# การใช้งานแทบบาร์และปิกเกอร์วิว

---

## วัตถุประสงค์

เรียนรู้การสร้างวิวหลายๆวิวด้วยแทบบาร์
เรียนรู้วิธีการเปลี่ยนวิวด้วยแทบบาร์
เรียนรู้วิธีการสร้างปิกเกอร์วิว

---

## ปัญหา

ต้องการสร้างแอพที่มีหลายวิวหรือหลายหน้าจอและควบคุมวิวด้วยแทบบาร์
สร้างปิกเกอร์แบบคอลัมน์เดียว แบบสองคอลัมน์ แบบอิสระ และแบบกำหนดเอง

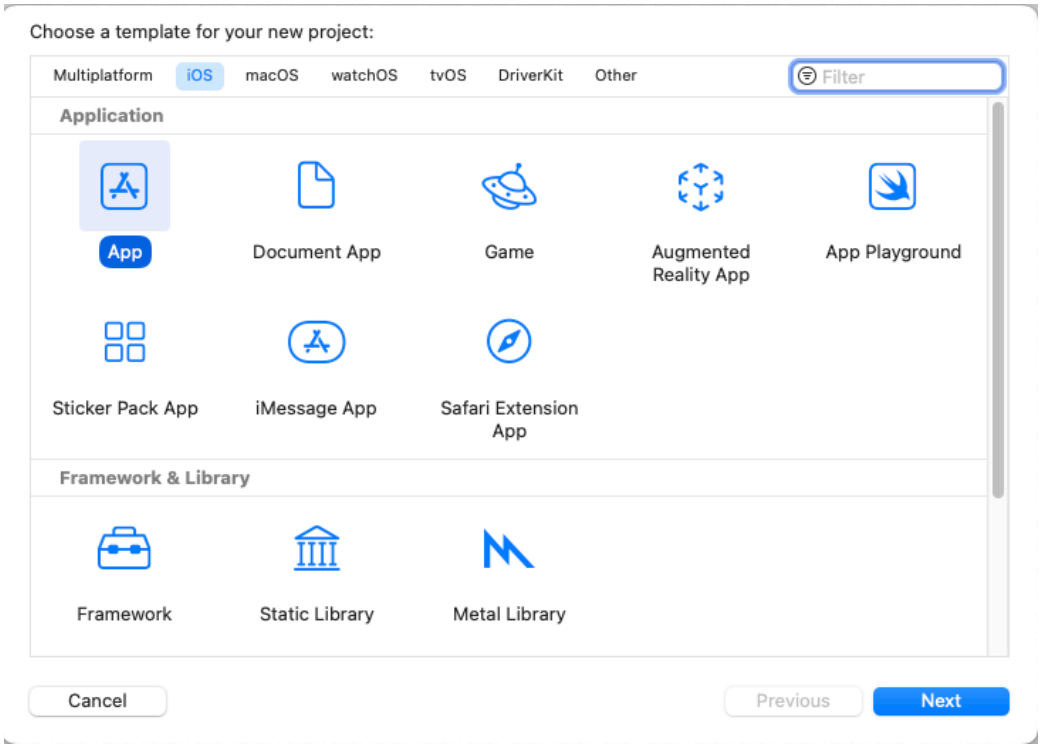
---

## วิธีการสร้างแอป

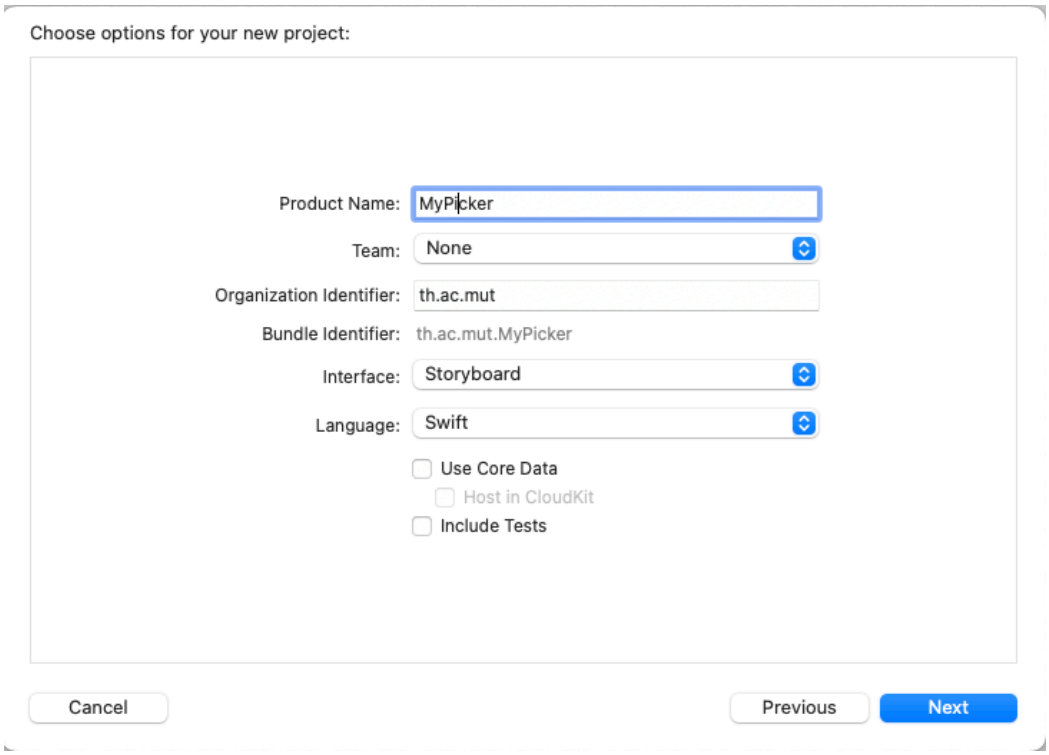
แอพนี้จะประกอบด้วยวิวสี่วิวที่ถูกควบคุมด้วยแทบบาร์คอนโทรลเลอร์
วิวหนึ่งแสดงปิกเกอร์เลือกวันที่
วิวสองแสดงปิกเกอร์แบบคอลัมน์เดียว หลายแถว
วิวสามแสดงปิกเกอร์แบบสองคอลัมน์หลายแถว แบบอิสระ
วิวสี่แสดงปิกเกอร์แบบกำหนดเอง ด้วยรูปภาพ

