APACHE CORDOVA™

Deployment Instructions and Usage Description

Sound Recorder Application

TEAM 2

Tom Birmingham, David Boyer, Joy Oyi, and Rob Van Zandt

CSC 550
Fundamentals of Mobile Application Development
Southern Connecticut State University
Spring 2021

TABLE OF CONTENTS

Getting Started

This document provides the steps for running the Sound Recorder application on Android and iOS devices. The Sound Recorder application allows users to record their own voices (or other sounds) and play back the corresponding audio file. The Sound Recorder can be run on both physical and virtual devices, and the steps for running the application on each type are provided below.

Prerequisites

This guide assumes you have installed the **Apache Cordova CLI** (v10.0.0). Cordova can be installed using Node package manager (run ***npm install cordova***). Please refer to the official Apache Cordova documentation for complete instructions: https://cordova.apache.org/docs/en/latest/guide/cli/

This guide also assumes you have installed the required components to build and run applications on both Android and iSO platforms. A requirements check can be performed in the command-line by running the ***cordova requirements*** command. This command will return a list of components that are installed on or missing from your machine. For further instruction, please refer to each of the target platform guides official Cordova documentation:

https://cordova.apache.org/docs/en/10.x/guide/platforms/android/index.html
https://cordova.apache.org/docs/en/10.x/guide/platforms/ios/index.html

There are two methods for setting up the Sound Recorder application. One method requires that the Git command-line tool be installed and involves cloning the entire project from GitHub. This is the simplest and preferred method for obtaining the project directory, as it eliminates many possible points of failure. We also provide instructions on how to roll-your-own Sound Recorder app from scratch and drop Team 2's changes on top.

Some users report issues with their platform after cloning the project. It is important that all users after cloning the project or adding new files **add and remove their target platform**. Steps are detailed below.

Clone the Project from Team 2's GitHub

There are two methods for cloning Team 2's Sound Recorder application. The first step involves a traditional clone of the project from GitHub using the *git clone* command. If you do not have git installed on your development machine, then you can download the project source code in a web browser and extract it to a local directory.

Create a new folder in your home directory like ***Cordova***. Enter the folder from the command line and run the following command to clone Team 2's Sound Recorder project:

git clone https://github.com/bigcoffeemug/soundRecorder

Git will fetch all the project files and place them in a new folder called ***soundRecorder***. Open the folder and inspect the contents. Folders node_modules, platforms, plugins, www, and project files have all been provided for you.

NOTE: If you do not have git installed, go to https://github.com/bigcoffeemug/soundRecorder. Click on the green “Code” button and select *Download ZIP*. The contents of the repository are downloaded. Extract the soundRecorder folder and place it inside the ***Cordova*** folder.

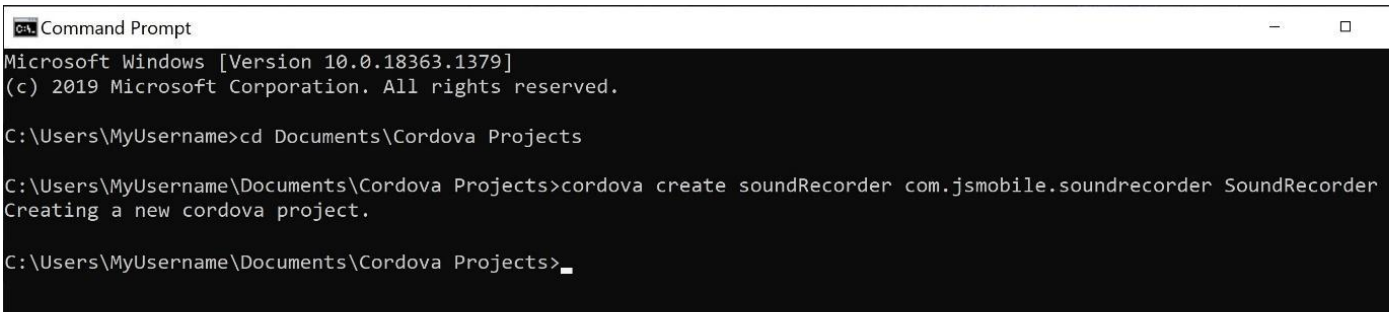
If you would prefer to “roll-your-own” Sound Recorder application using Team 2’s changes, please proceed with the following steps.

Building Sound Recorder App from Scratch

The following steps demonstrate how to build and run the Sound Recorder application from scratch, including fetching necessary project files and Team 2 customizations from Team 2’s shared directory on Google Drive.

