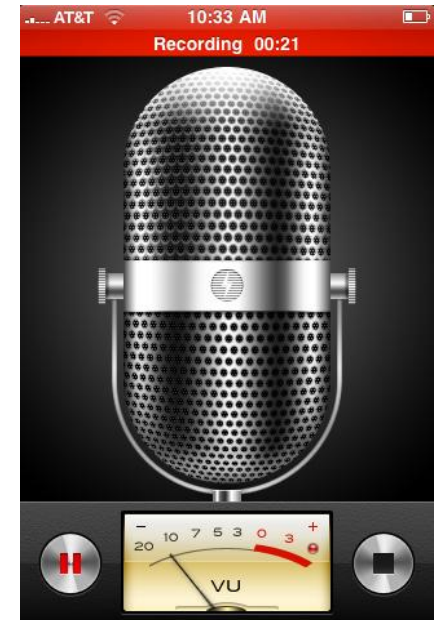




Multimedia

Multimedia

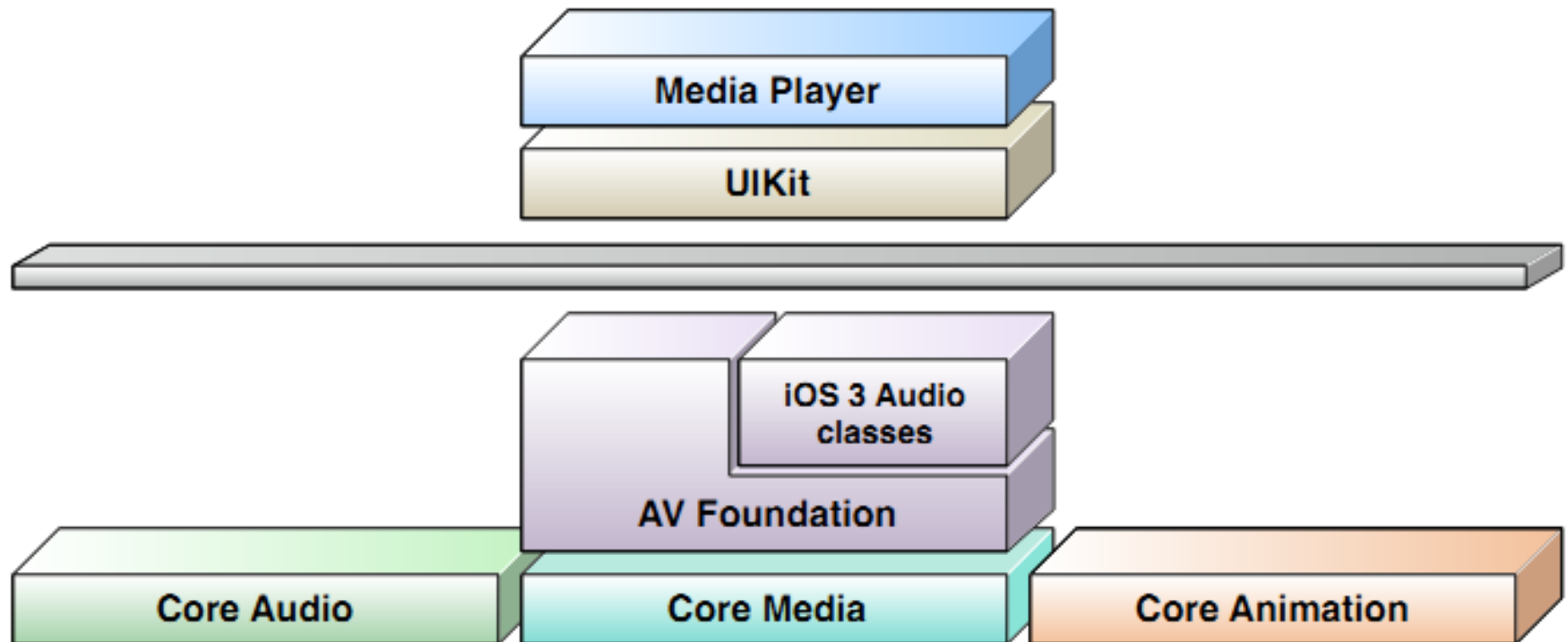




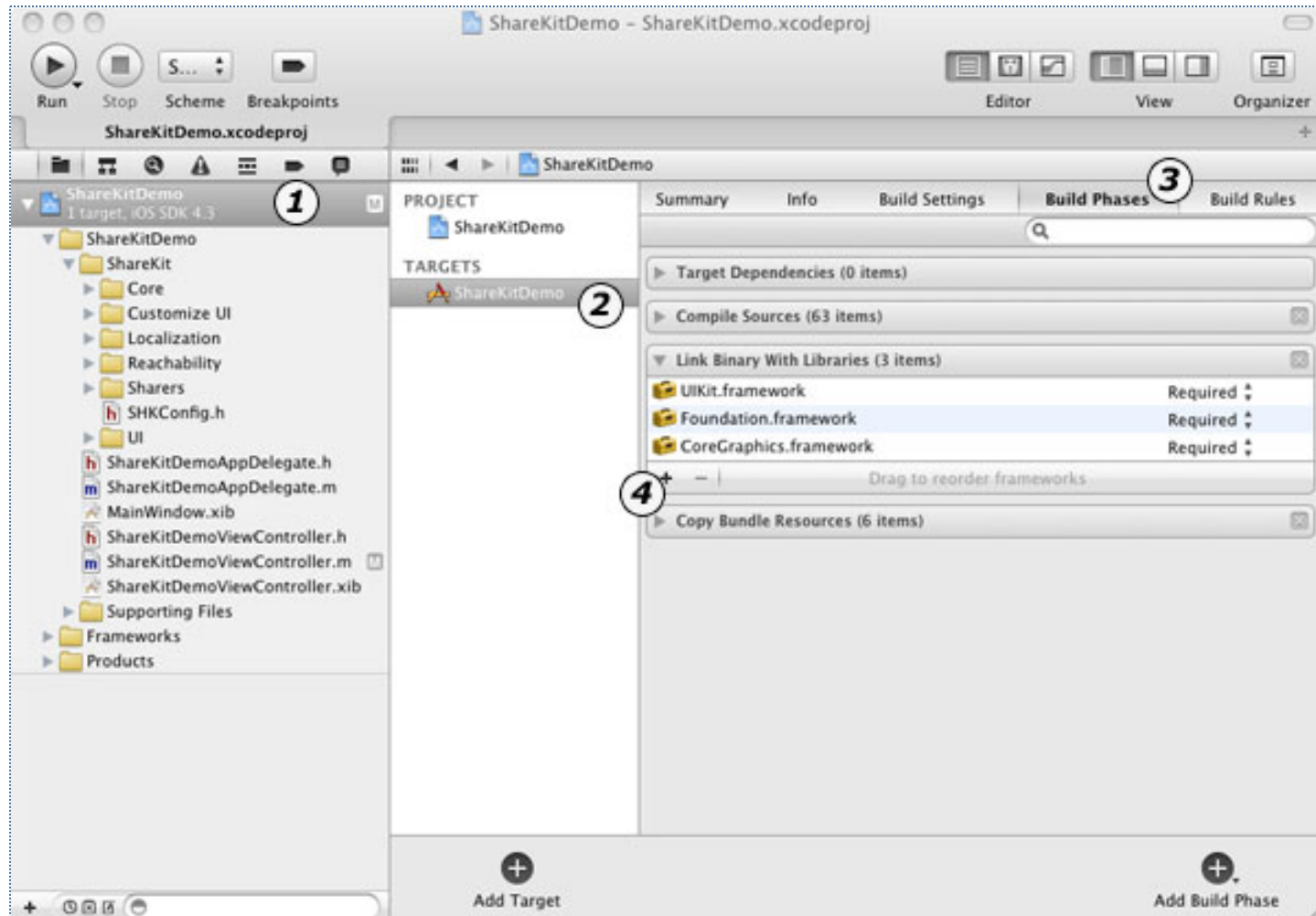
Các thư viện và lớp thường dùng

- AVFoundation Framework
- MediaPlayer Framework
- AVAudioPlayer class
- AVAudioRecorder class
- AVAudioSession class
- AVCaptureSession class
- MPMoviePlayerController class
- MPMediaPickerController class

AVFoundation Framework



Thêm Framework





AVAudioPlayer

```
@interface Playing_Audio_FilesViewController
    : UIViewController <AVAudioPlayerDelegate>
@property (nonatomic, strong) AVAudioPlayer *audioPlayer;
@end

NSBundle *mainBundle = [NSBundle mainBundle];
NSString *filePath = [mainBundle pathForResource:@"MySong"
                                         ofType:@"mp3"];
NSData *fileData = [NSData dataWithContentsOfFile:filePath];
NSError *error = nil;
/* Start the audio player */
self.audioPlayer = [[AVAudioPlayer alloc] initWithData:fileData
                                         error:&error];
```



Ví dụ AVAudioPlayer

- Viết ứng dụng mở một tập tin MP3



Handling Interruptions While Playing Audio

```
- (void)audioPlayerBeginInterruption:(AVAudioPlayer *)player{  
  
    /* Audio Session is interrupted. The player will be paused  
    here */  
  
}  
- (void)audioPlayerEndInterruption:(AVAudioPlayer *)player  
    withFlags:(NSUInteger)flags{  
  
    if (flags == AVAudioSessionInterruptionFlags_ShouldResume  
        && player != nil){  
        [player play];  
    }  
}
```