## Creating the Tab Bar Controller and Pickers

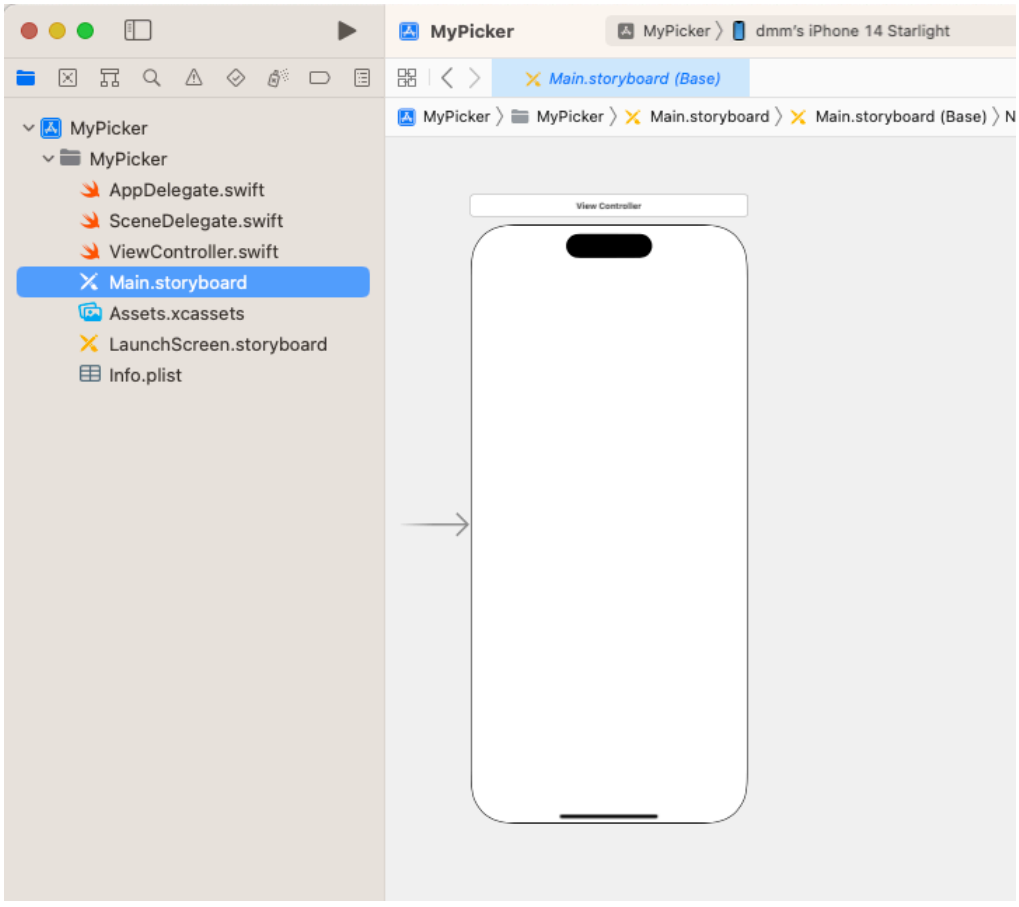
1. Create a new project: File—>New—>Project…
2. Choose Platform: iOS
3. Choose Application: App