Create the Project

Once you have installed Apache Cordova, use the following command to create the sound recorder application: ***cordova create soundRecorder com.jsmobile.soundrecorder soundRecorder***

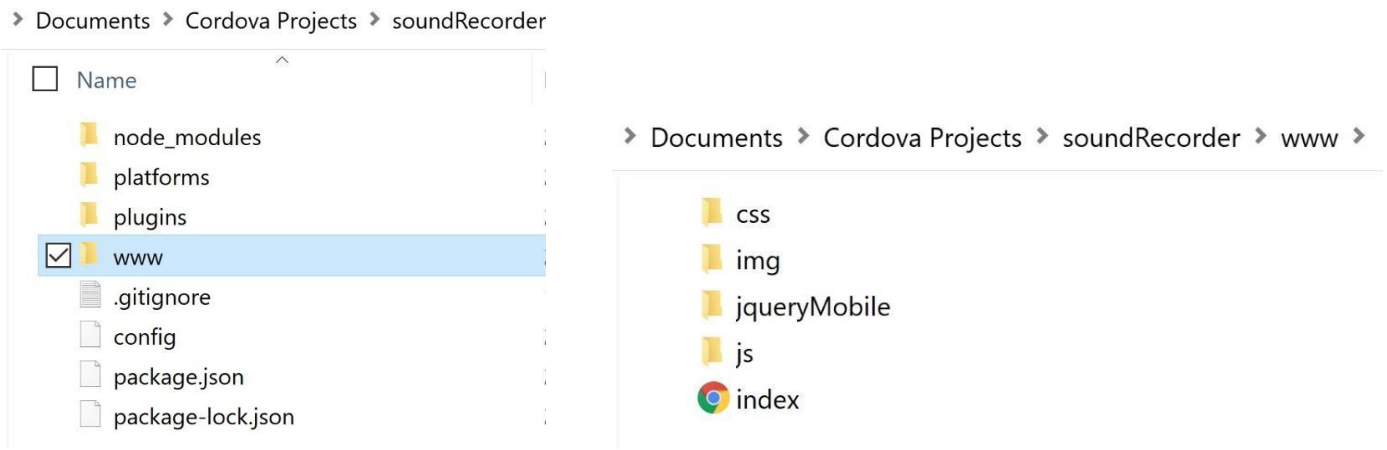


The folders shown below will now appear within your project directory. Note that the *www* folder contains HTML, CSS, and JavaScript files, which are necessary for running our application. It also contains an *image* folder.

Add Necessary Project Files and Team 2 Customizations

Before running the Sound Recorder application, an additional folder called jqueryMobile is required within the www/ directory. Team 2 has made this folder available along with their customizations for the application on Google Drive. Remove your default www/ folder and config.xml file in the project root directory and replace it with Team 2’s www/ folder and config.xml file found at the URL below:

https://drive.google.com/drive/folders/11Y90d7CGCrBpMclzKv_t9cgHIApmbB2L?usp=sharing



Once the www/ folder has been updated, you will notice that a jqueryMobile folder has been added to the project. These files allow jQuery mobile to be utilized within the JavaScript portion of the application. The code within the index.html, app.css, and app.js files will also have changed, since the code is now specific to our Sound Recorder application.

Add Plugins

The Sound Recorder application can now be run on the Android and iOS platforms, but a few plugins are required in order for Apache Cordova to communicate with native device components. The media plugin allows sounds to be recorded and played, the device plugin accesses device information, and the file plugin accesses the filesystem of the device. The following three commands need to be run from the project directory: ***cordova plugin add cordova-plugin-media***, ***cordova plugin add cordova-plugin-device***, ***cordova plugin add cordova-plugin-file***.

