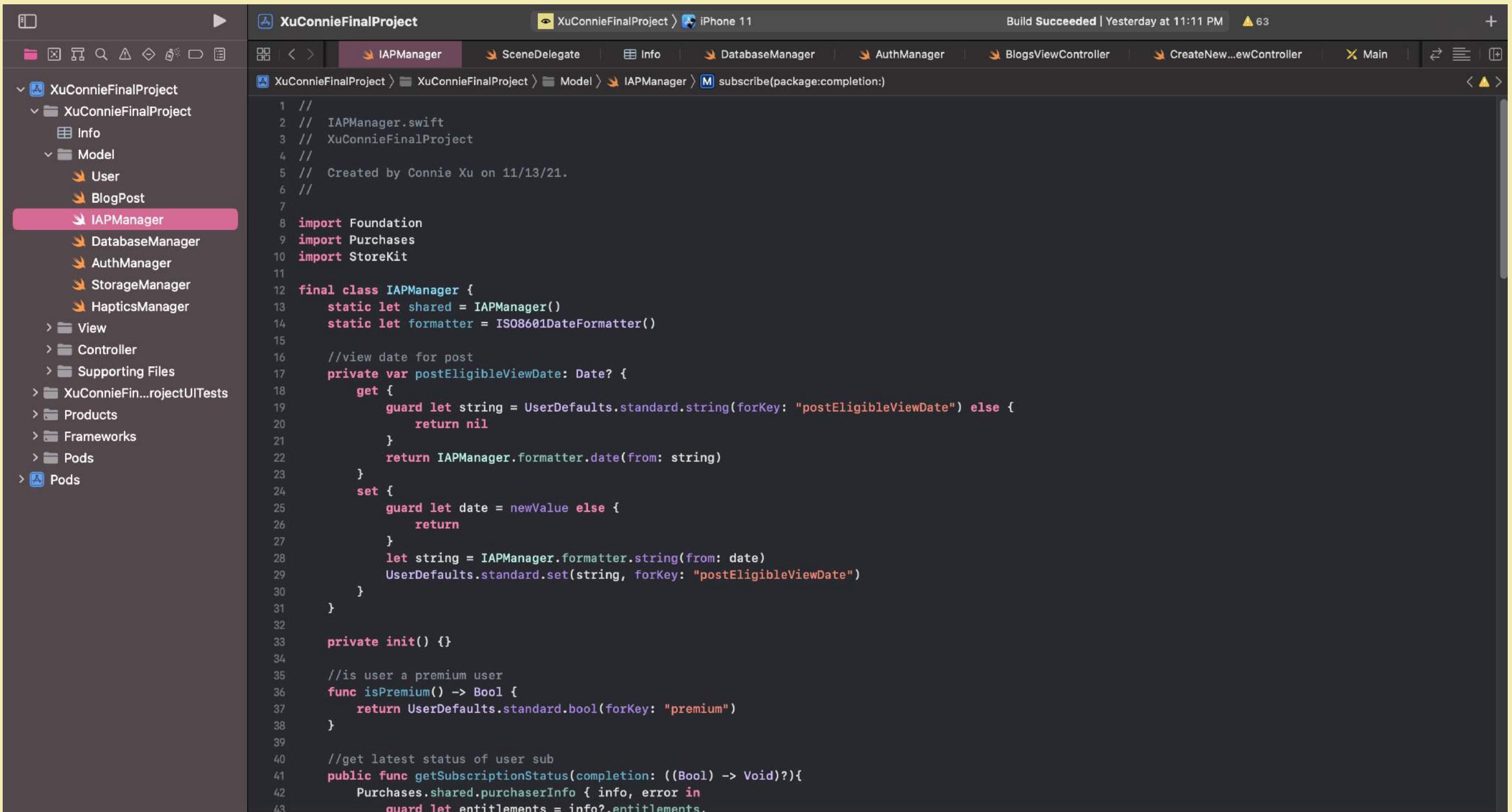
PROJECT SPECS

MVC Design Pattern

Model - Manager

IAPManager

- Manager for in-app purchases
- In-app purchase is subscription to view unlimited blog posts



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar displays the project structure, including 'Info', 'Model' (which contains 'User', 'BlogPost', and 'IAPManager'), 'View', 'Controller', 'Supporting Files', 'XuConnieFinalProjectUITests', 'Products', 'Frameworks', and 'Pods'. The right pane shows the 'IAPManager.swift' file. The code defines a final class 'IAPManager' with static properties for shared instance and formatter, and methods for getting and setting post eligible view date, checking if user is premium, and getting subscription status.

```
1 //  
2 //  IAPManager.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/13/21.  
6 //  
7  
8 import Foundation  
9 import Purchases  
10 import StoreKit  
11  
12 final class IAPManager {  
13     static let shared = IAPManager()  
14     static let formatter = ISO8601DateFormatter()  
15  
16     //view date for post  
17     private var postEligibleViewDate: Date? {  
18         get {  
19             guard let string = UserDefaults.standard.string(forKey: "postEligibleViewDate") else {  
20                 return nil  
21             }  
22             return IAPManager.formatter.date(from: string)  
23         }  
24         set {  
25             guard let date = newValue else {  
26                 return  
27             }  
28             let string = IAPManager.formatter.string(from: date)  
29             UserDefaults.standard.set(string, forKey: "postEligibleViewDate")  
30         }  
31     }  
32  
33     private init() {}  
34  
35     //is user a premium user  
36     func isPremium() -> Bool {  
37         return UserDefaults.standard.bool(forKey: "premium")  
38     }  
39  
40     //get latest status of user sub  
41     public func getSubscriptionStatus(completion: ((Bool) -> Void)?){  
42         Purchases.shared.purchaserInfo { info, error in  
43             guard let entitlements = info?.entitlements.  
44         }  
45     }  
46 }
```

> XuConnieFinalProject > Model > User.swift

> BlogPost.swift

> IAPManager.swift

> DatabaseManager.swift

> AuthManager.swift

> StorageManager.swift

> HapticsManager.swift

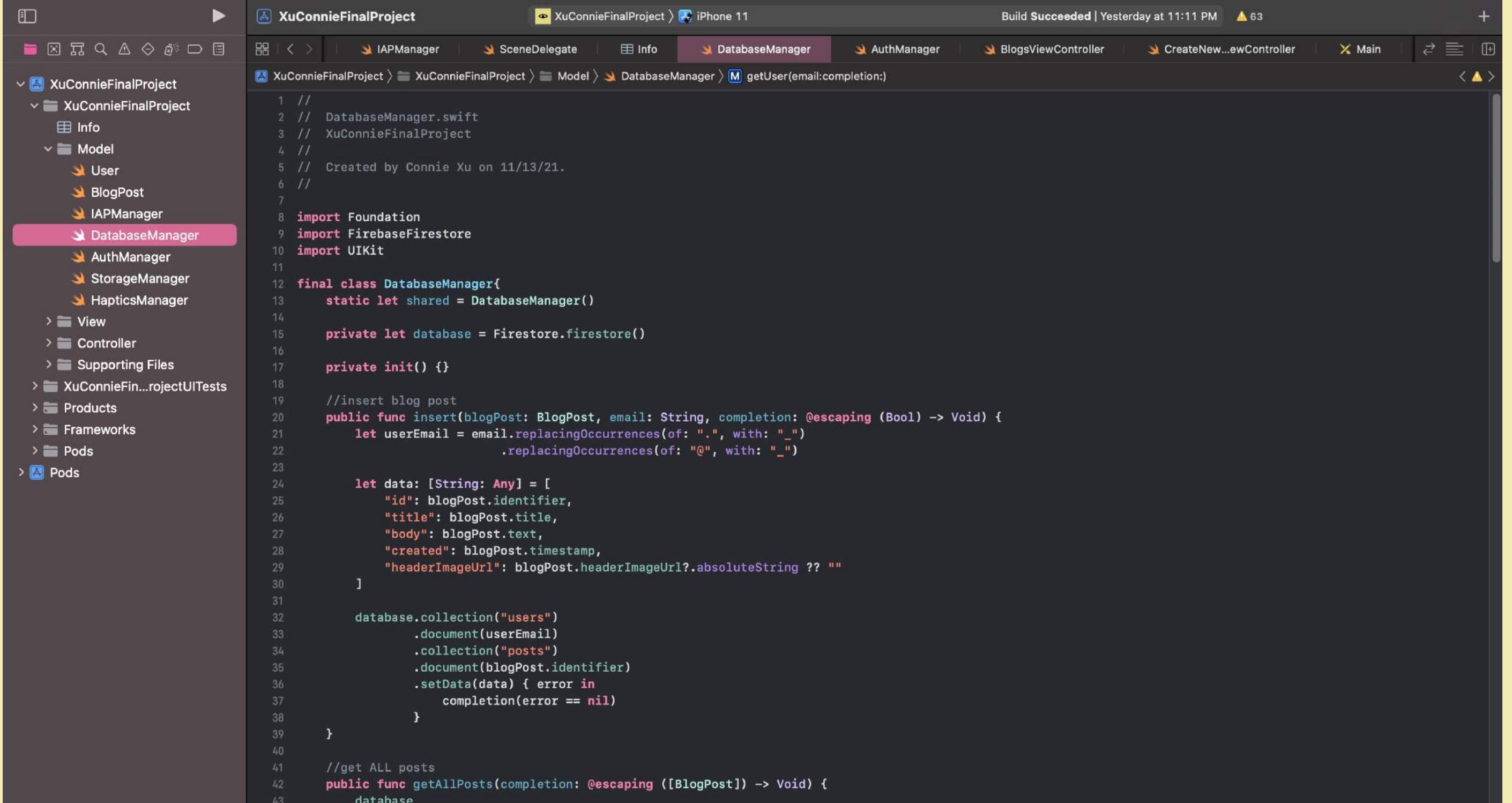
PROJECT SPECS

MVC Design Pattern

Model - Manager

DatabaseManager

- Manager for database
- Database content includes users and blogs



The screenshot shows the Xcode interface with the project navigation bar at the top. Below it is a tree view of the project structure under 'XuConnieFinalProject'. A pink box highlights the 'Model' folder, which contains 'User', 'BlogPost', 'IAPManager', 'DatabaseManager' (the current file being viewed), 'AuthManager', 'StorageManager', and 'HapticsManager'. The 'View', 'Controller', 'Supporting Files', and 'Pods' folders are also listed. The main editor area displays the 'DatabaseManager.swift' file. The code is written in Swift and interacts with Firestore. It includes imports for Foundation, FirebaseFirestore, and UIKit. The 'final class DatabaseManager' has a static let 'shared' property. It contains methods for inserting blog posts and getting all posts. The code uses string replacement and Firestore's collection and document methods.

```
1 // DatabaseManager.swift
2 // XuConnieFinalProject
3 // Created by Connie Xu on 11/13/21.
4 //
5
6 import Foundation
7 import FirebaseFirestore
8 import UIKit
9
10 final class DatabaseManager{
11     static let shared = DatabaseManager()
12
13     private let database = Firestore.firestore()
14
15     private init() {}
16
17     //insert blog post
18     public func insert(blogPost: BlogPost, email: String, completion: @escaping (Bool) -> Void) {
19         let userEmail = email.replacingOccurrences(of: ".", with: "_")
20             .replacingOccurrences(of: "@", with: "_")
21
22         let data: [String: Any] = [
23             "id": blogPost.identifier,
24             "title": blogPost.title,
25             "body": blogPost.text,
26             "created": blogPost.timestamp,
27             "headerImageUrl": blogPost.headerImageUrl?.absoluteString ?? ""
28         ]
29
30         database.collection("users")
31             .document(userEmail)
32                 .collection("posts")
33                     .document(blogPost.identifier)
34                         .setData(data) { error in
35                             completion(error == nil)
36                         }
37
38     }
39
40     //get ALL posts
41     public func getAllPosts(completion: @escaping ([BlogPost]) -> Void) {
42         database
43     }
}
```

> XuConnieFinalProject > Model > User.swift

> BlogPost.swift

> IAPManager.swift

> DatabaseManager.swift

> AuthManager.swift

> StorageManager.swift

> HapticsManager.swift

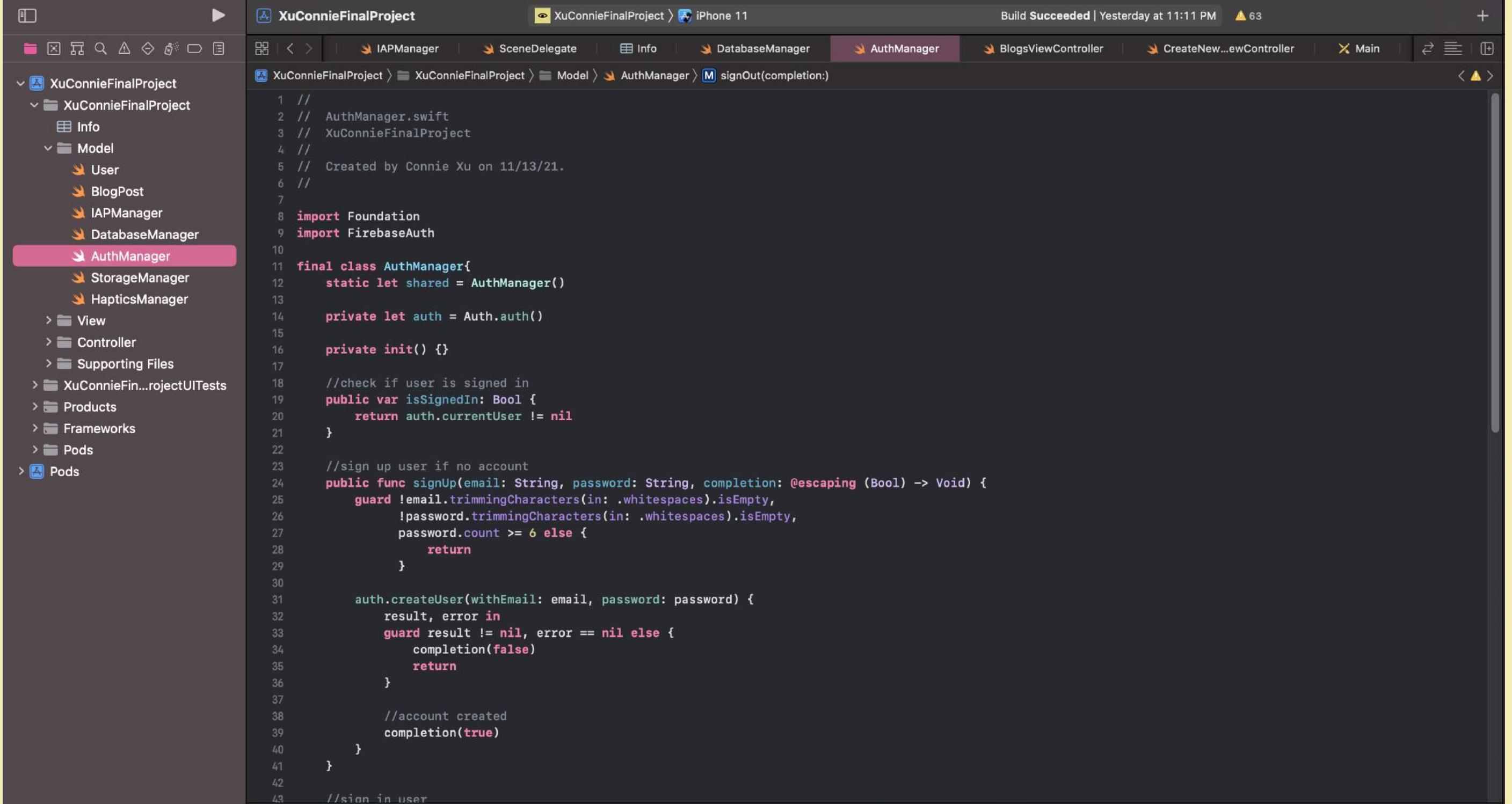
PROJECT SPECS

MVC Design Pattern

Model - Manager

AuthManager

- Manager for authorization
- Sign in and sign up



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar displays the project structure, including 'Info', 'Model' (which contains 'User', 'BlogPost', 'IAPManager', 'DatabaseManager', and 'AuthManager'), 'View', 'Controller', 'Supporting Files', 'XuConnieFinalProjectUITests', 'Products', 'Frameworks', and 'Pods'. The right pane shows the 'AuthManager.swift' file under 'Model'. The code implements a static shared instance of the AuthManager class, which handles user authentication. It includes methods for checking if a user is signed in, signing up a new user, and signing in an existing user.

```
1 //  
2 //  AuthManager.swift  
3 //  XuConnieFinalProject  
4 //  
5 //  Created by Connie Xu on 11/13/21.  
6 //  
7  
8 import Foundation  
9 import FirebaseAuth  
10  
11 final class AuthManager{  
12     static let shared = AuthManager()  
13  
14     private let auth = Auth.auth()  
15  
16     private init() {}  
17  
18     //check if user is signed in  
19     public var isSignedIn: Bool {  
20         return auth.currentUser != nil  
21     }  
22  
23     //sign up user if no account  
24     public func signUp(email: String, password: String, completion: @escaping (Bool) -> Void) {  
25         guard !email.trimmingCharacters(in: .whitespaces).isEmpty,  
26             !password.trimmingCharacters(in: .whitespaces).isEmpty,  
27             password.count >= 6 else {  
28             return  
29         }  
30  
31         auth.createUser(withEmail: email, password: password) {  
32             result, error in  
33             guard result != nil, error == nil else {  
34                 completion(false)  
35                 return  
36             }  
37  
38             //account created  
39             completion(true)  
40         }  
41     }  
42  
43     //sign in user
```

> XuConnieFinalProject > Model > User.swift
> BlogPost.swift

> IAPManager.swift
> DatabaseManager.swift
> AuthManager.swift
> StorageManager.swift
> HapticsManager.swift

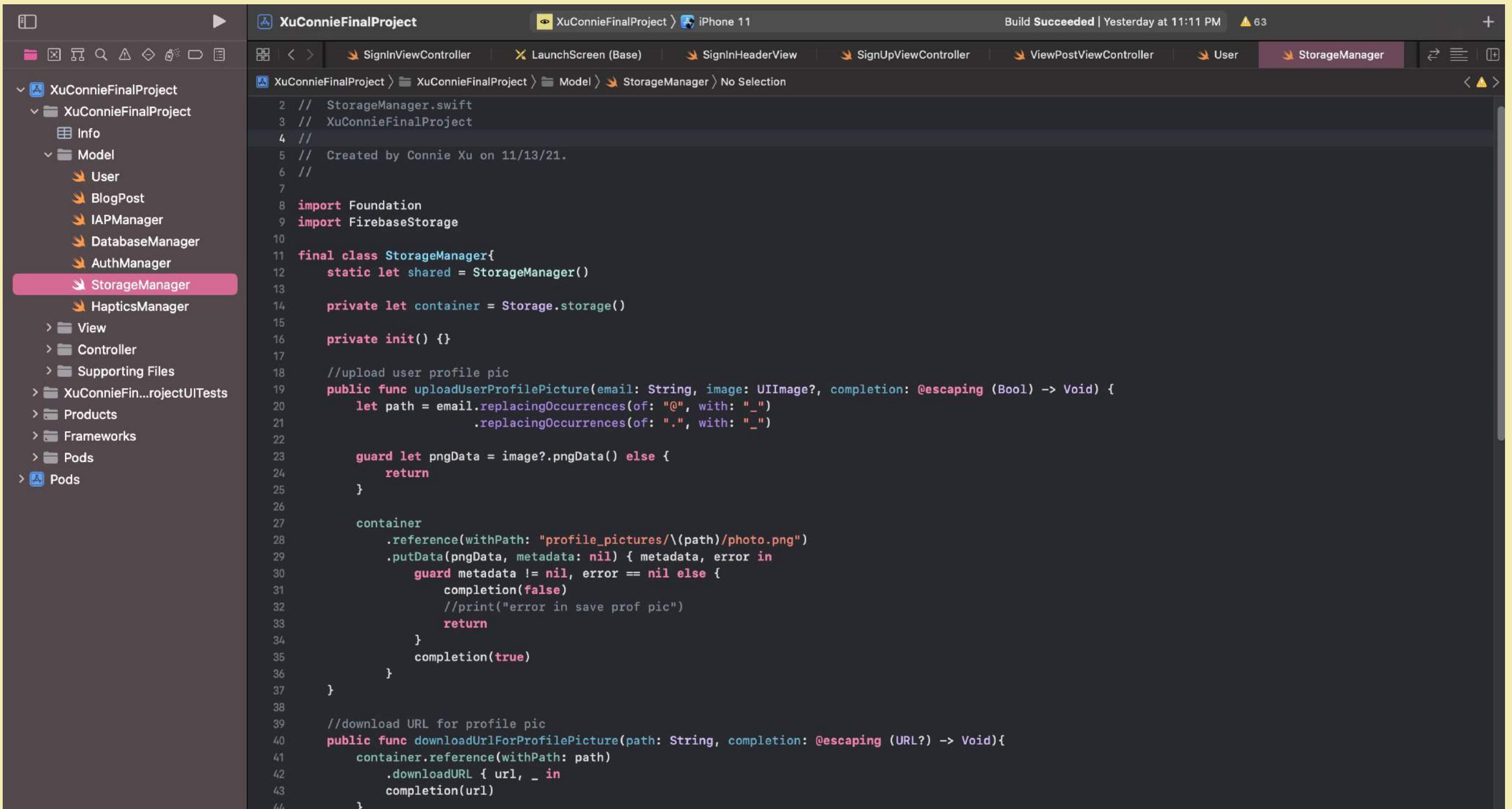
PROJECT SPECS

MVC Design Pattern

Model - Manager

StorageManager

- Manager for storage
- Images for profile picture and blog hero images



The screenshot shows the Xcode interface with the project structure on the left and the code editor on the right.

Project Structure:

- XuConnieFinalProject
- ↳ Info
- ↳ Model
 - ↳ User
 - ↳ BlogPost
 - ↳ IAPManager
 - ↳ DatabaseManager
 - ↳ AuthManager
 - ↳ StorageManager
 - ↳ HapticsManager
- ↳ View
- ↳ Controller
- ↳ Supporting Files
- ↳ XuConnieFinalProjectUITests
- ↳ Products
- ↳ Frameworks
- ↳ Pods
- ↳ A Pods folder

Code Editor (StorageManager.swift):

```
2 // StorageManager.swift
3 // XuConnieFinalProject
4 //
5 // Created by Connie Xu on 11/13/21.
6 //
7
8 import Foundation
9 import FirebaseStorage
10
11 final class StorageManager{
12     static let shared = StorageManager()
13
14     private let container = Storage.storage()
15
16     private init(){}
17
18     //upload user profile pic
19     public func uploadUserProfilePicture(email: String, image: UIImage?, completion: @escaping (Bool) -> Void) {
20         let path = email.replacingOccurrences(of: "@", with: "_")
21             .replacingOccurrences(of: ".", with: "_")
22
23         guard let pngData = image?.pngData() else {
24             return
25         }
26
27         container
28             .reference(withPath: "profile_pictures/\(path)/photo.png")
29             .putData(pngData, metadata: nil) { metadata, error in
30                 guard metadata != nil, error == nil else {
31                     completion(false)
32                     //print("error in save prof pic")
33                     return
34                 }
35                 completion(true)
36             }
37
38
39     //download URL for profile pic
40     public func downloadUrlForProfilePicture(path: String, completion: @escaping (URL?) -> Void){
41         container.reference(withPath: path)
42             .downloadURL { url, _ in
43                 completion(url)
44             }
45     }
46 }
```

> XuConnieFinalProject > Model > User.swift
> BlogPost.swift

> IAPManager.swift
> DatabaseManager.swift
> AuthManager.swift
> StorageManager.swift
> HapticsManager.swift

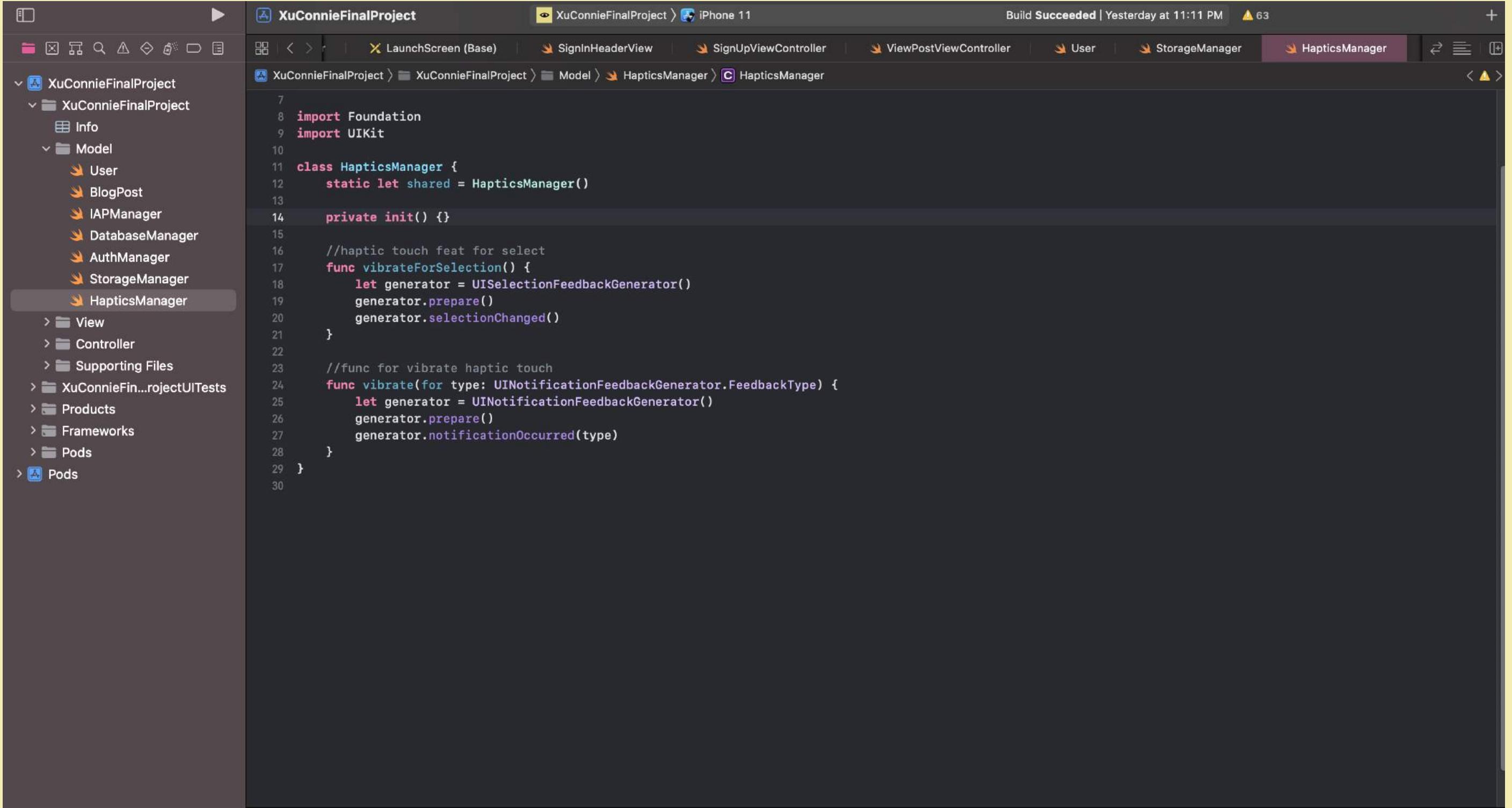
PROJECT SPECS

MVC Design Pattern

Model - Manager

HapticsManager

- Manager for haptic touch
- Vibrate when notification pops up



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar displays the project structure, including the 'Model' folder which contains 'User', 'BlogPost', 'IAPManager', 'DatabaseManager', 'AuthManager', 'StorageManager', and 'HapticsManager'. The right pane shows the code editor with the 'HapticsManager.swift' file selected. The code defines a class 'HapticsManager' with static methods for vibration.

```
7 import Foundation
8 import UIKit
9
10 class HapticsManager {
11     static let shared = HapticsManager()
12
13     private init() {}
14
15     //haptic touch feat for select
16     func vibrateForSelection() {
17         let generator = UISelectionFeedbackGenerator()
18         generator.prepare()
19         generator.selectionChanged()
20     }
21
22     //func for vibrate haptic touch
23     func vibrate(for type: UINotificationFeedbackGenerator.FeedbackType) {
24         let generator = UINotificationFeedbackGenerator()
25         generator.prepare()
26         generator.notificationOccurred(type)
27     }
28
29 }
```

> XuConnieFinalProject > Model > User.swift
> BlogPost.swift

> IAPManager.swift
> DatabaseManager.swift
> AuthManager.swift
> StorageManager.swift
> HapticsManager.swift