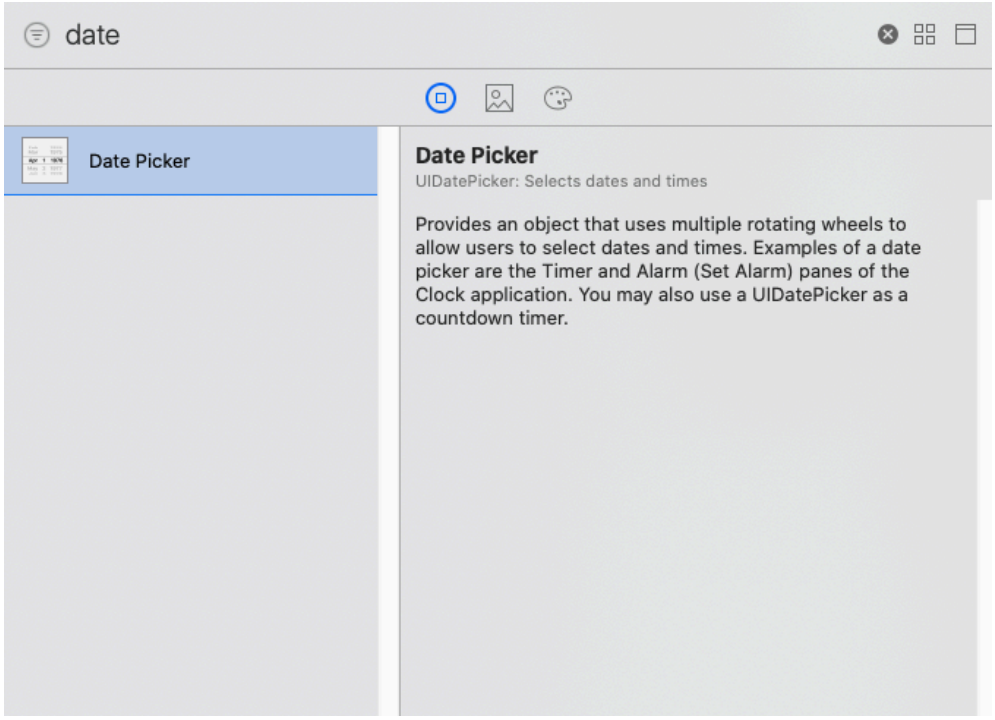
3. Name the project: MyPicker

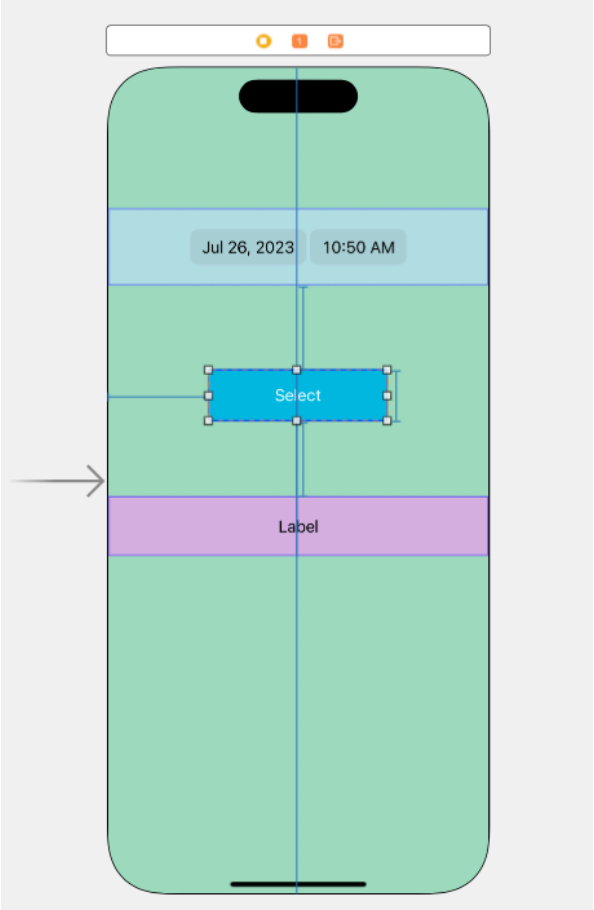



4. Click on Main.Storyboard.



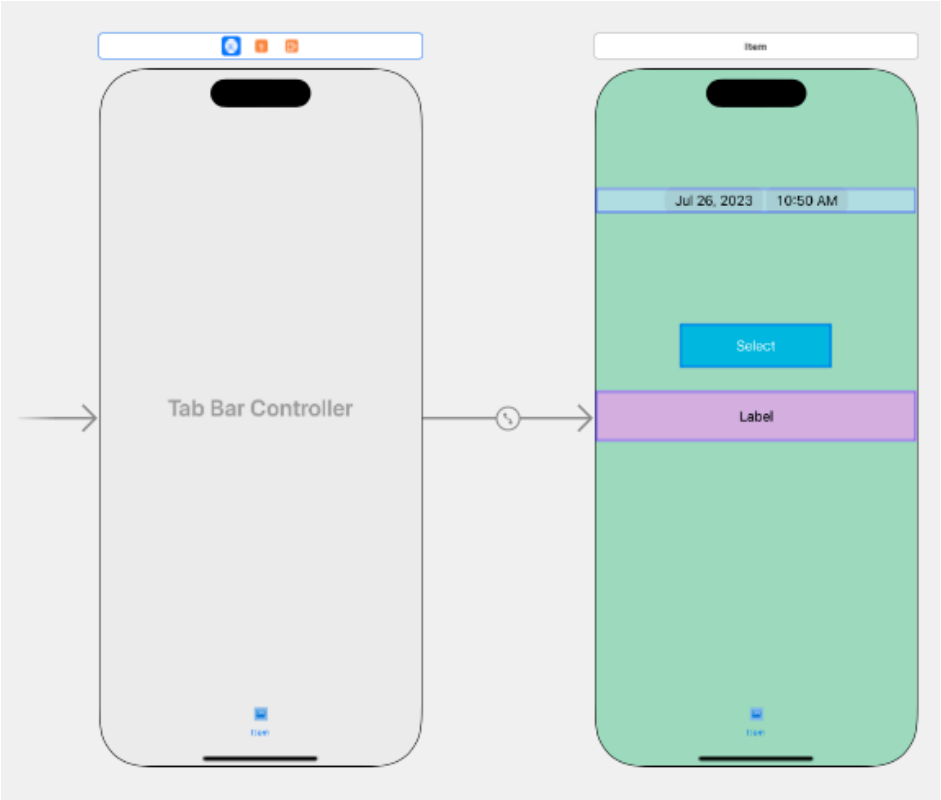
5. Add a Date Picker, a button and a label in the view.