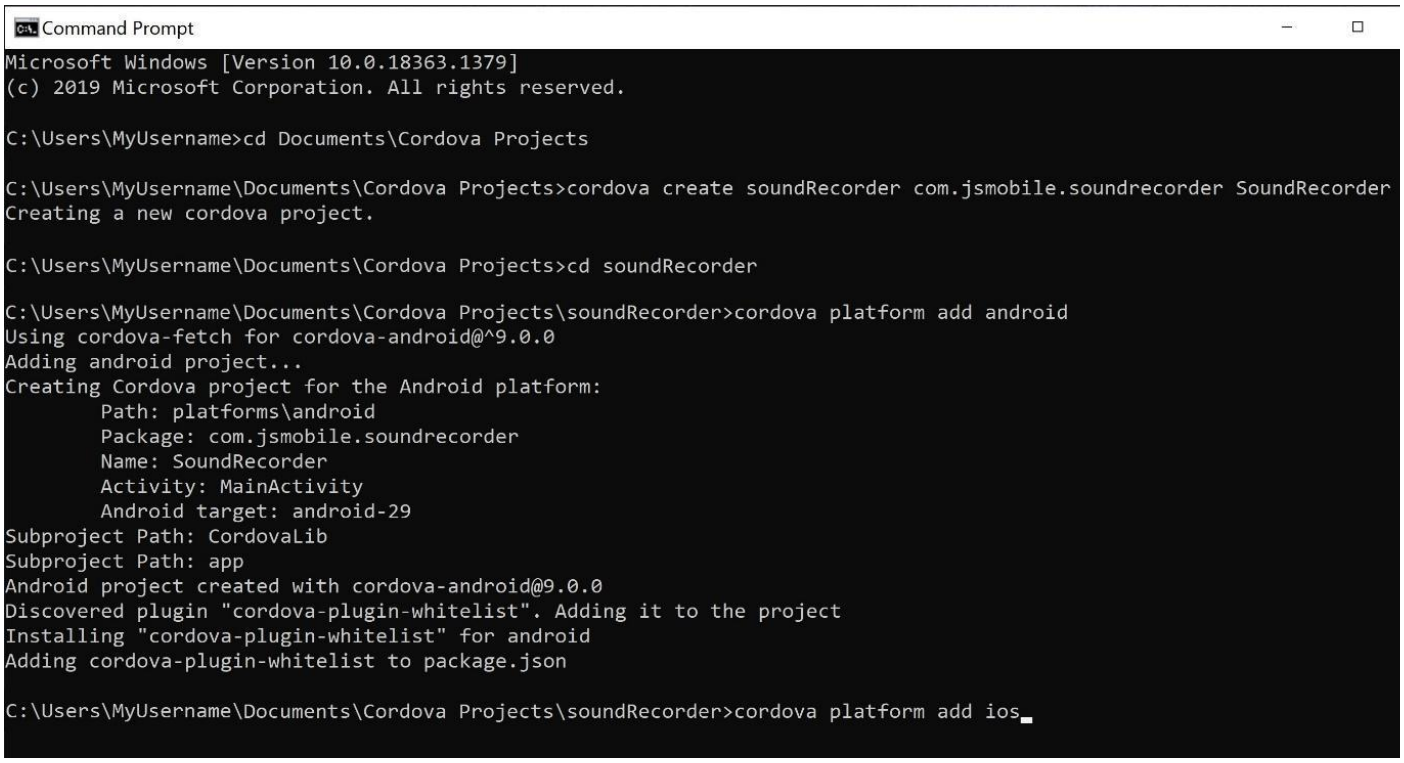
```
C:\Users\MyUsername\Documents\Cordova Projects\soundRecorder>cordova plugin add cordova-plugin-media
Installing "cordova-plugin-media" for android
Installing "cordova-plugin-file" for android

The Android Persistent storage location now defaults to "Internal". Please check this plugin's README to see if your app
lication needs any changes in its config.xml.

If this is a new application no changes are required.

If this is an update to an existing application that did not specify an "AndroidPersistentFileLocation" you may need to
add:

      "<preference name="AndroidPersistentFileLocation" value="Compatibility" />"

to config.xml in order for the application to find previously stored files.

Adding cordova-plugin-media to package.json

C:\Users\MyUsername\Documents\Cordova Projects\soundRecorder>cordova plugin add cordova-plugin-device
Installing "cordova-plugin-device" for android
Adding cordova-plugin-device to package.json

C:\Users\MyUsername\Documents\Cordova Projects\soundRecorder>cordova plugin add cordova-plugin-file
Plugin "cordova-plugin-file" already installed on android. Making it top-level.
Adding cordova-plugin-file to package.json
```

At this point the project has been created, necessary project files and Team 2 customizations have been added, platforms have been added, and plugins have been added. The next steps include adding the platform directories and running the application on Android and iOS devices.

Add Platform Directories

Regardless of whether the project directory was cloned from GitHub or created from scratch, it is necessary to add the Android and iOS platform specific directories. This allows the project to be built for each operating system. Any time that changes are made to the config.xml and impact one of the platforms, the platform must be reloaded. This can be accomplished by removing and adding the platform again, such as running the commands ***cordova platform rm android*** and ***cordova platform add android***. The result is that the platforms are rebuilt, and necessary files are included within the platform-specific directories.

Navigate to the project directory using the command line, and use the ***cordova platform add android*** command to add the Android platform. Once the Android platform has been added successfully, the similar command ***cordova platform add ios*** can be used to add the iOS platform.