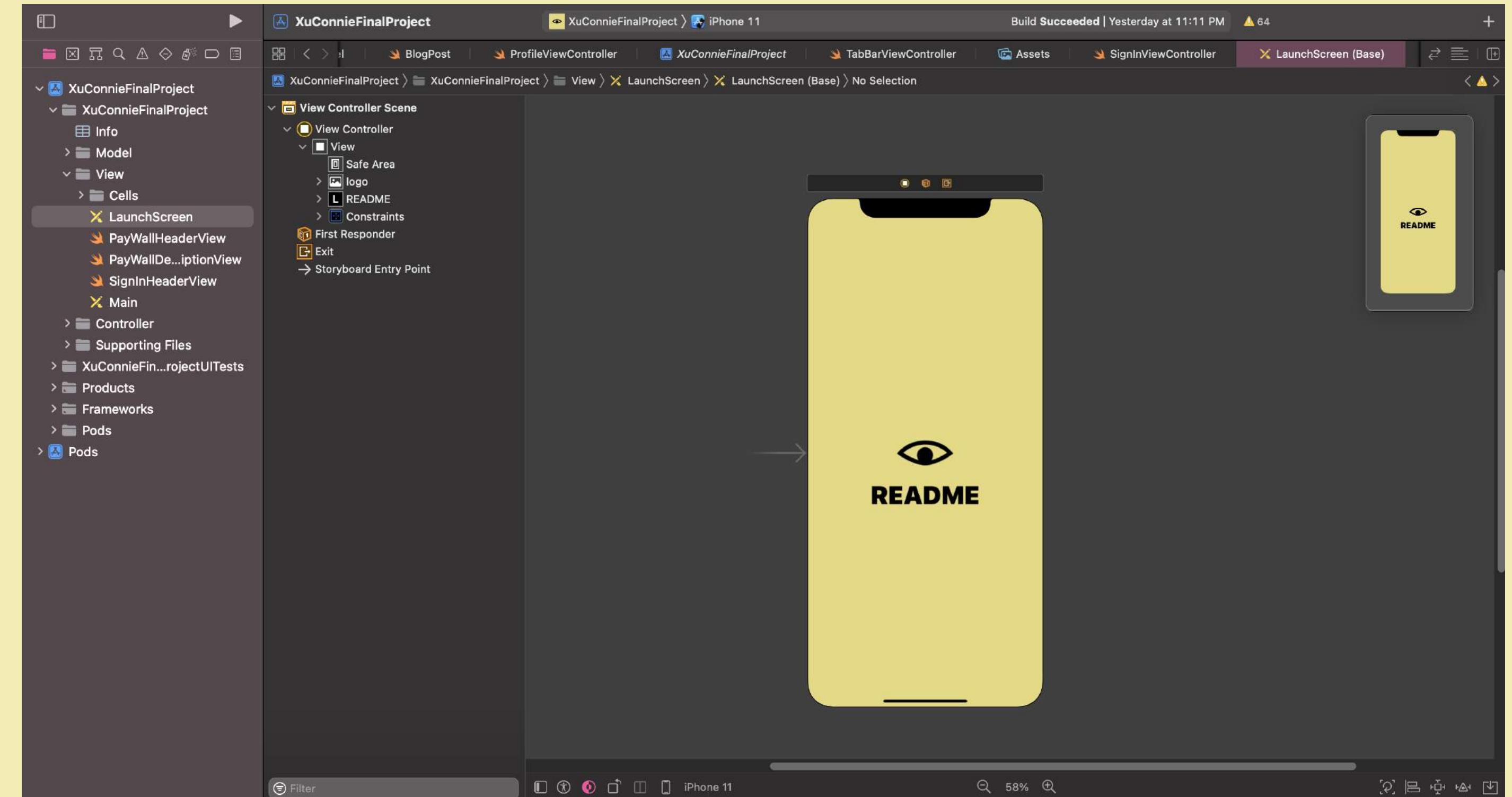
PROJECT SPECS

MVC Design Pattern

View - Storyboard

LaunchScreen

- Splash screen



> XuConnieFinalProject > View > LaunchScreen.storyboard

> PayWallHeaderView.swift
> PayWallDescriptionView.swift
> SignInHeaderView.swift

> Cells > PostPreviewTableViewCellViewModel.swift
> Cells > PostHeaderTableViewCellViewModel.swift

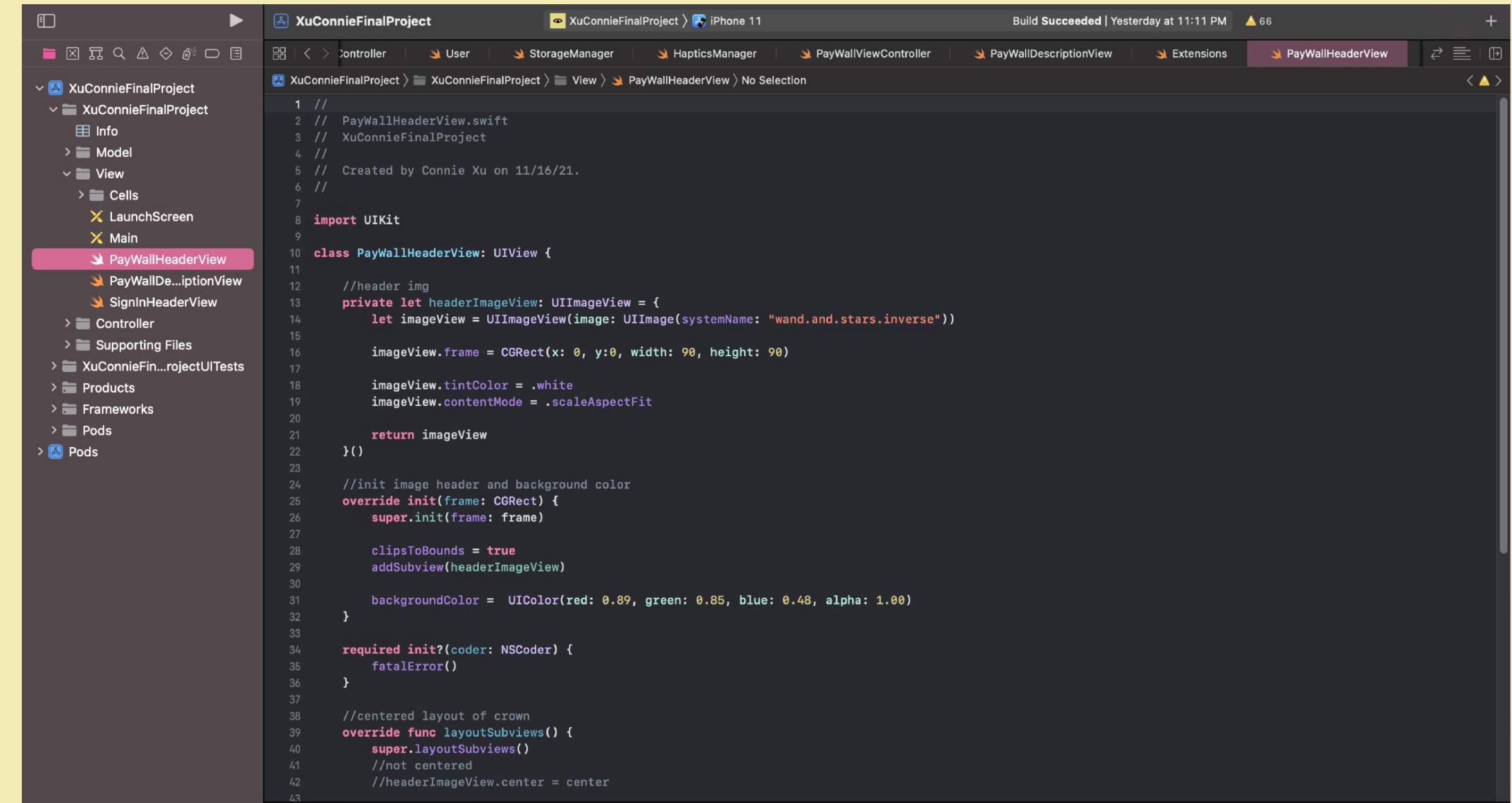
PROJECT SPECS

MVC Design Pattern

View

PayWallHeaderView

- Hero image of pay wall



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The file 'PayWallHeaderView.swift' is selected in the list of files. The code in the editor is as follows:

```
1 //  
2 // PayWallHeaderView.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/16/21.  
6 //  
7  
8 import UIKit  
9  
10 class PayWallHeaderView: UIView {  
11  
12     //header img  
13     private let headerImageView: UIImageView = {  
14         let imageView = UIImageView(image: UIImage(systemName: "wand.and.stars.inverse"))  
15  
16         imageView.frame = CGRect(x: 0, y:0, width: 90, height: 90)  
17  
18         imageView.tintColor = .white  
19         imageView.contentMode = .scaleAspectFit  
20  
21         return imageView  
22     }()  
23  
24     //init image header and background color  
25     override init(frame: CGRect) {  
26         super.init(frame: frame)  
27  
28         clipsToBounds = true  
29         addSubview(headerImageView)  
30  
31         backgroundColor = UIColor(red: 0.89, green: 0.85, blue: 0.48, alpha: 1.00)  
32     }  
33  
34     required init?(coder: NSCoder) {  
35         fatalError()  
36     }  
37  
38     //centered layout of crown  
39     override func layoutSubviews() {  
40         super.layoutSubviews()  
41         //not centered  
42         //headerImageView.center = center  
43 }
```

> XuConnieFinalProject > View > LaunchScreen.storyboard

> PayWallHeaderView.swift
> PayWallDescriptionView.swift
> SignInHeaderView.swift

> Cells > PostPreviewTableViewCellViewModel.swift
> Cells > PostHeaderTableViewCellViewModel.swift

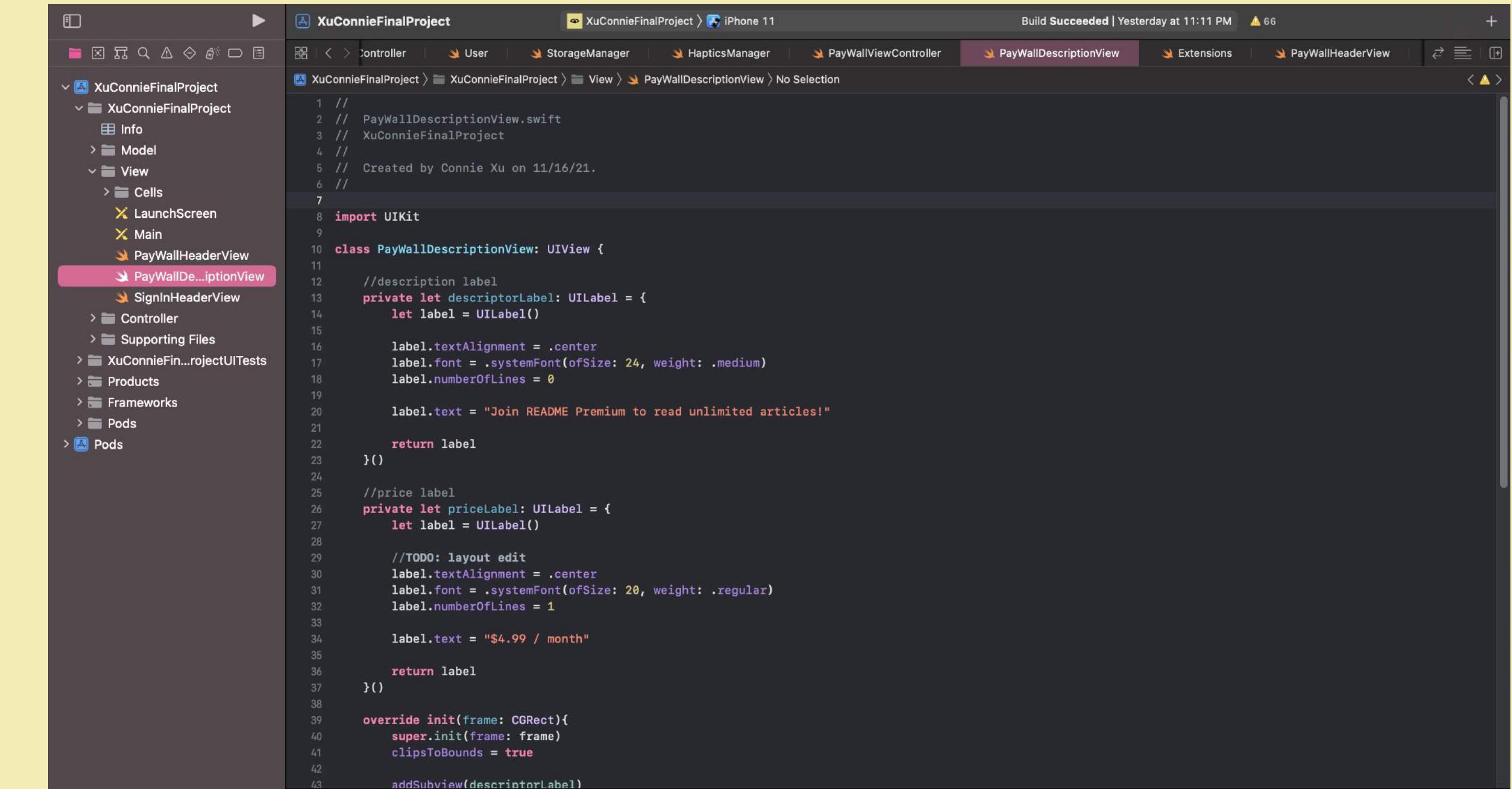
PROJECT SPECS

MVC Design Pattern

View

PayWallDescriptionView

- Content of pay wall



The screenshot shows the Xcode interface with the project navigation bar at the top. Below it is a file browser showing the project structure. A pink box highlights the 'PayWallDescriptionView' file in the 'View' folder. The main area of the screen displays the code for the PayWallDescriptionView class.

```
1 //  
2 // PayWallDescriptionView.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/16/21.  
6 //  
7  
8 import UIKit  
9  
10 class PayWallDescriptionView: UIView {  
11  
    //description label  
12    private let descriptorLabel: UILabel = {  
13        let label = UILabel()  
14  
15        label.textAlignment = .center  
16        label.font = .systemFont(ofSize: 24, weight: .medium)  
17        label.numberOfLines = 0  
18  
19        label.text = "Join README Premium to read unlimited articles!"  
20  
21        return label  
22    }()  
23  
    //price label  
24    private let priceLabel: UILabel = {  
25        let label = UILabel()  
26  
27        //TODO: layout edit  
28        label.textAlignment = .center  
29        label.font = .systemFont(ofSize: 20, weight: .regular)  
30        label.numberOfLines = 1  
31  
32        label.text = "$4.99 / month"  
33  
34        return label  
35    }()  
36  
37    override init(frame: CGRect){  
38        super.init(frame: frame)  
39        clipsToBounds = true  
40  
41        addSubview(descriptorLabel)  
42    }  
43}
```

> XuConnieFinalProject > View > LaunchScreen.storyboard

> PayWallHeaderView.swift
> PayWallDescriptionView.swift
> SignInHeaderView.swift

> Cells > PostPreviewTableViewCellViewModel.swift
> Cells > PostHeaderTableViewCellViewModel.swift

PROJECT SPECS

MVC Design Pattern

View

SignInHeaderView

- Header of sign in page

The screenshot shows the Xcode interface with the project structure on the left and the code editor on the right. The code editor displays the `SignInHeaderView.swift` file, which defines a view containing an image and a label. The preview window on the right shows the 'Sign In' screen with a logo, the text 'README', an email input field ('conniexu@gmail.com'), a password input field ('.....'), and a 'Sign In' button.

```
1 //  
2 // SignInHeaderView.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/29/21.  
6 //  
7 |  
8 import UIKit  
9  
10 //sign in to user  
11 class SignInHeaderView: UIView {  
12  
13     private let imageView: UIImageView = {  
14         let imageView = UIImageView(image: UIImage(named: "logo"))  
15  
16         imageView.contentMode = .scaleAspectFit  
17  
18         imageView.backgroundColor = UIColor(red: 0.89, green: 0.85, blue: 0.48, alpha: 1.00)  
19  
20         return imageView  
21     }  
22  
23     private let label: UILabel = {  
24         let label = UILabel()  
25  
26         label.textAlignment = .center  
27         label.numberOfLines = 0  
28         label.font = .systemFont(ofSize: 30, weight: .black)  
29         label.text = "README"  
30  
31         return label  
32     }  
33  
34     override init(frame: CGRect) {  
35         super.init(frame: frame)  
36         clipsToBounds = true  
37  
38         addSubview(label)  
39         addSubview(imageView)  
40     }  
41  
42     required init?(coder: NSCoder) {  
43         fatalError("init(coder:) has not been implemented")  
44     }  
45 }
```

> XuConnieFinalProject > View > LaunchScreen.storyboard

> PayWallHeaderView.swift
> PayWallDescriptionView.swift
> SignInHeaderView.swift

> Cells > PostPreviewTableViewCellViewModel.swift
> Cells > PostHeaderTableViewCellViewModel.swift

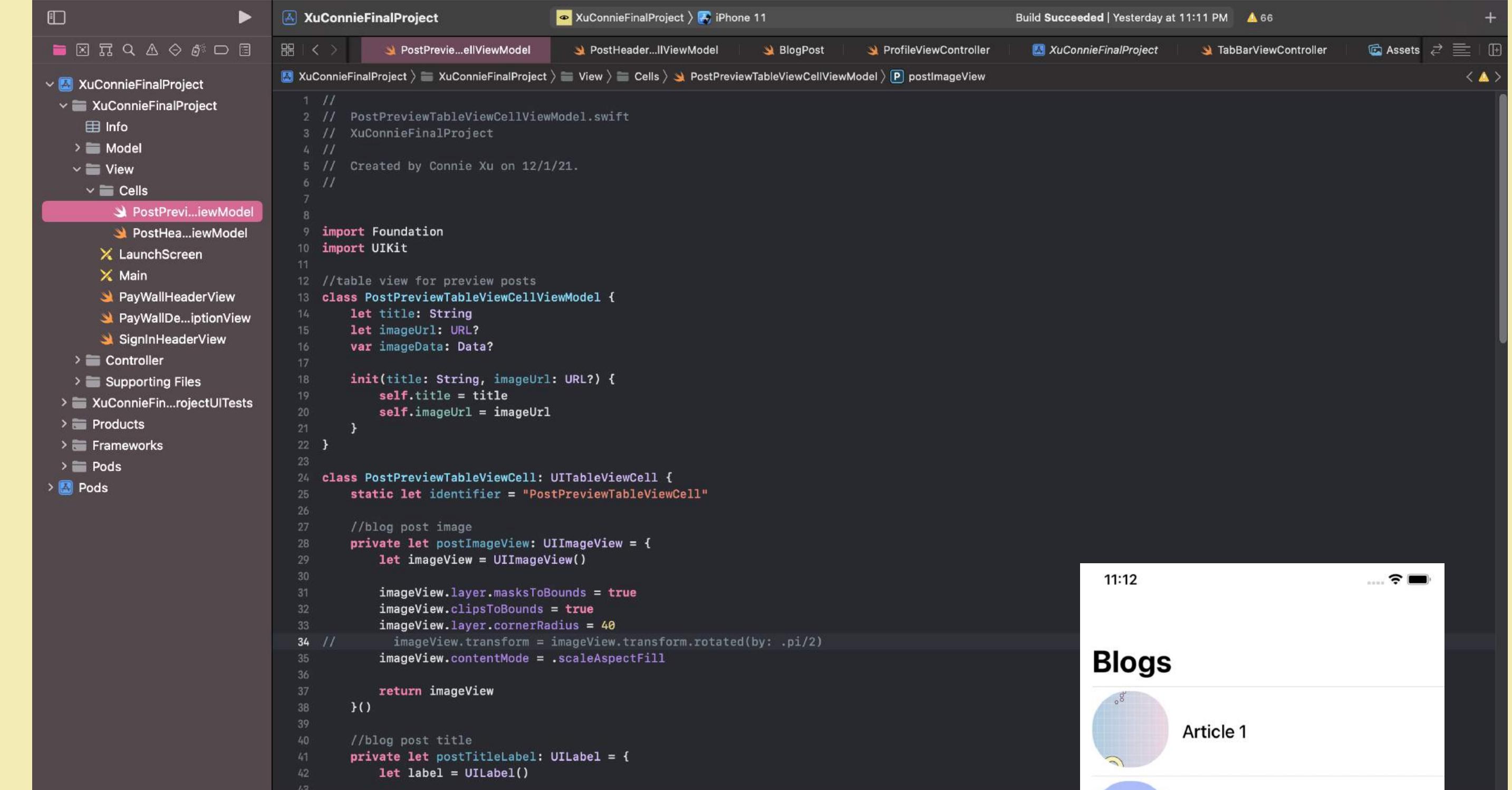
PROJECT SPECS

MVC Design Pattern

View

PostPreviewTableViewCellView

- List of posts view model



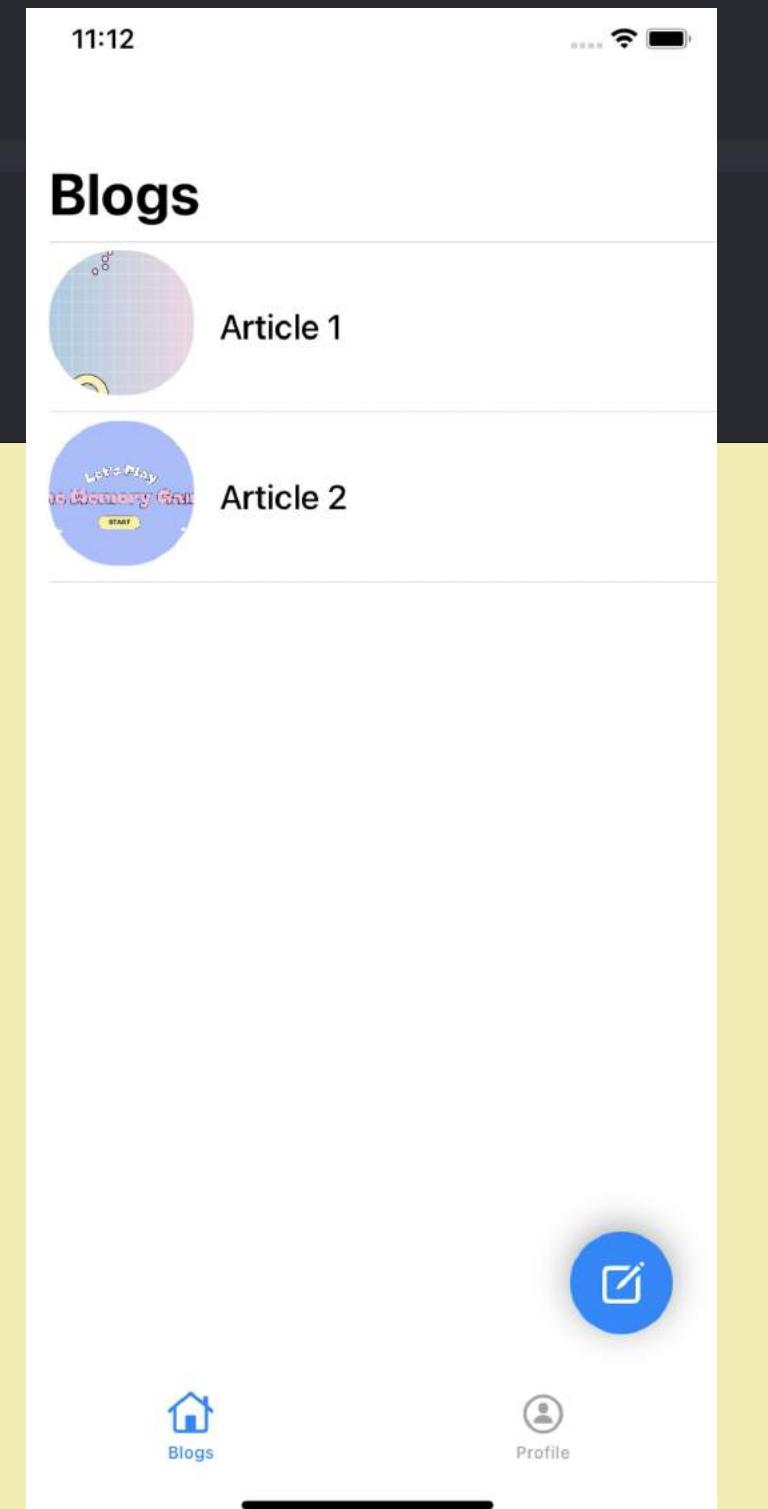
The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The file 'PostPreviewTableViewCellViewModel.swift' is selected in the list. The code defines a view model for a table view cell, handling post titles and images.

```
1 //  
2 // PostPreviewTableViewCellViewModel.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 12/1/21.  
6 //  
7  
8 import Foundation  
9 import UIKit  
10  
11 //table view for preview posts  
12 class PostPreviewTableViewCellViewModel {  
13     let title: String  
14     let imageUrl: URL?  
15     var imageData: Data?  
16  
17     init(title: String, imageUrl: URL?) {  
18         self.title = title  
19         self.imageUrl = imageUrl  
20     }  
21  
22 }  
23  
24 class PostPreviewTableViewCell: UITableViewCell {  
25     static let identifier = "PostPreviewTableViewCell"  
26  
27     //blog post image  
28     private let postImageView: UIImageView = {  
29         let imageView = UIImageView()  
30  
31         imageView.layer.masksToBounds = true  
32         imageView.clipsToBounds = true  
33         imageView.layer.cornerRadius = 40  
34         //         imageView.transform = imageView.transform.rotated(by: .pi/2)  
35         imageView.contentMode = .scaleAspectFill  
36  
37         return imageView  
38     }()  
39  
40     //blog post title  
41     private let postTitleLabel: UILabel = {  
42         let label = UILabel()  
43         return label  
44     }()  
45  
46 }
```

> XuConnieFinalProject > View > LaunchScreen.storyboard

> PayWallHeaderView.swift
> PayWallDescriptionView.swift
> SignInHeaderView.swift

> Cells > PostPreviewTableViewCellViewModel.swift
> Cells > PostHeaderTableViewCellViewModel.swift



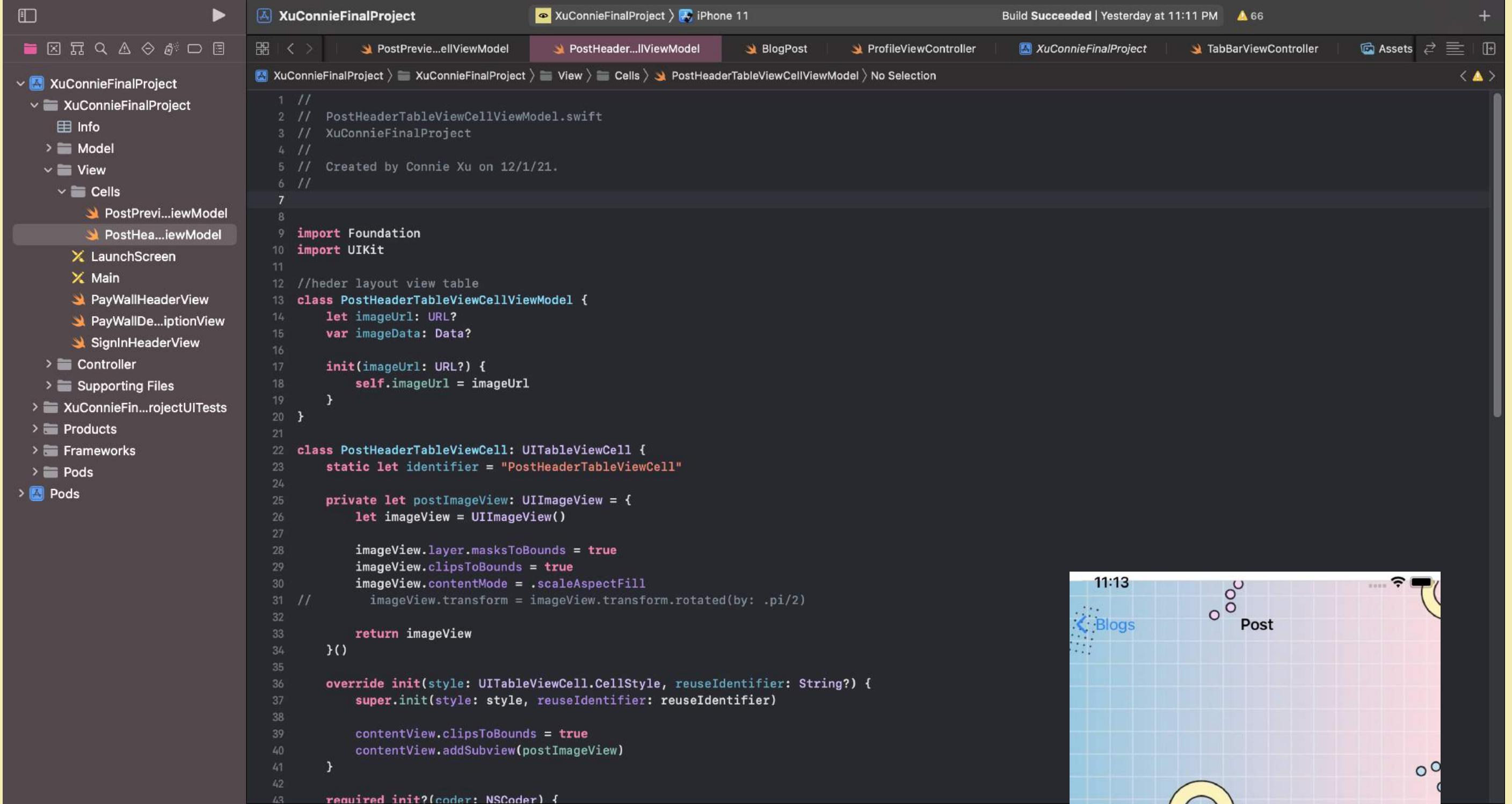
PROJECT SPECS

MVC Design Pattern

View

PostHeaderTableViewCellView

- Hero image of post view model



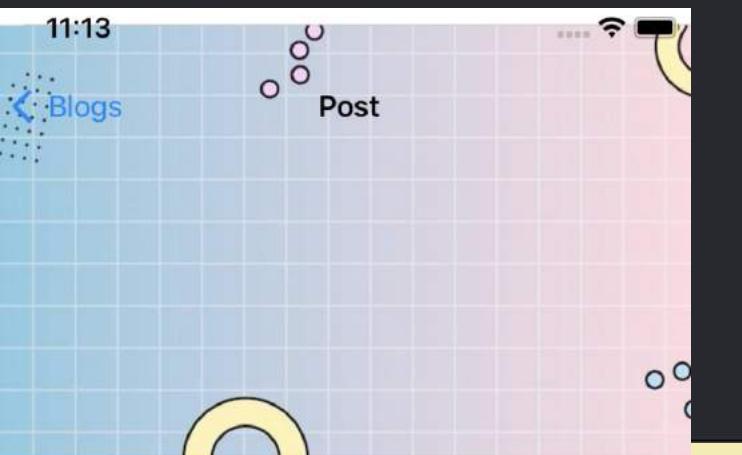
The screenshot shows the Xcode interface with the project structure on the left and the code editor on the right. The code editor displays the `PostHeaderTableViewCellViewModel.swift` file, which defines a class for a table view cell. The code includes properties for an image URL and data, and methods for initializing the cell and setting up its content view.

```
1 //  
2 // PostHeaderTableViewCellViewModel.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 12/1/21.  
6 //  
7  
8 import Foundation  
9 import UIKit  
10  
11 //header layout view table  
12 class PostHeaderTableViewCellViewModel {  
13     let imageUrl: URL?  
14     var imageData: Data?  
15  
16     init(imageUrl: URL?) {  
17         self.imageUrl = imageUrl  
18     }  
19 }  
20  
21 class PostHeaderTableViewCell: UITableViewCell {  
22     static let identifier = "PostHeaderTableViewCell"  
23  
24     private let postImageView: UIImageView = {  
25         let imageView = UIImageView()  
26  
27         imageView.layer.masksToBounds = true  
28         imageView.clipsToBounds = true  
29         imageView.contentMode = .scaleAspectFill  
30         // imageView.transform = imageView.transform.rotated(by: .pi/2)  
31         return imageView  
32     }  
33  
34     override init(style: UITableViewCell.CellStyle, reuseIdentifier: String?) {  
35         super.init(style: style, reuseIdentifier: reuseIdentifier)  
36  
37         contentView.clipsToBounds = true  
38         contentView.addSubview(postImageView)  
39     }  
40  
41     required init?(coder: NSCoder) {  
42         fatalError("init(coder:) has not been implemented")  
43     }  
44 }
```

> XuConnieFinalProject > View > LaunchScreen.storyboard

> PayWallHeaderView.swift
> PayWallDescriptionView.swift
> SignInHeaderView.swift

> Cells > PostPreviewTableViewCellViewModel.swift
> Cells > PostHeaderTableViewCellViewModel.swift



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Porta non pulvinar neque laoreet. Ipsum a arcu cursus vitae congue. Nunc lobortis mattis aliquam faucibus purus in massa tempor. Eget arcu dictum varius duis. Magna fermentum iaculis eu non diam phasellus vestibulum. Adipiscing elit ut aliquam purus sit amet luctus venenatis. Mi bibendum neque egestas congue quisque egestas diam in. Nisl vel pretium lectus quam id leo in vitae. Scelerisque mauris pellentesque pulvinar pellentesque habitant morbi tristique senectus et. Purus semper eget duis at. Eu sem integer vitae justo eget magna. Sed tempus urna et pharetra pharetra massa massa ultricies. Egestas integer eget aliquet nibh praesent.

