CS 501 Mobile Application Development

Spring 2023

Assignment 3 – Logcat, testing, multiple Activities

**Due:** Night of lecture 4, before class.

**Team Name:** Not Yet

**Members:** Renming Zhang, Liyu Qu, Melody Chan, Sarah Bonna, Samamtha Shih, Huanjia Liang, Yuhe Bian, Yunqi Ni

Do not email homework, please submit on Blackboard. You should also take screenshots or recording of your assignment and submit. Post the code on GitHub and submit the link and the videos in a zip file. Other than the reading, this is a group assignment. When working as a group, do not divide and conquer, do not assign one problem to one or two people. Everyone must work on each problem. Please be sure to include your team’s name and members for the group submission. Only one group submission per team please.

Note that all programming assignments from now on must include unit (JVM) and instrument tests.

**Assignments:**

1. **(Individually)** Read all of chapters 8, 9 and 10 before the next class.
2. **(Team)** At this point you should be all set on your teams. In the appropriate thread on Piazza, please post your team’s name and members.
3. **(Team)** Using the GeoQuiz in chapter 7 as the starting point, fix the loophole in it whereby rotating the CheatActivity the clear out the cheating result so it looks like they didn’t cheat. Why does this happen?

When the screen rotates, the initial activity instance is destroyed and a new one is created. Therefore, any value stored during the initial activity will be wiped away. This caused the cheating result to clear out.

**GitHub link:**

<https://github.com/melodyy0128/BU2023SPR-CS501/tree/master/week3/Assignment3-Q3%26Q4-GeoQuiz>

**Video link:**

<https://drive.google.com/file/d/126Kcn7IQqcuUBQGtwhW3lWYGhP0P_7as/view?usp=sharing>

1. **(Team)** Again using the GeoQuiz in chapter 7 as the starting point, update the code to fix the problem that if a user cheats on one question, they get the judgement toast for all the questions. Modify the code so if a user answers a question without using the cheat button, the correct or incorrect toast is shown.

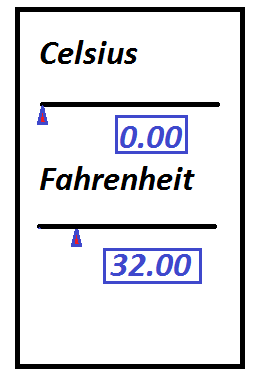
**GitHub link:**

<https://github.com/melodyy0128/BU2023SPR-CS501/tree/master/week3/Assignment3-Q3%26Q4-GeoQuiz>

**Video link:**

<https://drive.google.com/file/d/128QlcfEJjwB_cCMAdniAA8qeAVllX_mm/view?usp=sharing>

1. **(Team)** For this problem you will be required to read the documentation for the Seekbar Component. Create a temperature converter that works both ways. Use a ConstraintLayout for this Activity. Implement a Celsius to Fahrenheit converter to utilize two SeekBar Components, one to represent Celsius the other for Fahrenheit.



Interesting Message

* 1. Moving one SeekBar, will not only change its own value, but also update its counterpart. For Example dragging the Fahrenheit SeekBar to 32° F will update the Celsius SeekBar to 0° C and vice versa.
  2. Use a Celsius range of 0° - 100° C. Use a Fahrenheit Range of 0° - 212° F.
  3. The Fahrenheit SeekBar cannot remain below 32° F. That is, if a user were to drag the Fahrenheit SeekBar below 32° F, it should automatically snap back to 32° F, leaving the Celsius value at 0° C.
  4. A SnackBar should popup and read “I wish it were warmer.” when the temperature is less than or equal to 20ºC degrees, otherwise it should read “I wish it were colder.”. Choose whatever min/max ranges you like for each seekbar.

**GitHub link:**

<https://github.com/melodyy0128/BU2023SPR-CS501/tree/master/week3/Assignment3-Q5-TemperatureConverter>

**Video link:**

<https://drive.google.com/file/d/129FbkKtgbW7AJX0DQ4dTpfgj3cQ1eZMT/view?usp=sharing>

1. **(Team)** Create a flashcard App with two activities, a simple login, for which you may simulate logging in with a hardcoded username and password (please tell us what these are). After entering the correct user/pass combination, the user should be routed to a Flash Card Activity.
   1. When the user presses the “Generate 10 Random problems” button, 10 random addition and subtraction problems will be generated. Out of 10 problems, approximately half should be addition and half subtraction, these are random with a 50/50 probability of getting an addition or subtraction problem, each time.
   2. Your program will utilize Kotlin as the code-behind layer of your Activity and will generate 10 random problems, students will enter the answer into the textbox, then depress submit. When the user completes all 10 answers, just provide a Toast with their score, eg, “9 out of 10”.
   3. The top operand, (i.e., - Operand 1) must always be between 1-99. Operand 2 must always be between 1-20.
   4. Be sure the user can restart the game and play multiple times. Feel free to use the template below as a guide (just a guide, not a requirement).
   5. The “Generate 10 Random problems” button should become disabled once the game starts. After the users submits answers to all 10 problems, will the button become re-enabled.
   6. Don’t worry about telling students which problems they got right or wrong, or giving them immediate or delayed feedback.
   7. Provide a descriptive Storyboard for your App. Much of the design and user interaction is up to you. Use the below template as a guide.
   8. Update the App so that once the user successfully logs in, a Toast is displayed when the second Activity starts up. “Welcome <user>”, where <user> is the username entered in the first Activity.
   9. The user should not have to start over just because the screen was rotated. Update the code to store the user’s score and other important state information, such as the current problem on the screen, when the screen is rotated, state of the “Generate 10 Random problems” button, etc. This only applies to the second Activity (not the Login).
   10. The App should “Look Good” in both portrait and landscape mode. This only applies to the second Activity (not the Login).

**GitHub link:**

<https://github.com/whisperzh/FlashCard>

**Video link:**

A picture containing text

Description automatically generatedText

Description automatically generated