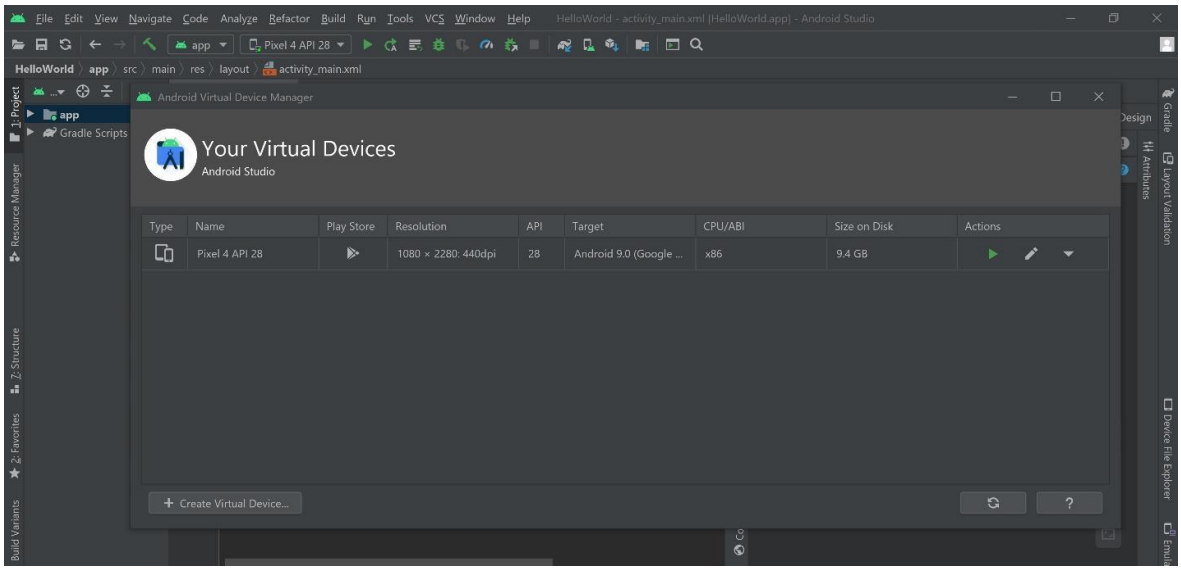


Running the Application on Android

Please note that the Sound Recorder application is compatible with Android 5.0 (API 21) or later.

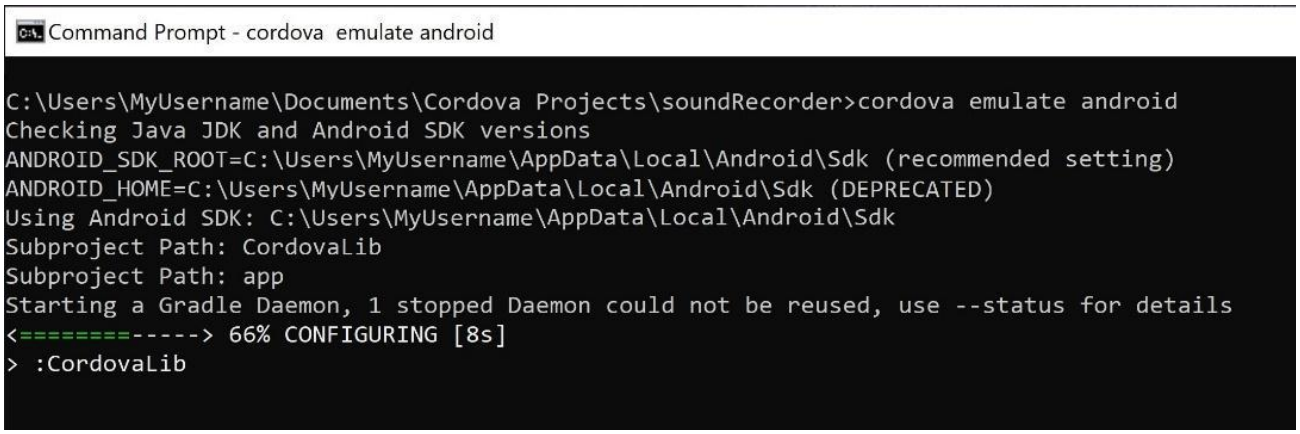
Configure an Android Virtual Device

Always ensure that an Android Virtual Device exists on your computer before attempting to emulate the Sound Recorder application. A new virtual device can be created using Android Studio's built-in AVD Manager (found under Tools > AVD Manager). You can specify the API for your virtual device.



