

Alarm

```
class Alarm: PersistableRecord, Codable, FetchableRecord
```

Inheritance

```
Codable, FetchableRecord, PersistableRecord
```

Initializers

```
init()
```

```
init()
```

```
init(label:hour:minute:year:month:day:active:)
```

```
init(label: String, hour: Int, minute: Int, year: Int, month: Int, day: Int, active: Bool)
```

Properties

```
id
```

```
var id: Int?
```

```
firebaseID
```

```
var firebaseID: String?
```

```
label
```

```
var label: String?
```

```
year
```

```
var year: Int?
```

```
month
```

```
var month: Int?
```

```
day
```

```
var day: Int?
```

```
hour
```

```
var hour: Int?
```

```
minute
```

```
var minute: Int?
```

active

```
var active: Bool?
```

status

```
var status: String?
```

deleteRequest

```
var deleteRequest: String?
```

Methods

encode(to:)

```
func encode(to container: inout PersistenceContainer)
```

alarmDB

```
let alarmDB
```

AlarmScheduler

Alarm scheduling class to schedule alarm

```
class AlarmScheduler
```

Methods

scheduleNotification(title:dateComponents:id:)

Function to schedule alarm for the first time after doanloaded from Firebase. Simply call this function and it will push the alarm to the OS notification center

```
func scheduleNotification(title: String, dateComponents: DateComponents, id: String?)
```

Parameters

- title: alarm identifier string
- dateComponents: The date of the alarm when it should be triggered
- id: Firebase ID fetched from firebase just tracking the alarm data

rescheduleNotification(title:id:)

Reschedule alam once the the alarm is delayed

```
public func rescheduleNotification(title: String, id: String?)
```

Parameters

- title: alarm identifier string
- id: Firebase ID fetched from firebase just tracking the alarm data

clearNotifications(title:)

Request OS to remove notification from pending list and clear the notification list already delivered

```
func clearNotifications(title: String)
```

Parameters

- title: Alarm identifier string

AlarmVC

The alarm view controller that is show after a row on the table is tapped

```
class AlarmVC: UIViewController, UINavigationControllerDelegate, UIImagePickerControllerDelegate
```

Inheritance

UIImagePickerControllerDelegate, UINavigationControllerDelegate, UIViewController

Properties

backButton

```
let backButton
```

saveButton

```
let saveButton
```

cancelButton

```
let cancelButton
```

deleteButton

```
let deleteButton
```

buttonStack

```
let buttonStack
```

alarmDetailsLabel

```
let alarmDetailsLabel
```

instructionLabel

```
let instructionLabel
```

dateLabel

```
let dateLabel
```

timeLabel

```
let timeLabel
```

alarmDelegate

```
var alarmDelegate: AlarmVCDelegate?
```

alarm

```
var alarm
```

alarmNameTextField

```
let alarmNameTextField
```

alarmDateTextField

```
let alarmDateTextField
```

alarmTimeTextField

```
let alarmTimeTextField
```

statusLabel

```
let statusLabel
```

statusStatusLabel

```
let statusStatusLabel
```

datePicker

```
let datePicker
```

timePicker

```
let timePicker
```

slider

```
let slider
```

pictureSlider

```
let pictureSlider
```

imageView

```
let imageView
```

notificationTitle

```
var notificationTitle: Any?
```

takenImageViewer

```
let takenImageViewer
```

alarmIndex

```
var alarmIndex
```

Methods

viewDidLoad()

Main function of the view controller, all the important UI and rest of the alarm meta data are downloaded here.

```
override func viewDidLoad()
```

imageViewSetup()

Sets up the size/location/shape and style of the imageview

```
func imageViewSetup()
```

showDatePicker()

Sets up the size/location/shape and style of the date picker

```
func showDatePicker()
```

takenImageViewSetup()

Sets up the size/location/shape and style of the imageview for take image

```
func takenImageViewSetup()
```

showTimePicker()

Sets up the size/location/shape and style of the time picker

```
func showTimePicker()
```

doneDatePicker()

action when done button is tapped on date picker

```
@objc func doneDatePicker()
```

donetimePicker()

action when done button is tapped on time picker

```
@objc func donetimePicker()
```

cancelDatePicker()

Action to cancel date picker and make it disappear

```
@objc func cancelDatePicker()
```

changeValue(_:)