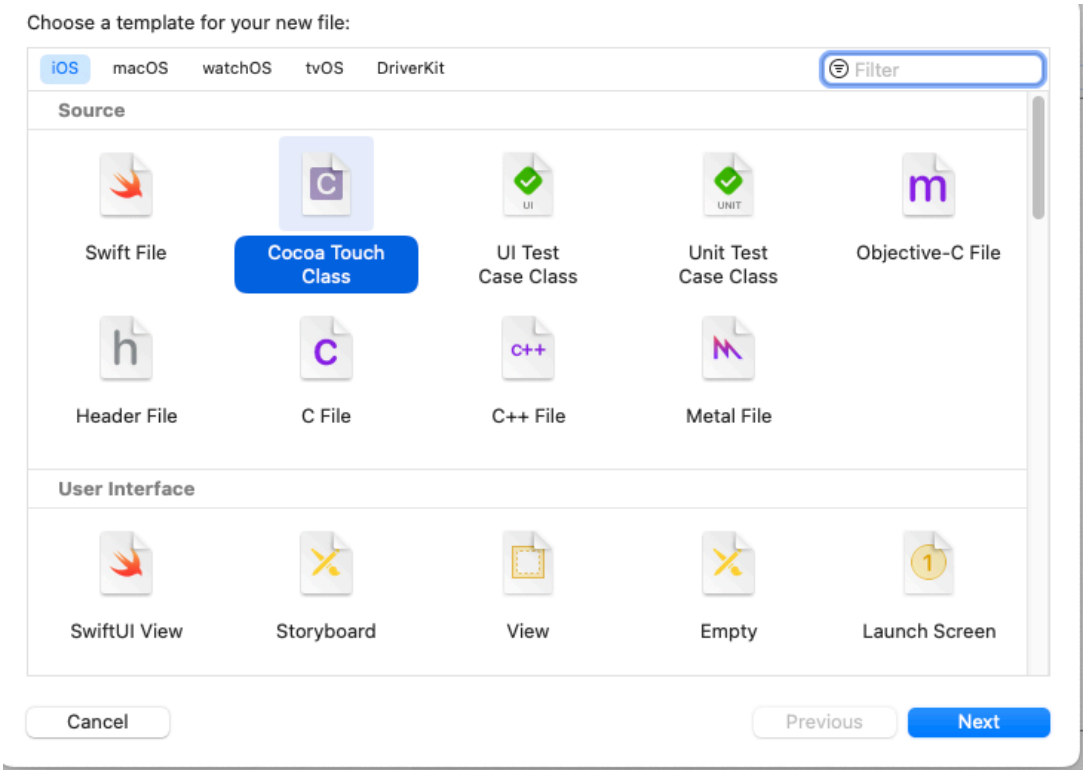



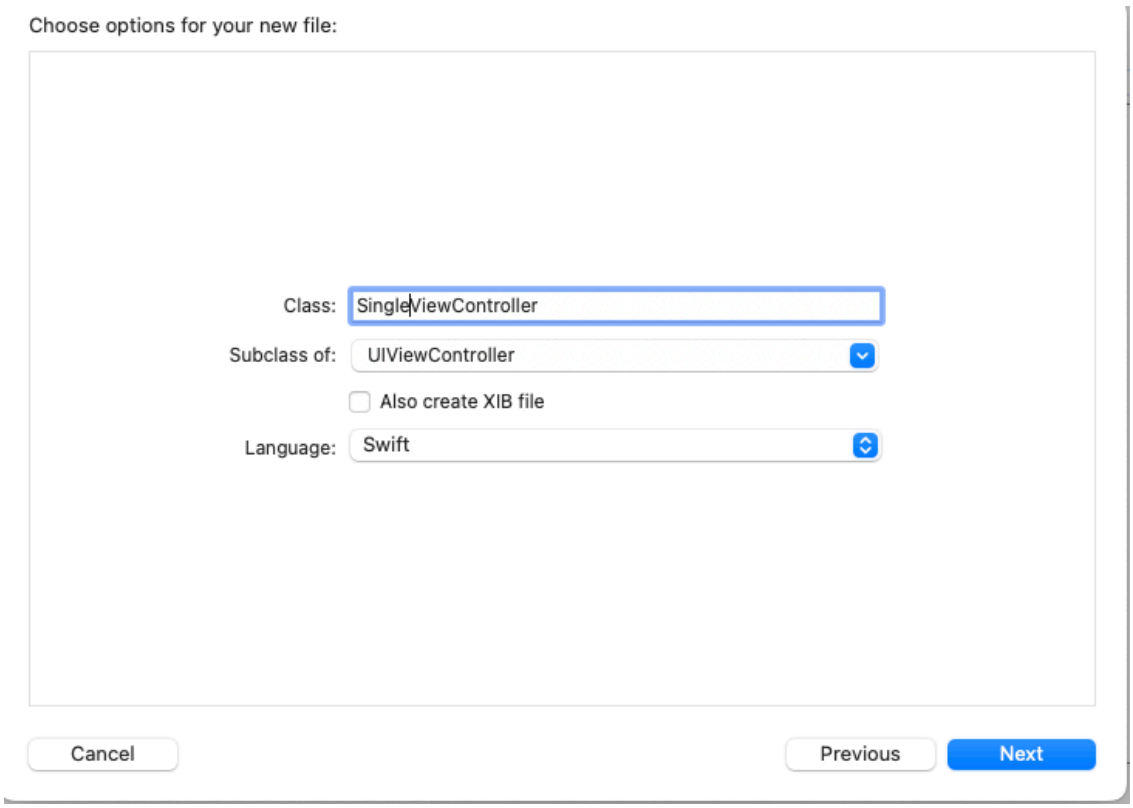

6. Set constraints: Editor—>Resolve … —> Reset …

7. Embed ViewController in Tab Bar Controller. Select menu Editor—>Embed In—>Tab Bar Controller

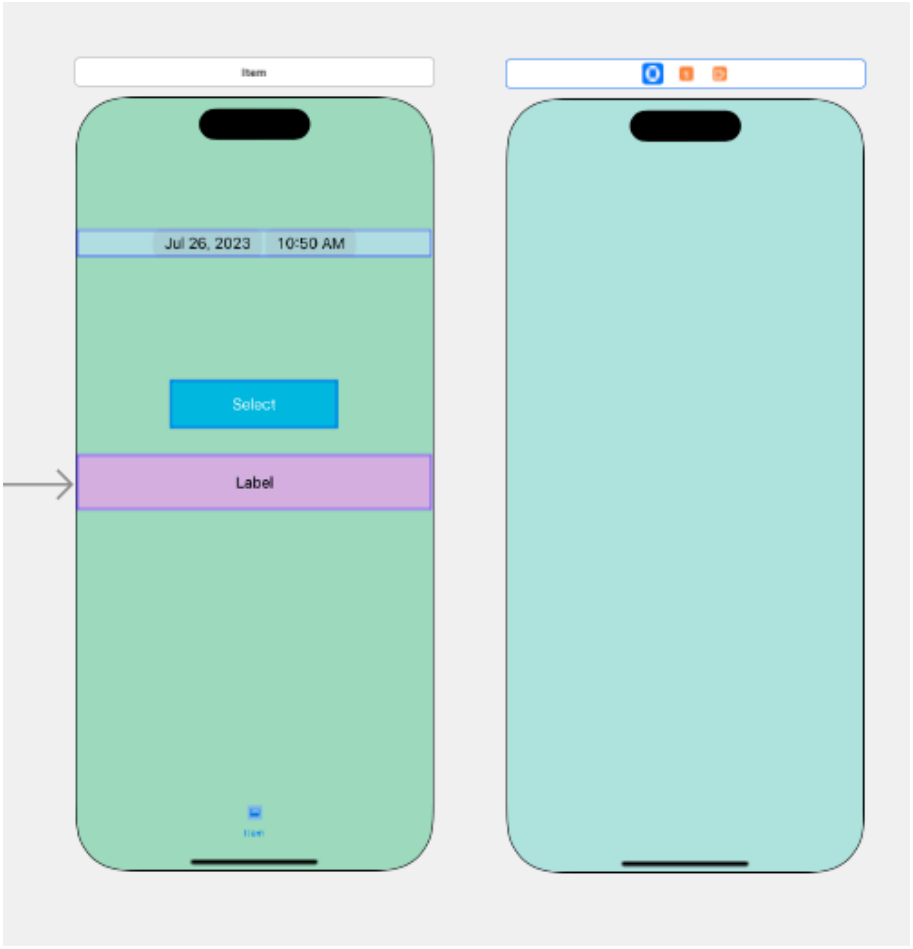



8. Add a class name: SingleViewController. Select menu File—> Niew —> File, Then select CocoTouch template.

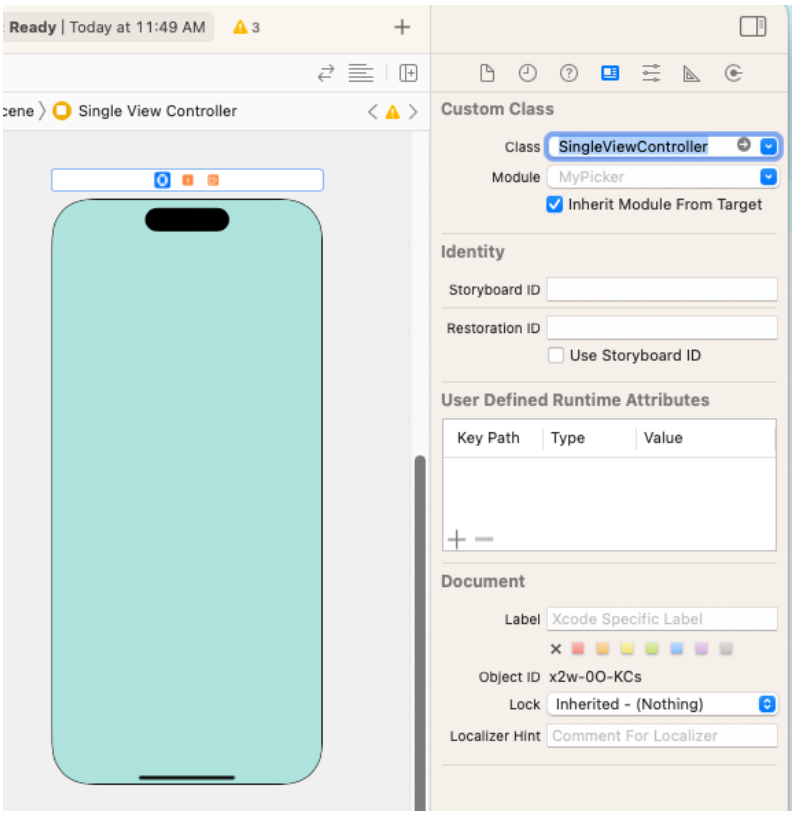





9. Create a new ViewController in storyboard.



10, Connect SingleViewController class to this new ViewController.