Deploy to Android Emulator

Once you have confirmed that an Android Virtual Device exists, the application can be emulated using the ***cordova emulate android*** command. Use ***cordova run android --list*** or ***adb devices*** to list available Android devices, and use ***cordova run android --target=<targetID>*** to run the application on a specific device. If no device is specified, the application will run on the default device.

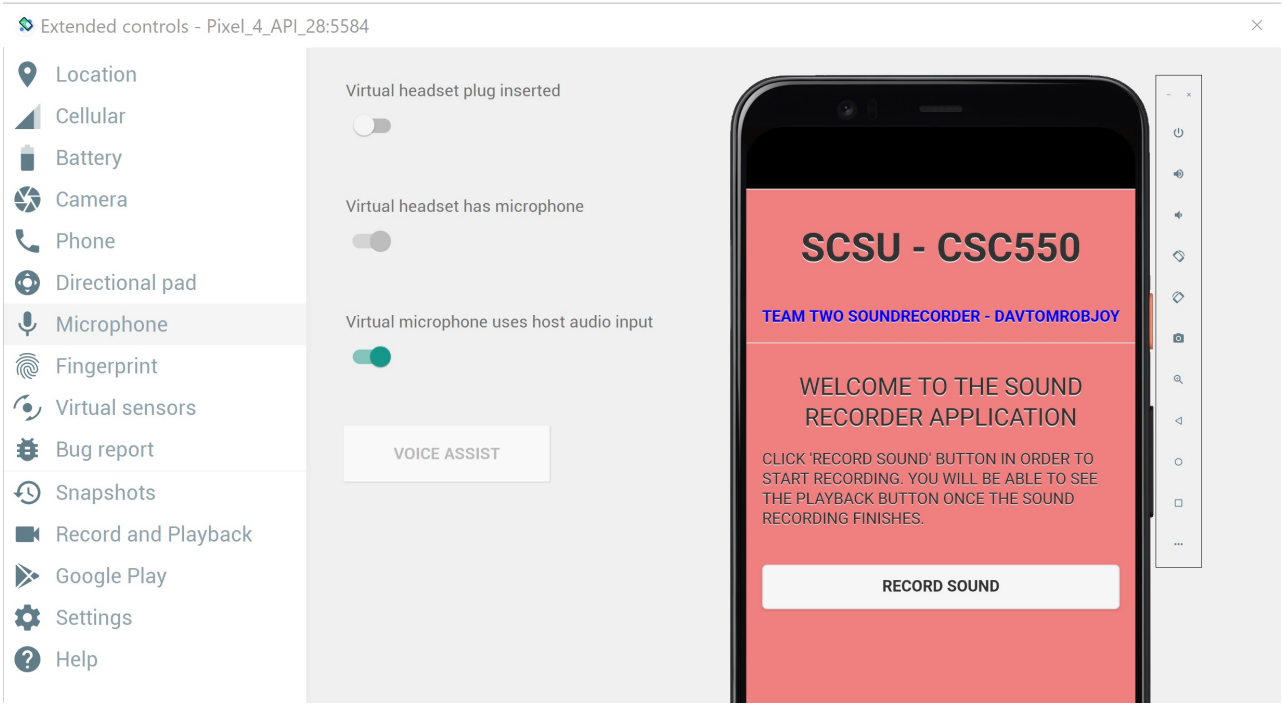


```
ANDROID_SDK_ROOT=C:\Users\MyUsername\AppData\Local\Android\Sdk (recommended setting)
ANDROID_HOME=C:\Users\MyUsername\AppData\Local\Android\Sdk (DEPRECATED)
Using Android SDK: C:\Users\MyUsername\AppData\Local\Android\Sdk
Subproject Path: CordovaLib
Subproject Path: app
Starting a Gradle Daemon, 1 stopped Daemon could not be reused, use --status for details

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/6.5/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 21s
40 actionable tasks: 40 up-to-date
Built the following apk(s):
   C:\Users\MyUsername\Documents\Cordova Projects\soundRecorder\platforms\android\app\build\outputs\apk\debug\app-debug.
apk
Checking Java JDK and Android SDK versions
ANDROID_SDK_ROOT=C:\Users\MyUsername\AppData\Local\Android\Sdk (recommended setting)
ANDROID_HOME=C:\Users\MyUsername\AppData\Local\Android\Sdk (DEPRECATED)
Using Android SDK: C:\Users\MyUsername\AppData\Local\Android\Sdk
No emulator specified, defaulting to Pixel_4_API_28
Waiting for emulator to start...
Waiting for emulator to boot (this may take a while)...BOOT COMPLETE
Using apk: C:\Users\MyUsername\Documents\Cordova Projects\soundRecorder\platforms\android\app\build\outputs\apk\debug\
app-debug.apk
Package name: com.jsmobile.soundrecorder
INSTALL SUCCESS                                                \app-deb
LAUNCH SUCCESS
```

After the build succeeds it will attempt to start the emulated device and deploy the application. If the deployment is successful, it will run the application on the device.