Lorem ipsum dolor sit amet consectetur adipiscing. Condimentum mattis pellentesque id nibh tortor. Dictum sit amet justo donec enim diam vulputate ut pharetra. Rutrum tellus pellentesque eu tincidunt tortor. Facilisi nullam vehicula ipsum a arcu cursus. Egestas dui id ornare arcu odio ut sem nulla pharetra. Sed risus ultricies tristique nulla aliquet enim tortor at auctor. Senectus et netus et malesuada fames

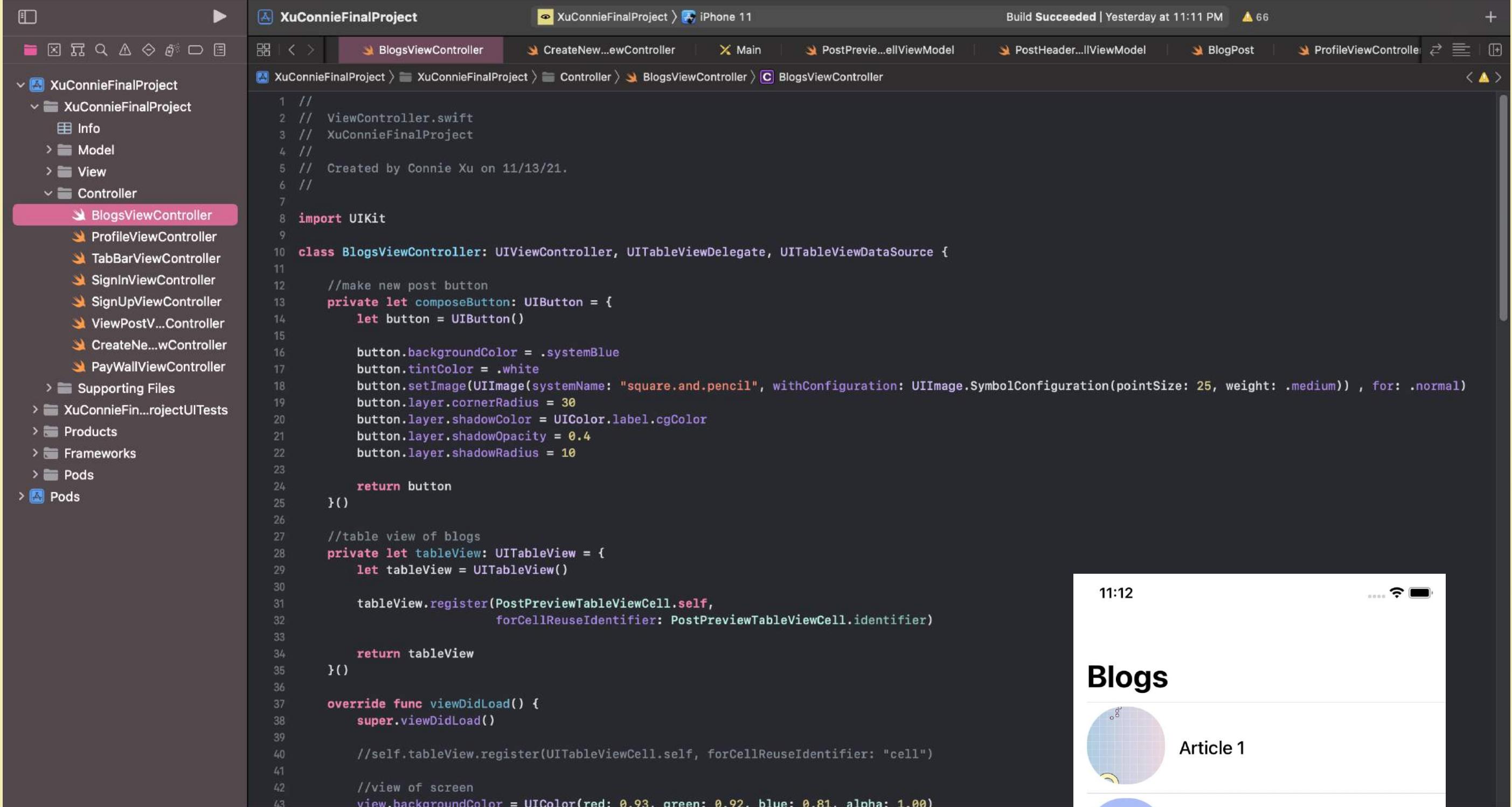
PROJECT SPECS

MVC Design Pattern

Controller

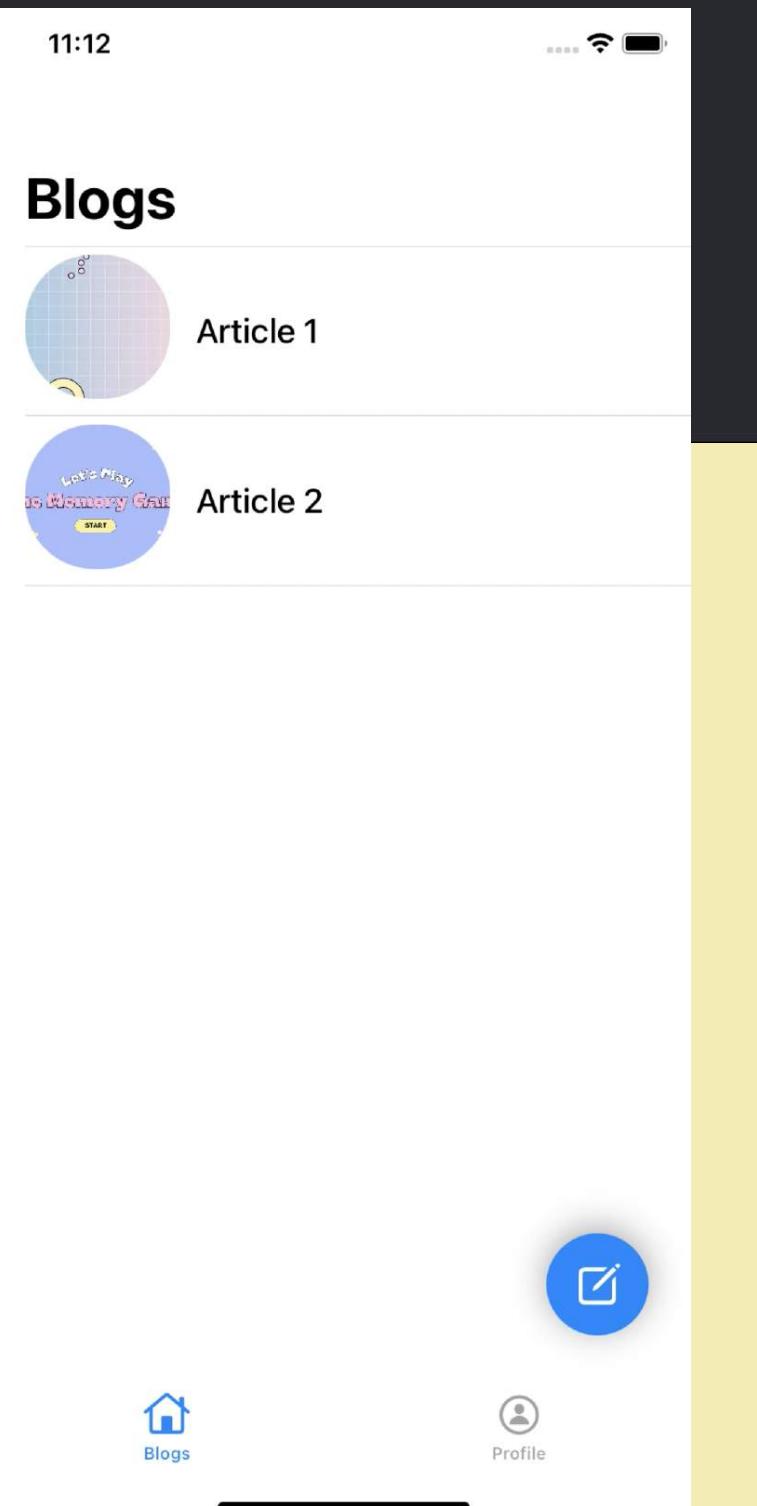
BlogPostViewController

- Blogs Tab
- Lists all blog posts



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar shows the project structure with a pink highlight on 'BlogsViewController'. The main editor area displays the code for 'BlogsViewController.swift'. The code initializes a compose button and a UITableView to list blog posts.

```
1 // ViewController.swift
2 // XuConnieFinalProject
3 // Created by Connie Xu on 11/13/21.
4 //
5
6 import UIKit
7
8 class BlogsViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
9
10    //make new post button
11    private let composeButton: UIButton = {
12        let button = UIButton()
13
14        button.backgroundColor = .systemBlue
15        button.tintColor = .white
16        button.setImage(UIImage(systemName: "square.and.pencil", withConfiguration: UIImage.SymbolConfiguration(pointSize: 25, weight: .medium)) , for: .normal)
17        button.layer.cornerRadius = 30
18        button.layer.shadowColor = UIColor.label.cgColor
19        button.layer.shadowOpacity = 0.4
20        button.layer.shadowRadius = 10
21
22        return button
23    }()
24
25
26    //table view of blogs
27    private let tableView: UITableView = {
28        let tableView = UITableView()
29
30        tableView.register(PostPreviewTableViewCell.self,
31                           forCellReuseIdentifier: PostPreviewTableViewCell.identifier)
32
33        return tableView
34    }()
35
36
37    override func viewDidLoad() {
38        super.viewDidLoad()
39
40        //self.tableView.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
41
42        //view of screen
43        view.backgroundColor = UIColor(red: 0.93, green: 0.92, blue: 0.81, alpha: 1.00)
44    }
45}
```



> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

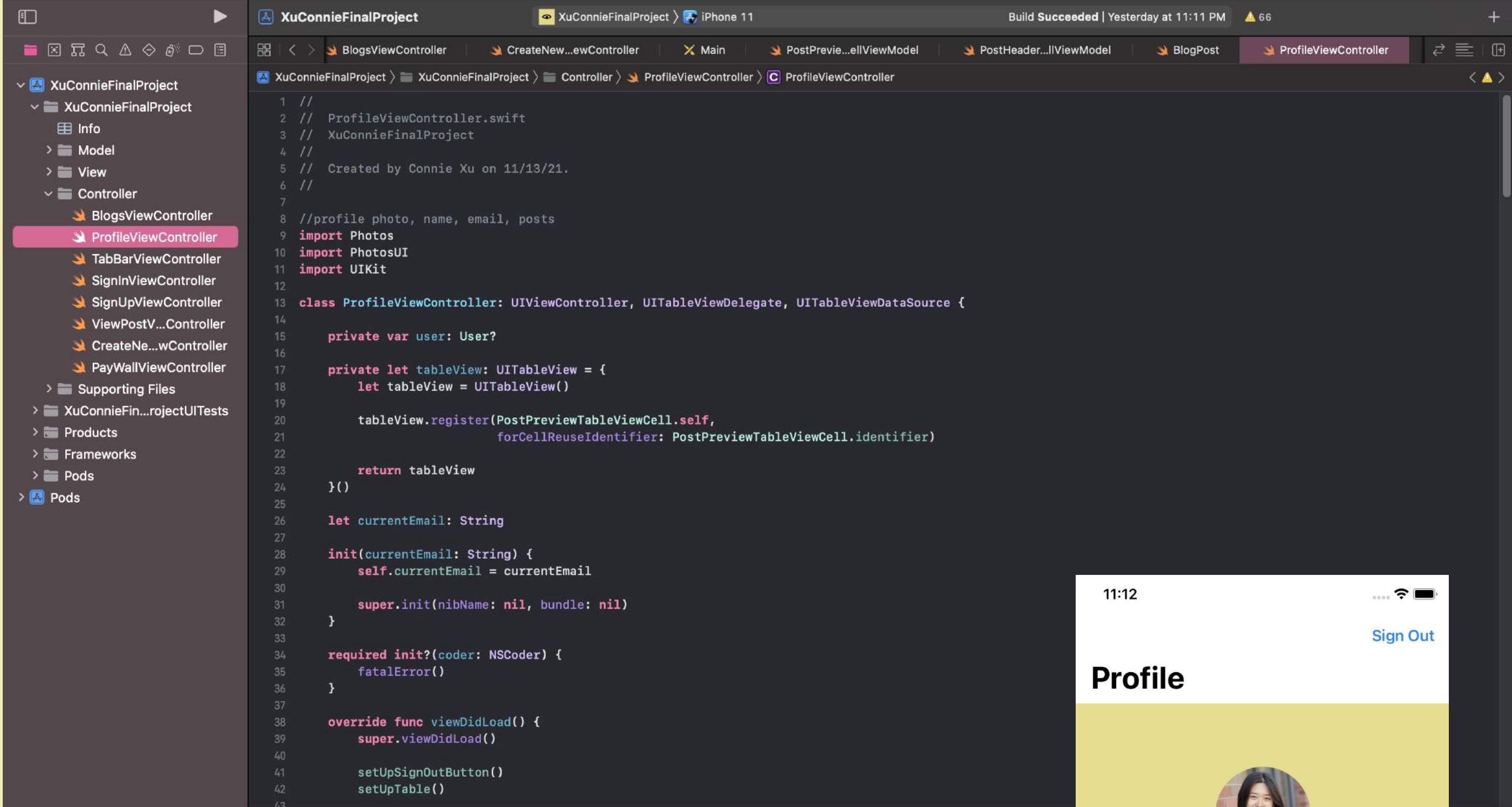
PROJECT SPECS

MVC Design Pattern

Controller

ProfileViewController

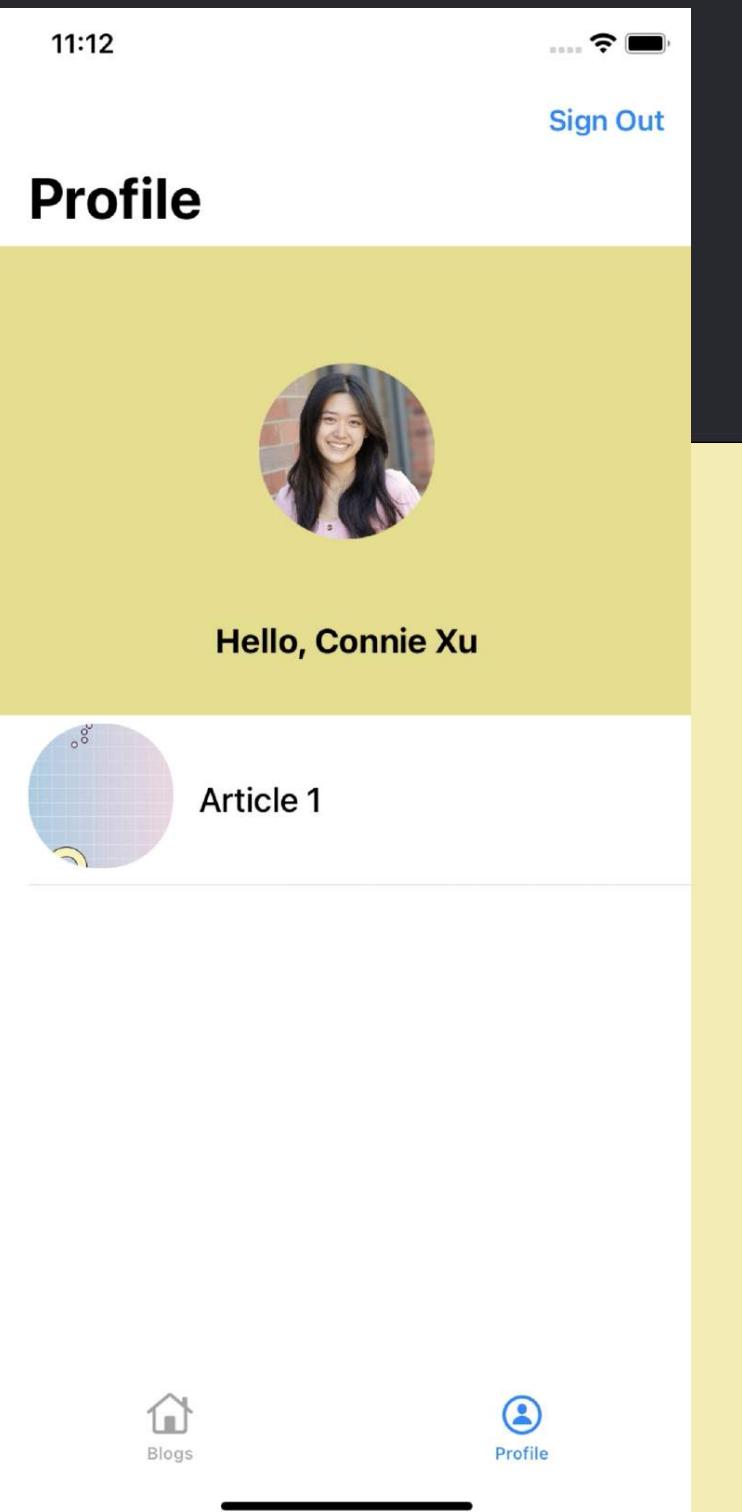
- Profile Tab
- Lists user's blog posts



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. In the Project Navigator, the 'Controller' folder is expanded, and the 'ProfileViewController' file is selected, highlighted with a pink rectangle. The main editor area displays the code for ProfileViewController.swift:

```
1 // ProfileViewController.swift
2 // XuConnieFinalProject
3 // Created by Connie Xu on 11/13/21.
4 //
5
6 //profile photo, name, email, posts
7 import Photos
8 import PhotosUI
9 import UIKit
10
11 class ProfileViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
12
13     private var user: User?
14
15     private let tableView: UITableView = {
16         let tableView = UITableView()
17
18         tableView.register(PostPreviewTableViewCell.self,
19                         forCellReuseIdentifier: PostPreviewTableViewCell.identifier)
20
21         return tableView
22     }()
23
24     let currentUserEmail: String
25
26     init(currentUserEmail: String) {
27         self.currentUserEmail = currentUserEmail
28
29         super.init(nibName: nil, bundle: nil)
30     }
31
32     required init?(coder: NSCoder) {
33         fatalError()
34     }
35
36     override func viewDidLoad() {
37         super.viewDidLoad()
38
39         setUpSignOutButton()
40         setUpTable()
41     }
42
43 }
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift



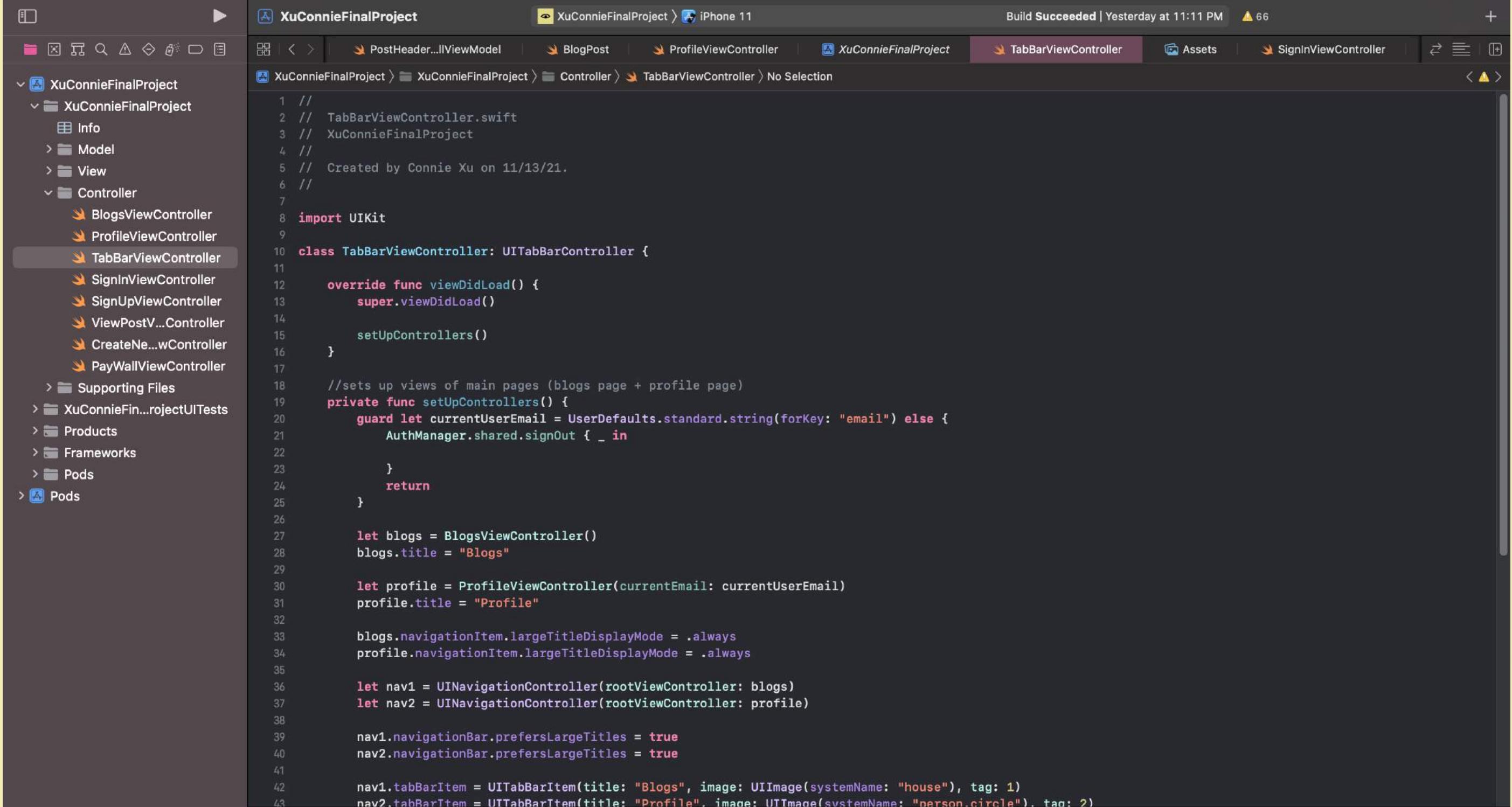
PROJECT SPECS

MVC Design Pattern

Controller

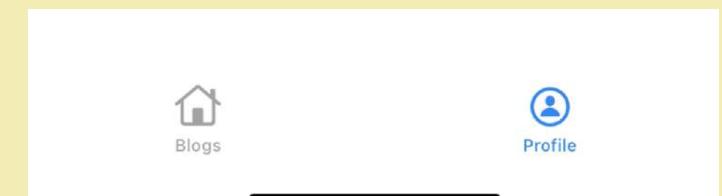
TabBarViewController

- 2 Tabs
 - Blogs Tab
 - Profile Tab



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar displays the project structure, including 'Info', 'Model', 'View', and 'Controller' folders containing various view controller classes like 'BlogsViewController', 'ProfileViewController', 'TabBarViewController', etc. The right pane shows the 'TabBarViewController.swift' file with the following code:

```
1 //  
2 // TabBarViewController.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/13/21.  
6 //  
7  
8 import UIKit  
9  
10 class TabBarViewController: UITabBarController {  
11  
12     override func viewDidLoad() {  
13         super.viewDidLoad()  
14  
15         setUpControllers()  
16     }  
17  
18     //sets up views of main pages (blogs page + profile page)  
19     private func setUpControllers() {  
20         guard let currentUserEmail = UserDefaults.standard.string(forKey: "email") else {  
21             AuthManager.shared.signOut { _ in  
22                 return  
23             }  
24         }  
25  
26         let blogs = BlogsViewController()  
27         blogs.title = "Blogs"  
28  
29         let profile = ProfileViewController(currentEmail: currentUserEmail)  
30         profile.title = "Profile"  
31  
32         blogs.navigationItem.largeTitleDisplayMode = .always  
33         profile.navigationItem.largeTitleDisplayMode = .always  
34  
35         let nav1 = UINavigationController(rootViewController: blogs)  
36         let nav2 = UINavigationController(rootViewController: profile)  
37  
38         nav1.navigationBar.prefersLargeTitles = true  
39         nav2.navigationBar.prefersLargeTitles = true  
40  
41         nav1.tabBarItem = UITabBarItem(title: "Blogs", image: UIImage(systemName: "house"), tag: 1)  
42         nav2.tabBarItem = UITabBarItem(title: "Profile", image: UIImage(systemName: "person.circle"), tag: 2)  
43 }
```



> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

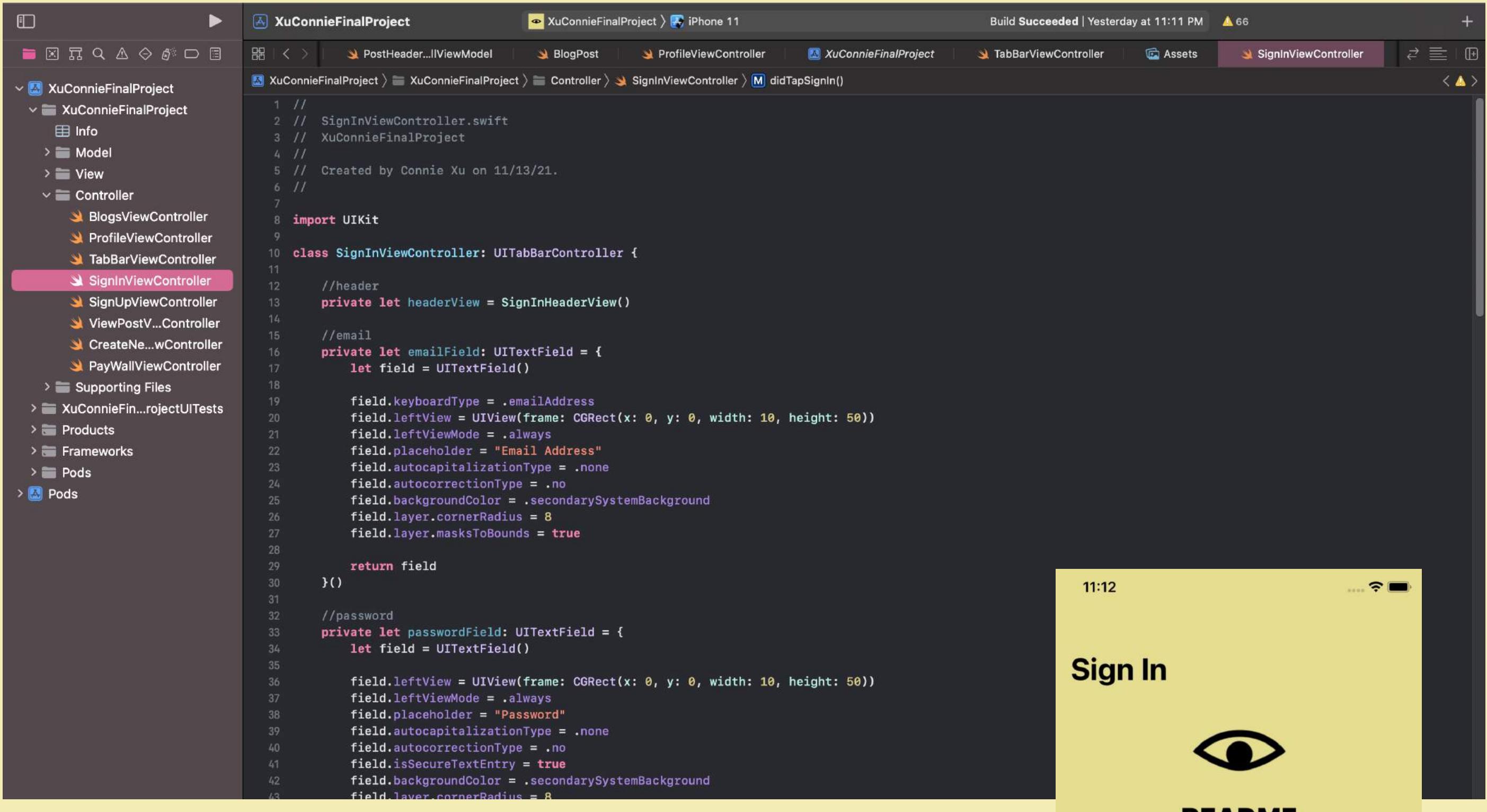
PROJECT SPECS

MVC Design Pattern

Controller

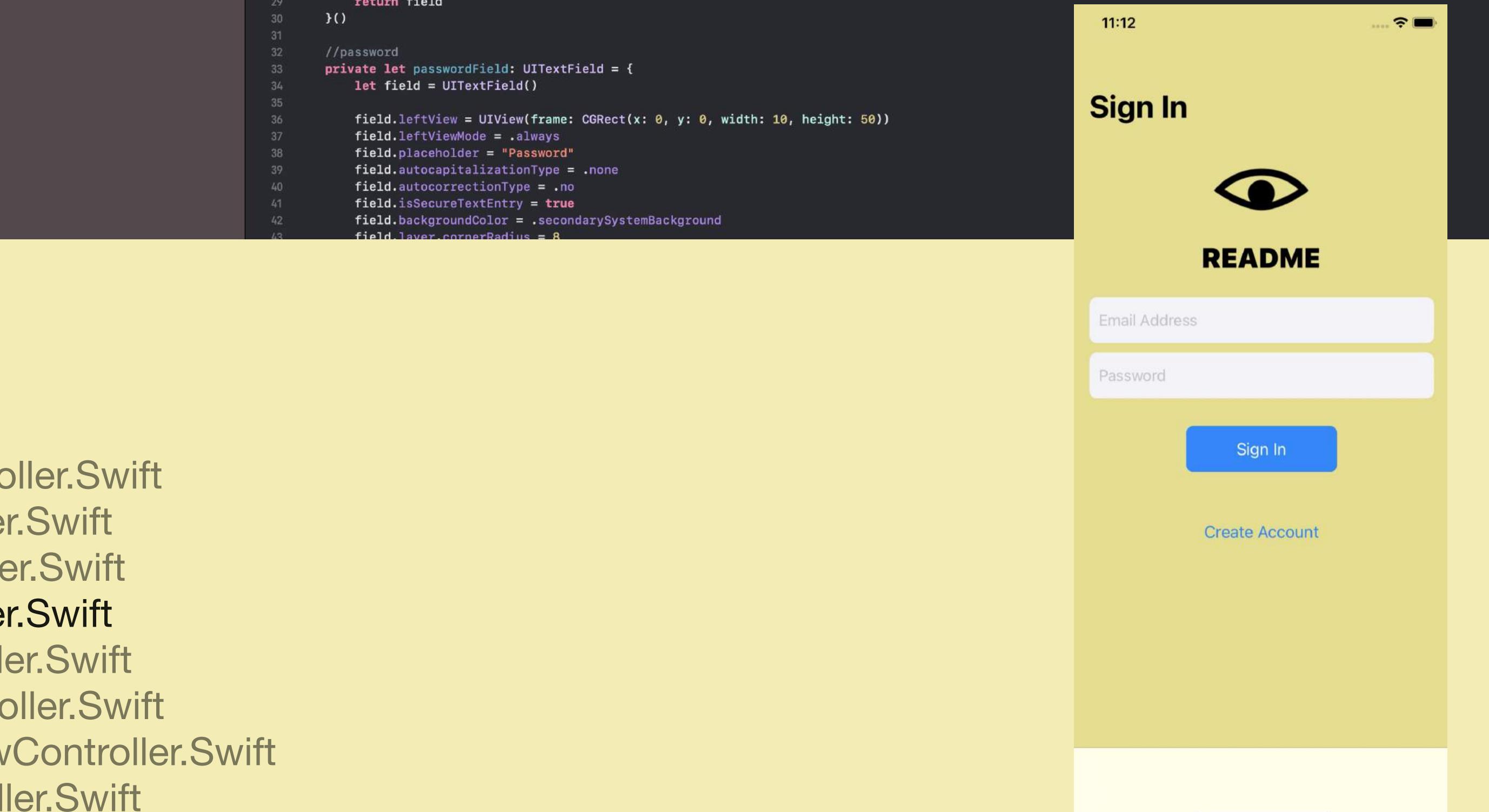
SignInViewController

- Sign into app
- Account already exists



The screenshot shows the Xcode interface with the project structure on the left and the code editor on the right. The code editor displays `SignInViewController.swift` with the following content:

```
1 //  
2 // SignInViewController.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/13/21.  
6 //  
7  
8 import UIKit  
9  
10 class SignInViewController: UITabBarController {  
11  
    //header  
    private let headerView = SignInHeaderView()  
  
    //email  
    private let emailField: UITextField = {  
        let field = UITextField()  
  
        field.keyboardType = .emailAddress  
        field.leftView = UIView(frame: CGRect(x: 0, y: 0, width: 10, height: 50))  
        field.leftViewMode = .always  
        field.placeholder = "Email Address"  
        field.autocapitalizationType = .none  
        field.autocorrectionType = .no  
        field.backgroundColor = .secondarySystemBackground  
        field.layer.cornerRadius = 8  
        field.layer.masksToBounds = true  
  
        return field  
    }()  
  
    //password  
    private let passwordField: UITextField = {  
        let field = UITextField()  
  
        field.leftView = UIView(frame: CGRect(x: 0, y: 0, width: 10, height: 50))  
        field.leftViewMode = .always  
        field.placeholder = "Password"  
        field.autocapitalizationType = .none  
        field.autocorrectionType = .no  
        field.isSecureTextEntry = true  
        field.backgroundColor = .secondarySystemBackground  
        field.layer.cornerRadius = 8  
  
        return field  
    }()  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        self.view.addSubview(headerView)  
        self.view.addSubview(emailField)  
        self.view.addSubview(passwordField)  
    }  
}
```



> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

PROJECT SPECS

MVC Design Pattern

Controller

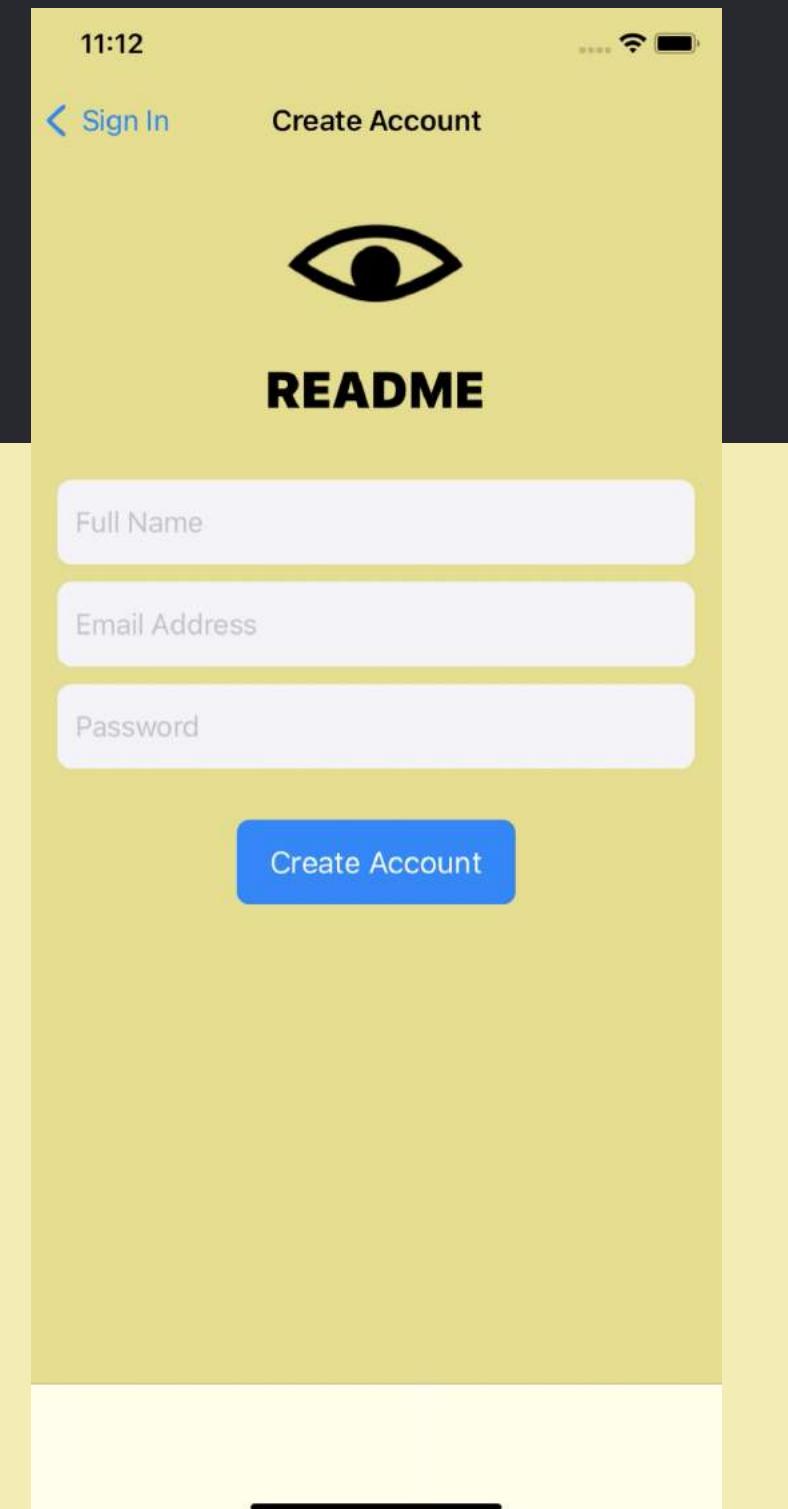
SignUpViewController

- Create new account
 - Account does not exists yet

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar displays the project structure, including files like Info, Model, View, Controller (with SignUpViewController selected), and Supporting Files. The main editor area shows the code for SignUpViewController.swift, which extends UITabBarController and defines two UITextField properties: nameField and emailField. The right side of the interface shows a preview of the app's sign-up screen. The screen has a light yellow background with a large eye icon in the center. At the top, there are navigation buttons: 'Sign In' (blue) and 'Create Account' (black). Below the eye icon, the word 'README' is displayed in large, bold, black capital letters.

```
1 //  
2 //  SignUpViewController.swift  
3 //  XuConnieFinalProject  
4 //  
5 //  Created by Connie Xu on 11/13/21.  
6 //  
7  
8 import UIKit  
9  
10 class SignUpViewController: UITabBarController {  
11  
12     //header  
13     private let headerView = SignInHeaderView()  
14  
15     //name text field  
16     private let nameField: UITextField = {  
17         let field = UITextField()  
18  
19         field.leftView = UIView(frame: CGRect(x: 0, y: 0, width: 10, height: 50))  
20         field.leftViewMode = .always  
21         field.placeholder = "Full Name"  
22         field.backgroundColor = .secondarySystemBackground  
23         field.layer.cornerRadius = 8  
24         field.layer.masksToBounds = true  
25  
26         return field  
27     }()  
28  
29     //email text field  
30     private let emailField: UITextField = {  
31         let field = UITextField()  
32  
33         field.keyboardType = .emailAddress  
34         field.autocapitalizationType = .none  
35         field.autocorrectionType = .no  
36         field.leftView = UIView(frame: CGRect(x: 0, y: 0, width: 10, height: 50))  
37         field.leftViewMode = .always  
38         field.placeholder = "Email Address"  
39         field.backgroundColor = .secondarySystemBackground  
40         field.layer.cornerRadius = 8  
41         field.layer.masksToBounds = true  
42  
43         return field  
44     }()
```



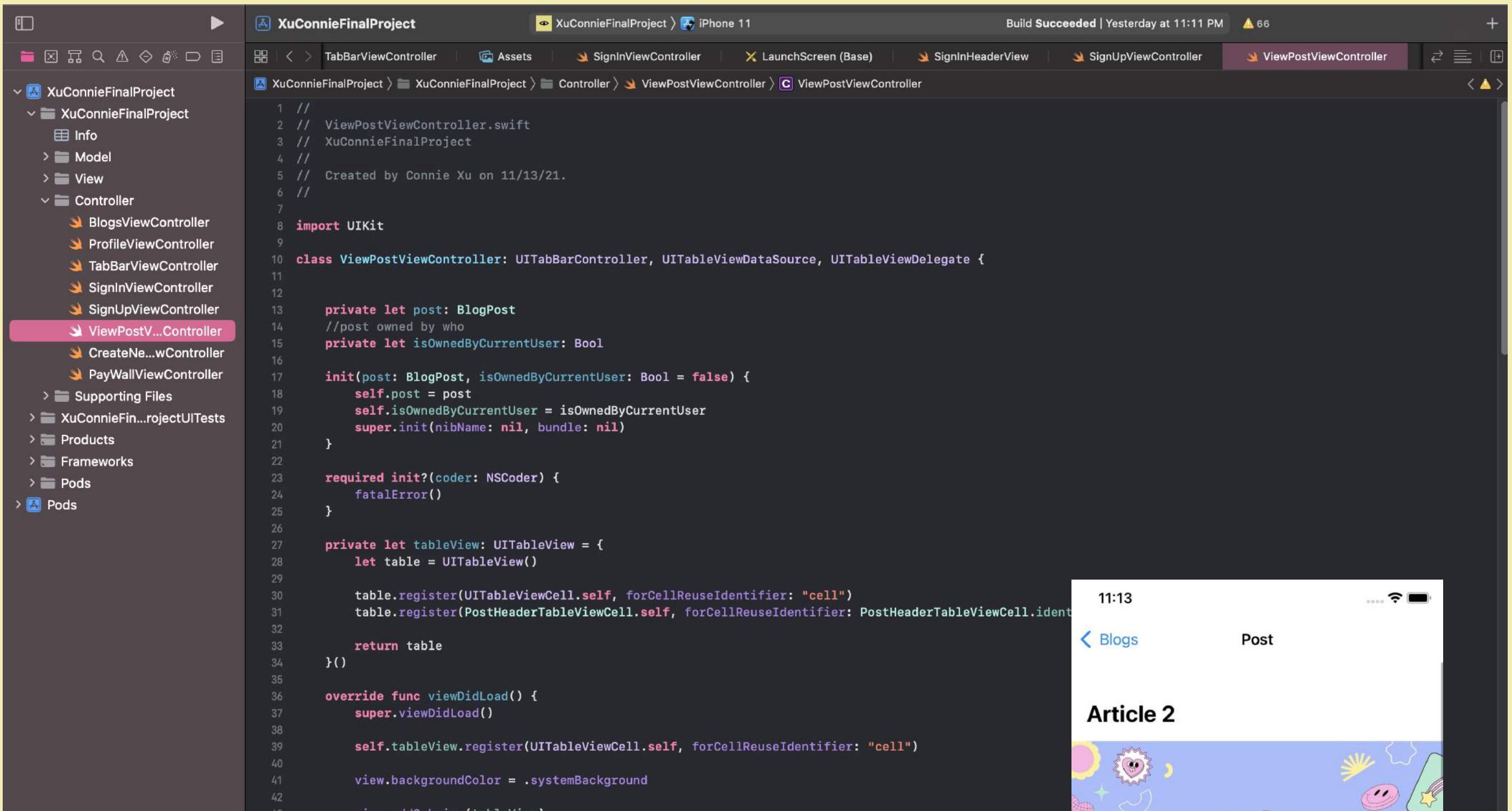
PROJECT SPECS

MVC Design Pattern

Controller

ViewPostViewController

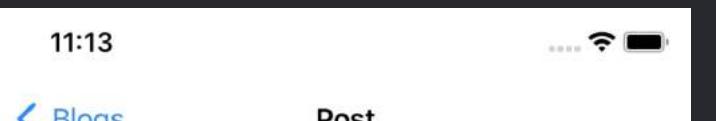
- View blog post



The screenshot shows the Xcode interface with the project structure on the left and the code editor on the right. The code editor displays the `ViewPostViewController.swift` file, which is a subclass of `UITabBarController`. The code initializes a `BlogPost` object and sets up a `UITableView` to display it. The storyboard preview on the right shows a blog post titled "Article 2" by "Bryan Ge".

```
1 // XuConnieFinalProject
2 // ViewPostViewController.swift
3 // XuConnieFinalProject
4 //
5 // Created by Connie Xu on 11/13/21.
6 //
7
8 import UIKit
9
10 class ViewPostViewController: UITabBarController, UITableViewDataSource, UITableViewDelegate {
11
12     private let post: BlogPost
13     //post owned by who
14     private let isOwnedByCurrentUser: Bool
15
16     init(post: BlogPost, isOwnedByCurrentUser: Bool = false) {
17         self.post = post
18         self.isOwnedByCurrentUser = isOwnedByCurrentUser
19         super.init(nibName: nil, bundle: nil)
20     }
21
22     required init?(coder: NSCoder) {
23         fatalError()
24     }
25
26
27     private let tableView: UITableView = {
28         let table = UITableView()
29
30         table.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
31         table.register(PostHeaderTableViewCell.self, forCellReuseIdentifier: PostHeaderTableViewCell.identifier)
32
33         return table
34     }()
35
36     override func viewDidLoad() {
37         super.viewDidLoad()
38
39         self.tableView.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
40
41         view.backgroundColor = .systemBackground
42
43         view.addSubview(tableView)
44     }
45
46
47 }
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

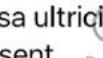
11:13  Build Succeeded | Yesterday at 11:11 PM | 66

Article 2

Let's Play The Memory Game

START

By: Bryan Ge

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Porta non pulvinar neque laoreet. Ipsum a arcu cursus vitae congue. Nunc lobortis mattis aliquam faucibus purus in massa tempor. Eget arcu dictum varius duis. Magna fermentum iaculis eu non diam phasellus vestibulum. Adipiscing elit ut aliquam purus sit amet luctus venenatis. Mi bibendum neque egestas congue quisque egestas diam in. Nisl vel pretium lectus quam id leo in vitae. Scelerisque mauris pellentesque pulvinar pellentesque habitant morbi tristique senectus et. Purus semper eget duis at. Eu sem integer vitae justo eget magna. Sed tempus urna et pharetra pharetra massa massa ultricies. Egestas integer eget aliquet nibh praesent. 

Profile

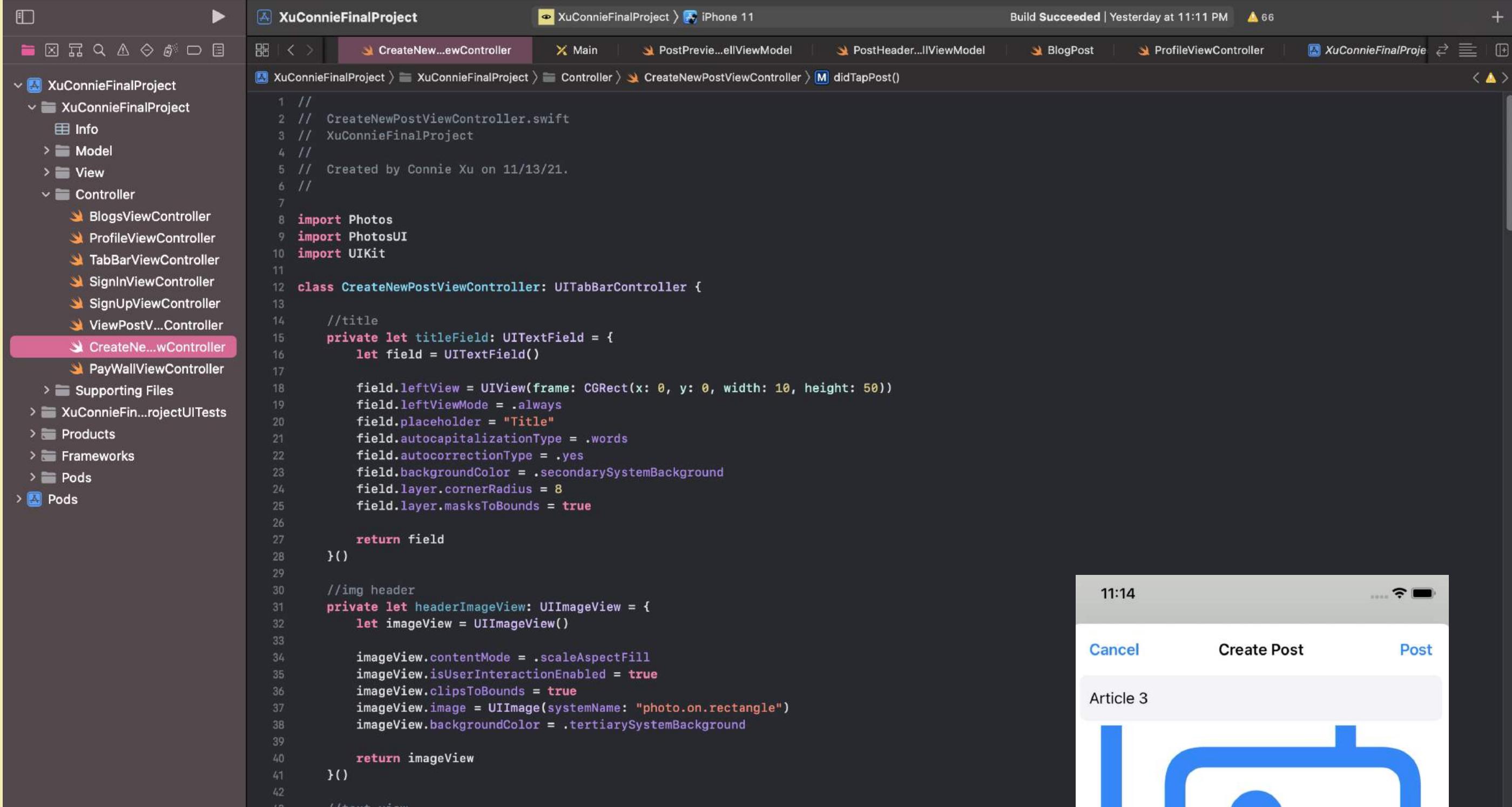
PROJECT SPECS

MVC Design Pattern

Controller

CreateNewPostViewController

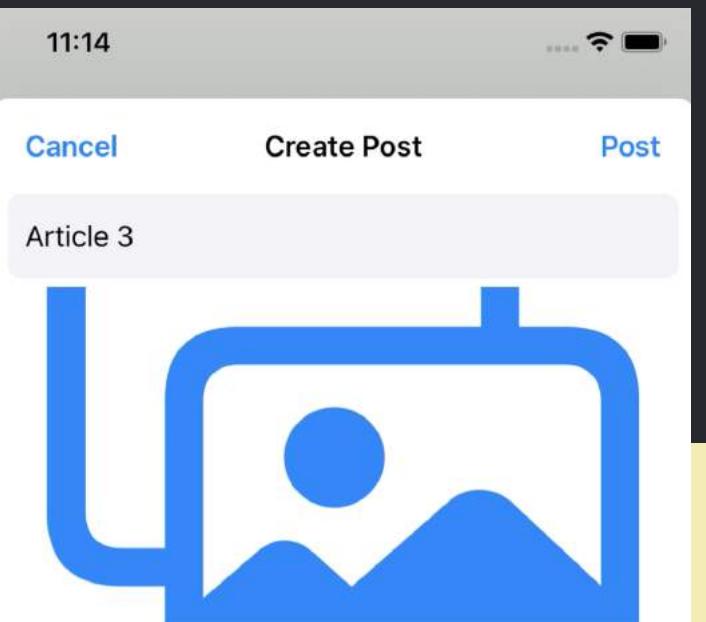
- Draft new blog post



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar shows the project structure with several controller files listed under 'Controller': BlogsViewController, ProfileViewController, TabBarController, SignInViewController, SignUpViewController, ViewPostViewController, CreateNewPostViewController (which is selected and highlighted in pink), and PayWallViewController. The main editor area displays the Swift code for 'CreateNewPostViewController'. The code defines a class that inherits from UITabBarController. It includes private properties for a title field (UITextView) and a header image view (UIImageView). The title field is initialized with a placeholder 'Title', autocapitalization set to 'words', and a secondary system background color. The header image view is set to scale aspect fill and has a tertiary system background color. The right side of the Xcode window shows a preview of the iOS application interface, which is a simple form titled 'Create Post' with fields for 'Article' and 'Image'.

```
1 //  
2 // CreateNewPostViewController.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/13/21.  
6 //  
7  
8 import Photos  
9 import PhotosUI  
10 import UIKit  
11  
12 class CreateNewPostViewController: UITabBarController {  
13  
14     //title  
15     private let titleField: UITextField = {  
16         let field = UITextField()  
17  
18         field.leftView = UIView(frame: CGRect(x: 0, y: 0, width: 10, height: 50))  
19         field.leftViewMode = .always  
20         field.placeholder = "Title"  
21         field.autocapitalizationType = .words  
22         field.autocorrectionType = .yes  
23         field.backgroundColor = .secondarySystemBackground  
24         field.layer.cornerRadius = 8  
25         field.layer.masksToBounds = true  
26  
27         return field  
28     }()  
29  
30     //img header  
31     private let headerImageView: UIImageView = {  
32         let imageView = UIImageView()  
33  
34         imageView.contentMode = .scaleAspectFill  
35         imageView.isUserInteractionEnabled = true  
36         imageView.clipsToBounds = true  
37         imageView.image = UIImage(systemName: "photo.on.rectangle")  
38         imageView.backgroundColor = .tertiarySystemBackground  
39  
40         return imageView  
41     }()  
42  
43     //text view
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift



consectetur adipiscing elit quis tristique.
Fringilla phasellus faucibus scelerisque eleifend
donec. Ultricies tristique nulla aliquet enim.
Feugiat in fermentum posuere urna nec. Magna
fermentum iaculis eu non. Purus faucibus
ornare suspendisse sed nisi lacus sed. Voluptat
est velit egestas duí id ornare arcu. Felis eget
velit aliquet sagittis id consectetur purus. Mi
eget mauris pharetra et ultrices. Blandit massa
enim nec duí nunc mattis enim ut. Ut diam
quam nulla porttitor massa id neque. Sit amet
consectetur adipiscing elit pellentesque.

Eget arcu dictum varius duis at consectetur
lorem. Urna cursus eget nunc scelerisque
viverra mauris in. Sed risus ultrices tristique
nulla aliquet enim tortor. Sit amet dictum sit
amet justo donec enim diam. Commodo viverra
maecenas accumsan lacus vel. Rhoncus urna
neque viverra justo nec ultrices duí sapien eget.
Tristique nulla aliquet enim tortor at auctor urna
nunc id. Justo donec enim diam vulputate.
Facilisis magna etiam tempor orci. Sit amet
mattis vulputate enim. Nec feugiat in

PROJECT SPECS

MVC Design Pattern

Controller

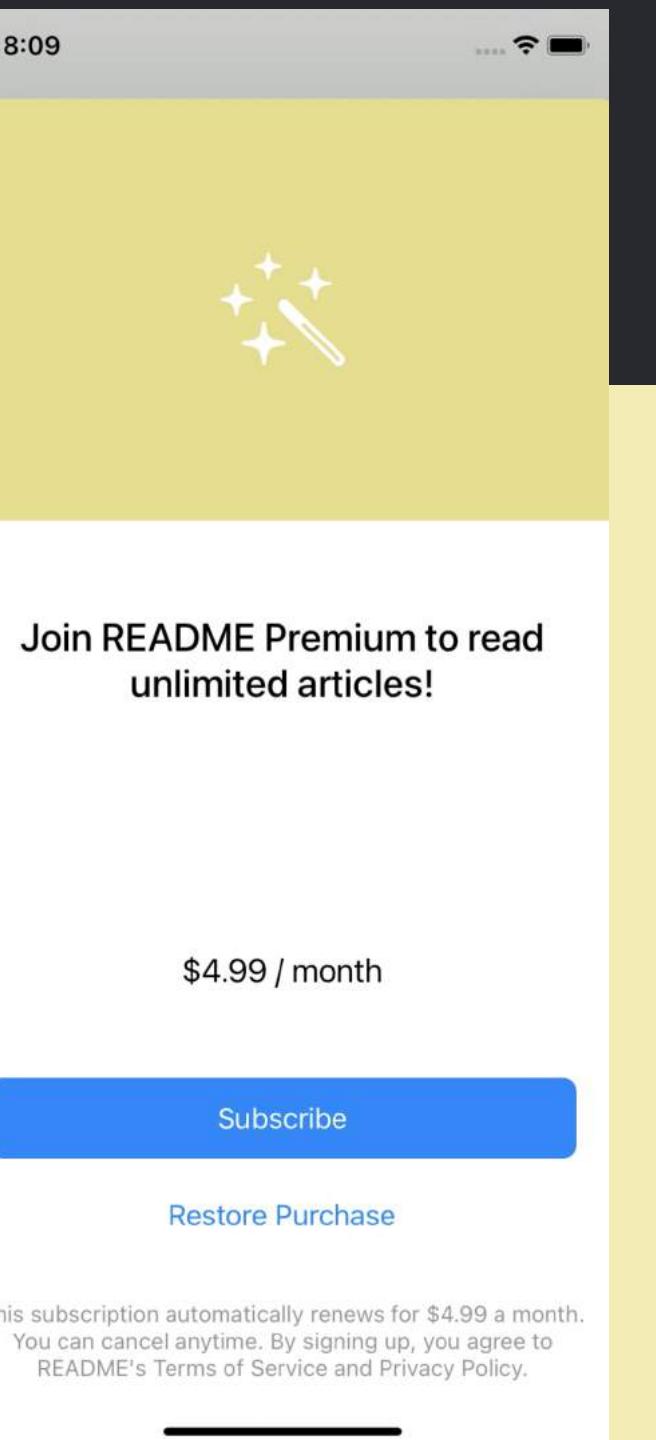
PayWallViewController

- Pop up screen for user to purchase README premium

The screenshot shows the Xcode interface with the project structure on the left and the code editor on the right. The code editor displays the `PayWallViewController.swift` file, which contains Swift code for creating a paywall screen. The preview window on the right shows a yellow screen with a 'Subscribe' button and a 'Restore Purchase' button.