Ví dụ Handling Interruptions While Playing Audio

- Viết ứng dụng xử lý sự kiện tạm dừng mở tập tin Audio và tiếp tục mở khi hết sự kiện



Recording Audio

```
#import <UIKit/UIKit.h>
#import <CoreAudio/CoreAudioTypes.h>
#import <AVFoundation/AVFoundation.h>

@interface Recording_AudioViewController : UIViewController
    <AVAudioPlayerDelegate, AVAudioRecorderDelegate>
@property (nonatomic, strong) AVAudioRecorder
                                *audioRecorder;
@property (nonatomic, strong) AVAudioPlayer *audioPlayer;
- (NSString *)    audioRecordingPath;
- (NSDictionary *) audioRecordingSettings;
@end
```



Recording Audio

```
NSError *error = nil;
NSString *pathAsString = [self audioRecordingPath];
NSURL *audioRecordingURL = [NSURL
fileURLWithPath:pathAsString];
self.audioRecorder = [[AVAudioRecorder alloc]
initWithURL:audioRecordingURL
settings:[self audioRecordingSettings]
error:&error];
```



Recording Audio

```
- (NSString *) audioRecordingPath{

    NSString *result = nil;
    NSArray *folders =
        NSSearchPathForDirectoriesInDomains
        (NSDocumentDirectory,
         NSUserDomainMask, YES);
    NSString *documentsFolder = [folders
                                   objectAtIndex:0];
    result = [documentsFolder
               stringByAppendingPathComponent:@"Recording.m4a"];
    return result;

}
```



Recording Audio

```
- (NSDictionary *) audioRecordingSettings{
    NSDictionary *result = nil;
    NSMutableDictionary *settings = [[NSMutableDictionary alloc] init];
    [settings setValue:[NSNumber numberWithInt:
                        kAudioFormatAppleLossless]
                    forKey:AVFormatIDKey];
    [settings setValue:[NSNumber numberWithFloat:44100.0f]
                    forKey:AVSampleRateKey];
    [settings setValue:[NSNumber numberWithInt:1]
                    forKey:AVNumberOfChannelsKey];
    [settings setValue:[NSNumber numberWithInt:
                        AVAudioQualityLow]
                    forKey:AVEncoderAudioQualityKey];
    result = [NSDictionary dictionaryWithDictionary:settings];
    return result; }
```



Recording Audio

- `audioRecordingSettings`
 - `AVFormatIDKey`
 - `kAudioFormatLinearPCM`
 - `kAudioFormatAppleLossless`
 - `AVSampleRateKey`
 - `AVNumberOfChannelsKey`
 - `AVEncoderAudioQualityKey`
 - `AVAudioQualityMin`
 - `AVAudioQualityLow`
 - `AVAudioQualityMedium`
 - `AVAudioQualityHigh`
 - `AVAudioQualityMax`



Ví dụ Recording Audio

- Viết ứng dụng thu âm



Handling Interruptions While Recording Audio

```
- (void)audioRecorderBeginInterruption:(AVAudioRecorder *)recorder{
    NSLog(@"Recording process is interrupted");
}
- (void)audioRecorderEndInterruption:(AVAudioRecorder *)recorder
    withFlags:(NSUInteger)flags{
    if (flags == AVAudioSessionInterruptionFlags_ShouldResume){
        NSLog(@"Resuming the recording...");
        [recorder record];
    }
}
```




Playing Video Files

```
#import <UIKit/UIKit.h>
#import <MediaPlayer/MediaPlayer.h>
@interface Playing_Video_FilesViewController :
    UIViewController
@property (nonatomic, strong) MPMoviePlayerController
    *moviePlayer;
@property (nonatomic, strong) UIButton *playButton;
@end
```



Playing Video Files

```
NSBundle *mainBundle = [NSBundle mainBundle];
NSString *urlAsString = [mainBundle
                        pathForResource:@"Sample"
                        ofType:@"m4v"];
NSURL *url = [NSURL fileURLWithPath:urlAsString];
self.moviePlayer = [[MPMoviePlayerController alloc]
                    initWithContentURL:url];
```

Ví dụ Playing Video Files

- Viết ứng dụng mở tập tin Video



Capturing Thumbnails from a Video File

```
NSNumber *thirdSecondThumbnail = [NSNumber  
                                   numberWithFloat:3.0f];  
NSArray *requestedThumbnails =  
    [NSArray arrayWithObject:thirdSecondThumbnail];  
  
[self.moviePlayer requestThumbnailImagesAtTimes:  
                                   requestedThumbnails  
                                   timeOption:MPMovieTimeOptionExact];
```