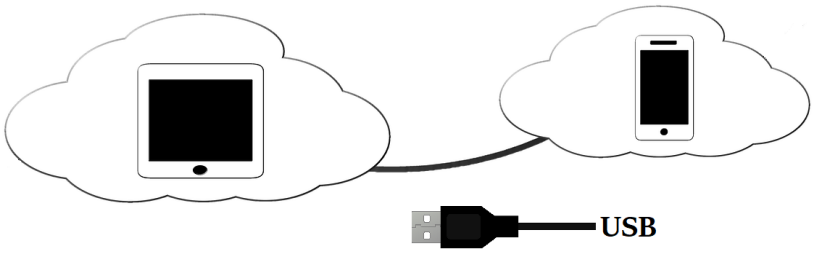
Configure the Emulator to use Microphone and Speaker

In order to record and play back sound on the emulator, the “Virtual microphone uses host audio input” setting must be toggled on under Microphone settings.



Deploy to a Physical Android Device

The application can be run on an Android physical device using the ***cordova run android --device*** command with the device connected to the computer via USB.



Ensure that the device is available once connected by running the ***cordova run --list*** command and verifying that the physical device is found. If no device is found, try disconnecting and reconnecting your USB cable. Please also ensure that USB debugging is enabled on your device. The Sound Recorder application will appear on your phone once successful.

Additional instructions on enabling USB debugging can be found on Android's official page: https://developer.android.com/studio/run/device.

Running the Application on iOS

Users running the macOS operating system have the ability to deploy the Sound Recorder Application to a physical iOS device such as an iPhone or iPad. Since iOS 10, Apple requires an additional step to run the application over Android devices. This includes setting a *Usage Description* before the microphone can be accessed.

Define a Usage Description

Both Android and iOS platforms require the user to provide permission for the app to access the internal microphone. This functionality works out-of-the-box for Android, but iOS requires the developer to provide a *Usage Description* to tell the user *why* access to the microphone is required. If this description is not present, the application will exit before asking the user permission to use the microphone; the description is not present. The usage description can be added to the ***config.xml*** file in the project root directory.

Add an edit-config entry to the ***config.xml*** file in the project root directory. Include the following three lines within the iOS platform element:

```
<edit-config target="NSMicrophoneUsageDescription" file="*-Info.plist" mode="merge">
    <string>need microphone access to record sounds</string>
</edit-config>
```

Once the config.xml file has been edited, the application can be run on both virtual and physical iOS devices. The ***cordova emulate ios*** command will run the application on the default virtual device. In order to run the application on a physical iOS device, a developer's provisioning profile must be configured.

Provisioning Profile

Xcode must be installed in order to configure a provisioning profile. After setting up the provisioning profile and linking it to the Cordova project, the application can be used for 6 days. After 6 days, the developer's certificate will expire and needs to be renewed. The user can renew the certificate and reload the application for testing for another 6 days.

Create a Provisioning Profile

1. Open Finder and go to the Sound Recorder root folder
2. Go to **/platforms/ios**
3. Open **SoundRecorder.xcodeproj** in Xcode
4. From the file menu, go to Xcode > Preferences > Accounts and add your Apple ID
5. Close preferences
6. Go to Target > Signing & Capabilities
7. Set *Automatically manage signing* and add the personal team account
8. Set Bundle Identifier like com.companyname.<yourName>.soundRecorder
9. Go to Project > Build Settings > Signing and set to *Automatic*

   Now that a Provisioning Profile has been set up, configure the Sound Recorder project to use it.

Configure the Project to Use the Provisioning Profile

1. Open **config.xml** in the Sound Recorder root directory
2. On line 2 replace the widget ID like *com.example.soundRecorder* with your bundle unique identifier like **com.companyname.<yourName>.soundRecorder**
3. Plug in the device to the computer via USB cable
4. From the command line, run the command ***cordova run ios --device***

At this point the application should be loaded on the device but the user cannot launch the app until the developer certificate is accepted in settings.