```
1 //  
2 // PayWallViewController.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/14/21.  
6 //  
7  
8 import UIKit  
9  
10  
11 class PayWallViewController: UIViewController {  
12  
13     //instance of views  
14     private let header = PayWallHeaderView()  
15     private let heroView = PayWallDescriptionView()  
16  
17     //header img  
18     private let headerImageView: UIImageView = {  
19         let imageView = UIImageView(image: UIImage(systemName: "crown.fill"))  
20         imageView.tintColor = .white  
21         imageView.contentMode = .scaleAspectFit  
22         return imageView  
23     }()  
24  
25     //cta buttons  
26     //subscribe  
27     private let buyButton: UIButton = {  
28         let button = UIButton()  
29  
30         button.setTitle("Subscribe", for: .normal)  
31         button.backgroundColor = .systemBlue  
32         button.setTitleColor(.white, for: .normal)  
33         button.layer.cornerRadius = 8  
34         button.layer.masksToBounds = true  
35  
36         return button  
37     }()  
38  
39     //restore  
40     private let restoreButton: UIButton = {  
41         let button = UIButton()  
42  
43         button.setTitle("Restore Purchase", for: .normal)  
44         button.backgroundColor = .systemBlue  
45         button.setTitleColor(.white, for: .normal)  
46         button.layer.cornerRadius = 8  
47         button.layer.masksToBounds = true  
48  
49         return button  
50     }()  
51  
52     override func viewDidLoad() {  
53         super.viewDidLoad()  
54         self.view.addSubview(header)  
55         self.view.addSubview(heroView)  
56         self.view.addSubview(buyButton)  
57         self.view.addSubview(restoreButton)  
58     }  
59  
60     func didTapRestore() {  
61         // Implement restore logic here  
62     }  
63 }
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift



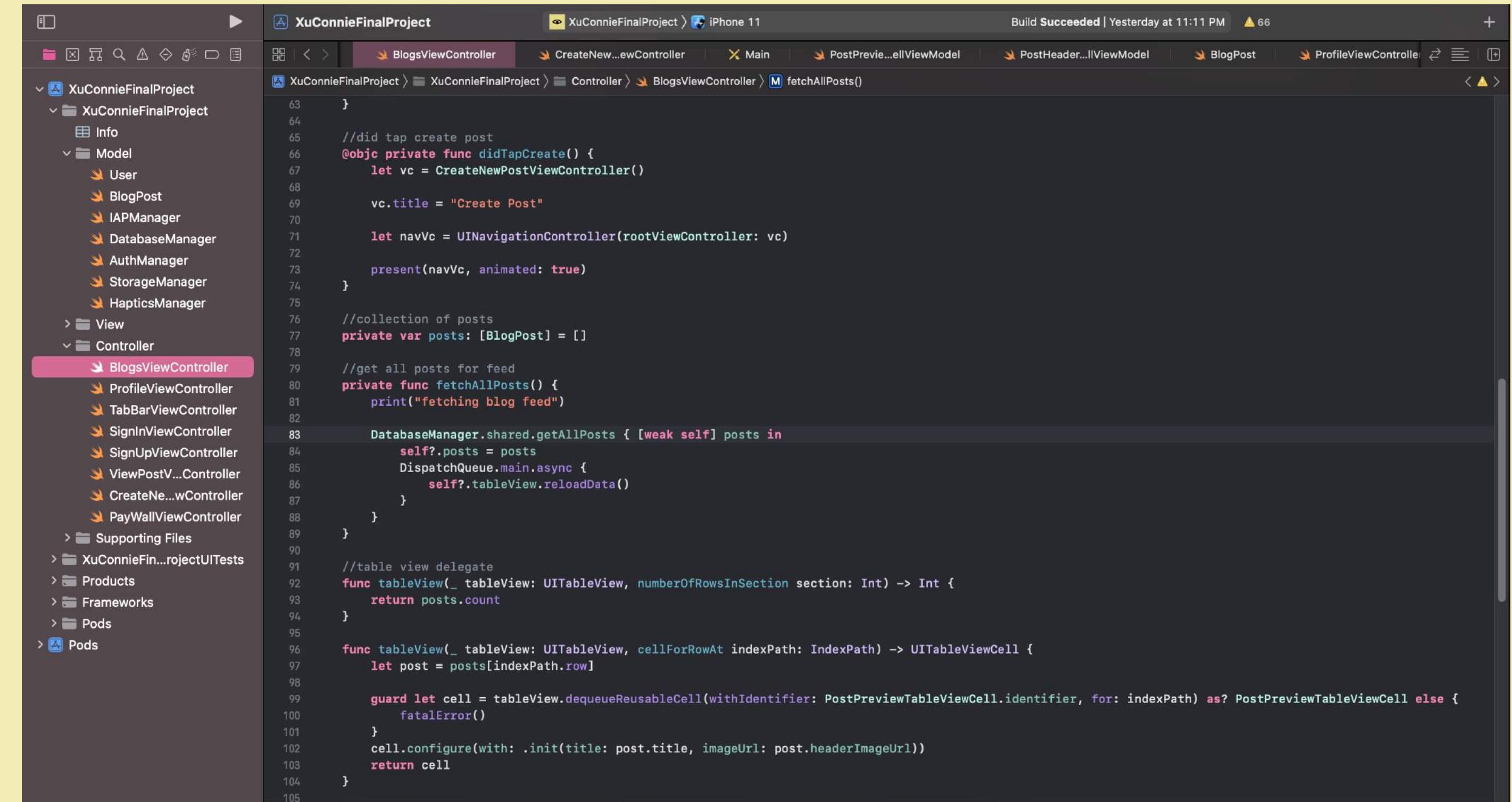
PROJECT SPECS

Singleton

BlogPostViewController

- DatabaseManager
- HapticsManager
- IAPManager

Usage of .shared



The screenshot shows the Xcode interface with the project navigation bar at the top. The left sidebar displays the project structure under 'XuConnieFinalProject'. The 'Controller' folder contains several Swift files, with 'BlogPostViewController' highlighted by a pink rectangle. The main editor area shows the code for 'BlogPostViewController.swift'. The code includes a private variable 'posts' of type [BlogPost], a function 'fetchAllPosts()' that uses 'DatabaseManager.shared.getAllPosts()', and table view delegate methods 'tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int' and 'tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell'.

```
63 }
64 //did tap create post
65 @objc private func didTapCreate() {
66     let vc = CreateNewPostViewController()
67     vc.title = "Create Post"
68     let navVc = UINavigationController(rootViewController: vc)
69     present(navVc, animated: true)
70 }
71
72 //collection of posts
73 private var posts: [BlogPost] = []
74
75 //get all posts for feed
76 private func fetchAllPosts() {
77     print("fetching blog feed")
78
79     DatabaseManager.shared.getAllPosts { [weak self] posts in
80         self?.posts = posts
81         DispatchQueue.main.async {
82             self?.tableView.reloadData()
83         }
84     }
85 }
86
87 //table view delegate
88 func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
89     return posts.count
90 }
91
92 func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
93     let post = posts[indexPath.row]
94
95     guard let cell = tableView.dequeueReusableCell(withIdentifier: PostPreviewTableViewCell.identifier, for: indexPath) as? PostPreviewTableViewCell else {
96         fatalError()
97     }
98     cell.configure(with: .init(title: post.title, imageUrl: post.headerImageUrl))
99     return cell
100 }
101
102
103
104
105 }
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

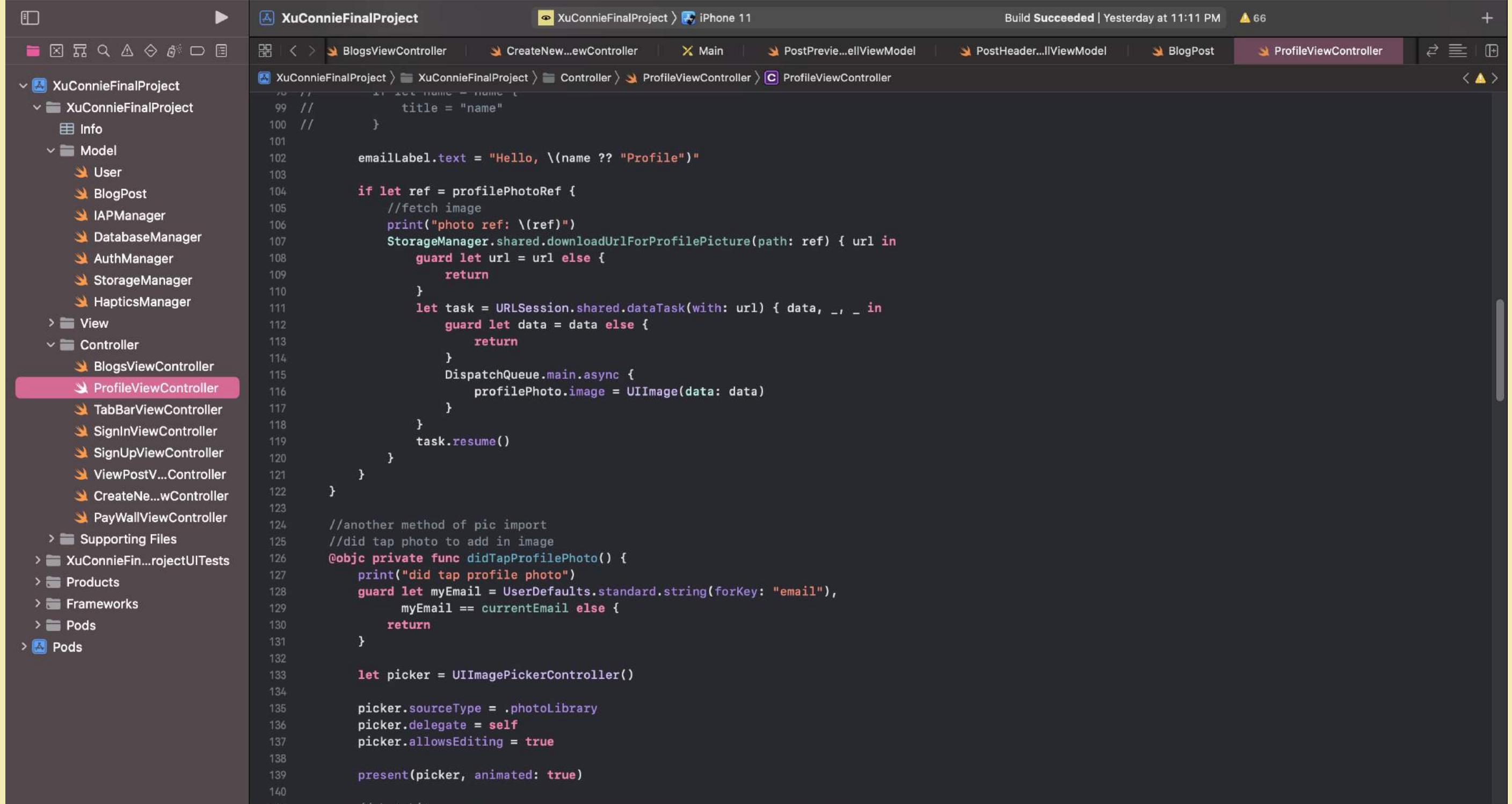
PROJECT SPECS

Singleton

ProfileViewController

- DatabaseManager
- IAPManager
- StorageManager
- AuthManager
- HapticsManager

Usage of .shared



The screenshot shows the Xcode interface with the project navigation bar at the top. Below it, the project structure is displayed in a tree view:

- XuConnieFinalProject
 - Info
 - Model
 - User
 - BlogPost
 - IAPManager
 - DatabaseManager
 - AuthManager
 - StorageManager
 - HapticsManager
 - View
 - Controller
 - BlogsViewController
 - ProfileViewController (highlighted with a pink rectangle)
 - TabBarController
 - SignInViewController
 - SignUpViewController
 - ViewPostViewController
 - CreateNewPostViewController
 - PayWallViewController
 - Supporting Files
 - XuConnieFinalProjectUITests
 - Products
 - Frameworks
 - Pods
 - Pods (selected)

The code editor on the right contains the implementation of the `ProfileViewController`. A portion of the code is shown below, demonstrating the use of `StorageManager.shared.downloadUrlForProfilePicture` to fetch a profile picture URL.

```
99 // title = "name"
100 // }
101
102 emailLabel.text = "Hello, \(name ?? "Profile")"
103
104 if let ref = profilePhotoRef {
105     //fetch image
106     print("photo ref: \(ref)")
107     StorageManager.shared.downloadUrlForProfilePicture(path: ref) { url in
108         guard let url = url else {
109             return
110         }
111         let task = URLSession.shared.dataTask(with: url) { data, _, _ in
112             guard let data = data else {
113                 return
114             }
115             DispatchQueue.main.async {
116                 profilePhoto.image = UIImage(data: data)
117             }
118         }
119         task.resume()
120     }
121 }
122
123 //another method of pic import
124 //did tap photo to add in image
125 @objc private func didTapProfilePhoto() {
126     print("did tap profile photo")
127     guard let myEmail = UserDefaults.standard.string(forKey: "email"),
128           myEmail == currentEmail else {
129         return
130     }
131
132     let picker = UIImagePickerController()
133
134     picker.sourceType = .photoLibrary
135     picker.delegate = self
136     picker.allowsEditing = true
137
138     present(picker, animated: true)
139
140 }
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

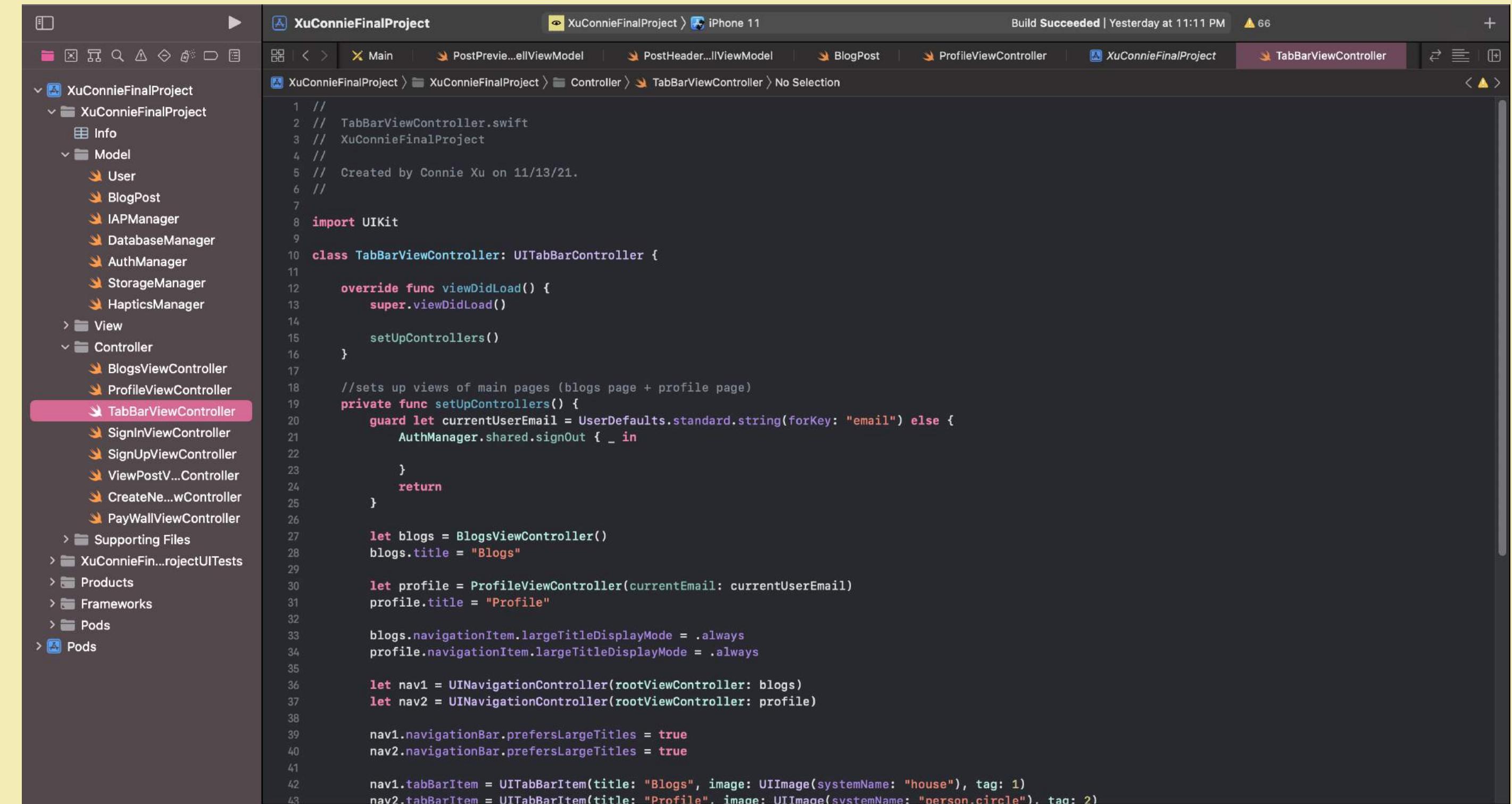
PROJECT SPECS

Singleton

TabBarController

- AuthManager

Usage of .shared



The screenshot shows the Xcode interface with the project navigation bar at the top. The project name is "XuConnieFinalProject". Below it, the file structure is shown:

- XuConnieFinalProject (selected)
- XuConnieFinalProject (Info)
- Model
 - User
 - BlogPost
 - IAPManager
 - DatabaseManager
 - AuthManager
 - StorageManager
 - HapticsManager
- View
- Controller
 - BlogsViewController
 - ProfileViewController
 - TabBarController (highlighted with a pink rectangle)
 - SignInViewController
 - SignUpViewController
 - ViewPostViewController
 - CreateNewPostViewController
 - PayWallViewController
- Supporting Files
- XuConnieFinalProjectUITests
- Products
- Frameworks
- Pods
- Pods (selected)

The right pane displays the code for `TabBarController.swift`:

```
1 //  
2 // TabBarController.swift  
3 // XuConnieFinalProject  
4 //  
5 // Created by Connie Xu on 11/13/21.  
6 //  
7  
8 import UIKit  
9  
10 class TabBarController: UITabBarController {  
11  
12     override func viewDidLoad() {  
13         super.viewDidLoad()  
14  
15         setUpControllers()  
16     }  
17  
18     //sets up views of main pages (blogs page + profile page)  
19     private func setUpControllers() {  
20         guard let currentUserEmail = UserDefaults.standard.string(forKey: "email") else {  
21             AuthManager.shared.signOut { _ in  
22                 return  
23             }  
24         }  
25  
26         let blogs = BlogsViewController()  
27         blogs.title = "Blogs"  
28  
29         let profile = ProfileViewController(currentEmail: currentUserEmail)  
30         profile.title = "Profile"  
31  
32         blogs.navigationItem.largeTitleDisplayMode = .always  
33         profile.navigationItem.largeTitleDisplayMode = .always  
34  
35         let nav1 = UINavigationController(rootViewController: blogs)  
36         let nav2 = UINavigationController(rootViewController: profile)  
37  
38         nav1.navigationBar.prefersLargeTitles = true  
39         nav2.navigationBar.prefersLargeTitles = true  
40  
41         nav1.tabBarItem = UITabBarItem(title: "Blogs", image: UIImage(systemName: "house"), tag: 1)  
42         nav2.tabBarItem = UITabBarItem(title: "Profile", image: UIImage(systemName: "person-circle"), tag: 2)  
43 }
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

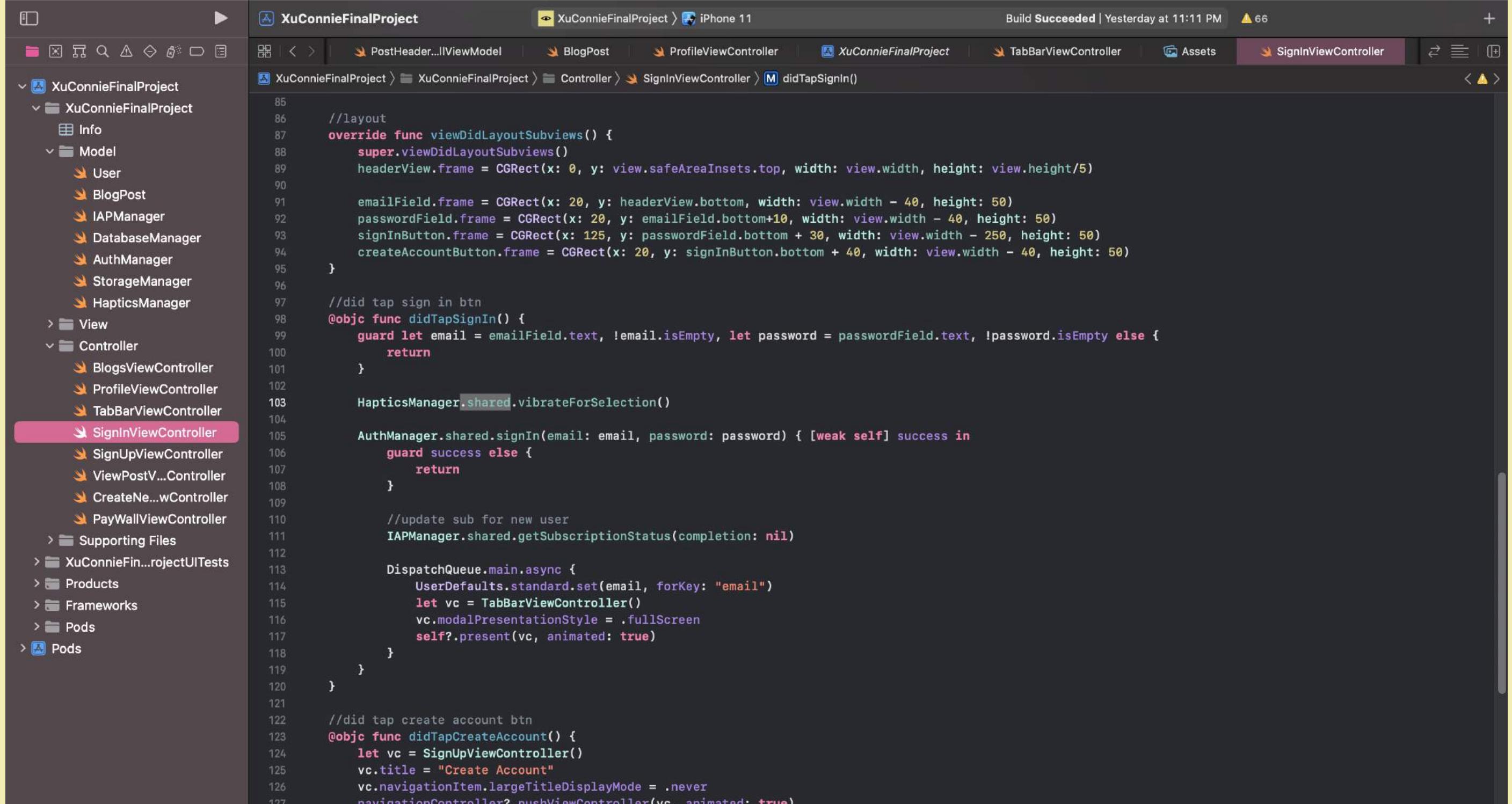
PROJECT SPECS

Singleton

SignInViewController

- HapticsManager
- AuthManager
- IAPManager

Usage of .shared



The screenshot shows the Xcode interface with the project navigation bar at the top. The left sidebar displays the project structure under 'XuConnieFinalProject'. A pink rectangular highlight is placed over the 'SignInViewController' entry in the 'Controller' folder. The main editor area shows a Swift file named 'SignInViewController.swift'. The code within the file demonstrates the use of the '.shared' property of the 'HapticsManager' class. Specifically, it calls `HapticsManager.shared.vibrateForSelection()`. The code also handles user authentication and subscription status updates.

```
85 //layout
86 override func viewDidLoad() {
87     super.viewDidLoad()
88     headerView.frame = CGRect(x: 0, y: view.safeAreaInsets.top, width: view.width, height: view.height/5)
89
90     emailField.frame = CGRect(x: 20, y: headerView.bottomAnchor, width: view.width - 40, height: 50)
91     passwordField.frame = CGRect(x: 20, y: emailField.bottomAnchor+10, width: view.width - 40, height: 50)
92     signInButton.frame = CGRect(x: 125, y: passwordField.bottomAnchor + 30, width: view.width - 250, height: 50)
93     createAccountButton.frame = CGRect(x: 20, y: signInButton.bottomAnchor + 40, width: view.width - 40, height: 50)
94 }
95
96 //did tap sign in btn
97 @objc func didTapSignIn() {
98     guard let email = emailField.text, !email.isEmpty, let password = passwordField.text, !password.isEmpty else {
99         return
100    }
101
102    HapticsManager.shared.vibrateForSelection()
103
104    AuthManager.shared.signIn(email: email, password: password) { [weak self] success in
105        guard success else {
106            return
107        }
108
109        //update sub for new user
110        IAPManager.shared.getSubscriptionStatus(completion: nil)
111
112        DispatchQueue.main.async {
113            UserDefaults.standard.set(email, forKey: "email")
114            let vc = TabBarController()
115            vc.modalPresentationStyle = .fullScreen
116            self?.present(vc, animated: true)
117        }
118    }
119 }
120
121 //did tap create account btn
122 @objc func didTapCreateAccount() {
123     let vc = SignUpViewController()
124     vc.title = "Create Account"
125     vc.navigationItem.largeTitleDisplayMode = .never
126     navigationController?.pushViewController(vc, animated: true)
127 }
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

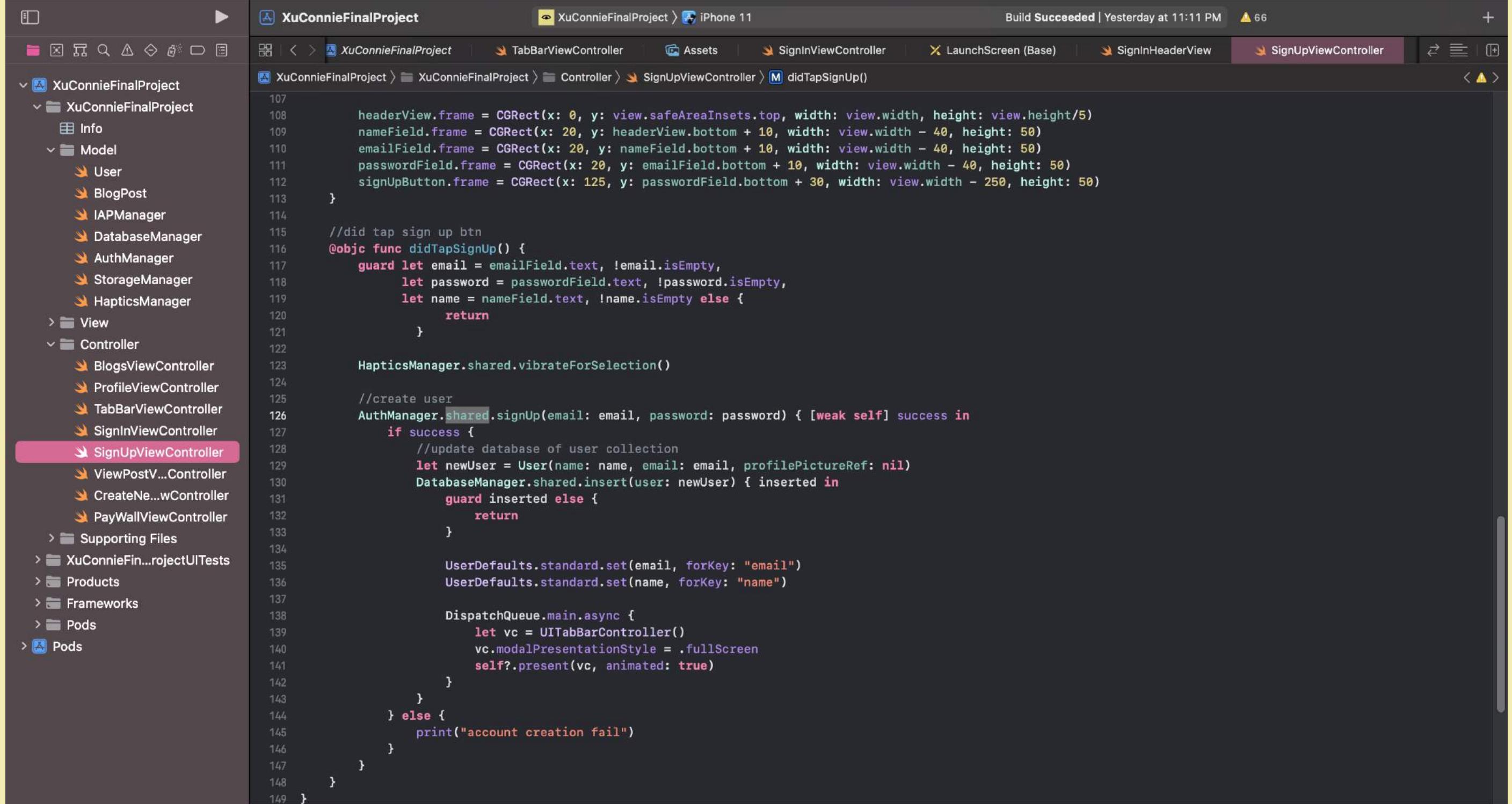
PROJECT SPECS

Singleton

SignUpViewController

- HapticsManager
- AuthManager
- IAPManager

Usage of .shared



The screenshot shows the Xcode interface with the project navigation bar at the top. The left sidebar displays the project structure under 'XuConnieFinalProject'. A pink box highlights the 'SignUpViewController' file in the 'Controller' folder. The right pane shows the code for the 'didTapSignUp()' method in 'SignUpViewController.swift'. The code uses 'AuthManager.shared' to sign up a user, update the database, and present a UITabBarController.

```
headerView.frame = CGRect(x: 0, y: view.safeAreaInsets.top, width: view.width, height: view.height/5)
nameField.frame = CGRect(x: 20, y: headerView.bottom + 10, width: view.width - 40, height: 50)
emailField.frame = CGRect(x: 20, y: nameField.bottom + 10, width: view.width - 40, height: 50)
passwordField.frame = CGRect(x: 20, y: emailField.bottom + 10, width: view.width - 40, height: 50)
signUpButton.frame = CGRect(x: 125, y: passwordField.bottom + 30, width: view.width - 250, height: 50)

//did tap sign up btn
@objc func didTapSignUp() {
    guard let email = emailField.text, !email.isEmpty,
          let password = passwordField.text, !password.isEmpty,
          let name = nameField.text, !name.isEmpty else {
        return
    }

    HapticsManager.shared.vibrateForSelection()

    //create user
    AuthManager.shared.signUp(email: email, password: password) { [weak self] success in
        if success {
            //update database of user collection
            let newUser = User(name: name, email: email, profilePictureRef: nil)
            DatabaseManager.shared.insert(user: newUser) { inserted in
                guard inserted else {
                    return
                }

                UserDefaults.standard.set(email, forKey: "email")
                UserDefaults.standard.set(name, forKey: "name")

                DispatchQueue.main.async {
                    let vc = UITabBarController()
                    vc.modalPresentationStyle = .fullScreen
                    self?.present(vc, animated: true)
                }
            }
        } else {
            print("account creation fail")
        }
    }
}

