single view project
set up firebase → p-list copyed to the folder enable email, enable database, initialize storage
close xcode
cd projectfolder → pod init
open Podfile → Analytics, Auth, Firestore, Core, Storage
pod install
open the workspace

// group ⇒ ViewControllers, Delegates,
SignIn VC
Feed VC
Upload VC
Settings VC
Snap VC
App Delegate
Scene Delegate

// AppDelegate.swift
```
import UIKit
import Firebase

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate{

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
        [UIApplication.launchOptionsKey: Any]?) -> Bool{
        FirebaseApp.configure()

        return true
    }
```

// SceneDelegate.swift
```
import UIKit
import

class SceneDelegate: UIResponder, UIWindowSceneDelegate{
    var window: UIWindow?
    func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options connectionOptions: UIScene.ConnectingOptions){
        let currentUser = PFUser.current()
        if currentUser != nil{
            let board = UIStoryboard(name: "Main", bundle: nil)
            let navigationController = board.instantiateViewController(identifier: "navigationController") as!
                UINavigationController
            window?.rootViewController = navigationController
        }
        guard let _ = (scene as? UIWindowScene) else {return}
    }
```

give identifier to navigationcontroller in main.storyboard

// SettingsViewController.swift

```
import UIKit
import Firebase

class SettingsViewController: UIViewController{

    override func viewDidLoad(){
        super.viewDidLoad()

    }
    @IBAction func logoutClicked(_ sender: Any){
        do{
            try Auth.auth().signOut()
            self.performSegue(withIdentifier: "toViewController", sender: nil)

        }catch{
            print("error")
        }
    }
}
```

// Model (folder) -> newfile -> UserSingleton.swift

```
import Foundation
class UserSingleton{
    static let sharedUserInfo = UserSingleton()
    var email = ""
    var username = ""

    private init(){
    }
}
```

// Snap.swift

```
import Foundation

struct Snap{
    var username: String
    var imageUserArray: [String]
    var date: Date
    var timeDifference: Int
}
```

// new File Cocoa Touch Class → FeedCell
subclass of UITableViewCell

// FeedCell.swift

```
import UIKit
import Firebase

class FeedCell : UITableViewCell{

    // labels
    @IBOutlet weak var documentIdLabel: UILabel! // hidden label
    override func awakeFromNib(){
        super.awakeFromNib()

    }

    override func setSelected(_ selected:Bool, animated:Bool){
        super.setSelected(selected, animated:animated)


    }

    @IBAction func likeButtonClicked(_ sender:Any){
        let fireStoreDatabase = Firestore.firestore()
        if let likeCount = Int(likeLabel.text!){
            let likeStore = ["likes": likeCount+1] as [String:Any]
            fireStoreDatabase.collection("Posts").document(documentIdLabel.text!).setData(likeStore, merge:true)
        }
        let userEmail = userEmailLabel.text!
        fireStoreDatabase.collection("PlayerId").whereField("email", isEqualTo:userEmail).getDocuments{(snapshot, error) in
            if error == nil{
                if snapshot?.isEmpty == false && snapshot != nil{
                    for document in snapshot!.documents{
                        if let playId = document.get("player_id") as? String{
                            OneSignal.postNotification(["contents":["en":"\(Auth.auth().currentUser!.email!) liked"],
                                "include_player_ids":["\(playerId)"]])

                        }
                    }
                }
            }
        }
    }
}
```

// create a new cell
→ prototype cell → change from 0 to 1
select the cell and add contents in it