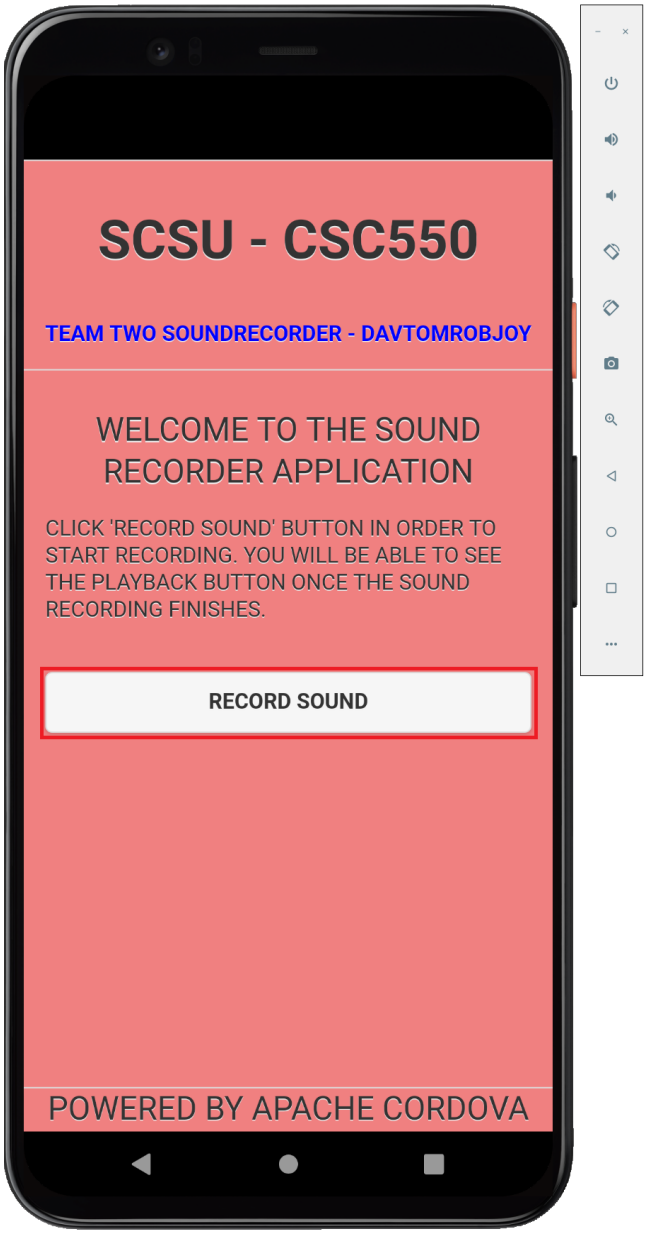
Accept the Development Account on the Device

1. Open the Settings app
2. Go to General > Device Management
3. Accept the Apple Development account
4. Run project in Cordova CLI again using the command ***cordova run ios --device***

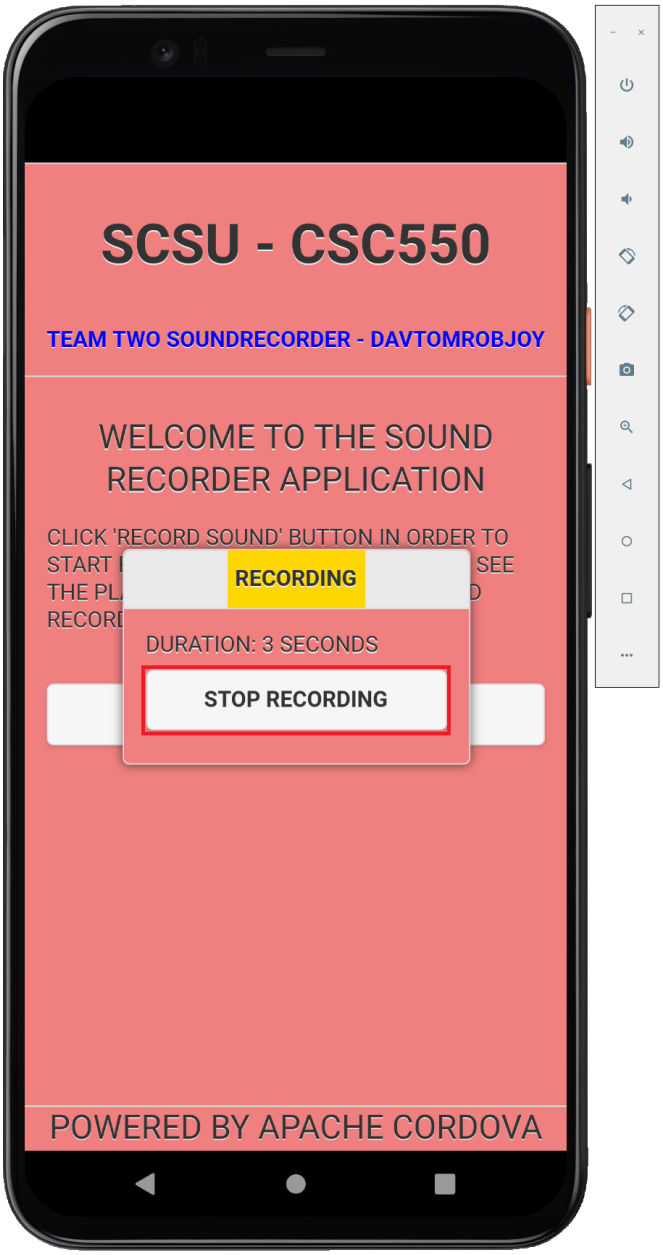
After running the command in the command line again, the application will open automatically on the device. When the user clicks the button to record sound, they will be prompted with the message requesting access to the microphone. After 6 days the developer's certificate will expire and the app will fail to open. The certificate can be renewed in Xcode and the app be redeployed to the phone with the Cordova CLI. The app will work for an additional 6 days and then the certificate will need to be renewed again.