```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

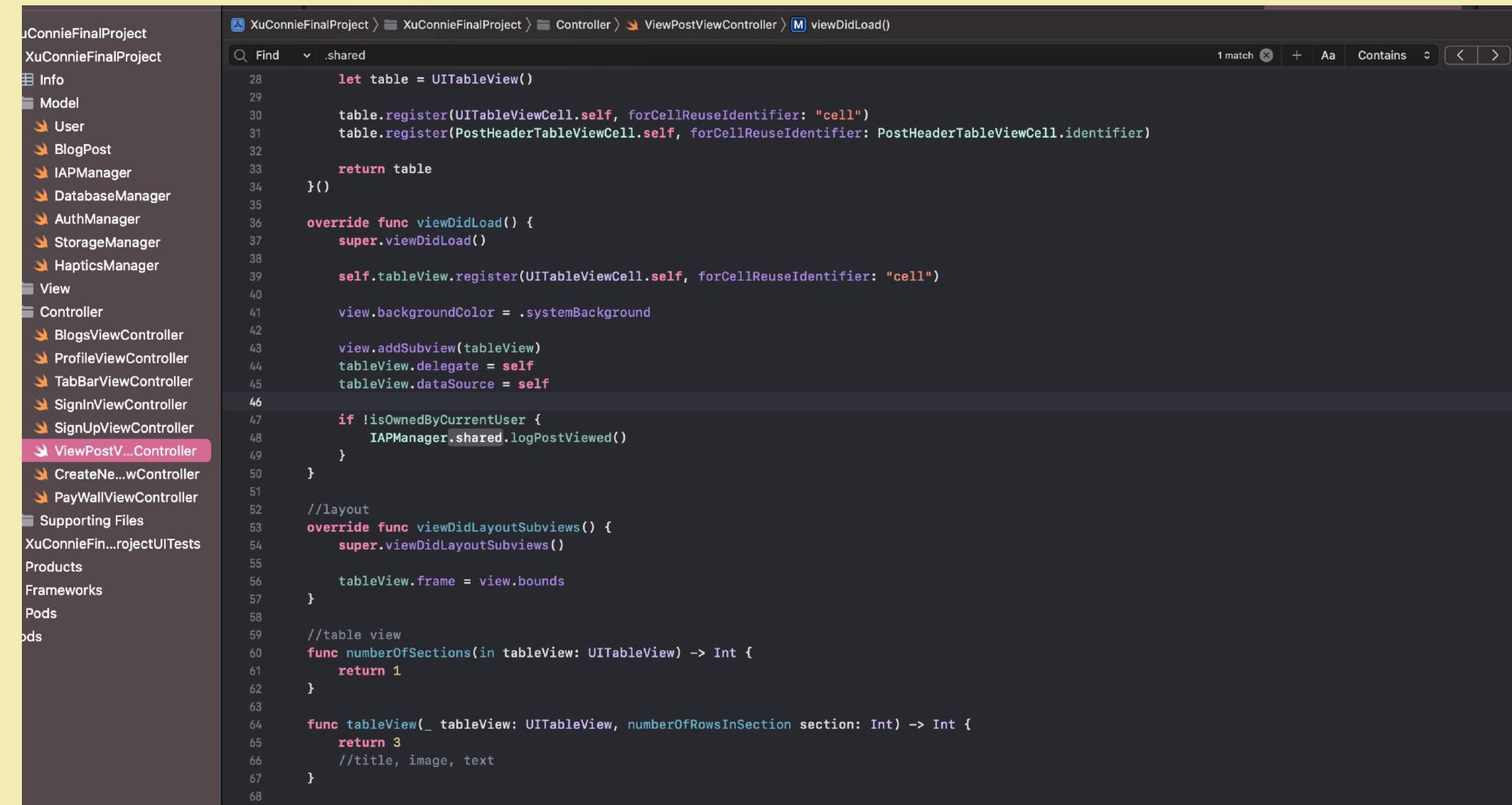
PROJECT SPECS

Singleton

ViewPostViewController

- IAPManager

Usage of .shared



The screenshot shows the Xcode interface with the project navigation bar at the top. Below it is a tree view of the project structure:

- XuConnieFinalProject
- XuConnieFinalProject
- Info
- Model
 - User
 - BlogPost
 - IAPManager
 - DatabaseManager
 - AuthManager
 - StorageManager
 - HapticsManager
- View
- Controller
 - BlogsViewController
 - ProfileViewController
 - TabBarViewController
 - SignInViewController
 - SignUpViewController
 - ViewPostV...Controller
 - CreateNe...wController
 - PayWallViewController
- Supporting Files
- XuConnieFin...rojectUITests
- Products
- Frameworks
- Pods
- Pods

The right side of the screen displays the code for `ViewPostViewController.swift`. The line `IAPManager.shared.logPostViewed()` is highlighted with a pink rectangle. The code is as follows:

```
let table = UITableView()
table.register(UITableViewCell.self, forCellReuseIdentifier: "cell")
table.register(PostHeaderTableViewCell.self, forCellReuseIdentifier: PostHeaderTableViewCell.identifier)
return table
}

override func viewDidLoad() {
    super.viewDidLoad()

    self.tableView.register(UITableViewCell.self, forCellReuseIdentifier: "cell")

    view.backgroundColor = .systemBackground
    view.addSubview(tableView)
    tableView.delegate = self
    tableView.dataSource = self

    if !isOwnedByCurrentUser {
        IAPManager.shared.logPostViewed()
    }
}

//layout
override func viewDidLayoutSubviews() {
    super.viewDidLayoutSubviews()

    tableView.frame = view.bounds
}

//table view
func numberOfSections(in tableView: UITableView) -> Int {
    return 1
}

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return 3
    //title, image, text
}
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

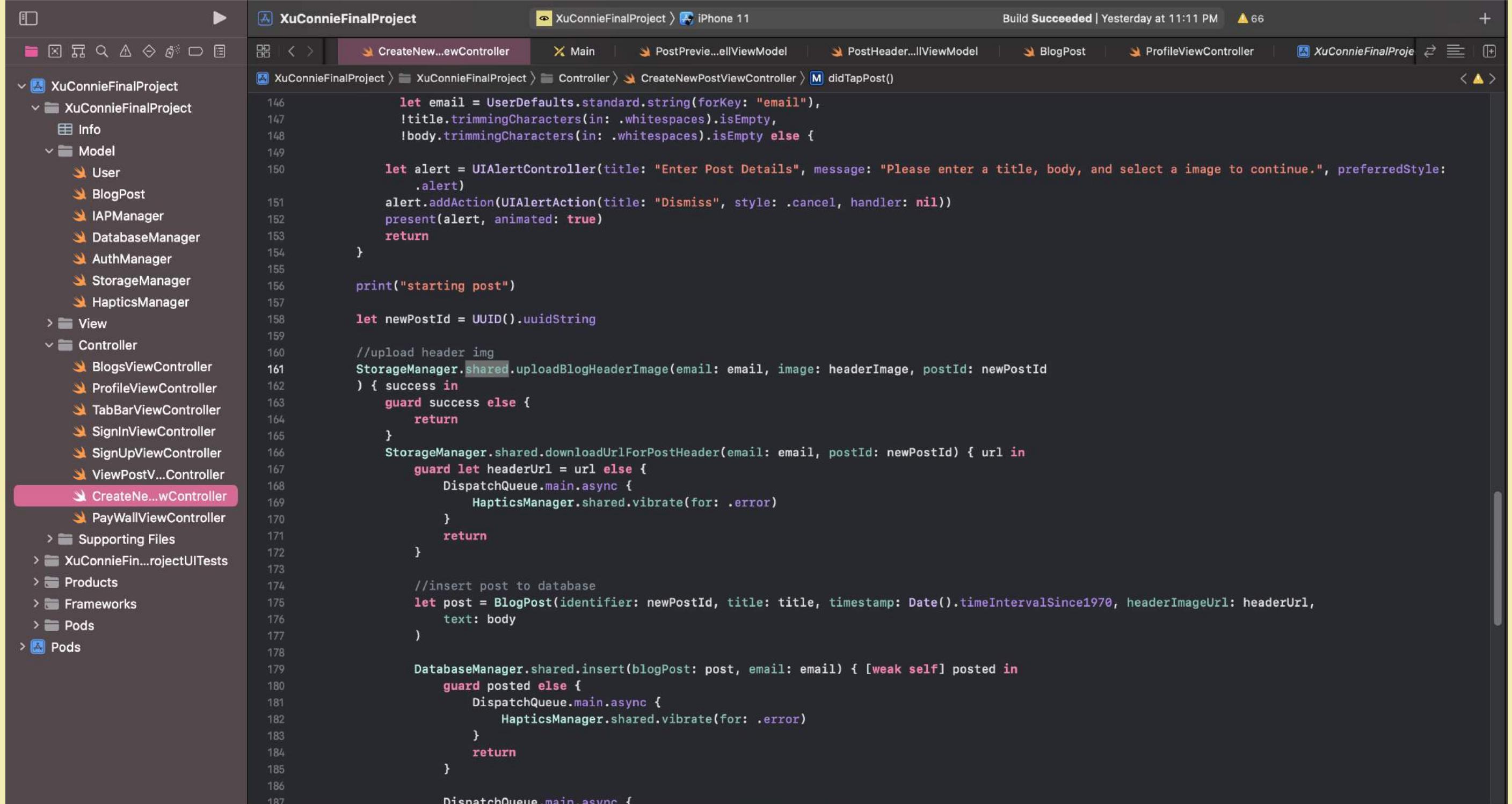
PROJECT SPECS

Singleton

CreateNewPostViewController

- StorageManager
- HapticsManager
- DatabaseManager

Usage of .shared



The screenshot shows the Xcode interface with the project navigation bar at the top. Below it is the file browser showing the project structure. A pink box highlights the 'CreateNewPostViewController' file under the 'Controller' folder. To the right is the code editor displaying the Swift code for this controller. The code uses several shared managers from the 'Model' folder.

```
146 let email = UserDefaults.standard.string(forKey: "email"),
147 !title.trimmingCharacters(in: .whitespaces).isEmpty,
148 !body.trimmingCharacters(in: .whitespaces).isEmpty else {
149
150     let alert = UIAlertController(title: "Enter Post Details", message: "Please enter a title, body, and select a image to continue.", preferredStyle:
151         .alert)
152     alert.addAction(UIAlertAction(title: "Dismiss", style: .cancel, handler: nil))
153     present(alert, animated: true)
154     return
155 }
156 print("starting post")
157
158 let newPostId = UUID().uuidString
159
160 //upload header img
161 StorageManager.shared.uploadBlogHeaderImage(email: email, image: headerImage, postId: newPostId
162 ) { success in
163     guard success else {
164         return
165     }
166     StorageManager.shared.downloadUrlForPostHeader(email: email, postId: newPostId) { url in
167         guard let headerUrl = url else {
168             DispatchQueue.main.async {
169                 HapticsManager.shared.vibrate(for: .error)
170             }
171             return
172         }
173
174         //insert post to database
175         let post = BlogPost(identifier: newPostId, title: title, timestamp: Date().timeIntervalSince1970, headerImageUrl: headerUrl,
176                             text: body
177         )
178
179         DatabaseManager.shared.insert(blogPost: post, email: email) { [weak self] posted in
180             guard posted else {
181                 DispatchQueue.main.async {
182                     HapticsManager.shared.vibrate(for: .error)
183                 }
184             }
185         }
186     }
187     DispatchQueue.main.async {
188 }
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

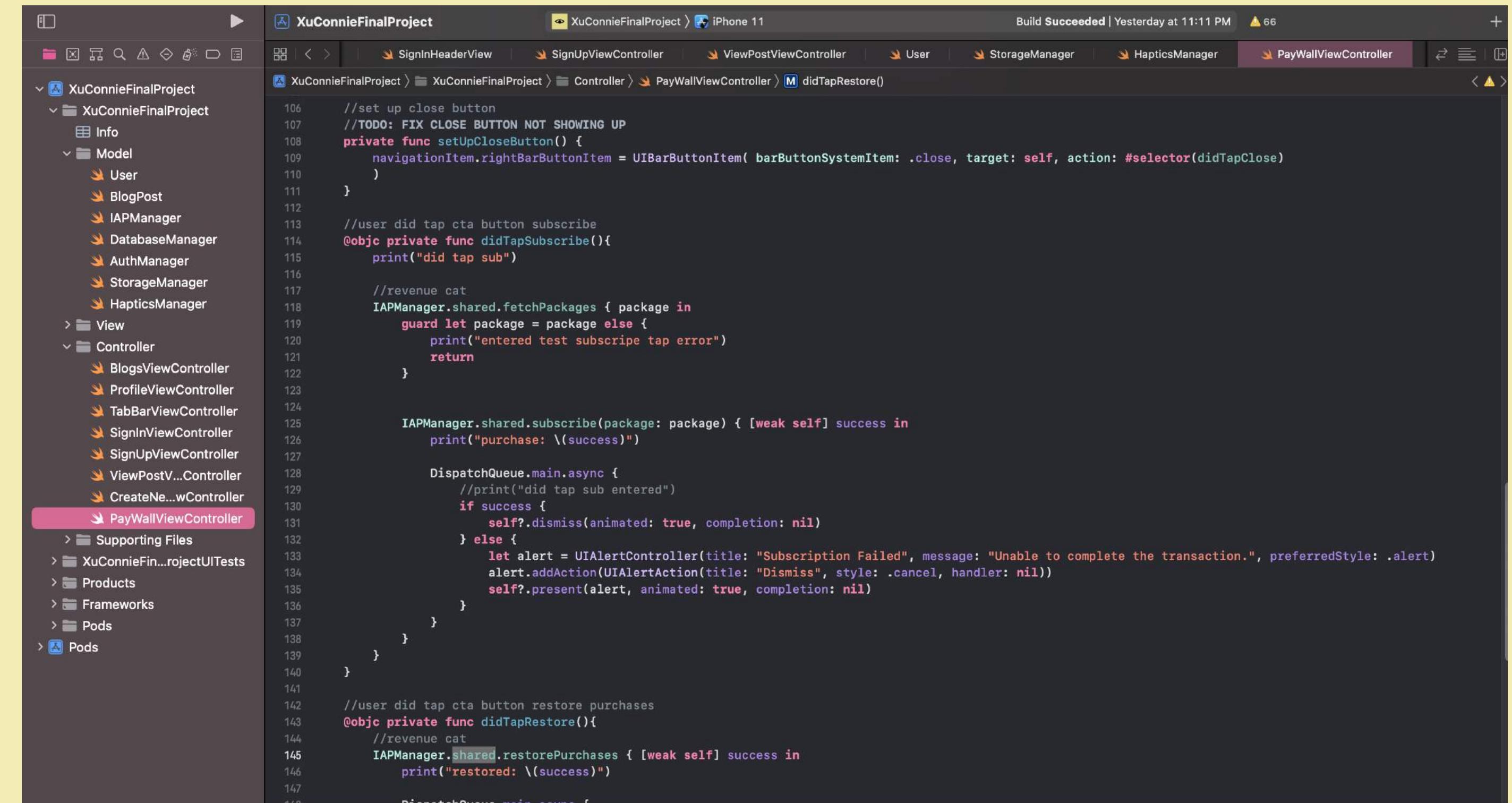
PROJECT SPECS

Singleton

PayWallViewController

- IAPManager

Usage of .shared



The screenshot shows the Xcode interface with the project navigation bar at the top. Below it is a tree view of the project structure under 'XuConnieFinalProject'. A pink box highlights 'PayWallViewController' in the 'Controller' folder. To the right is a code editor window showing 'PayWallViewController.swift'. The code uses the shared instance of IAPManager to handle purchases and restores.

```
106 //set up close button
107 //TODO: FIX CLOSE BUTTON NOT SHOWING UP
108 private func setUpCloseButton() {
109     navigationItem.rightBarButtonItem = UIBarButtonItem(barButtonSystemItem: .close, target: self, action: #selector(didTapClose))
110 }
111
112 //user did tap cta button subscribe
113 @objc private func didTapSubscribe(){
114     print("did tap sub")
115
116     //revenue cat
117     IAPManager.shared.fetchPackages { package in
118         guard let package = package else {
119             print("entered test subscribe tap error")
120             return
121         }
122
123         IAPManager.shared.subscribe(package: package) {[weak self] success in
124             print("purchase: \(success)")
125
126             DispatchQueue.main.async {
127                 //print("did tap sub entered")
128                 if success {
129                     self?.dismiss(animated: true, completion: nil)
130                 } else {
131                     let alert = UIAlertController(title: "Subscription Failed", message: "Unable to complete the transaction.", preferredStyle: .alert)
132                     alert.addAction(UIAlertAction(title: "Dismiss", style: .cancel, handler: nil))
133                     self?.present(alert, animated: true, completion: nil)
134                 }
135             }
136         }
137     }
138 }
139
140
141 //user did tap cta button restore purchases
142 @objc private func didTapRestore(){
143     //revenue cat
144     IAPManager.shared.restorePurchases {[weak self] success in
145         print("restored: \(success)")
146
147         DispatchQueue.main.async {
148             self?.dismiss(animated: true, completion: nil)
149         }
150     }
151 }
```

> XuConnieFinalProject > Controller > BlogPostViewController.swift
> ProfileViewController.swift
> TabBarViewController.swift
> SignInViewController.swift
> SignUpViewController.swift
> ViewPostViewController.swift
> CreateNewPostViewController.swift
> PayWallViewController.swift

PROJECT SPECS

Third Party APIs/Frameworks

Firebase

- Authentication database
- Persistent storage
 - Storage of user information
 - Storage of blog post information

The screenshot shows the Firebase console interface. On the left, there's a sidebar with icons for Home, Settings, Users, and Storage. The main area has tabs for 'Users', 'Sign-in method', 'Templates', and 'Usage'. A banner at the top says 'Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication' with a 'Get started' button. Below this, the 'Users' section lists three accounts: 'bge@gmail.com' (Created Dec 6, 2021, Signed In Dec 6, 2021, User UID I4W9H0w1XPftVggnYYelceActm63), 'conniexu@gmail.com' (Created Dec 6, 2021, Signed In Dec 6, 2021, User UID I4W9H0w1XPftVggnYYelceActm63), and 'conniexutest1@gmail.com' (Created Dec 6, 2021, Signed In Dec 6, 2021, User UID I4W9H0w1XPftVggnYYelceActm63). To the right, the 'Storage' section shows a file tree for 'README.ios-a42d9.appspot.com' with 'posts_headers/' and 'profile_pictures/' folders. At the bottom, there's a code editor for 'AppDelegate.swift'.

The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar shows the project structure with 'XuConnieFinalProject' expanded, showing 'Info', 'Model', 'View', 'Controller', 'Supporting Files', 'Extensions', 'AppDelegate' (which is selected), 'SceneDelegate', 'XuConnieFinalProjectTests', 'Assets', 'Configuration', 'StoreKitTestCertificate', 'GoogleService-Info', 'XuConnieFinalProjectUITests', 'Products', 'Frameworks', 'Pods', and 'Pods'. The right pane shows the 'AppDelegate.swift' file with the following code:

```
1 //  
2 // AppDelegate.swift  
3 // XuConnieFinalProject  
4 // Created by Connie Xu on 11/13/21.  
5 //  
6 //  
8 import Purchases  
9 import Firebase  
10 import UIKit  
12 @main  
13 class AppDelegate: UIResponder, UIApplicationDelegate {  
14  
15     func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {  
16  
17         //add firebase to app  
18         FirebaseApp.configure()  
19  
20         //initialize sdk from revenue cat  
21         Purchases.configure(withAPIKey: "dFwDMpBkRRystTZjKvWdtDRsEuozhWU")  
22  
23         //call IAPManager funcs to make sure they work  
24         IAPManager.shared.getSubscriptionStatus(completion: nil)  
25  
26         return true  
27     }  
28  
29     // MARK: UISceneSession Lifecycle  
31     func application(_ application: UIApplication, configurationForConnecting connectingSceneSession: UISceneSession, options: UIScene.ConnectionOptions) ->  
32         UISceneConfiguration {  
33             // Called when a new scene session is being created.  
34             // Use this method to select a configuration to create the new scene with.  
35             return UISceneConfiguration(name: "Default Configuration", sessionRole: connectingSceneSession.role)  
36         }  
37  
38     func application(_ application: UIApplication, didDiscardSceneSessions sceneSessions: Set<UISceneSession>) {  
39             // Called when the user discards a scene session.  
40             // If any sessions were discarded while the application was not running, this will be called shortly after application:didFinishLaunchingWithOptions.  
41             // Use this method to release any resources that were specific to the discarded scenes, as they will not return.  
42         }  
43 }
```

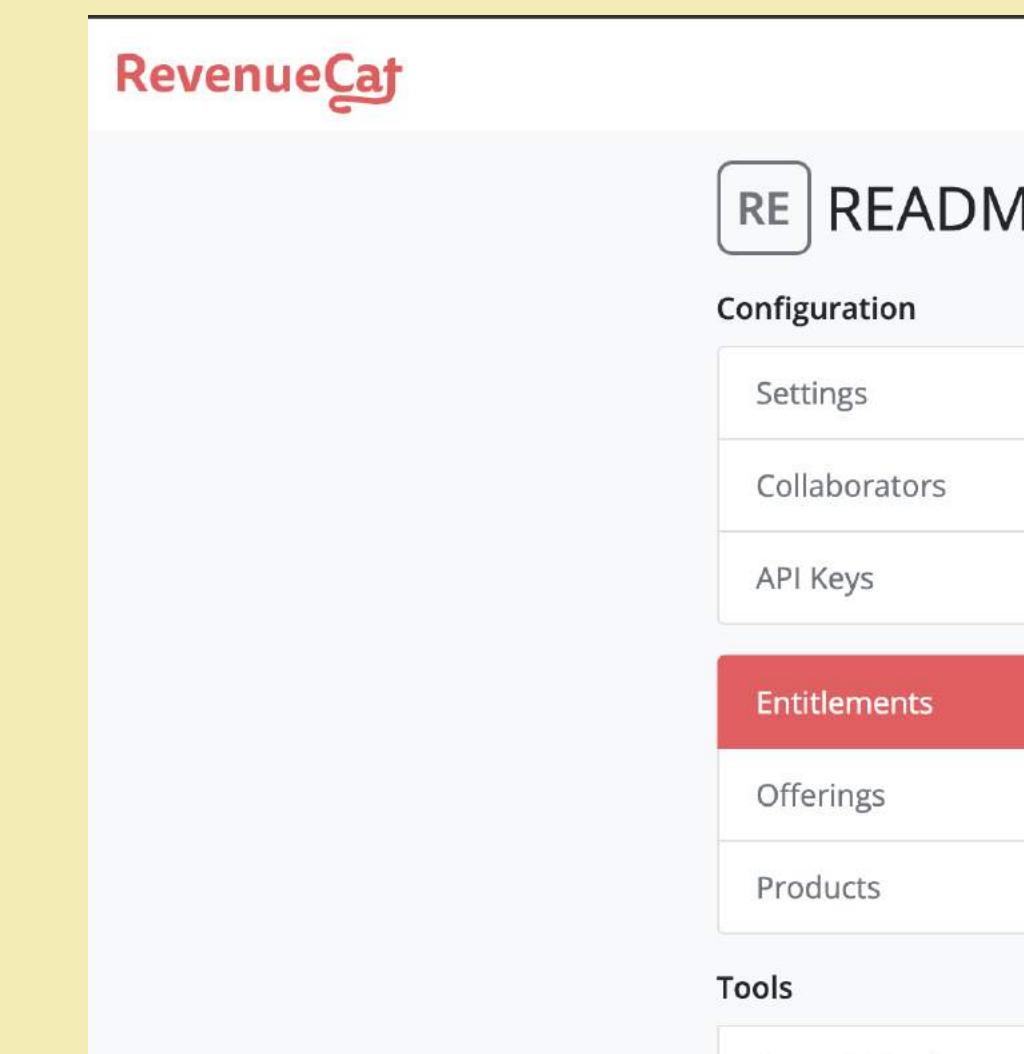
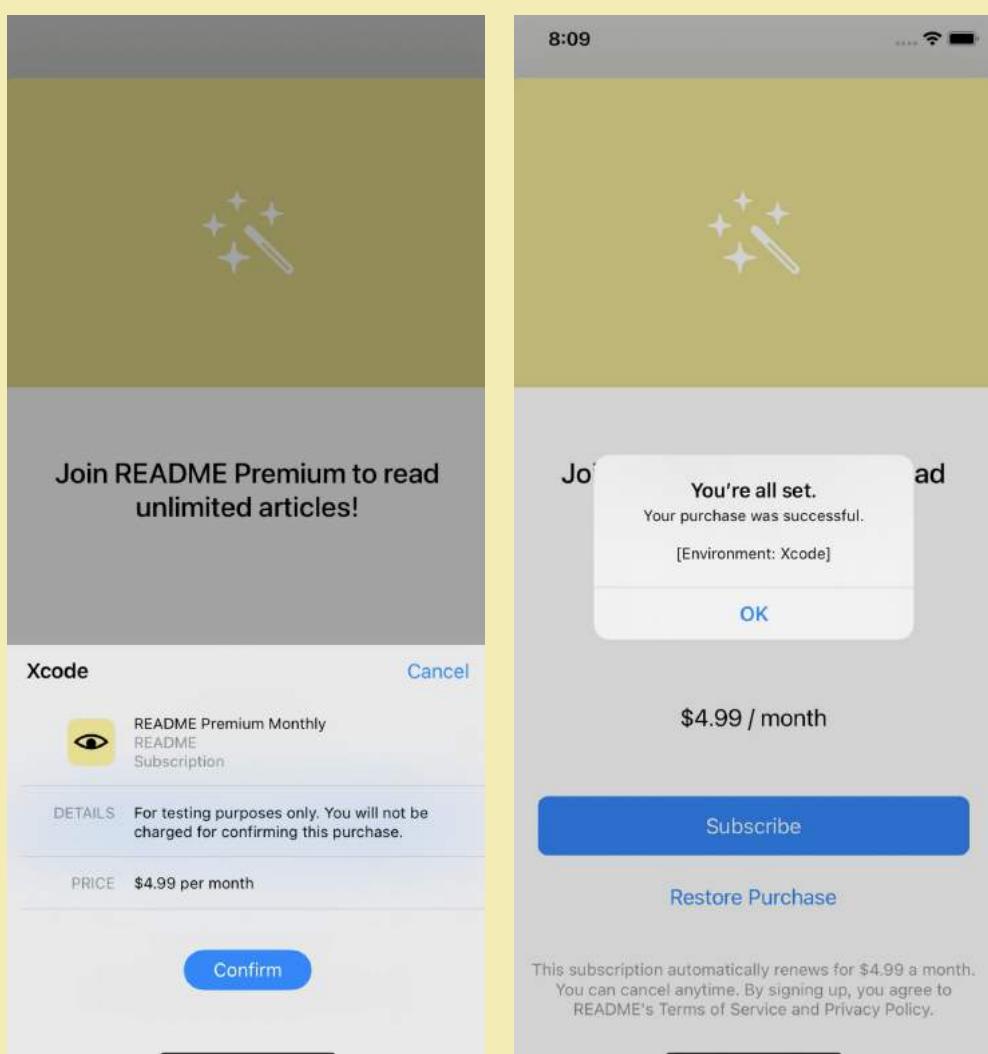
> XuConnieFinalProject > Supporting Files > AppDelegate.swift

