```
//
//  RecordSoundsViewController.swift
//  my pitch perfect
//
//  Created by Kathleen Stukenborg on 9/28/15.
//  Copyright © 2015 Kathleen Stukenborg. All rights reserved.
//

import UIKit
import AVFoundation

class RecordSoundsViewController: UIViewController, AVAudioRecorderDelegate {
    var audioRecorder:AVAudioRecorder!
    //create new object for class RecordedAudio
    var recordedAudio:RecordedAudio! //I have to put the ! on it

    override func viewWillAppear(animated: Bool) {
        recordingLabel.hidden = true
        stopOutlet.hidden = true
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBAction func record(sender: UIButton) {
        let dirPath = NSSearchPathForDirectoriesInDomains(.DocumentDirectory, .
                    UserDomainMask, true)[0] as String

        //let currentDateTime = NSDate()
        //let formatter = NSDateFormatter()
        //formatter.dateFormat = "ddMMyyyy-HHmmss"
        //let recordingName = formatter.stringFromDate(currentDateTime)+".wav"
        let recordingName = "my_audio.wav"
        let pathArray = [dirPath, recordingName]
        let filePath = NSURL.fileURLWithPathComponents(pathArray)
        print(filePath)

        let session = AVAudioSession.sharedInstance()
        try! session.setCategory(AVAudioSessionCategoryPlayAndRecord)

        try! audioRecorder = AVAudioRecorder(URL: filePath!, settings: [:])
        // when we become audioRecorder's delegate, we can run all of the
                    functions of audioRecorder
        audioRecorder.delegate = self
        audioRecorder.meteringEnabled = true
        audioRecorder.prepareToRecord()
        audioRecorder.record()
        recordingLabel.hidden = false
        stopOutlet.hidden = false
        recordOutlet.enabled = false
        print("in recordButton")

    }

    func audioRecorderDidFinishRecording(recorder: AVAudioRecorder, successfully
             flag: Bool) {
        print("in audioRecorderDidFinishRecording")
        //TODO: save audio
        if flag {
            recordedAudio = RecordedAudio() //initialize it
            print("initialized recordedAudio")
            recordedAudio.filePathUrl = recorder.url //recorder is the parameter
                          passed in
            print("set filePathUrl to parameter recroder.url")

            recordedAudio.title = recorder.url.lastPathComponent! //this gives us
                          the title
            print("got title")
        //TODO: Move audio to segue
            self.performSegueWithIdentifier("stopRecording", sender:
                          recordedAudio)
        } else {
            print("Recording was not successful")

        }
        print("leaving audioRecorderDidFinishRecording")
    }

    override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {


        if(segue.identifier == "stopRecording"){
            print("in prepareForSegue")
            let playSoundsVC:PlaySoundsViewController = segue.
                          destinationViewController as! PlaySoundsViewController
            let data = sender as! RecordedAudio //get the data that was passed
                          in.
            playSoundsVC.receivedAudio  = data
        }

    }
    @IBAction func stopAction(sender: UIButton) {
        recordingLabel.hidden = true
        stopOutlet.hidden = true
        recordOutlet.enabled = true
        audioRecorder.stop()
        let audioSession = AVAudioSession.sharedInstance()
        try! audioSession.setActive(false)
        print("in stopAction")

    }

    @IBOutlet weak var recordOutlet: UIButton!
    @IBOutlet weak var stopOutlet: UIButton!
    @IBOutlet weak var recordingLabel: UILabel!
    }
```