Using the Sound Recorder Application

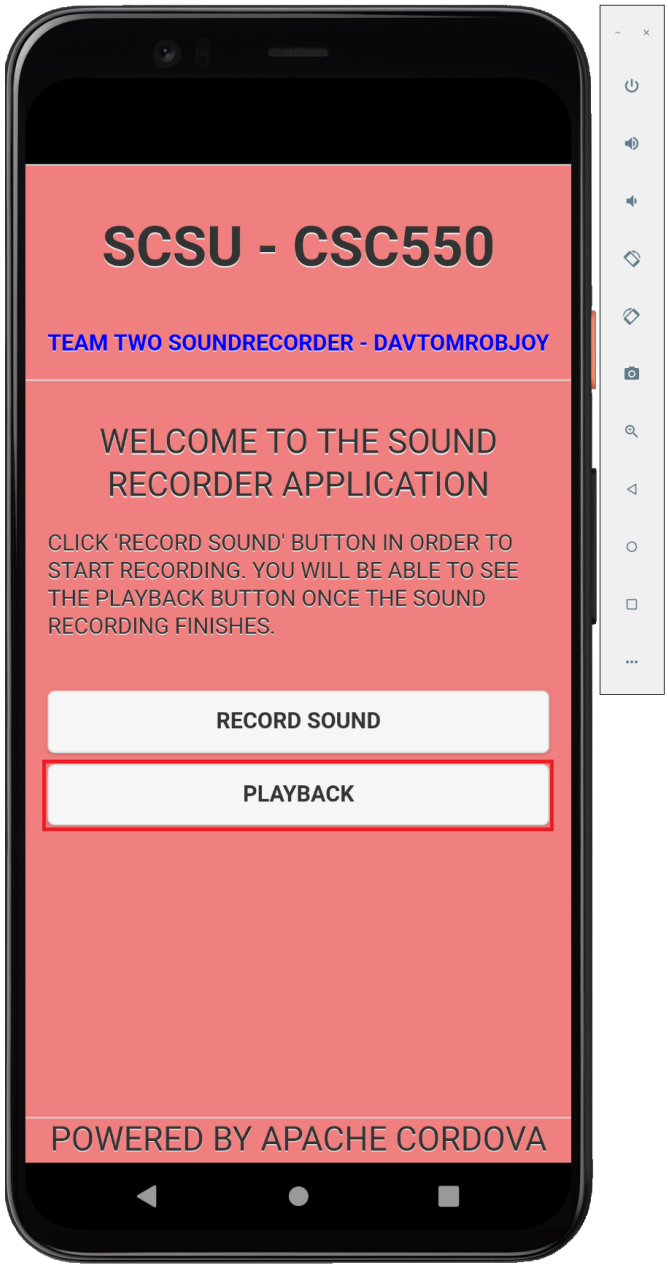
The Sound Recorder allows users to record sounds and play them back for listening. The microphone receives audio input, storage hosts recordings, and speakers produce audio output.
There is no limit to the number of times that a sound can be recorded and played.
Press the RECORD SOUND button to begin recording sound.



Once you have captured your desired sound, press STOP RECORDING.



The PLAYBACK button can then be pressed to retrieve and play the recorded sound file.



Only the latest sound recording can be retrieved and played back. The recordings are captured in Waveform audio format (.wav file) and the sample rate differs between devices. On Android, the files are saved in the *Data/cs-recorder* folder. On iOS, each application runs in its own sandbox for security purposes. This enables each application to operate separate from other applications and apps do not need to share data. The audio files from the sound recorder can be found in the Sound Recorder app sandbox, in the *Developer/tmp* folder.