the cell's identifier → cell
select Table View Cell

// assign the new cell class to

Create a hidden label

// relationship segway

// SignInVC.swift

// if not connect => clean the build > product > clean build
// re open xcode

```
import UIKit
import Firebase

class SignInVC : UIViewController{
    @IBOutlet weak var emailText : UITextField!
    @IBOutlet weak var passwordText : UITextField!
    @IBOutlet weak var userNameText : UITextField!
    override func viewDidLoad(){
        super.viewDidLoad()

    }

    @IBAction func signInClicked(_ sender:Any){
        if emailText.text! != "" && passwordText.text != "" {
            Auth.auth().signIn(withEmail: emailText.text!, password: passwordText.text!){ (authdata, error) in
                if error != nil{
                    makeAlert(titleInput: "Error!", messageInput: error?.localizedDescription ?? "Error")   // set a default value
                }else{
                    self.performSegue(withIdentifier: "toFeedVC", sender: nil)
                }
            }
        }else{
            makeAlert(titleInput: "Error!", messageInput: "Username/Password?")
        }

    }

    @IBAction func signUpClicked(_ sender:Any){
        if emailText.text! != "" && passwordText.text != "" {
            Auth.auth().createUser(withEmail: emailText.text!, password: passwordText.text!){ (authdata, error) in
                if error != nil{
                    makeAlert(titleInput: "Error!", messageInput: error?.localizedDescription ?? "Error")   // set a default value
                }else{
                    let fireStore = Firestore.firestore()
                    let userDictionary = ["email": self.emailText.text!, "userName": self.userNameText.text!] as [String:Any]
                    fireStore.collection("userInfo").addDocument(data: userDictionary){ (error) in
                        if error != nil{
                            // Alert
                        }
                        self.performSegue(withIdentifier: "toFeedVC", sender: nil)
                    }
                }
            }
        }else{
            makeAlert(titleInput: "Error!", messageInput: "Username/Password?")
        }
    }
```

```
func makeAlert(titleInput: String, messageInput: String){
    let alert = UIAlertController(title: titleInput, message: messageInput, preferredStyle:
        UIAlertController.Style.alert)
    let okButton = UIAlertAction(title: "Ok", style: UIAlertAction.Style.default, handler: nil)
    alert.addAction(okButton)
    self.present(alert, animated: true, completion: nil)
}
```

```
// UploadViewController.swift
import UIKit
import Firebase


class UploadViewController : UIViewController, UIImagePickerControllerDelegate , UINavigationControllerDelegate {
    @IBOutlet weak var imageView : UIImageView!
    ↓ similar ones

    override func viewDidLoad(){
        super.viewDidLoad()

        imageView.isUserInteractionEnabled = true
        let imageTapRecognizer = UITapGestureRecognizer(target: self, action : #selector(selectImage))
        imageView.addGestureRecognizer(imageTapRecognizer)
    }

    @objC func selectImage(){
        let picker = UIImagePickerController()
        picker.delegate = self
        picker.sourceType = .photoLibrary
        picker.allowEditing = true
        present(picker, animated: true, completion: nil)
    }

    func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info:
        [UIImagePickerController.InfoKey : Any]){
            imageView.image = info[.originalImage] as? UIImage

            saveButton.isEnabled = true

            self.dismiss(animated : true, completion: nil)
    }
                                                                    ← explaination
    //before IOS12 ⇒ info.plist ⇒ plus button ⇒ Privacy - Photo Library Usage Description → Value: to access photos


    @IBAction func saveButtonCliched(_ sender: Any){

        let storage = Storage.storage()
        let storageReference = storage.reference()

        let mediaFolder = storageReference.child("media")

        if let data = imageView.image?.jpegData(compressionQuality : 0.5){
            let uuid = UUID().uuidString
            let imageReference = mediaFolder.child("\(uuid).jpg")
            imageReference.putData(data, metadata: nil){ (metaData, error) in
                if error != nil{
                    self.makeAlert(titleInput : "Error!", messageInput : error?.localizedDescription ?? "error")
                }else{
                    imageReference.downloadURL{ (url, error) in
                        if error == nil{
```

```
let imageUrl = url?.absoluteString
// database        collection => document => field or collections
let firestoreDatabase = Firestore.firestore()

firestore.collection("Snaps").whereField("snapOwner", isEqualTo: UserSingleton.sharedUserInfo.username).
    getDocuments{ (snap, error) in
    if error != nil{
        self.makeAlert(titleInput: "Error!", messageInput: error?.localizedDescription ?? "error")
    }else{
        if snapshot?.isEmpty == false && snapshot != nil{
            for document in snapshot!.documents{
                let documentId = document.documentID
                if var imageUrlArray = document.get("imageUrlArray") as? [String]{
                    imageUrlArray.append(imageUrl!)
                    let additionalDictionary = ["imageUrlArray" : imageUrlArray] as [String : Any]
                    firestore.collection("Snap").document(documentId).setData(additionalDic-
                        -tionary, merge: true){ (error) in
                        if error != nil{
                            self.imageView.image = UIImage(named: "select.png")
                            self.tabBarController?.selectedIndex = 0
                        }
                    }
                }
            } -> // if snapshot == nil
        }else{
            let firestorePost = ["imageUrlArray": [imageUrl!], "snapOwner": UserSingleton.sharedUserInfo.username,
                "date": FieldValue.serverTimestamp()] as [String : Any]
            firestoreDatabase.collection("Posts").addDocument(data: firestorePost, completion:{(error) in
                if error != nil{
                    self.makeAlert(titleInput: "Error!", messageInput: error?.localizedDescription ?? "error")
                }else{
                    self.imageView.image = UIImage(named: "select.png")

                    self.tabBarController?.selectedIndex = 0
                }
            }
        }
    }
}

func makeAlert(titleInput: String, messageInput: String){
    let alert = UIAlertController(title: titleInput, message: messageInput, preferredStyle:
        UIAlertController.Style.alert)
    let okButton = UIAlertAction(title: "OK", style: UIAlertAction.Style.default, handler: nil)
    alert.addAction(okButton)
    self.present(alert, animated: true, completion: nil)
}
```

