Jaganathan Velraj

Prof. Paul Comitz

May 18, 2021

Final Mini-Project

Abstract:

This project will allow user to take notes (read and write) which will be stored in the device as a directory. The major components of the project will be the app can create, write, read, and edit a text file. I will be using button, list view, intent, activity, edit text and array adapter as my classes and objects for this project. Test cases for the project: check if the app can create, write, and store a file, and check if the app can edit, and read an existing file that been created. My schedule for getting this project done: understand the objective and draw a sample layout, write code, and test the app until all the objectives are met.

Project writeup:

For this project, I have used two activities: MainActivity and NoteActivity, and a normal class: ListTitle in order to make this project work as intended.

ListTitle:

It has two functions store array information and position of the arrayAdapter. First, I thought to have the array information within the mainActivity but while running the app the list gets resets to the initial array. Later, I figured out that when you come back from an activity, it will not retain its information but resets all the information in that activity. This made me to create a separate class that mediates between MainActivity and NoteActivity. In the class, I created an array called “titles” which holds initial values (i.e., “note1”, “note2”,…) and I will call it in mainActivity wherever it’s required. In NoteActivity, when user adds a title to the notes then I will add this information in the “titles” array from the data of edit text and array from MainActivity’s position. Well, you might think why I didn’t use put extra from main activity to note activity. The reason is first I had a time crunch, to prepare for my exams, and this might be a little less complex way to store the position from main activity.

The ListTitle class has four methods: setPosition(int pos), getPosition(), setArray(int pos, String str), and getArray().

Main Activity:

It sets the main page with list of notes and helps the user which note to pick as the title of each list changes once they made some change to that particular note. To achieve that, I used “textList” array using getArray() from ListTitle, listView, and arrayAdapter. First, to prevent resetting the activity I made a static Boolean names “firstPass” and set to “true” (Thanks to prof. Comitz). In an if statement, if the firstPass is true then I turn the variable to false, and else it gets a string with a key called title from NoteActivity, position from ListTitles and update the array in the ListTitles. Once the activity passed the if statement, it gets the array from the listTitles. Then, creates a variable: listView for listView from XML use find view and match its ID, a variable: arrayAdapter for Array Adapter with required parameters (arrayAdapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*,textList). Set the arrayAdapter to the listView. Set clickable to true in listView. Then, create a set On Item Click Listener; Inside it have a toast, call set position from ListTitles to change the current position, and call NoteIntent. I found this part of project easy since it is similar to the homework I did previously regarding array adapter and intent.

Note Activity:

Displays two edit text boxes with a submit button. First box where user will type their title for the note. Second box where user will type some message. If the user still in the app and wants to add something or to just to check their notes they can since the data will not erased while the app is still running.

I used same if statement that I used in main activity to prevent resetting the activity entirely. Inside the if case I created the text files required to store the data that user enters. Inside else case I initial both edit text, call read file method then set both the edit text with their respective data to be retained when user wants to see them. After typing their information in their notes, they should click submit button since this allows the app to write the data that user entered into the text files. Then, it will call the note Intent to return back to the main activity before it does that it will send title of that edit text to the other activity through put extra. To write the edit text data into text file, I used the file write code provided by the prof. Comitz and made my modification to it in order to work for this project. I did the same with read file as I mentioned in the previous sentence.

Overall, this app can create, write, read, and edit a text box using a button, list view, intent, activity, edit text and array adapter as my classes and objects in this project. I tested my app enough to tick boxes you have mentioned for this project. I didn’t follow my schedule as it is in the beginning but that bugged me later stage in the process. And I followed what I wrote in my schedule as it is then code just flows without an hesitation.