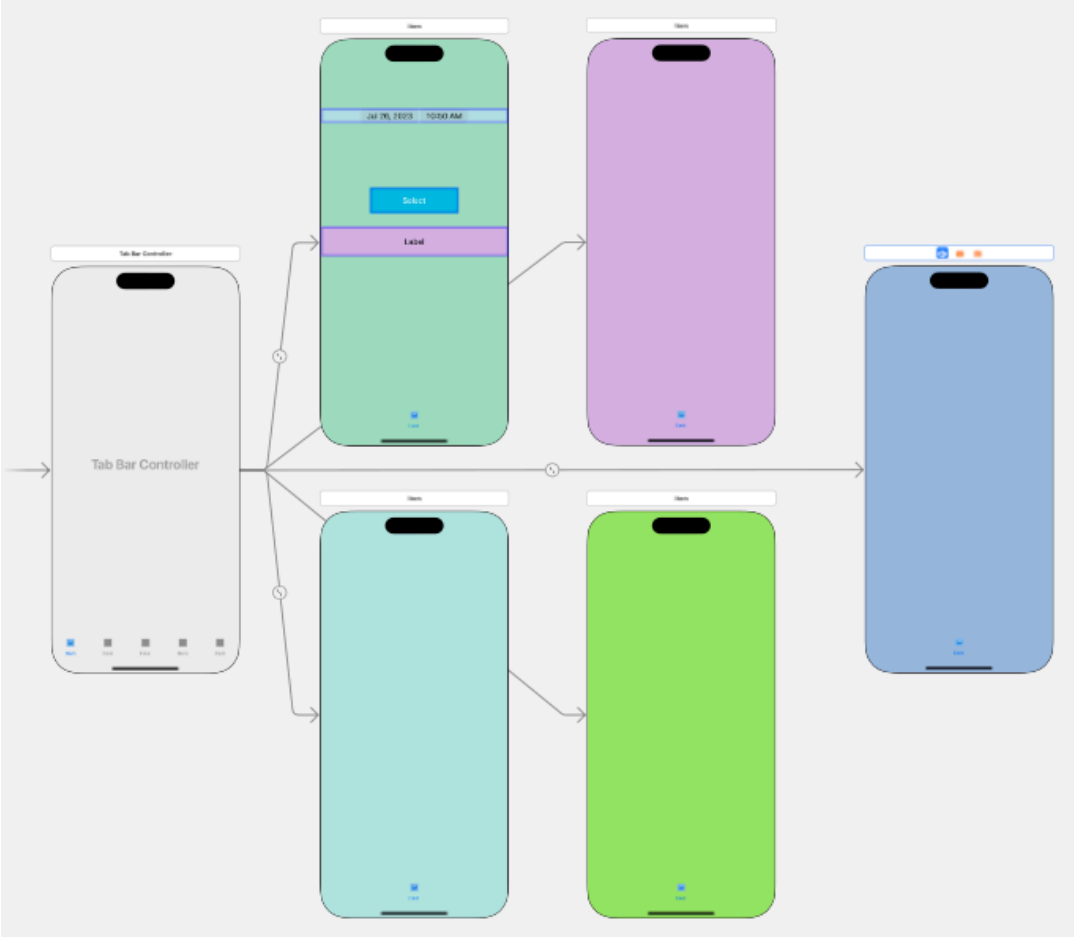



11. Add SingleViewController into Tab Bar Controller. Ctrl-Click on Tab Bar Controller then drag into SingleViewController and release. Select view controllers from Relationship Segue.

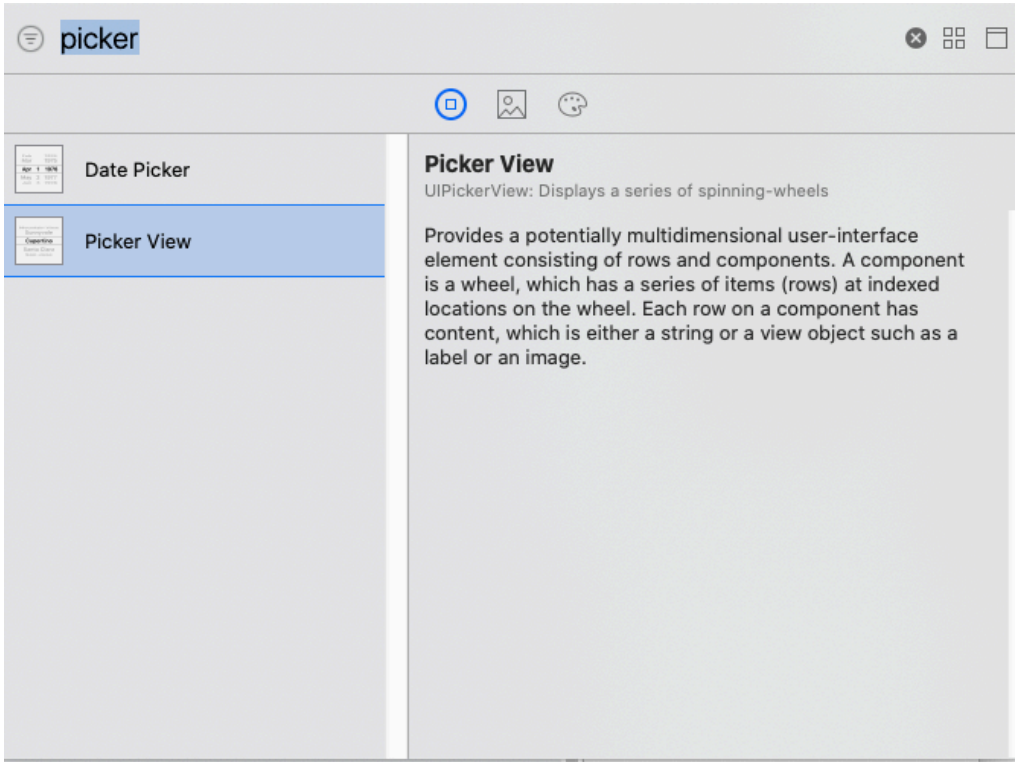
12. There are two view controllers under tab bar controller.

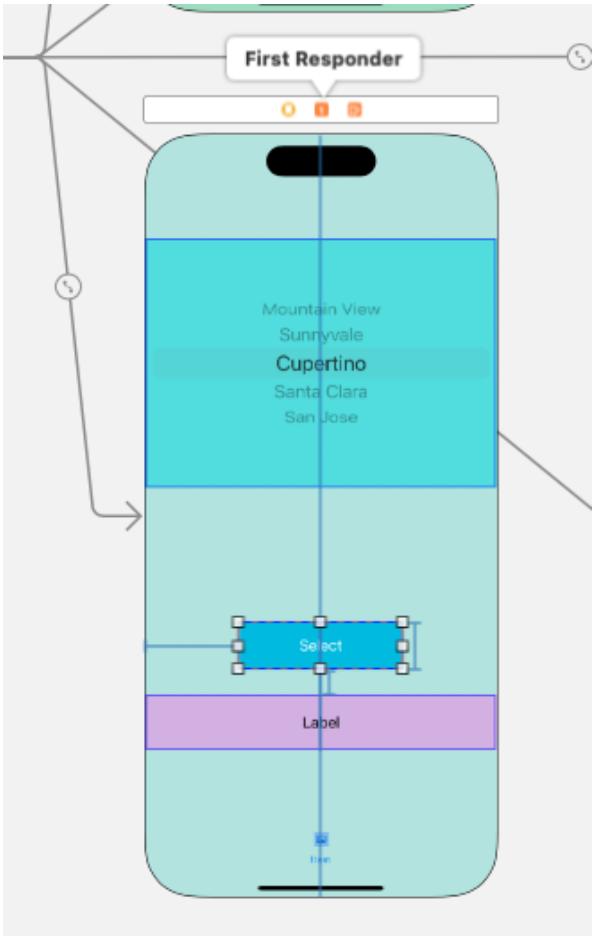

13. Repeat step 8-11 to create another three view controllers in storyboard:DoubleViewController, DependentViewController and CustomViewController.

14. There are five view controllers under tab bar controller.




15. Add a Picker View a button and a label in all view controller in storyboard. Then set constraints: Editor—>Resolve … —> Reset …






16. Add outlet to pickerview and label.

17. Add action to button.

18. Connect data source and delegate of the pickerview to the view controller in storyboard.

19. Add UIPickerViewDataSource and UIPickerViewDelegate protocol to the view controller in class.

20. Implement pickerview datasource and delegate in each view controller (see below code in each picker).

21. Add images to the project Assets..xcassets folder by dragging a folder *ImageSets* and drop it into the left column of the editing area, underneath AppIcon, select copy item into the project.