PROJECT SPECS

Third Party APIs/Frameworks

RevenueCat + Purchases SDK

- Streamline subscription-based in-app purchase for README Premium



```
1 // AppDelegate.swift
2 // XuConnieFinalProject
3 // XuConnieFinalProject
4 // Created by Connie Xu on 11/13/21.
5 //
6 //
7 import Purchases
8 import Firebase
9 import UIKit
10
11 @main
12 class AppDelegate: UIResponder, UIApplicationDelegate {
13
14     func application(_ application: UIApplication, didFinishLaunchingWithOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
15
16         //add firebase to app
17         FirebaseApp.configure()
18
19         //initialize sdk from revenue cat
20         Purchases.configure(withAPIKey: "dFWDMpBkRRystTZjKvWdtDRsEuozhWU")
21
22         //call IAPManager funcs to make sure they work
23         IAPManager.shared.getSubscriptionStatus(completion: nil)
24
25         return true
26     }
27
28     // MARK: UISceneSession Lifecycle
29
30     func application(_ application: UIApplication, configurationForConnecting connectingSceneSession: UISceneSession, options: UIScene.ConnectionOptions) ->
31     UISceneConfiguration {
32         // Called when a new scene session is being created.
33         // Use this method to select a configuration to create the new scene with.
34         return UISceneConfiguration(name: "Default Configuration", sessionRole: connectingSceneSession.role)
35     }
36
37     func application(_ application: UIApplication, didDiscardSceneSessions sceneSessions: Set<UISceneSession>) {
38         // Called when the user discards a scene session.
39         // If any sessions were discarded while the application was not running, this will be called shortly after application:didFinishLaunchingWithOptions.
40         // Use this method to release any resources that were specific to the discarded scenes, as they will not return.
41     }
42 }
```

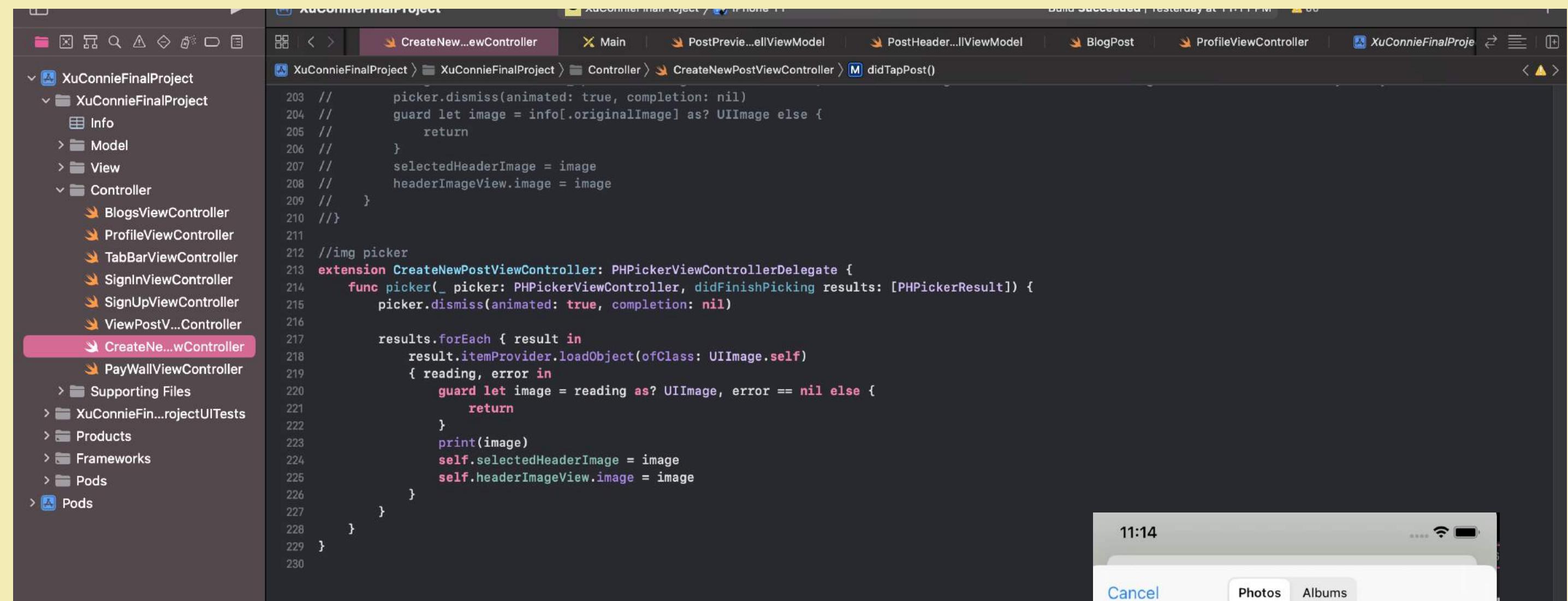
> XuConnieFinalProject > Supporting Files > AppDelegate.swift

PROJECT SPECS

Apple Frameworks

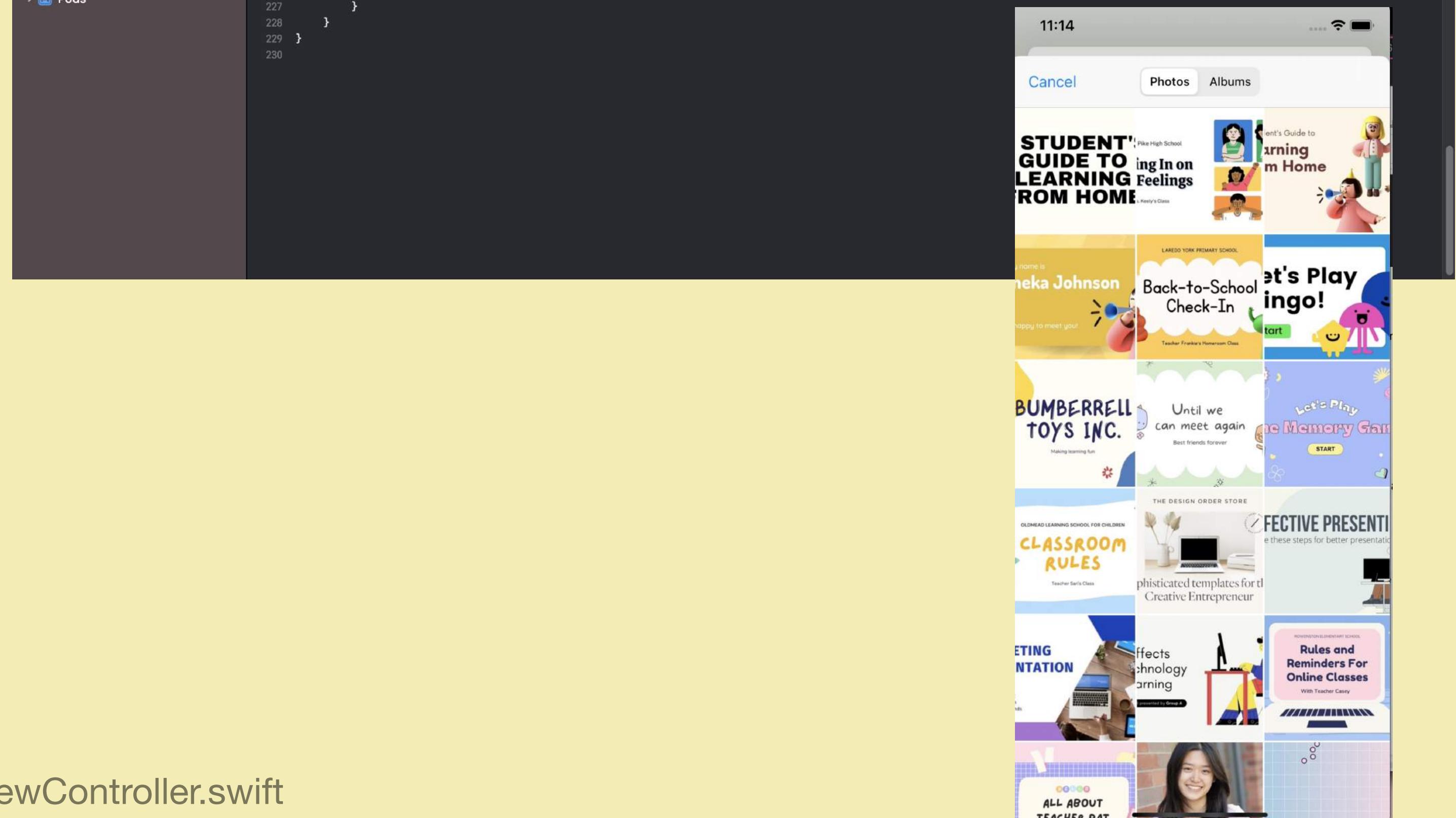
PhotoKit - Photo Upload Method 1

- User select photo from photo library for hero image when creating a new blog post



The screenshot shows the Xcode interface with the project navigation bar at the top. Below it is a tree view of the project structure under 'XuConnieFinalProject'. A file named 'CreateNewPostViewController.swift' is selected and highlighted with a pink background. The code editor on the right contains the implementation of this file, specifically the `didTapPost()` method which handles photo picking.

```
203 // picker.dismiss(animated: true, completion: nil)
204 // guard let image = info[.originalImage] as? UIImage else {
205 //     return
206 // }
207 // selectedHeaderImage = image
208 // headerImageView.image = image
209 //
210 //
211 //img picker
212 extension CreateNewPostViewController: PHPickerViewControllerDelegate {
213     func picker(_ picker: PHPickerViewController, didFinishPicking results: [PHPickerResult]) {
214         picker.dismiss(animated: true, completion: nil)
215
216         results.forEach { result in
217             result.itemProvider.loadObject(ofClass: UIImage.self)
218             { reading, error in
219                 guard let image = reading as? UIImage, error == nil else {
220                     return
221                 }
222                 print(image)
223                 self.selectedHeaderImage = image
224                 self.headerImageView.image = image
225             }
226         }
227     }
228 }
229 }
```



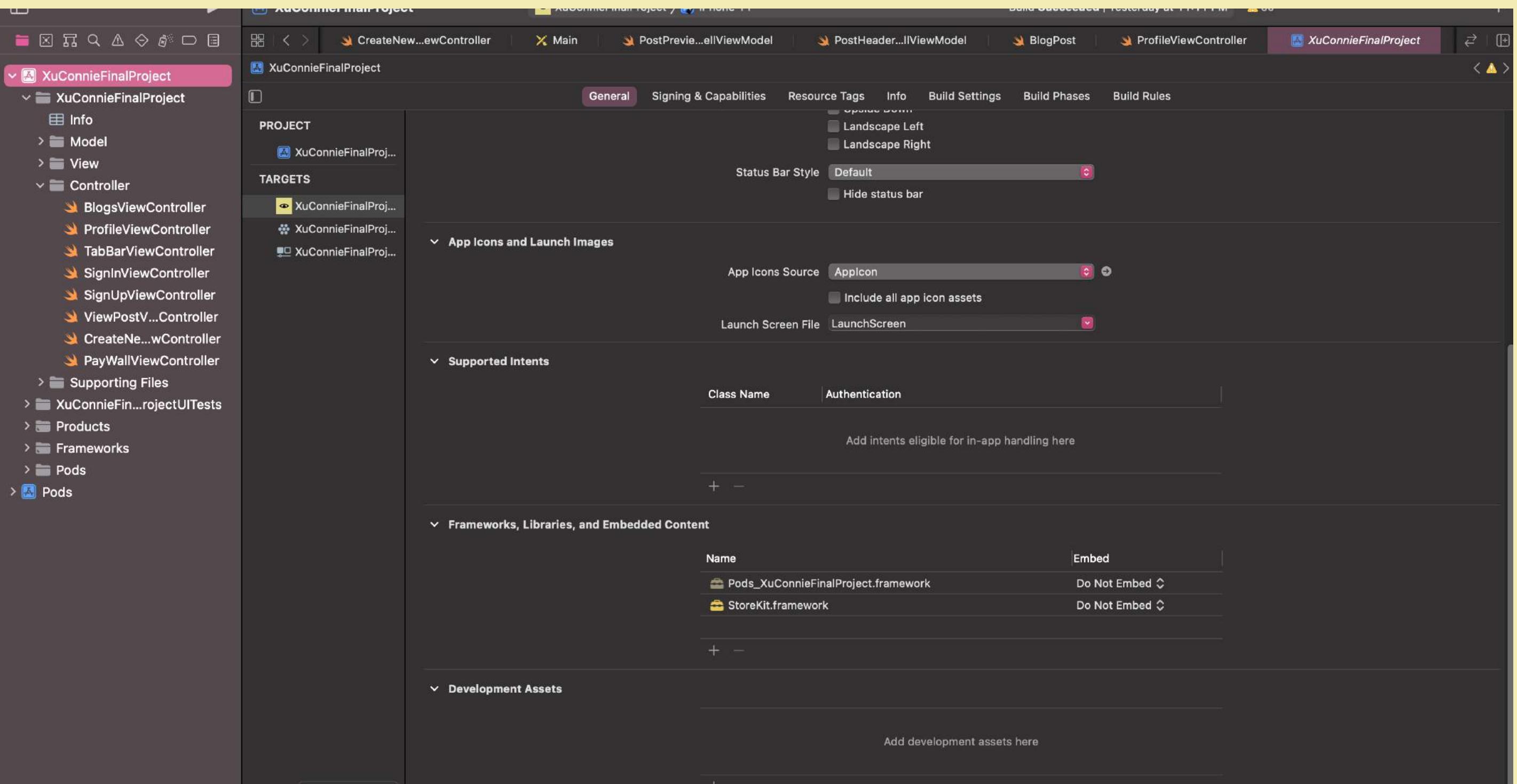
> XuConnieFinalProject > Controller > CreateNewPostViewController.swift

PROJECT SPECS

Apple Frameworks

StoreKit

- For in-app purchase for README Premium subscription



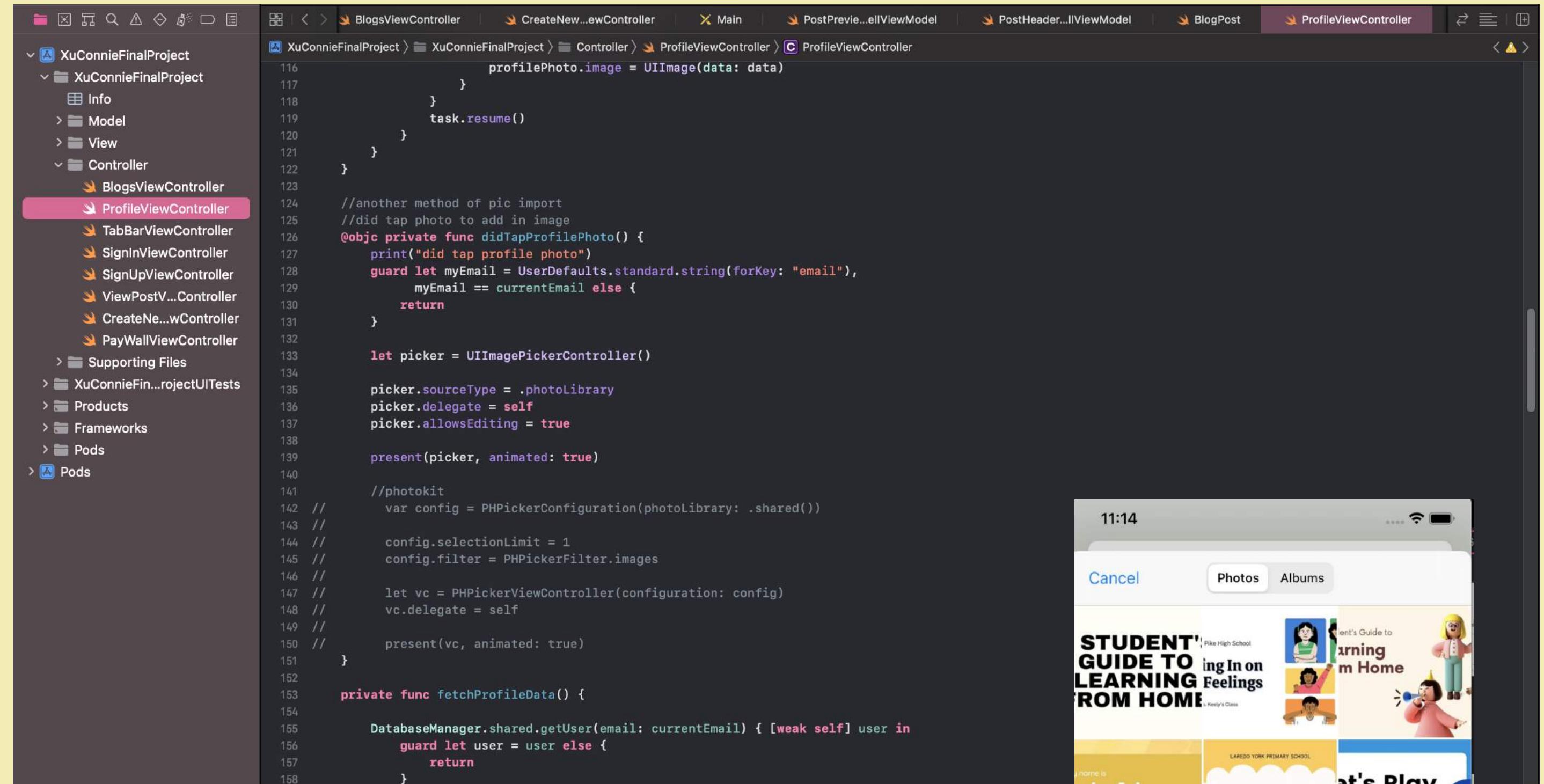
The screenshot shows the 'In-App Purchases' section in App Store Connect for the 'README Blogs' app. It displays a table with one item: 'README Premium' (Type: Non-Renewing Subscription, Product ID: edu.usc.XuConnieFinalProject.prem). A note indicates that the first in-app purchase must be submitted with a new app version.

PROJECT SPECS

Apple Frameworks

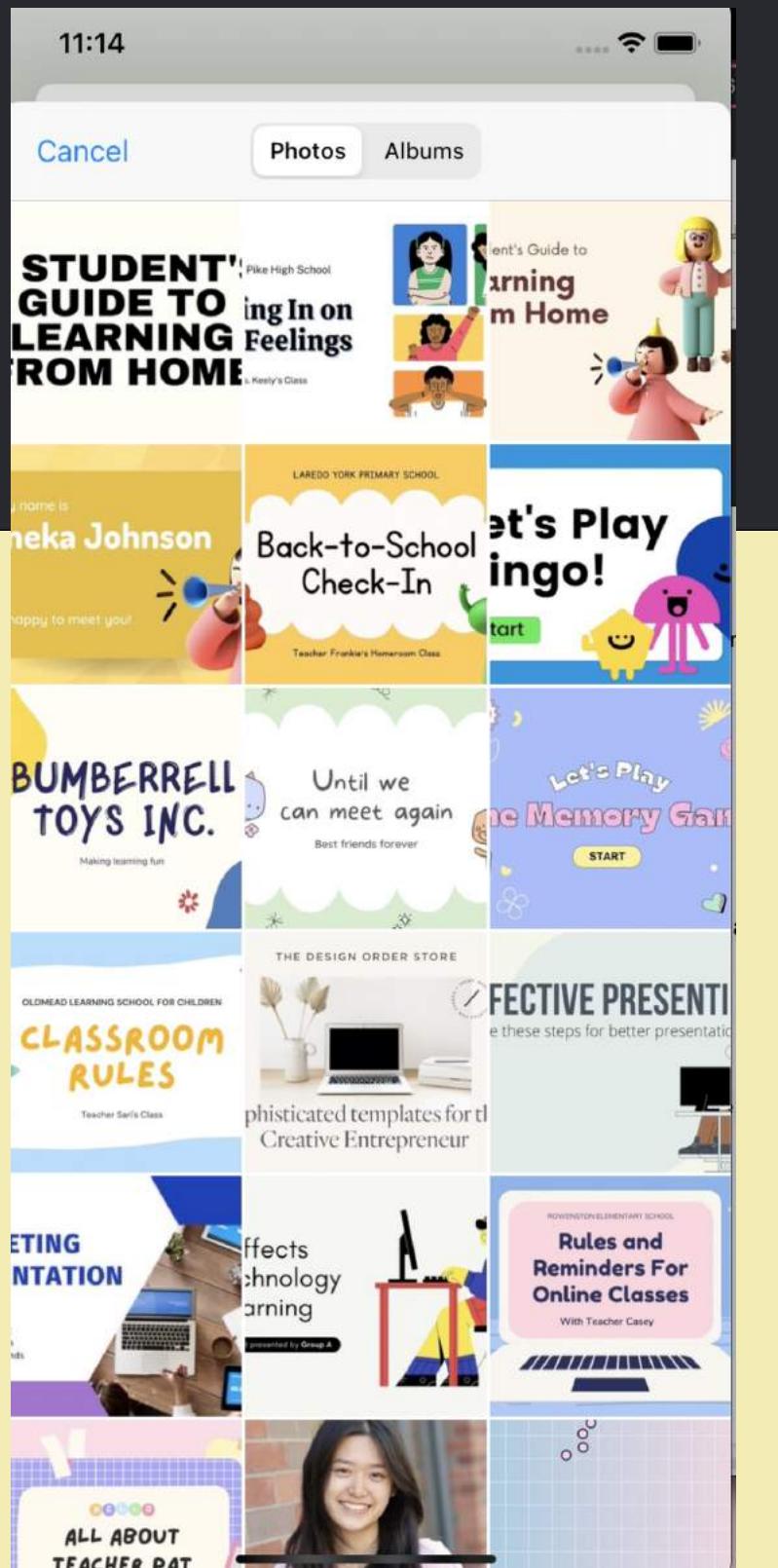
UIKit - Photo Upload Method 2

- User select photo from photo library for profile picture



The screenshot shows the Xcode interface with the Project Navigator on the left and a code editor on the right. The Project Navigator lists files under 'XuConnieFinalProject' including 'Info', 'Model', 'View', 'Controller' (with 'ProfileViewController' selected), 'Supporting Files', and 'Pods'. The code editor displays Swift code for 'ProfileViewController.swift':

```
116         }
117     }
118     task.resume()
119   }
120 }
121
122 //another method of pic import
123 //did tap photo to add in image
124 @objc private func didTapProfilePhoto() {
125   print("did tap profile photo")
126   guard let myEmail = UserDefaults.standard.string(forKey: "email"),
127         myEmail == currentEmail else {
128     return
129   }
130
131   let picker = UIImagePickerController()
132
133   picker.sourceType = .photoLibrary
134   picker.delegate = self
135   picker.allowsEditing = true
136
137   present(picker, animated: true)
138
139 //photokit
140 //  var config = PHPickerConfiguration(photoLibrary: .shared())
141 //  config.selectionLimit = 1
142 //  config.filter = PHPickerFilter.images
143 //
144 //  let vc = PHPickerViewController(configuration: config)
145 //  vc.delegate = self
146 //
147 //  present(vc, animated: true)
148 }
149
150 private func fetchProfileData() {
151
152   DatabaseManager.shared.getUser(email: currentEmail) { [weak self] user in
153     guard let user = user else {
154       return
155     }
156   }
157 }
```



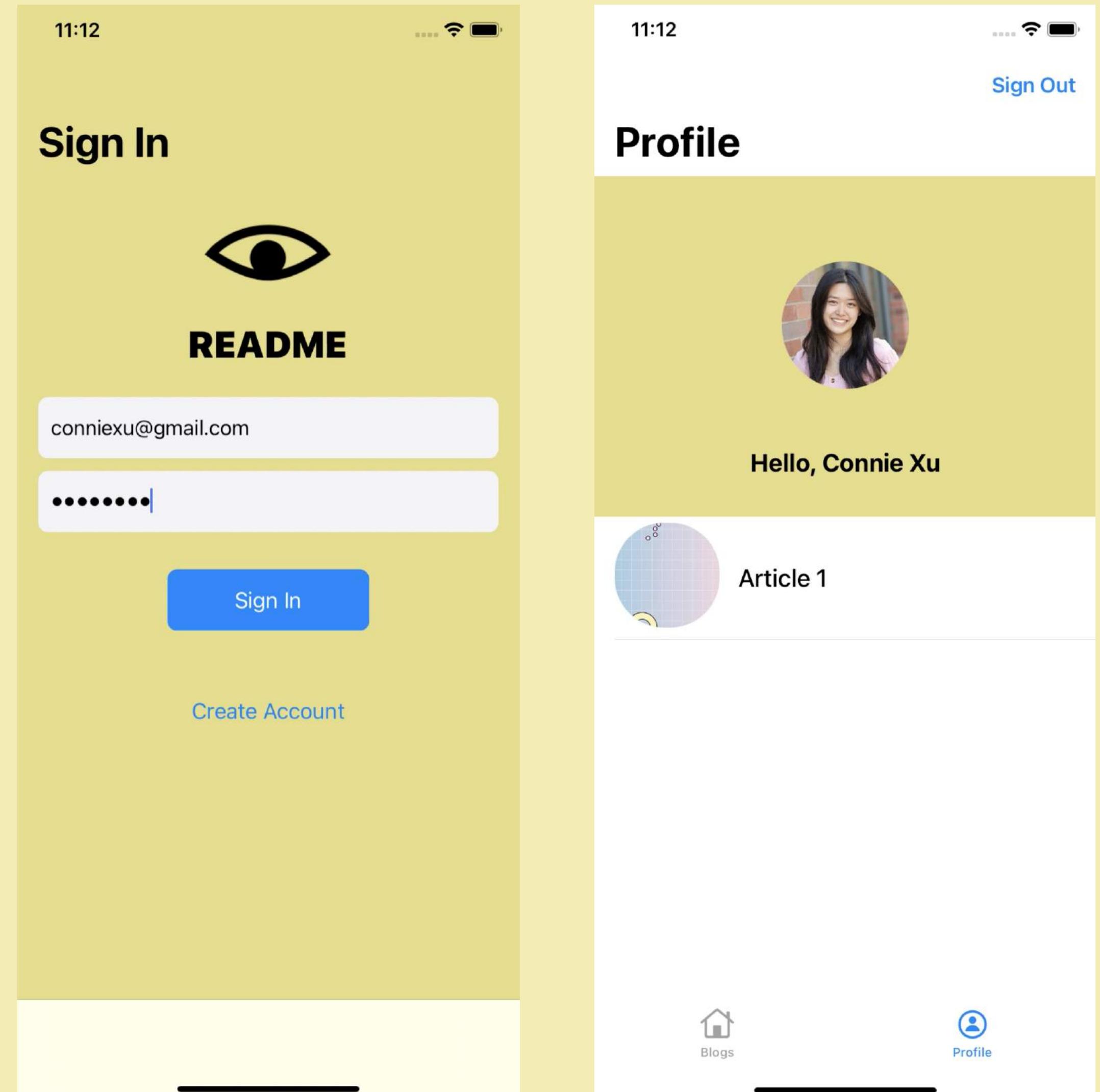
> XuConnieFinalProject > Controller > ProfileViewController.swift

PROJECT SPECS

Apple HIG

Branding

- Incorporate refined, unobtrusive branding
- Defer to content over branding.

