**CMSC 234**

**Project 1 - Piggy Bank**

**Derya O. Kurin**

**Final UI Design:**

Graphical user interface, application

Description automatically generated

**Pseudocode (Algorithm):**

1. Apply the UI design shared above in activity\_main.xml file

In the activity\_main.xml:

1. Register onClick eventHandlers to both buttons
2. Create unique ids for each RadioButtons, EditTexts and Buttons
3. Create a CalculatorUtility class
4. In the Calculator Utility class, create 3 static methods:
   1. + sum: double

Parameters: quarters:int, dimes:int, nickels: int, pennies:int

* 1. +add: double

Parameters: total: double, additionAmount: double

* 1. +subtract: double

Parameters: total: double, subtractionAmount: double

1. In the Main Activity class:
   1. Define data fields
      * quarter: EditText
      * dime: EditText
      * nickel: EditText
      * penny: EditText
      * totalMoney: EditText
      * saveOrSpend: EditText
   2. Create an onCreate method and setContentView as the activity\_main
   3. Initialize the class properties by registering them with their associated ids on Create.
   4. Write an onClick eventHandler
      * Create a switch statement for both buttons: Calculate Total and Process Transaction
      * In case button1 is clicked (Calculate Total); call the static method CalculationUtil.sum with the coin values
      * In case button2 is clicked (Process Transaction);
        1. Create instances of radio buttons and register them with the two radio button ids, for both saving and spending
        2. IF spending radio button is checked:

IF saving text length is 0

THEN Set the text of Total Money Edit View to the return value of CalculationUtil.subtract static method

ELSE

Create a toast message with a test:

“Please enter an amount to spend!”

ENDIF

ELSE IF spending radio button is checked:

IF spending text length is 0

THEN Set the text of Total Money Edit View to the return value of CalculationUtil.add static method

ELSE

Create a toast message with a test:

“Please enter an amount to save!”

ENDIF

ENDIF

**Lessons Learned:**

In this project I utilized what I previously learned in Temperatur Calculator app.

With this one too, I created a utility class and created some static methods for calculating the total sum of the coins, and in addition I added an add() and subtract() methods. I could have handled these simple calculations in the eventHandler method in MainActivity but in general I thought it would be a good idea to separate the calculations from the MainActivity by using the utility class even though the calculations were simple. I think it is a good practice to do so, because by the time our app needs to do more complicated calculations, we would not need to change the app’s structure but instead add new methods to the utility class. In terms of scalability, I believe this way is better.

While creating this app I initially created a bug and received an java.lang.IllegalStateException. After searching some resources online, I figured it out that the problem caused by forcing the app with wrong casting practices. I realized that In the activity\_main.xml file, I added a title text to EditText such as Quarters and when the user enter their input number, a conflict took place because the inputType was expected to be a number of double and what I had was a string “Quarter: %d”. After I separate the title as the TextView and EditView, it solved the issue and I could see the calculation result properly.

I created two buttons in this app and registered the same onClick event to these buttons. In the eventHandler, I used a switch statement to check which button was clicked. This way, I could handle all the events in one method.

While calculate total is used to sum up all the coins and shows it in the Total edit view,

Process transaction button is used to add savings or subract spendings and it again shows the remaining sum in the Total edit view.

With this project I gained more experience in working with widgets and manipulation the input data and showing them properly in my app.

**Screen Shots for Tests:**

Calculating total value of the coins with the Calculate Total Button:

A screenshot of a cell phone

Description automatically generated

Spending some amount from the total amount with the Process Transaction Button

6.4 -2.0 = 4.0

Graphical user interface, application

Description automatically generated A screenshot of a cell phone

Description automatically generated

Saving some amount with the Process Transaction Button:

A screenshot of a cell phone

Description automatically generatedGraphical user interface, application

Description automatically generated