// FeedViewController.swift

```
import UIKit
import Firebase
import SDWebImage
import OneSignal
```

// increase the prototype cell number to 1

→ if not showing up → Build Phases → link Binary with Libraries → + SDWebImage

// SDWebImage

pod 'SDWebImage' & close xcode first

```
class FeedViewController: UIViewController, UITableViewDelegate, UITableViewDataSource{
    @IBOutlet weak var tableView: UITableView!

    let fireStoreDatabase = Firestore.firestore()
    var snapArray = [Snap]()
    var chosenSnap : Snap?



    override func viewDidLoad(){
        super.viewDidLoad()
        tableView.delegate = self
        tableView.dataSource = self
        getDataFromFirestore()
        getUserInfo()
    }

    func getUserInfo(){
        fireStoreDatabase.collection("userInfo").whereField("email", isEqualTo: Auth.auth().currentUser!.email!).getDocuments{(snapshot, error) in
            if error != nil{
                self.makeAlert(titleInput: "Error!", messageInput: error?.localizedDescription ?? "error")
            }else{
                if snapshot?.isEmpty == false && snapshot != nil{
                    for document in snapshot!.documents{
                        if let username = document.get("username") as? String{
                            UserSingleton.sharedUserInfo.email = Auth.auth.currentUser!.email!
                            UserSingleton.sharedUserInfo.username = username
                        }
                    }
                }
            }
        }
    }


    func getDataFromFirestore(){
        /* let settings = fireStoreDatabase.settings
        settings.areTimestampsInSnapshotsEnabled = true // may not need this because the default setting is true
        fireStoreDatabase.settings = settings */

        fireStoreDatabase.collection("Snaps").order(by: "date", descending: true).addSnapshotListener{(snapshot, error) in
            if error != nil{
                print(error?.localizedDescription)
```

```
        } else {
            if snapshot?.isEmpty != true && snapshot != nil {
                self.snapArray.removeAll(keepingCapacity: false)
                for document in snapshot!.documents {
                    let documentId = document.documentID
                    if let username = document.get("snapOwner") as? String {
                        if let imageUrlArray = document.get("imageUrlArray") as? [String] {
                            if let date = document.get("date") as? Timestamp {
                                if let difference = Calendar.current.dateComponents([.hour], from: date.dateValue(), to: Date())
                                    .hour {
                                    if difference >= 24 {
                                        self.fireStoreDatabase.collection("Snaps").document(documentId).delete { (error) in
                                            // display the error
                                        }
                                    } else {
                                        let snap = Snap(username: username, imageUrlArray: imageUrlArray, date: date.dateValue()
                                                        , timeDifference: 24 - difference)
                                        self.snapArray.append(snap)
                                    }
                                }
                            }
                        }
                    }
                }
                self.tableView.reloadData() // call after the for loop
            }
        }
    }
}

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

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath) as! FeedCell
    cell.feedUserNameLabel.text = snapArray[indexPath.row].username
    cell.feedImageView.sd_setImage(with: URL(string: snapArray[indexPath.row].imageArray[0]))

    return cell
}

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "toSnapVC" {
        let destinationVC = segue.destination as SnapVC
        destinationVC.selectedSnap = chosenSnap

    }
}
```

```
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath){
    chosenSnap = self.snapArray[indexPath.row]
    performSegue(withIdentifier: "toSnapVC", sender: nil)

}
```

//SnapVC.swift

//pod → ImageSlideShow/Kingfisher

```
import UIKit
import ImageSlideShow

class SnapVC: UIViewController{
    @IBOutlet weak var timeLabel: UILabel!
```


```
    var selectedSnap: Snap?

    var inputArray = [KingfisherSource]()

    override func viewDidLoad(){
        super.viewDidLoad()

        if snap = selectedSnap{
            timeLabel.text = "Time Left: \(snap.timeDifference)"
            for imageUrl in snap.imageUrlArray{
                inputArray.append(KingfisherSource(urlString: imageUrl)!)
            }

            let imageSlideShow = ImageSlideShow(frame: CGRect(x: 10, y: 10, width: self.view.frame.width * 0.95, height:
                self.view.frame.height * 0.9))
            imageSlideShow.backgroundColor = UIColor.white

            let pageIndicator = UIPageControl()
            pageIndicator.currentPageIndicatorTintColor = UIColor.lightGray
            pageIndicator.pageIndicatorTintColor = UIColor.black
            imageSlideShow.pageIndicator = pageIndicator


            imageSlideShow.contentScaleMode = UIViewContentMode.scaleAspectFit
            imageSlideShow.setImageInputs(inputArray)

            self.view.addSubview(imageSlideShow)

            self.view.bringSubviewToFront(timeLabel)

        }
    }
}
```