**MP3Learn App Documentation**

Authors: Claire Bedard, Jeremiah Bray, Donald Shade, Daniel Garcia

Last Update: 8 March 2019

Contents

Overview

Important Classes/Activities

Other Key Features

Future To-Dos

Overview

The purpose of this document is to have a comprehensive documentation of the MP3Learn Android app, designed by LightSys Technology Services for Cybermissions. The app functions as a template-like audio learning tool, so that MP3 audio lessons, accompanying PDFs, and user notes can be accessed and stored offline. Key features include the MediaPlayer and PDFView classes for opening MP3 and PDF files, and a DatabaseConnection class for storing notes and other information within the application.

Important Classes/Activities

Course: Simply contains an array of Lesson objects and the course’s name.

Lesson: Contains lesson’s name, current timestamp of the audio, its associated course, a String with any notes taken by the user, and the filenames of the MP3 and PDF of the lesson.

DatabaseConnection: This is an SQLite database that stores a column for every lesson with its name, its course name, the current timestamp (see above), and any user notes.

MainActivity: Handles everything involving the course and lesson hierarchy and selection process, switching over to lessonActivity only once a lesson is selected.

lessonActivity: Opens a single lesson with its MP3, PDF, and notes. MediaPlayer and PDFView both are implemented in this activity.

Other Key Features

MediaPlayer: Android’s MediaPlayer library was used to open and play the MP3s. A separate library, SeekBar, was used to make the seek bar beneath the media controls.

PDFView: This library, obtained from GitHub open-source code, opens PDFs from an inputStream (which was the simplest way of accessing data stored directly in the app’s raw folder) and loads it onto the lesson activity.

file\_dir.txt: Since all MP3 and PDF files are stored in the app’s raw folder, a file\_dir.txt document was created to store a file hierarchy model, allowing correct association of lessons and courses without the need for a directory. The Main Activity’s getCourseData() function reads this document in order to create those associations when the app first gets launched.

How to create files for the app

In the repository, there is a directory(folder) named FileDirHelper. In this directory there are three files. A .jar, a .java and a .txt. The jar file is used for creating the files(and file\_dir.txt). The .java is the source code and the .txt serves as a README and contains similar information to this. To use the .jar first create the necessary directory system. The highest directory should be labeled courses. In that directory create sub-folders with the names of the courses. In those sub folders create, sub-folders for each lesson. Then place the lesson materials inside these folders. Your structure should look like this

Courses

--Name of Course

----Lesson1Name

------Lesson1.mp3

------Lesson1.txt( or .pdf)

----Lesson2Name

--Another Course

Once this is done, place the .jar file in the directory with the “Courses” directory. Double clicking the .jar will run it. You can also run it from the command line. The .jar will then parse through all of the directory and files inside of “Courses” and create file\_dir.txt. It will place this file in a new directory called files\_to\_add. Simply move all the files in this directory to the raw folder in android. Android Studio will do the rest when it compiles your new app.

Future To-Dos

All Notes: Add a menu item where all notes can be accessed separately, as well as manually saved or deleted

Add times to the media controls.

Spring Break Code-a-thon Programmers

Daniel Garcia – Project Lead ([danielggarcia@cedarville.edu](mailto:danielggarcia@cedarville.edu))

Donald Shade – Project Manager ([donaldrshade@cedarville.edu](mailto:donaldrshade@cedarville.edu))

Claire Bedard

Jeremiah Bray