

LCD Clock

Concept:

Create an LCD clock with HH:MM display.

The user should be able to set color of the digits, change the background image, set the time-zone, and the time format – 24hr/12hr. In 12-hour format, show AM/PM.

Make sure the clock rotates correctly when the device is rotated and is displayed properly on different devices.

See the designs (*included separately*) to see what your app should look like when it's turned in.

Requirements:

- Make your own seven-segment display using UIViews in Interface Builder. Create a single UIView for a digit and use it for the 4 numbers.
- Use NSTimer to trigger time changes on the screen. Make sure the displayed time is in sync (within a second) with the phone's clock. (It can be in a different time zone than the phone though)
- The user's preferences above should be saved using NSUserDefaults. Which means that even after the app is killed and restarted, the app should remember the settings.
- Use an alternate way to save by archiving your objects to your own file.
- You should use autolayout so that the view is displayed consistently across different devices and orientations. To simplify the autolayout step, you should create a view as a subview of your main view. On that subview, you should add the 4 views for the digits as subviews.

Optional:

- Implement blinking hour-minute separator which blinks twice a second
- Add seconds
- Implement triangular tips
- If you were to add an alarm clock feature, how would you make sure the alarm works even when the app is not running.

Resources:

Archives and Serializations Programming Guide

<https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/Archiving/Archiving.html>

Date and Time Programming Guide

<https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/DatesAndTimes/DatesAndTimes.html>

View Programming Guide

https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewPG_iPhoneOS/Introduction/Introduction.html

Resources:

Make sure you understand each of the following

1. What is meant by archiving?
2. Where is the data in UserDefaults stored? How can you look inside it?
3. What kind of class version issue can you expect with object archiving?