CMPS 121 – Assignment 3
Spring 2018
Due: June 1 @ 11:59pm

Overview

In this assignment you will be creating an app that records audio using buttons from a WebView, and displays a list of all audio that has been recorded. When the user taps on one of the list items, the audio is played. Essentially the app will be a Voice Memo app to keep track of voice memos we record. This Assignment will be discussed in-depth during class on May 21, 2018.

Details

ListView Activity

Like in Assignment 2, when the user opens the app, they will be greeted with a ListView showing all of the audio recordings (we will refer to these as voice memos) that have been captured thus far, shown below in Figure 1. If there are no voice memos to show (the user is opening the app for the first time or no voice memos have been created yet), there will be a TextView stating there is nothing to show (as was done in Assignment 2).

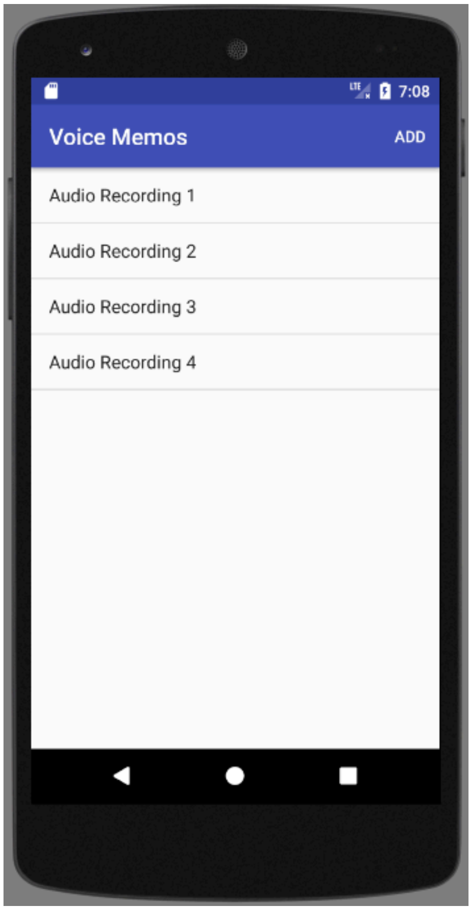

Figure 1

When the user taps an item in the list, the audio recording will play. The numbering convention for the items in the list is simply Audio Recording ___, such that the blank is the number in the list.

You will be keeping track of how many voice memos have been made through a very simple file that serializes a String. The file will be called "file.ser" (like in Assignment 2), however all that file.ser will contain is simply a String representation of the integer of the number of voice memos that have been captured. From the example above, file.ser would simply contain the String “4” and no other text. If file.ser doesn’t exist, then the list should not display any Voice Memos, instead, just a TextView stating “No Voice Memos”.

To build the ListView, parse the String from file.ser into an integer. Create an array of Strings, and initialize each element in the String to “Audio Recording 1”, “Audio Recording 2”, … , “Audio Recording n” such that n is the integer that was parsed from the String in file.ser. Assuming the name of the array of Strings is `listItems`, pass that array as an argument to ArrayAdapter object that will be used to build the list view (the same way that is used in the ListView example):

```
ArrayAdapter adapter = new ArrayAdapter(this,
android.R.layout.simple_list_item_1, listItems);
//mListView is the name of our ListView object
mListView.setAdapter(adapter);
```

Record Audio Activity

When the user taps the “Add” button from the menu, they are taken to the screen where the audio will be captured, shown in Figure 2.