Action when slider value is changed. Updates the current status of the alarm based on the direction of slide. If left, then add delay to the alarm. If right, update the data on firebase and prompt the user to the next step

```
@objc func changeValue(_ sender: UISlider)
```

Parameters

- sender: The slider object itself

changeValuePictureSlider(_:)

Action when slider value is changed. Updates the current status of the alarm based on the direction of slide. If left, then add delay to the alarm. If right, update the data on firebase and prompt the user to the next step

```
@objc func changeValuePictureSlider(_ sender: UISlider)
```

Parameters

- sender: The slider object itself

imagePickerController(_:didFinishPickingMediaWithInfo:)

Action when image picker is done picking image and returns it. It also uploads the image to firebase and updates the status of the alarm

```
func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [UIImagePickerController.InfoKey :
```



alarmDateTextFieldSetup()

Sets up the size/location/shape and style of the alarm date text field

```
func alarmDateTextFieldSetup()
```

instructionLabelSetup()

Sets up the size/location/shape and style of the instruction label

```
func instructionLabelSetup()
```

alarmDetailsLabelSetup()

Sets up the size/location/shape and style of the alarm details label

```
func alarmDetailsLabelSetup()
```

alarmNameTextFieldSetup()

Sets up the size/location/shape and style of the alarm name text field

```
func alarmNameTextFieldSetup()
```

backButtonSetup()

Sets up the size/location/shape and style of the back button

```
func backButtonSetup()
```

backButtonClicked()

Action for back button when clicked, takes the user back to the previous screen

```
@objc func backButtonClicked()
```

`statusStatusLabelSetup()`

Sets up the size/location/shape and style of the label for the status label

```
func statusStatusLabelSetup()
```

`sliderSetup()`

Sets up the size/location/shape and style of the slider

```
func sliderSetup()
```

`pictureSliderSetup()`

Sets up the size/location/shape and style of the slider

```
func pictureSliderSetup()
```

AlarmVCDelegate

```
protocol AlarmVCDelegate
```

Requirements

reloadTableDelegate()

```
func reloadTableDelegate()
```

AppDatabase

A type responsible for initializing the application database.

```
struct AppDatabase
```

See `AppDelegate.setupDatabase()`

Methods

`openDatabase(atPath:)`

Creates a fully initialized database at path

```
static func openDatabase(atPath path: String) throws -> DatabaseQueue
```

AppDelegate

```
@available(iOS 10.0, *) @UIApplicationMain class AppDelegate: UIResponder, UIApplicationDelegate, UNUserNotificationCenterDelegate
```

Inheritance

`UIApplicationDelegate`, `UIResponder`, `UNUserNotificationCenterDelegate`

Properties

`window`

```
var window: UIWindow?
```

Methods

`application(_:didFinishLaunchingWithOptions:)`

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?)
```

`application(_:performFetchWithCompletionHandler:)`

Background fetch function. Tries the fetch data when the app is in background

```
func application(_ application: UIApplication, performFetchWithCompletionHandler completionHandler: @escaping (UIBackgroundFetchResult)
```

Parameters

- application: The current application
- completionHandler: checks whether new data has been fetched

`applicationWillResignActive(_:)`

```
func applicationWillResignActive(_ application: UIApplication)
```

`applicationDidEnterBackground(_:)`

```
func applicationDidEnterBackground(_ application: UIApplication)
```

`applicationWillEnterForeground(_:)`

```
func applicationWillEnterForeground(_ application: UIApplication)
```

`applicationDidBecomeActive(_:)`

```
func applicationDidBecomeActive(_ application: UIApplication)
```

`applicationWillTerminate(_:)`

```
func applicationWillTerminate(_ application: UIApplication)
```

`application(_:didRegisterForRemoteNotificationsWithDeviceToken:)`

```
func application(_ application: UIApplication, didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data)
```

`application(_:didFailToRegisterForRemoteNotificationsWithError:)`

```
func application(_ application: UIApplication, didFailToRegisterForRemoteNotificationsWithError error: Error)
```

`setupDatabase(_:)`

Sets up the internal swl database which is currently not in use

```
private func setupDatabase(_ application: UIApplication) throws
```


registerForPushNotifications()

Registers the app for push notification

```
func registerForPushNotifications()
```

getNotificationSettings()

Gets the notification settings for local notification

```
func getNotificationSettings()
```

userNotificationCenter(_:willPresent:withCompletionHandler:)