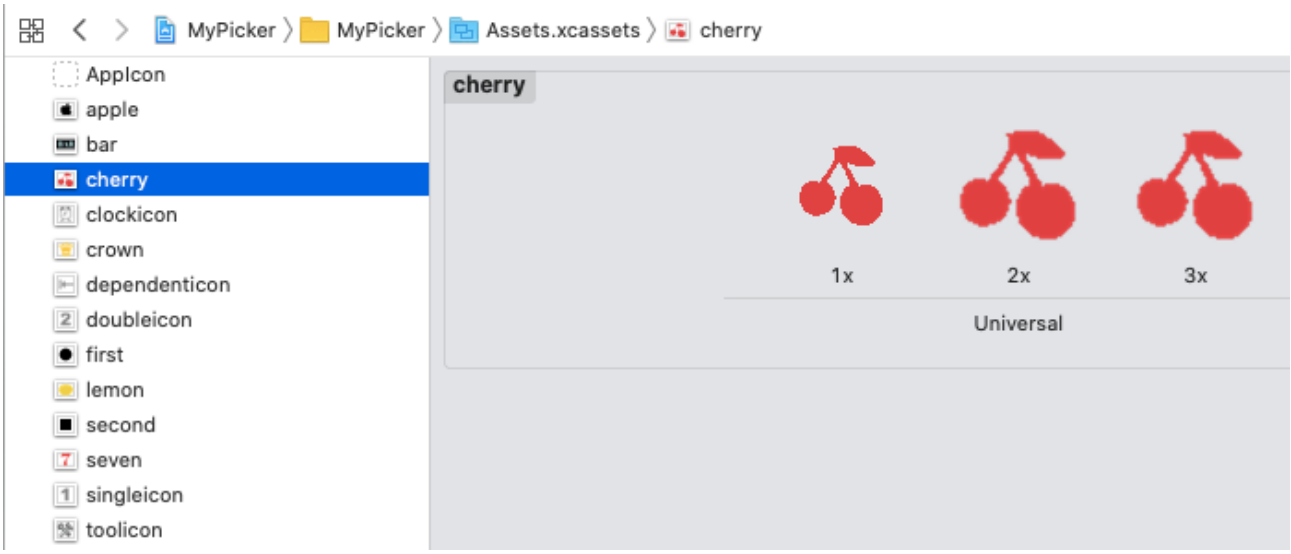
22. Set title and image for each tab bar item.

23. Add sound resources to the project.

24. Add statedictionay.plist to the project.

25. Implement others suppporting functions.

26. Run App.

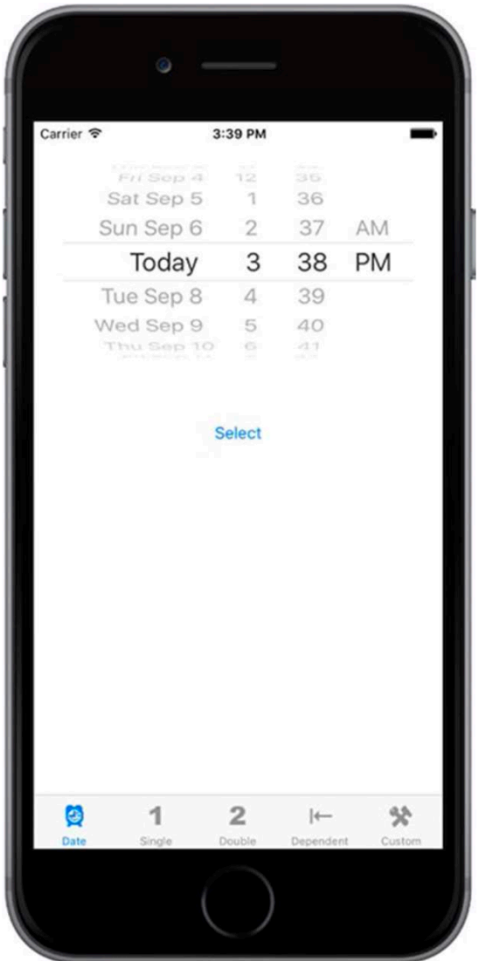



# Impmenting Pickers

1. Add Outlet for each picker in each class, and outlet for a label, and action for a button.
2. Add Data Source and Delegate for each of a picker view.

## Date Picker

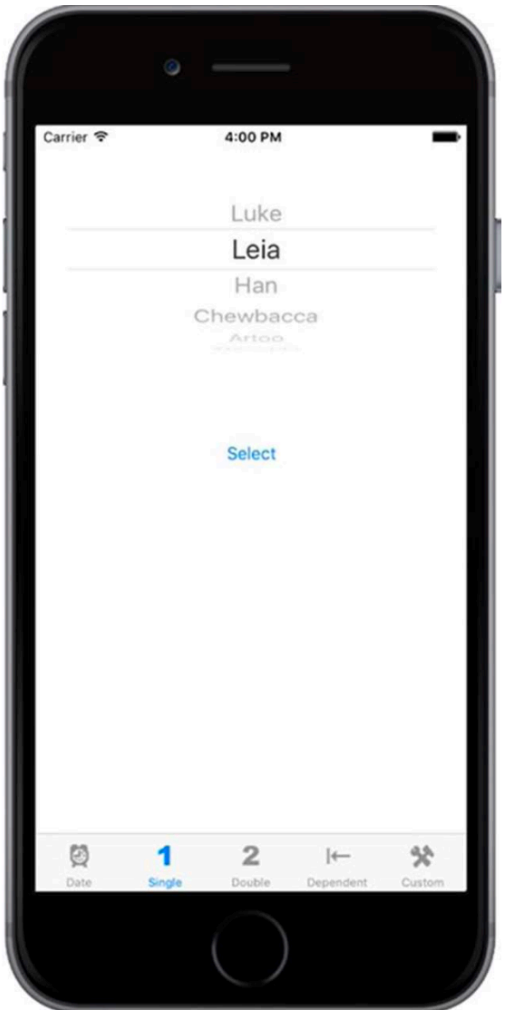
```
class FirstViewController: UIViewController {
    @IBOutlet var datePicker:UIDatePicker!
    @IBOutlet var lblSelect:UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
        let date = Date()
        datePicker.setDate(date, animated: false)
        lblSelect.text = ""
    }
    @IBAction func buttonPressed(sender: AnyObject) {
        let date = datePicker.date
        let message = "The date and time you selected is \(date)"
        let alert = UIAlertController(
            title: "Date and Time Selected",
            message: message,
            preferredStyle: .alert)
        let action = UIAlertAction(
            title: "That's so true!",
            style: .default,
            handler: nil)
        alert.addAction(action)
        present(alert, animated: true, completion: nil)
        lblSelect.text = message
    }
```




