**CS6326.001 User Interface Design**

**Spring 2022**

**Assignment 5 – Android App**

Designing for mobile is different from designing for a large-screen PC. This assignment is to write the part of a game that tracks high scores. The fields of interest are the name of the user, the score, and the date/time that the user got that score. For purposes of this assignment you will enter all three fields. Here are the details:

Fields:

1. Player name. This should allow a maximum of 25 characters. No error checking is needed other than not empty.
2. Score. This should accept only positive integers, and not zero.
3. Date/time. Must be validity-checked. You can auto-populate this with the current date/time but allow it to be changed.

When the program comes up, you should see a list of the top twenty high scores. That list must show the rank (1 through 20), name of the person, the date, and the score in columns, and must be in numerical order by high score. In the case of duplicate scores, the most recent date should take precedence. The list must scroll if there are more scores than will fit on the screen. You can use either a ListView or a RecyclerView for this list. This is your main Activity.

An **add** button on the action bar will bring up a separate screen (Android Activity) to add a new high score. When you finish adding and save, the *add* screen should close and the new score should be in the list. The Save button can be either on the entry screen or on the action bar, your choice. As required by good design, the Save button should be enabled only when there is valid data. There is no need for a Cancel button, since the Android “back” function does this.

Scores are stored in a tab-delimited text file, not in a SQLite database. You should completely rewrite this file with only the top twenty scores when any changes are made. The I/O should be in a separate class, not in the main activity. You are allowed to use Java collection class sorting functions rather than writing your own.

Apply both Android design principles and the various things we have learned about design so far in the course. Your screens need not be flashy, but they should look good and their functions should be clear to the user.

**This is an individual assignment**, since it can be written and tested entirely using the emulator and does not require sensors. Write your own code, but you are allowed to use code from the “Programming Tips” on my Web site, and from the slides.

A hint: You can use the StartActivityForResult function to go from the main activity with the list to the activity that allows entry of a high score, and return information to the main activity.

Good program design would suggest you have at least two classes in addition to the two Activities. We can discuss in class what those might be.

**To hand in:** You will be required to show the application on your device to the TA and/or the instructor. You will also be required to hand in your entire Android Studio project through eLearning. Create a Zip file containing your project with the name <netID>-Asg5.zip.

|  |  |
| --- | --- |
| **Grading criteria** | |
| Meets the above requirements | 60 |
| Object-oriented design | 30 |
| Internal program comments. Here is a link to what I consider to be good comments: https://www.utdallas.edu/~John.Cole/Documentation.htm | 10 |

Additional grading notes:

1. Program gives the error “Unfortunately, your program has stopped working” for any reason. -20
2. Program has only one activity: -30
3. Program calls I/O functions in more than one Activity: -15
4. Program does not have a separate class for file I/O. -15
5. Missing validity check: -5 per field. (Validity checks need not be in a separate class.)