Ví dụ Capturing Thumbnails from a Video File

- Viết ứng dụng Capture hình từ Video

Accessing the Music Library

```
#import <UIKit/UIKit.h>
#import <MediaPlayer/MediaPlayer.h>
@interface Accessing_the_Music_LibraryViewController
    :UIViewController <MPMediaPickerControllerDelegate>
@end
```

```
MPMediaPickerController *mediaPicker =
    [[MPMediaPickerController alloc]
     initWithMediaTypes:MPMediaTypeAny];
[self.navigationController presentViewController:
    mediaPicker
    animated:YES];
```

Accessing the Music Library

```
@interface Accessing_the_Music_LibraryViewController :  
UIViewController <MPMediaPickerControllerDelegate,  
                    AVAudioPlayerDelegate>  
  
@property (nonatomic, strong) MPMusicPlayerController  
                    *myMusicPlayer;  
  
@property (nonatomic, strong) UIButton  
                    *buttonPickAndPlay;  
  
@property (nonatomic, strong) UIButton  
                    *buttonStopPlaying;  
  
@end
```

Accessing the Music Library

- Trong phương thức viewDidLoad

```
[self.buttonPickAndPlay setTitle:@"Pick and Play"
                           forState:UIControlStateNormal];

[self.buttonPickAndPlay addTarget:self
                           action:@selector(displayMediaPlayerAndPlayItem)
                           forControlEvents:UIControlEventTouchUpInside];

[self.view addSubview:self.buttonPickAndPlay];

self.buttonStopPlaying = [UIButton
                           buttonWithType:UIButtonTypeRoundedRect];

[self.buttonStopPlaying setTitle:@"Stop Playing"
                           forState:UIControlStateNormal];

[self.buttonStopPlaying addTarget:self
                           action:@selector(stopPlayingAudio)
                           forControlEvents:UIControlEventTouchUpInside];

[self.view addSubview:self.buttonStopPlaying];

[self.navigationController setNavigationBarHidden:YES animated:NO];
```


Accessing the Music Library

- Trong phương thức **displayMediaPlayerAndPlayItem**

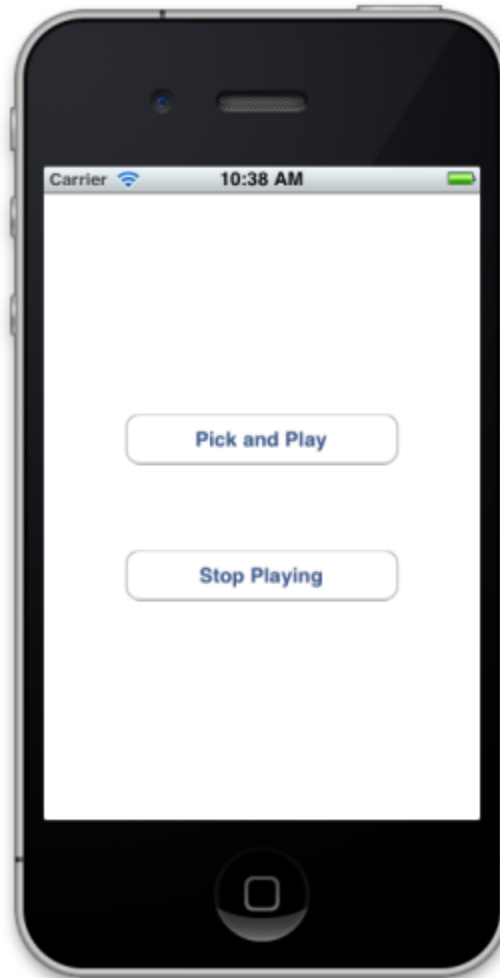
```
MPMediaPickerController *mediaPicker =  
    [[MPMediaPickerController alloc]  
     initWithMediaTypes:MPMediaTypeMusic];  
mediaPicker.delegate = self;  
mediaPicker.allowsPickingMultipleItems = YES;  
[self.navigationController  
 presentModalViewController:mediaPicker  
 animated:YES];
```

Accessing the Music Library

- Trong phương thức **stopPlayingAudio**

```
[[NSNotificationCenter defaultCenter] removeObserver:self  
name:MPMusicPlayerControllerPlaybackStateDidChangeNotification  
object:self.myMusicPlayer];  
  
[[NSNotificationCenter defaultCenter] removeObserver:self  
name:MPMusicPlayerControllerNowPlayingItemDidChangeNotification  
object:self.myMusicPlayer];  
  
[[NSNotificationCenter defaultCenter] removeObserver:self  
name:MPMusicPlayerControllerVolumeDidChangeNotification  
object:self.myMusicPlayer];  
[self.myMusicPlayer stop];
```

Ví dụ Accessing the Music Library





Q&A