## Single Picker

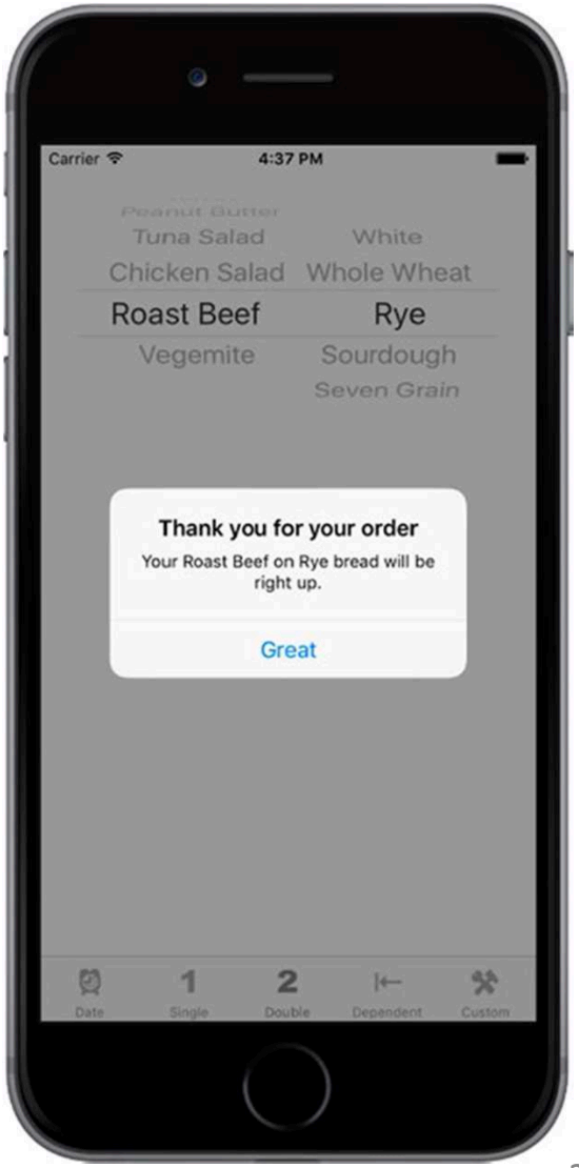
```
    @IBOutlet weak var myPicker: UIPickerView!
    @IBOutlet weak var lblResult: UILabel!

    private let characterNames = [
    "Luke", "Leia", "Han", "Chewbacca", "Artoo", "Threepio",
"Lando"]

 //MARK:- picker datasource
    func numberOfComponents(in pickerView: UIPickerView) -> Int {
        return 1
    }
    func pickerView(_ pickerView: UIPickerView,
numberOfRowsInComponent component: Int) -> Int {
        return characterNames.count

    }
    func pickerView(_ pickerView: UIPickerView, titleForRow row:
Int, forComponent component: Int) -> String? {
        return characterNames[row]

    }
```

```
    // MARK:- picker delegate
    func pickerView(_ pickerView: UIPickerView, didSelectRow row:
Int, inComponent component: Int) {
        lblResult.text = characterNames[row]
    }

@IBAction func bSelect(_ sender: Any) {
        let row = myPicker.selectedRow(inComponent: 0)
        let selected = characterNames[row]
        let title = "You selected \(selected)!"
        let alert = UIAlertController(
            title: title,
            message: "Thank you for choosing",
            preferredStyle: .alert)
        let action = UIAlertAction(
            title: "You're welcome",
            style: .default,
            handler: nil)
        alert.addAction(action)
        present(alert, animated: true, completion: nil)
        lblResult.text  = characterNames[row]
    }
```



## Double picker

```
@IBOutlet weak var doublePicker: UIPickerView!
@IBOutlet weak var lblResult: UILabel!
private let fillingComponent = 0
private let breadComponent = 1
private let fillingTypes = [
        "Ham", "Turkey", "Peanut Butter", "Tuna Salad",
        "Chicken Salad", "Roast Beef", "Vegemite"]
private let breadTypes = [ "White", "Whole Wheat", "Rye",
                    "Sourdough", "Seven Grain”]

// MARK: Picker Data Source Methods
    func numberOfComponents(in pickerView: UIPickerView) -> Int {
        return 2
    }

   func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent
component: Int) -> Int {
        if component == breadComponent {
            return breadTypes.count
        } else {
            return fillingTypes.count
        }
    }
    // MARK: Picker Delegate Methods
    func pickerView(_ pickerView: UIPickerView,titleForRow row: Int,
                    forComponent component: Int) -> String? {
        if component == breadComponent {
            return breadTypes[row]
        } else {
            return fillingTypes[row]
        }
    }

@IBAction func buttonPressed(sender: AnyObject) {
        let fillingRow = doublePicker.selectedRow(inComponent:
fillingComponent)
        let breadRow = doublePicker.selectedRow(inComponent:
breadComponent)
        let filling = fillingTypes[fillingRow]
        let bread = breadTypes[breadRow]
        let message = "Your \(filling) on \(bread) bread will be right
up."
        let alert = UIAlertController(
            title: "Thank you for your order",
            message: message,
            preferredStyle: .alert)
        let action = UIAlertAction(
            title: "Great", style: .default, handler: nil)
        alert.addAction(action)
        present(alert, animated: true, completion: nil)

        lblResult.text = message
    }
```

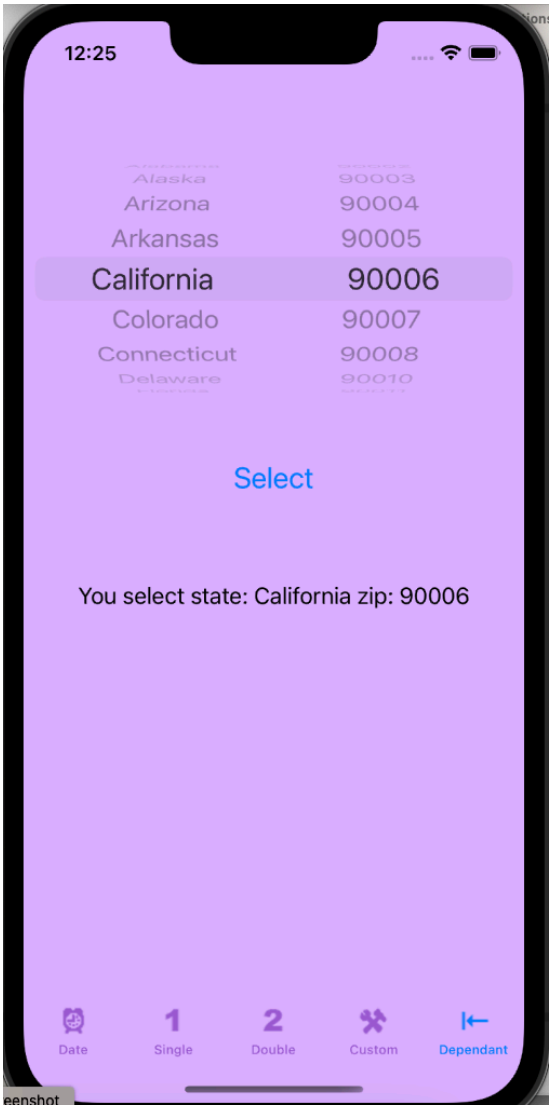



## Dependent picker

```
@IBOutlet weak var dependentPicker: UIPickerView!
    @IBOutlet weak var lblSelected: UILabel!
    private let stateComponent:Int = 0
    private let zipComponent = 1
    private var stateZips:[String : [String]]!
    private var states:[String]!
    private var zips:[String]!
    var selectedState = ""
    var selectedZip = ""

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
        let bundle = Bundle.main
```

```
        let plistURL = bundle.url(forResource: "statedictionary",
                                  withExtension: "plist")
        stateZips = NSDictionary(contentsOf: plistURL!) as? [String :
[String]]
        let allStates = stateZips.keys
        states = allStates.sorted(by: <)
        let selectedState = states[0]
        zips = stateZips[selectedState]
    }

    // MARK: User Functions
    @IBAction func buttonPressed(_ sender: Any) {
        let stateRow =
        dependentPicker.selectedRow(inComponent: stateComponent)
        let zipRow =
        dependentPicker.selectedRow(inComponent: zipComponent)
        let state = states[stateRow]
        let zip = zips[zipRow]
        let title = "You selected zip code \(zip)"
        let message = "\(zip) is in \(state)"
        let alert = UIAlertController(
            title: title,
            message: message,
            preferredStyle: .alert)
        let action = UIAlertAction(
            title: "OK",
            style: .default,
            handler: nil)
        alert.addAction(action)
        present(alert, animated: true, completion: nil)
    }

    // MARK: Picker Data Source Methods
    func numberOfComponents(in pickerView: UIPickerView) -> Int {
        return 2
    }

    func pickerView(_ pickerView: UIPickerView,
                    numberOfRowsInComponent component: Int) -> Int {
        if component == stateComponent {
            return states.count
        } else {
            return zips.count
        }
    }

    // MARK: Picker Delegate Methods
    func pickerView(_ pickerView: UIPickerView, titleForRow row: Int,
                    forComponent component: Int) -> String? {
        if component == stateComponent {
            return states[row]
        } else {
            return zips[row]
        }
    }
```

```
    func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int,
                    inComponent component: Int) {

        if component == stateComponent {
            selectedState = states[row]
            zips = stateZips[selectedState]
            selectedZip = zips[0]
            dependentPicker.reloadComponent(zipComponent)
            dependentPicker.selectRow(0, inComponent: zipComponent,
                                      animated: true)
        }
        else {
            zips = stateZips[selectedState]
            selectedZip = zips[row]
            dependentPicker.reloadComponent(zipComponent)
        }
        lblSelected.text = "You select state: \(selectedState) zip: \
(selectedZip)"
    }
```



