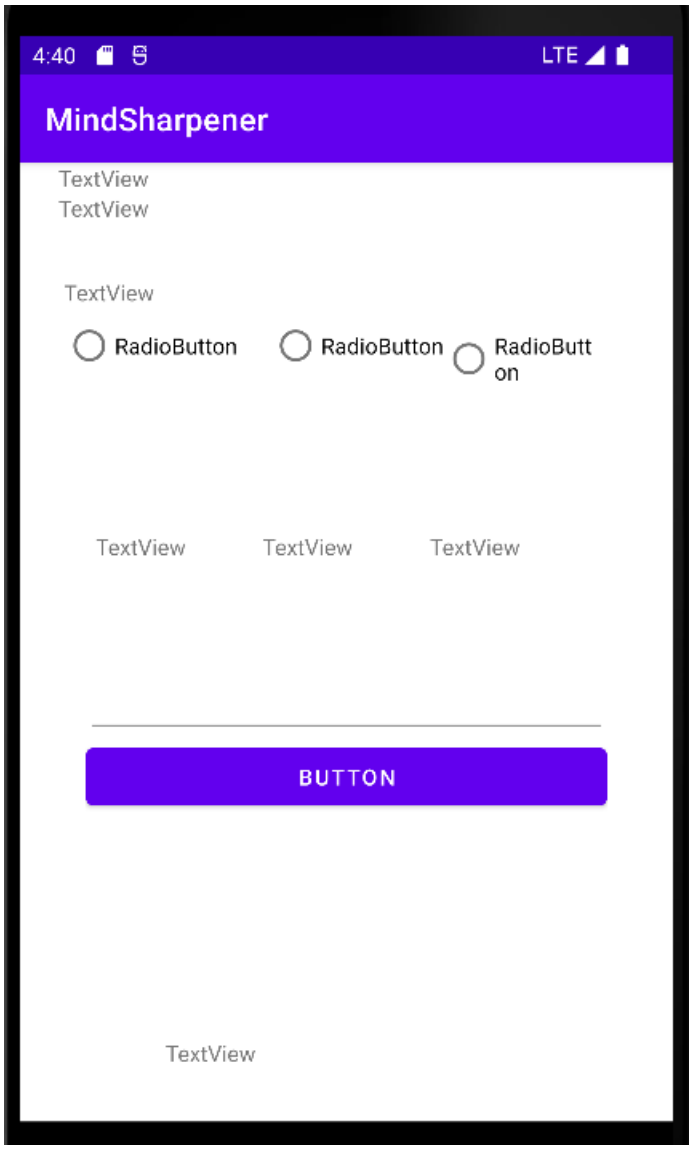
Lab Test CSM3505 18 January 2023

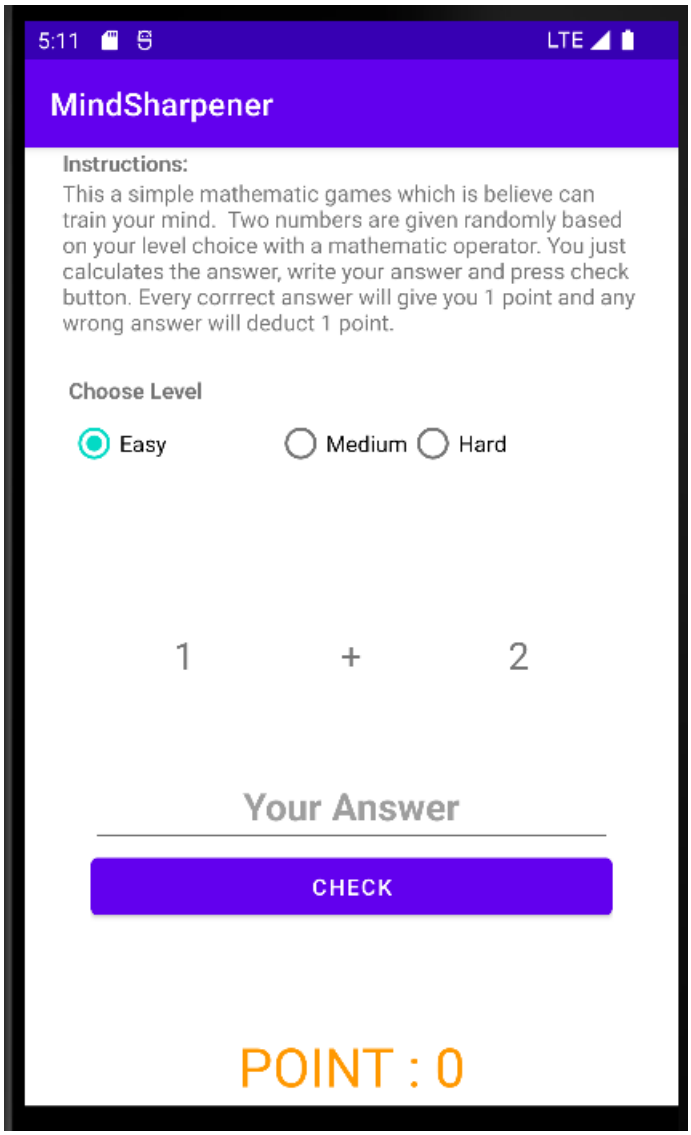
Create an Android Studio Empty Project and name the project as MindSharpener