```
@available(iOS 10.0, *) func userNotificationCenter(_ center: UNUserNotificationCenter, willPresent notification: UNNotification, wi
```

userNotificationCenter(_:didReceive:withCompletionHandler:)

```
@available(iOS 10.0, *) func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive response: UNNotificationResponse,
```

downloadNotificationSound()

Download the custom notification sound that's to be used

```
func downloadNotificationSound()
```

FirestoreUtil

```
class FirestoreUtil
```

Methods

fetchOneObject(firebaseID:)

Fetch one particular firebase object using the given firebase ID. Returns a single alarm

```
func fetchOneObject(firebaseID: String) -> Alarm
```

Parameters

- firebaseID: The id of the element trying to fetch from firebase

Types

- [Alarm](#)
- [AlarmDB](#)
- [AlarmScheduler](#)
- [AlarmVC](#)
- [AppDatabase](#)
- [AppDelegate](#)
- [FirestoreUtil](#)
- [MainVC](#)
- [RewardVC](#)
- [RootViewController](#)
- [SignInVC](#)

Protocols

- [AlarmVCDelegate](#)

Globals

Functions

- [getDocumentsDirectory\(\)](#)

Variables

- [activeAlarm](#)
- [alarmDB](#)
- [completedTask](#)
- [dbQueue](#)
- [fUtil](#)
- [inactiveAlarm](#)
- [ref](#)
- [scheduler](#)
- [totalTask](#)
- [userID](#)

MainVC

The main view controllr of the app. Users often see this view controller as a default. This is where all the alarms are fetched from firebase and alarm requests are made to the OS.

```
class MainVC: UIViewController
```

Inheritance

[AlarmVCDelegate](#), [UITableViewDataSource](#), [UITableViewDelegate](#), [UIViewController](#), [UNUserNotificationCenterDelegate](#)

Properties

welcomeTextView

```
let welcomeTextView
```

clockLabel

```
let clockLabel
```

topBar

```
let topBar
```

timeLabel

```
let timeLabel
```

timer

```
var timer
```

date

```
var date
```

calendar

```
let calendar
```

dateFormatter

```
let dateFormatter
```

logoutButton

```
let logoutButton
```

refreshControl

```
var refreshControl
```

summaryLabel

```
let summaryLabel
```

ref

```
var ref: DatabaseReference!
```

alarmTable

```
let alarmTable
```

Methods

reloadTableDelegate()

```
func reloadTableDelegate()
```

viewDidLoad()

This is the main function where all the important tasks are called, such as loading the UI, Fetching from firebase and loading data into array

```
override func viewDidLoad()
```

updateTimeLabel()

Updates the time label for the clock shown on screen. Also refreshes the table view after certain time period

```
@objc func updateTimeLabel()
```

reloadTable()

Reloads data in the table view

```
@objc func reloadTable()
```

summaryLabelSetup()

Sets up the summary label. This label shows how many alarms are remaining to be completed for the day.

```
func summaryLabelSetup()
```

logoutButtonClicked()

Logout button action. Logout the user from the app

```
@objc func logoutButtonClicked()
```

fetchFromFirebase()

Fetches data from firebase and loads them in an array to be supplied to table view

```
func fetchFromFirebase()
```

tableView(_:numberOfRowsInSection:)

Built in table view function. Determines how many rows to be shown in the table

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int
```

Parameters

- tableView: table view class
- section: number of rows

tableView(_:cellForRowAt:)

Sorts out the data to be shown in each row of the table

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

tableView(_:didSelectRowAt:)

Action to be done when a row is tapped

```
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath)
```

userNotificationCenter(_:willPresent:withCompletionHandler:)

```
@available(iOS 10.0, *) func userNotificationCenter(_ center: UNUserNotificationCenter, willPresent notification: UNNotification, withCompletionHandler: (UNNotificationPresentationOptions) -> Void) -> Void
```

userNotificationCenter(_:didReceive:withCompletionHandler:)

```
@available(iOS 10.0, *) func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive response: UNNotificationResponse, withCompletionHandler: () -> Void) -> Void
```

logoutButtonSetup()

Sets up the size/location/shape and style of the logout button

```
func logoutButtonSetup()
```

alarmTableSetup()

Sets up the size/location/shape and style of the alarm table

```
func alarmTableSetup()
```

topViewSetup()

Sets up the size/location/shape and style of the top view

```
func topViewSetup()
```

`timeLabelSetup()`

Sets up the size/location/shape and style of the time label

```
func timeLabelSetup()
```