## Custom picker

เพิ่มเสียงเมื่อกดปุ่มและเมื่อชนะเกมส์ ให้เพิ่มเฟรมเวิร์คจัดการเสียง

import UIKit

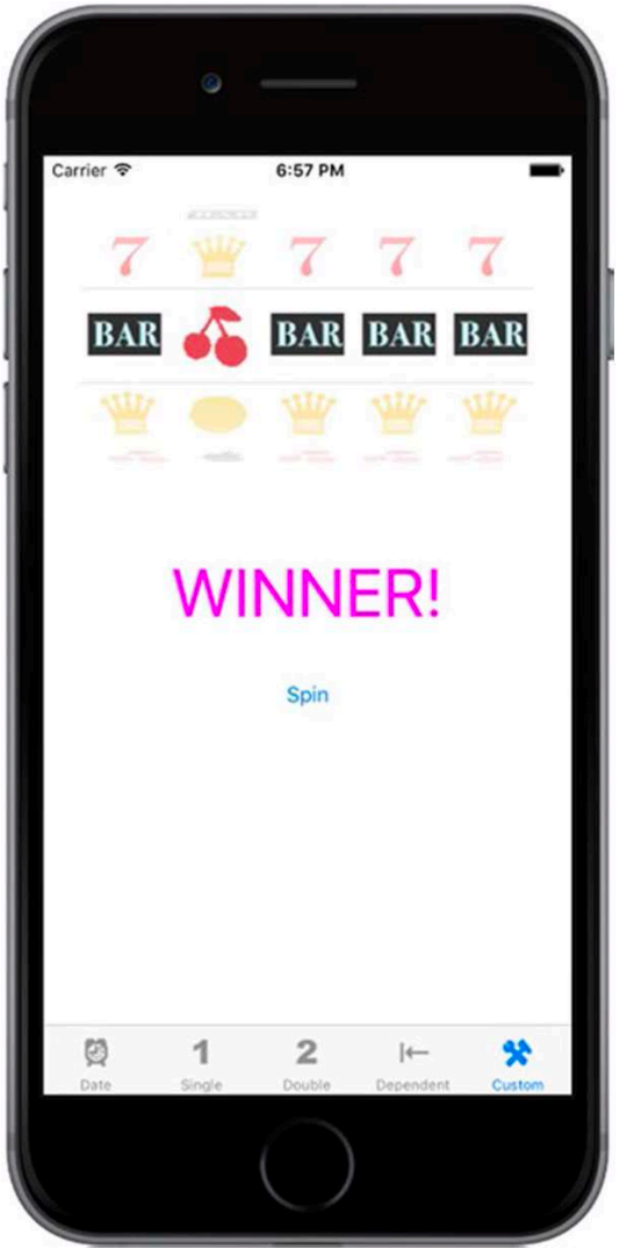
Import AudioToolbox

เพิ่มเอาท์เลตชื่อ button ให้กับปุ่มกด  เพื่อซ่อนและแสดงปุ่มกด

ปรับชื่อฟังก์ชันของปุ่มกดให้ตรงกัน (spin)

```
    @IBOutlet weak var myPicker: UIPickerView!
    @IBOutlet weak var lblWin: UILabel!
    private var images:[UIImage]!
    @IBOutlet weak var button: UIButton!
    private var winSoundID: SystemSoundID = 0
    private var crunchSoundID: SystemSoundID = 0
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
        images = [
        UIImage(named: "seven")!,
        UIImage(named: "bar")!,
        UIImage(named: "crown")!,
        UIImage(named: "cherry")!,
        UIImage(named: "lemon")!,
        UIImage(named: "apple")!
        ]
        lblWin.text = " " // Note the space between the quotes
        initSpin()
    }
    func initSpin() {
        for i in 0..<5 {
            let newValue = Int.random(in: 0..<images.count)
            myPicker.selectRow(newValue, inComponent: i, animated: true)
            myPicker.reloadComponent(i)
        }
    }

     func showButton() {
        button.isHidden = false
    }
    func playWinSound() {
        if winSoundID == 0 {
            let soundURL = Bundle.main.url(forResource:
                "win", withExtension: "wav")! as CFURL
            AudioServicesCreateSystemSoundID(soundURL, &winSoundID)
        }
        AudioServicesPlaySystemSound(winSoundID)
        lblWin.text = "WINNER!"
        DispatchQueue.main.asyncAfter(deadline: .now() + 1.5,execute: {
            self.showButton()
        })
    }
```

```
@IBAction func spin(_ sender: Any) {
        var win = false
        var numInRow = -1
        var lastVal = -1
        for i in 0..<5 {
            let newValue = Int.random(in: 0..<images.count)
            if newValue == lastVal {
                numInRow += 1
            } else {
                numInRow = 1
            }
            lastVal = newValue
            myPicker.selectRow(newValue, inComponent: i, animated: true)
            myPicker.reloadComponent(i)
            if numInRow >= 3 {
                win = true
            }
        }
        if crunchSoundID == 0 {
            let soundURL = Bundle.main.url(forResource:
                "crunch", withExtension: "wav")! as CFURL
            AudioServicesCreateSystemSoundID(soundURL, &crunchSoundID)
        }
        AudioServicesPlaySystemSound(crunchSoundID)
        if win {
            DispatchQueue.main.asyncAfter(deadline: .now() + 0.5,
                    execute:  {self.playWinSound()})
        } else {
            DispatchQueue.main.asyncAfter(deadline: .now() + 0.5,
                    execute: { self.showButton()})
        }
        button.isHidden = true
        lblWin.text = " " // Note the space between the quotes
    }
// MARK: Picker Data Source Methods
    func numberOfComponents(in pickerView: UIPickerView) -> Int {
        return 5
    }
    func pickerView(_ pickerView: UIPickerView,
                    numberOfRowsInComponent component: Int) -> Int {
        return images.count
    }

    // MARK: Picker Delegate Methods
    func pickerView(_ pickerView: UIPickerView, viewForRow row: Int,
                    forComponent component: Int,
                    reusing view: UIView?) -> UIView {
        let image = images[row]
        let imageView = UIImageView(image: image)
        return imageView
    }
    func pickerView(_ pickerView: UIPickerView,
                    rowHeightForComponent component: Int) -> CGFloat {
        return 64
    }
```


Carrier
6:57 PM
WINNER!
Spin
Date
1
Single
2
Double
Dependent
Custom