Task 1

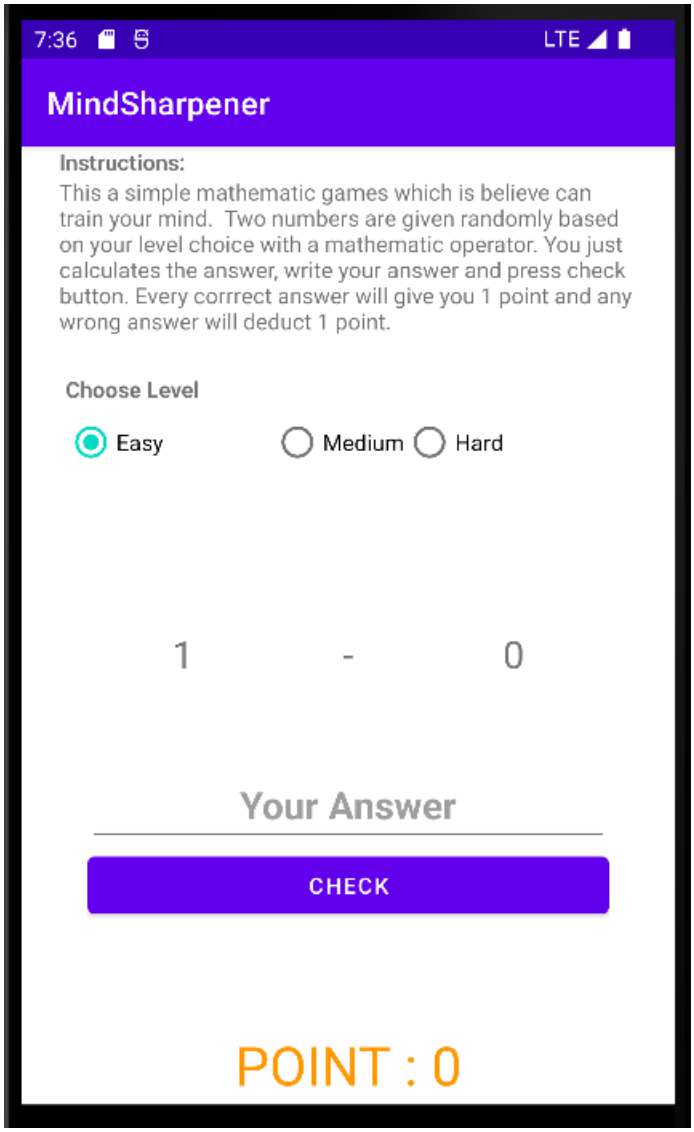
For the above project, design the activity screen as per given below. Group the control using the LinearLayout may make the arrangement more easy!



Change the label, text size text colour accordingly to the following design. Check easy as a default value.



Task 2



As above figure, when the app loaded, it will read the selected radio button and display the question according to chosen level. Chosen level will determine the number of digit generated by the random number generator. For example:

Random random = new Random()
int randomNumber = random.nextInt(9)

will returns 1 digit number from 0 to 9, and random.nextInt(999) will returns 3 digit numbers. Therefore, two numbers will be generated as first number and second number.

Meanwhile, since the operator is presented by number, then, it only have 4 numbers which representing +, - , * and / respectively. So, to generates number for operator, we

only need 1 digit which start form 0 and end at 3. Therefore, the random number generator for operator is:

```
  Random random = new Random()
int randomNumber = random.nextInt(3)
```

The following algorithm can be use to display operand and operator randomly on screen:

1. Instantiate Random class
2. Read RadioGroup to determine which radio button is selected
3. If radio button is easy generate two numbers of 1 digit each
4. Else if radio button is medium, generate two numbers of 2 digits each
5. Else if radio button is hard, generate two numbers of 3 digits each
6. Generate 1 number for operator and get operator symbol where 0 for (+) , 1 for (-), 2 for (*) and 3 for (/).
7. Display generated number and operator symbol

User then can enter their answer,then, it is checked against the question given as the following algorithm

1. Get user answer
2. Get firstNumber from previous
3. Get secondNumber from previous
4. Get operator from previous
5. Calculate the answer
6. Compare the answer with user answer
7. If correct, increase point by 1
8. Else, deduct point by 1
9. Display another question as the first algorithm above

Write a full program MainActivity.java to implements the logic behind this app either in Kotlin or in Java.

Evaluation Rubrics

	4	3	2	1
Program Execution	Program execute correctly with no syntax or runtime error	Program executes correctly with minor errors	Program executes with a minor (easily fix error)	Program does not execute
Correct Output	Program displays correct output with no errors	Output has minor errors	Output has multiple errors	Output is incorrect
Design of Output	Program display more than expected output	Program display minimally expected output	Program does not display the required output	Output is poorly designed
Design of Logic	Program is logically well design	Program has slight logic errors that do not significantly effect the results	Program has significant logic errors	Program is incorrect
Standards	Program is stylistically well designed	Few inappropriate designchoices (poor variable name, improper identation)	Several inappropriate design choices (poor variable name, improper identation	Program is poorly written
Documentation	Program is well documented	Missing one required comment	Missing two or more required comments	Most or all documentation missing

Submission

1. Upload your <main> folder into the github
2. Put the link into a word document
3. Put your matric number into the word document
4. Upload the word document into Lab Test Submission link in e-learn