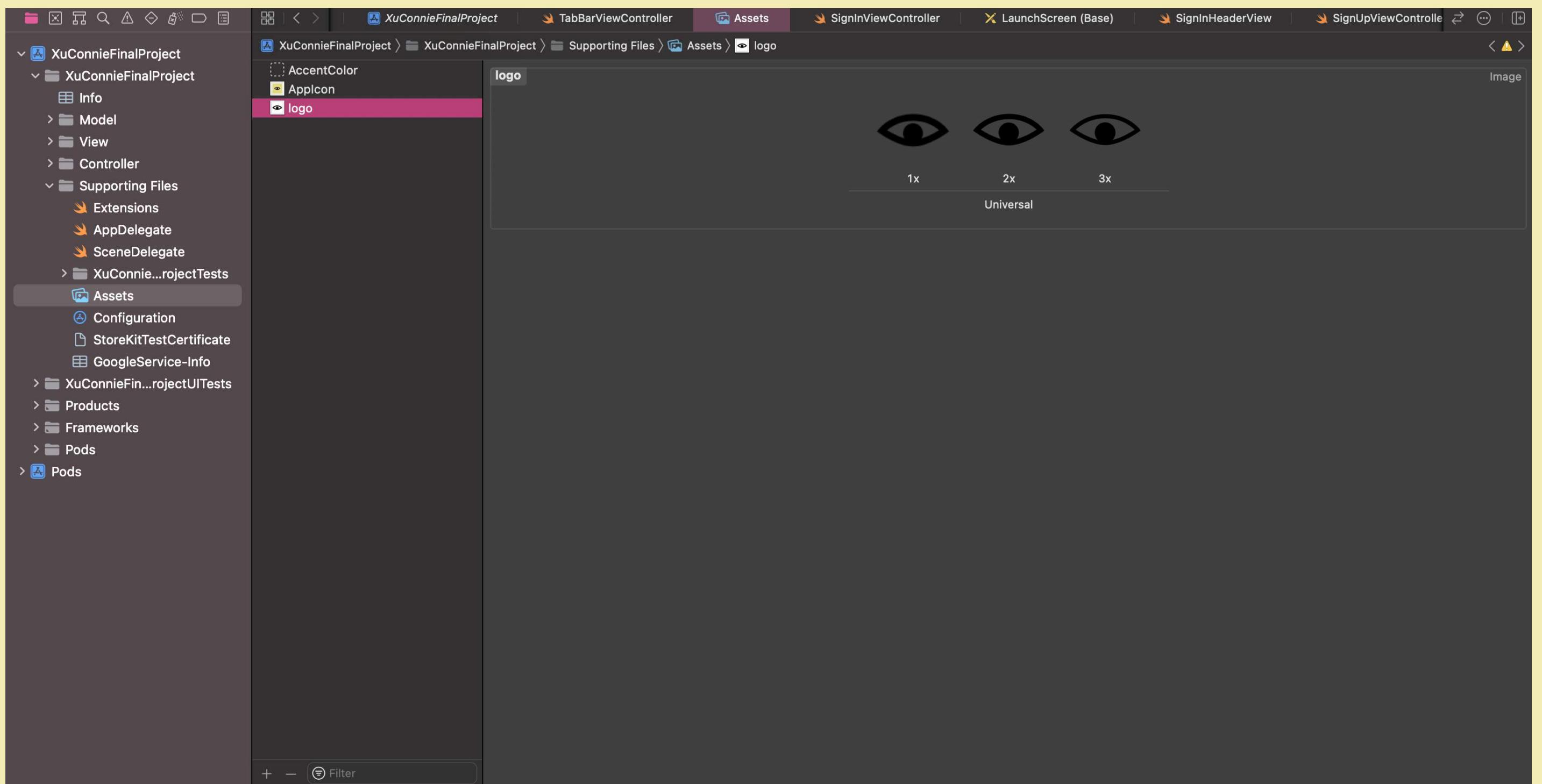


PROJECT SPECS

Apple HIG

Icons & Images

- Supply high-resolution images for all artwork in your app, for all devices your app supports

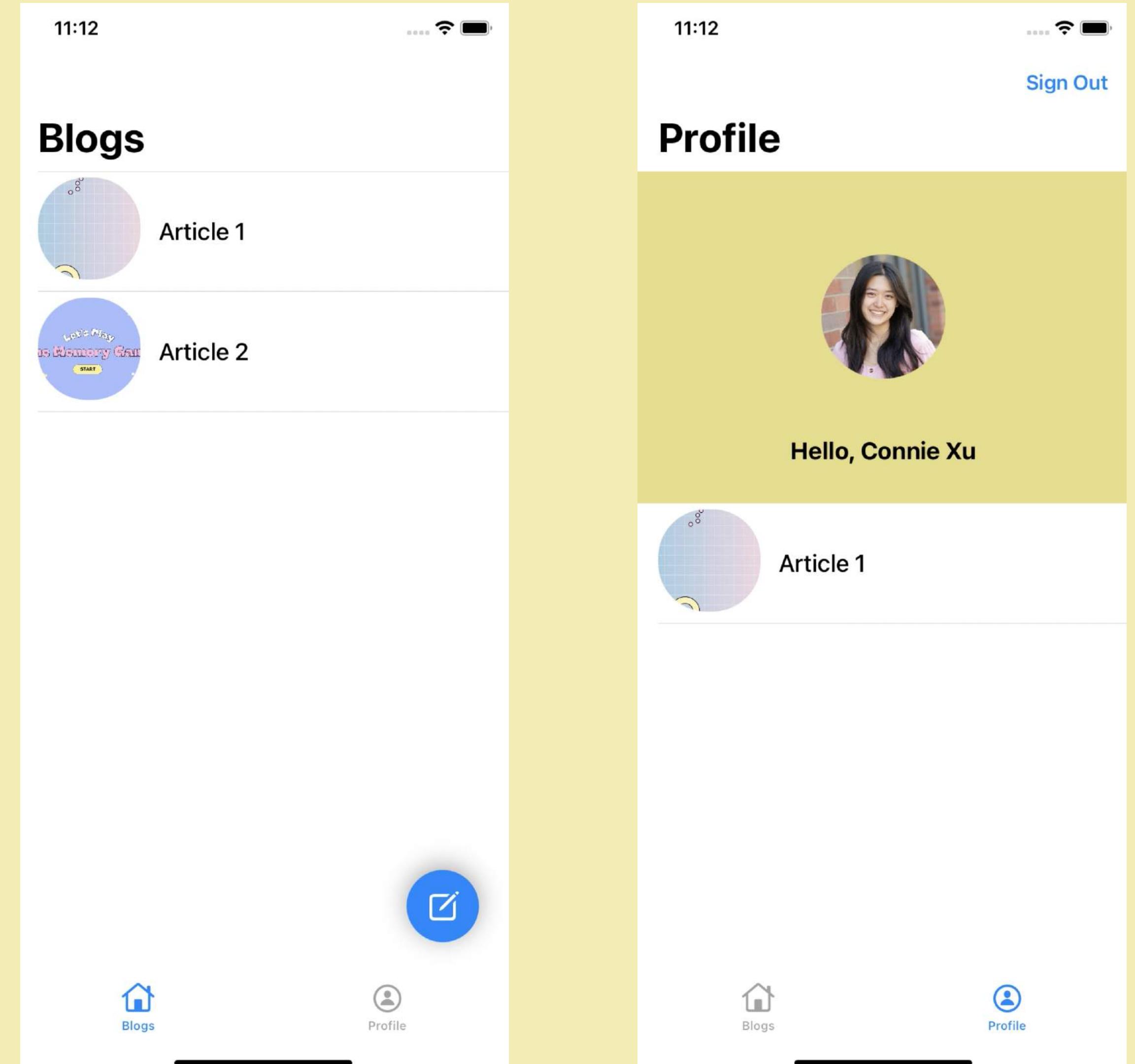


PROJECT SPECS

Apple HIG

Bars - Tab Bar

- Avoid hiding the tab bar when people navigate to different areas in your app



PROJECT SPECS

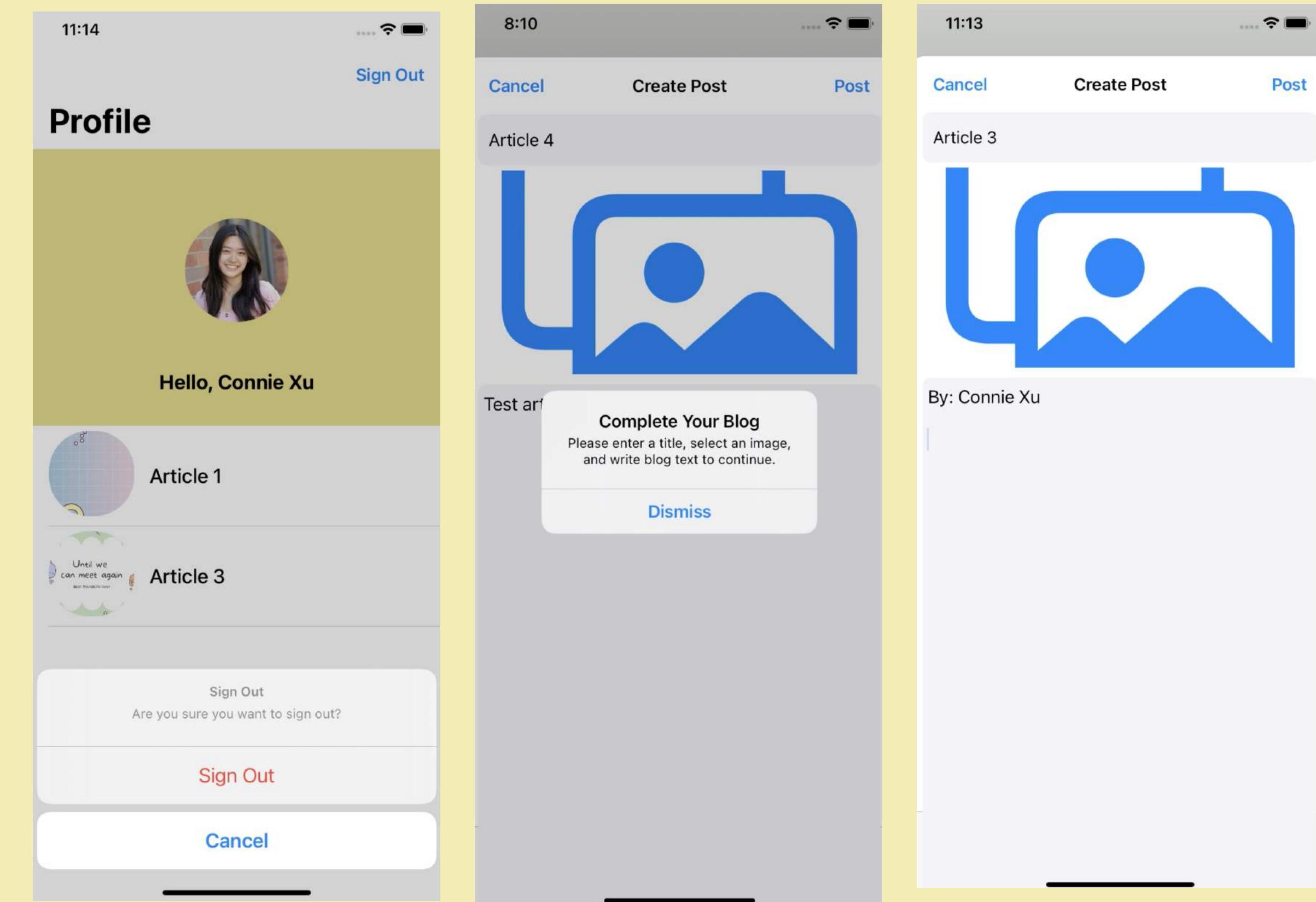
Apple HIG

Views - Action Sheet

- Use an action sheet to request confirmation before performing a potentially destructive operation

Views - Text View

- Keep text legible



PROJECT SPECS

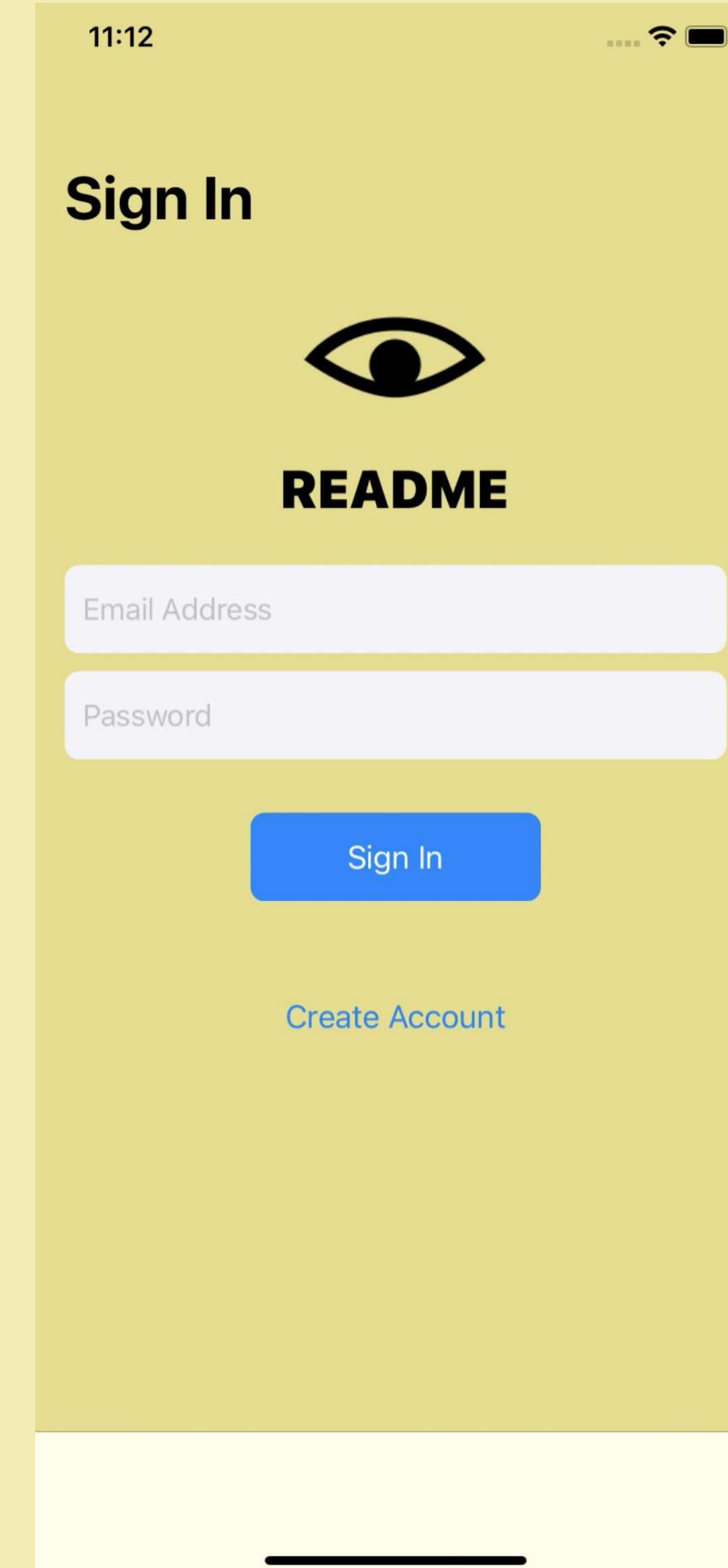
Apple HIG

Controls - Buttons

- Use a filled button for the most likely action in a view.
- Use style — not size — to visually distinguish the preferred choice among multiple options

Controls - Text Field

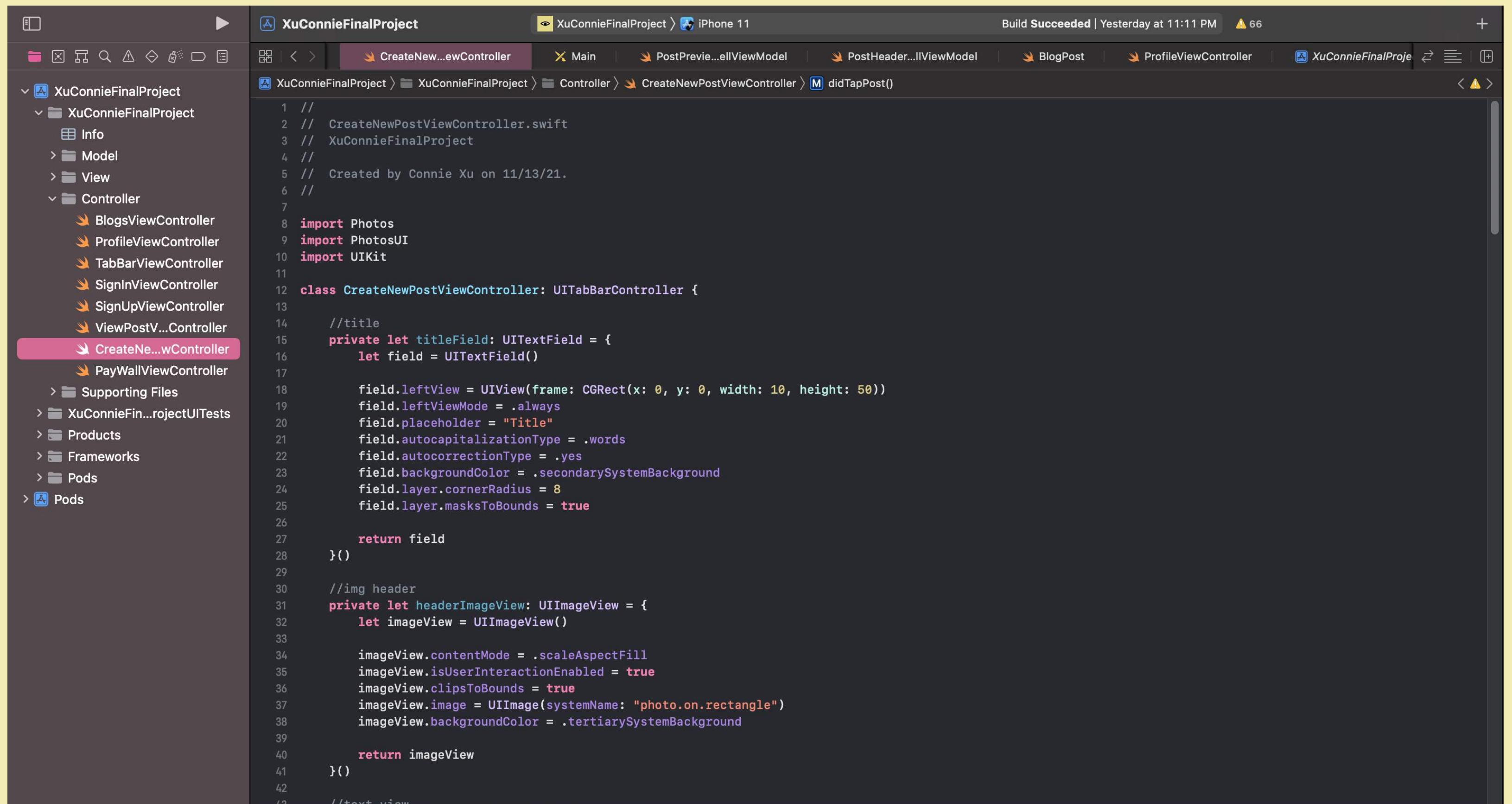
- Show a hint in a text field to help communicate its purpose.



PROJECT SPECS

Comments

proof of use of comments throughout code in addition to header of code



The screenshot shows the Xcode interface with the project 'XuConnieFinalProject' open. The left sidebar displays the project structure, including 'XuConnieFinalProject' at the root level, followed by 'Info', 'Model', 'View', and 'Controller'. Under 'Controller', several files are listed: 'BlogsViewController', 'ProfileViewController', 'TabBarController', 'SignInViewController', 'SignUpViewController', 'ViewPost...', 'CreateNewPostController' (which is highlighted with a pink rectangle), and 'PayWallViewController'. Below these are 'Supporting Files', 'XuConnieFin...', 'Products', 'Frameworks', and 'Pods'. The main editor area shows the code for 'CreateNewPostViewController.swift'. The code includes several comments:

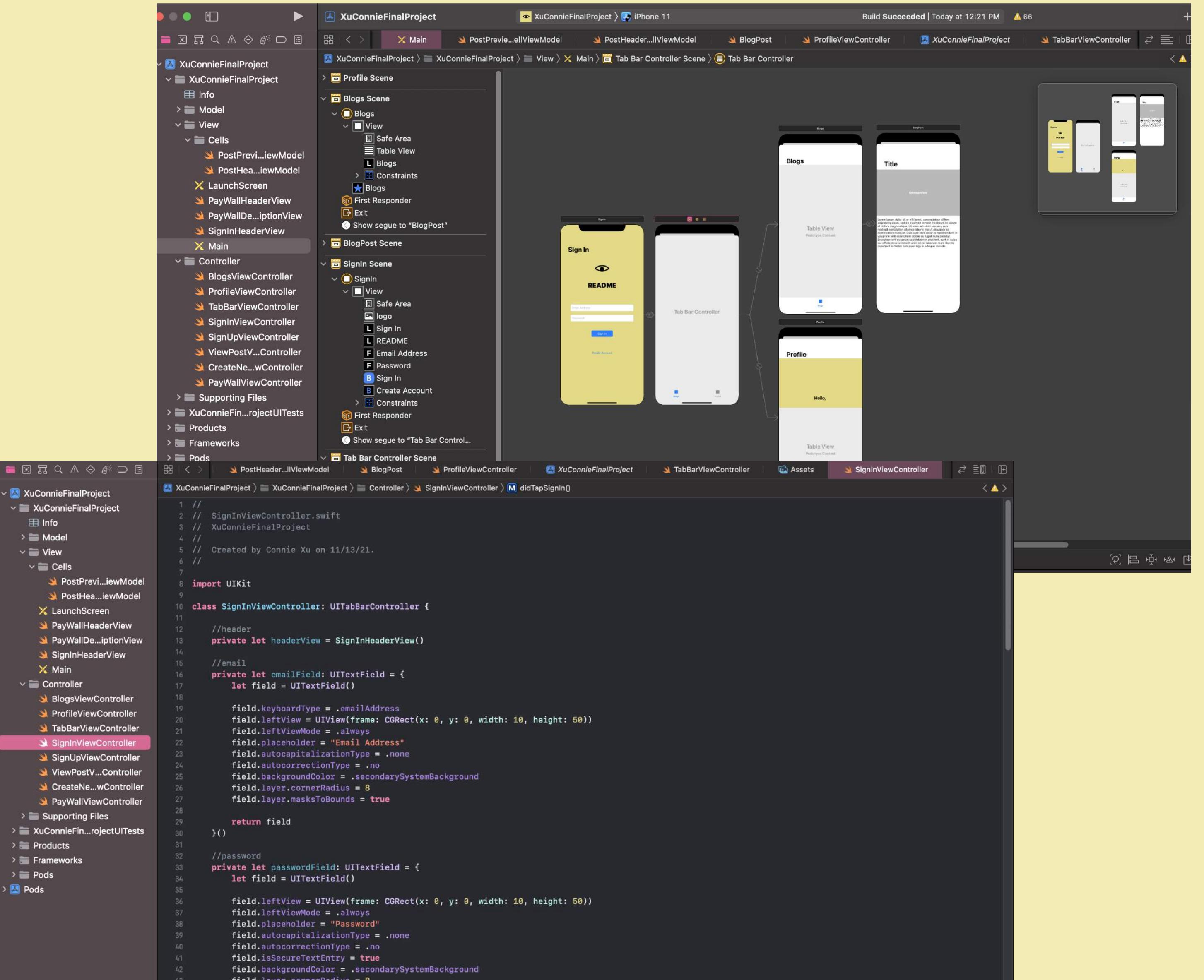
```
1 // CreateNewPostViewController.swift
2 // XuConnieFinalProject
3 // Created by Connie Xu on 11/13/21.
4 //
5 import Photos
6 import PhotosUI
7 import UIKit
8
9 class CreateNewPostViewController: UITabBarController {
10
11     //title
12     private let titleField: UITextField = {
13         let field = UITextField()
14
15         field.leftView = UIView(frame: CGRect(x: 0, y: 0, width: 10, height: 50))
16         field.leftViewMode = .always
17         field.placeholder = "Title"
18         field.autocapitalizationType = .words
19         field.autocorrectionType = .yes
20         field.backgroundColor = .secondarySystemBackground
21         field.layer.cornerRadius = 8
22         field.layer.masksToBounds = true
23
24         return field
25     }()
26
27     //img header
28     private let headerImageView: UIImageView = {
29         let imageView = UIImageView()
30
31         imageView.contentMode = .scaleAspectFill
32         imageView.isUserInteractionEnabled = true
33         imageView.clipsToBounds = true
34         imageView.image = UIImage(systemName: "photo.on.rectangle")
35         imageView.backgroundColor = .tertiarySystemBackground
36
37         return imageView
38     }()
39
40     //text view
41 }
```

PROJECT SPECS

Extra Notes

Pivots - Storyboard & Swift

- Drafted out elements on main.storyboard file
 - Utilized autolayout
 - Named classes for each VC appropriately
 - Connected segue on main.storyboard
- PIVOT: discovered segue creations between classes are easier via .swift files

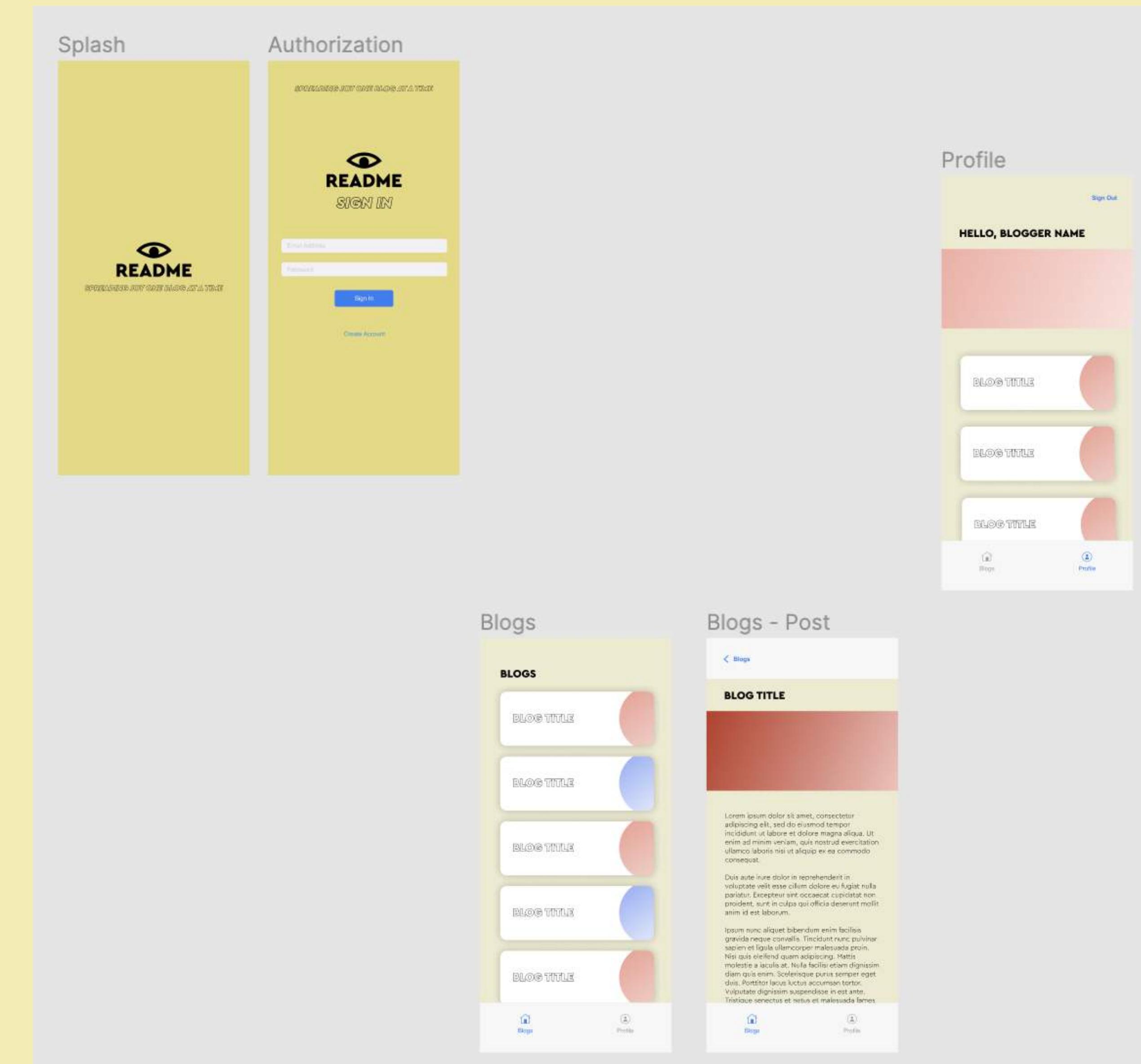


PROJECT SPECS

Extra Notes

Pivots - Design

- Designed initial screen layouts on Figma (seen on the right)
- PIVOT: reviewed Apple's Human Interface Guidelines
 - Decided to pivot design to match HIG



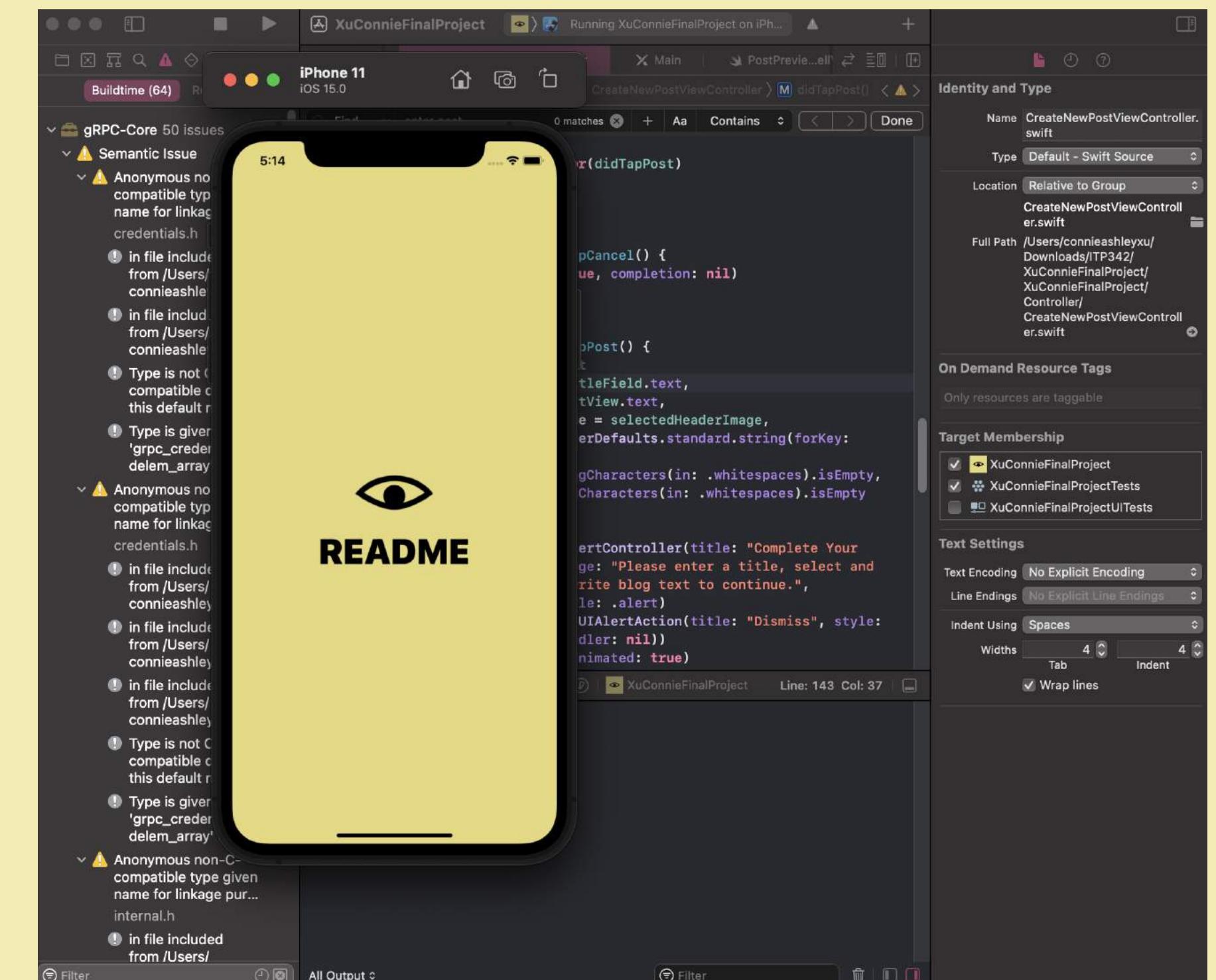
PROJECT SPECS

Extra Notes

Running the App - StoreKit

- You need to sign in to your Apple ID to test the subscription purchase aspect

recording of app demo linked [here](#)



proof that the code does compile and run