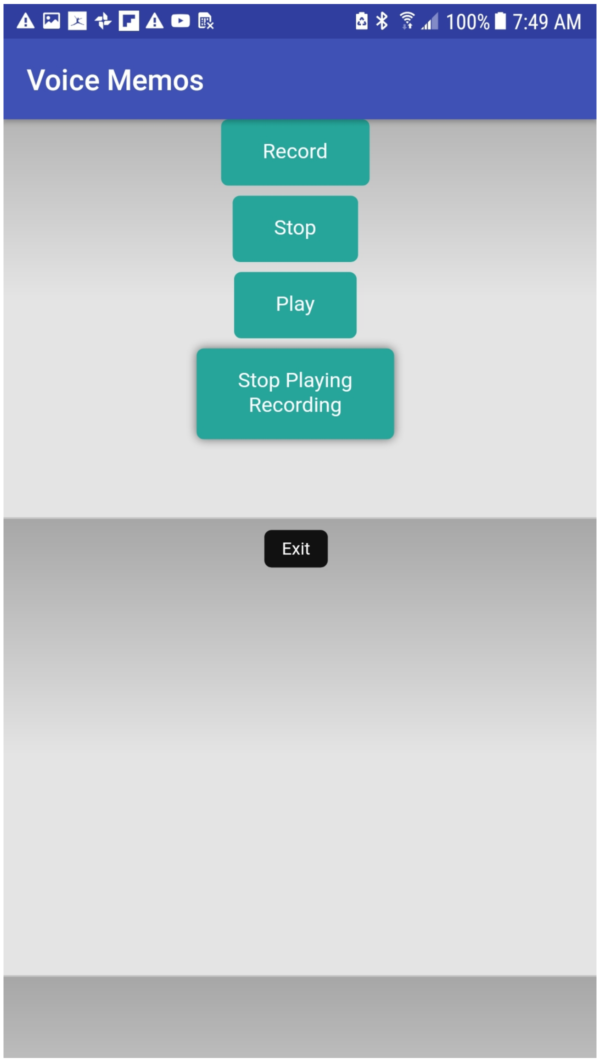

Figure 2

What is shown in this activity is simply a WebView taking up the whole screen; the URL the WebView is visiting should be the following:
https://users.soe.ucsc.edu/~dustinadams/CMPS121/assignment3/www/index.html

I would recommend visiting this URL and viewing the page-source to see what is going on behind the scenes. In this webpage, there are five buttons, each of which calls a javascript function – the javascript function will call a function that you will be defining in a JavaScriptInterface, an inner class in the Activity. These will be the functions you need to define in your JavaScriptInterface:

`record()` – this will start the audio recording. Here, open file.ser, read in the String to an int, add one to that int, convert back to a String, and this will be the filename for your audio file (don't use the CreateRandomAudioFilename method that is shown in the AudioRecord example). The audio file's extension will be 3gp (the same as in the AudioRecord example).
`stop()` – stops the audio recording. Here, also open file.ser, read in the String to an int, add one to that int, convert back to a String, and serialize that String into a file named file.ser.
`play()` – plays the recent-most audio recording captured.
`stoprec()` – stops audio that is currently playing.
`exit()` – exits the activity (pops the activity off the activity stack).

The recording of the audio will take place in the exact same manner that happened in the audio recording example shown in class (source code available on the course webpage). The main difference here is that you are not defining OnClickListener objects for your Android Buttons, instead you are defining methods within the JavaScriptInterface that will be called when the buttons in the WebView are tapped. Also, you will not need to disable/enable any Android Buttons (as is done in the AudioRecord example); the buttons are already being enabled/disabled for you in the javascript code in the webpage that is being visited.

Extra Credit

For 8 points extra credit to count toward your overall assignment points, capture the location, time, and date each time the user records a voice memo (presumably store this information in a json file). In the menu bar being shown in the list view activity, there will also be a "Map" button (in addition to the "Add" button) that will start a new activity showing a MapView of all the places a voice memo has been captured as markers on the map. When the user taps a marker, they will hear the audio recording and be shown an Android Toast of the time and date the audio recording was captured. Partial points will not be given for extra credit.

Turning it in

Submit your app project in a zip file into Canvas.

Grading Rubric

Are all screens present formatted in a readable and usable way?	2 pts
Does the list view add a new item as a result of tapping record then stop?	2 pts
Does tapping on the individual item in the list play the correct voice memo?	2 pts
Are all of the buttons in WebView performing the correct action?	2 pts
You are using the provided URL for your WebView?	1 pt
The app does not crash?	1 pt