`welcomeTextViewSetup()`

Sets up the size/location/shape and style of the welcome text

```
func welcomeTextViewSetup()
```

`clockLabelSetup()`

Sets up the size/location/shape and style of the clock label

```
func clockLabelSetup()
```

`viewWillAppear(_:)`

Refreshes the data in the alarm table when the view is loaded

```
override func viewWillAppear(_ animated: Bool)
```

RewardVC

```
class RewardVC: UIViewController
```

Inheritance

`UIViewController`

Properties

`circleHolder`

```
let circleHolder
```

`percentageLabel`

```
let percentageLabel
```

`button`

```
let button
```

`textView`

```
let textView
```

`fraction`

```
var fraction
```

imageView

```
let imageView
```

imageArray

```
let imageArray
```

Methods

viewDidLoad()

```
override func viewDidLoad()
```

imageViewSetup()

Sets up the size/location/shape and style of the imageView

```
func imageViewSetup()
```

progressCircleSetup()

Sets up the size/location/shape and style of the progress circle

```
func progressCircleSetup()
```

buttonSetup()

Sets up the size/location/shape and style of the button

```
func buttonSetup()
```

buttonClicked()

Action when the button is tapped. Takes user back

```
@objc func buttonClicked()
```

textViewSetup()

Sets up the size/location/shape and style of the textView

```
func textViewSetup()
```

RootViewController

```
class RootViewController: UIViewController
```

Inheritance

UIViewController

Methods

viewDidLoad()

```
override func viewDidLoad()
```

SignInVC

```
class SignInVC: UIViewController
```

Inheritance

UIViewController

Properties

loginButton

```
let loginButton
```

signupButton

```
let signupButton
```

guestButton

```
let guestButton
```

emailTF

```
let emailTF
```

passwordTF

```
let passwordTF
```

backButton

```
let backButton
```

logoImage

```
let logoImage
```

errorMessageView

```
let errorMessageView
```

Methods

viewWillAppear(_:)

Adds a authentication listener to see if the user is signed in

```
override func viewWillAppear(_ animated: Bool)
```

keyboardWillShow(notification:)

Raises the view when the keyboard appears

```
@objc func keyboardWillShow(notification: NSNotification)
```

Parameters

- notification: The notification observer that the keyboard has appeared

keyboardWillHide(notification:)

Lowers the view when the keyboard disappears

```
@objc func keyboardWillHide(notification: NSNotification)
```

Parameters

- notification: The notification observer that the keyboard has disappeared

viewDidLoad()

Main function of this view controller. All the important tasks are done here such as loading the UI, adding listener for the keyboard etc.

```
override func viewDidLoad()
```

errorMessageViewSetup()

Sets up the size/location/shape and style of the error message textview

```
func errorMessageViewSetup()
```

logoImageSetup()

Sets up the size/location/shape and style of the logo image

```
func logoImageSetup()
```

loginButtonSetup()

Sets up the size/location/shape and style of the login button

```
func loginButtonSetup()
```

loginButtonTapped()

Action when the login button is tapped. Checks if all of the required fields are full and checks with firebase to authenticate

```
@objc func loginButtonTapped()
```

signUpButtonSetup()

Sets up the size/location/shape and style of the signup button

```
func signUpButtonSetup()
```

signUpButtonTapped()

Action when the signup button is tapped. Registers the user in firebase

```
@objc func signUpButtonTapped()
```


emailTFSetup()

Sets up the size/location/shape and style of the email textfield

```
func emailTFSetup()
```

passwordTFSetup()

Sets up the size/location/shape and style of the password textfield

```
func passwordTFSetup()
```

backButtonSetup()

Sets up the size/location/shape and style of the back button

```
func backButtonSetup()
```

Generated at 2020-05-02T13:21:48-0400 using [swift-doc](#) 1.0.0-beta.2.

- Types
- Protocols
- Global Variables
- Global Functions

activeAlarm

```
var activeAlarm
```

completedTask

```
var completedTask
```

dbQueue

```
var dbQueue: DatabaseQueue!
```

fUtil

```
let fUtil
```

getDocumentsDirectory()

Get the document directory

```
func getDocumentsDirectory() -> URL
```

inactiveAlarm

```
var inactiveAlarm
```

ref

```
var ref: DatabaseReference!
```

scheduler

```
let scheduler
```

totalTask

```
var totalTask
```

userID

```
let userID
```