UOW MALAYSIA KDU PENANG UNIVERSITY COLLEGE

PART OF THE UNIVERSITY OF WOLLONGONG AUSTRALIA GLOBAL NETWORK

ASSIGNMENT COVER PAGE

Validated by UNIVERSITY OF LINCOLN UNITED KINGDOM

Programme	**Course Code and Title**
Bachelor of Computer Science (Hons) Bachelor of Computer Science (Hons) In Computer & Network Technology Bachelor of Information Systems (Hons)Bachelor of Software Engineering (Hons)	CET3013 (Mobile Application Development)
Student's name / student's id	**Lecturer's name**
CHAN SEOW FEN / 0207368	Lai Kim Min

Date issued	**Submission Deadline**	**Indicative Weighting**
Week 4 -02/10/2023	Week 6 - 16/10/2023	30%
Assignment [1]	Pizza Calculator Application	

This assessment assesses the following course learning outcomes

# as in Course Guide	**UOWM KDU Penang University College Learning Outcome**
CLO1	Discuss Android applications components and resources
CLO4	Create a complete app with appropriate software engineering techniques that complies with Android Design guidelines
# as in Course Guide	**University of Lincoln Learning Outcome**
CLO1	N/A
CLO2	
CLO3	
CLO4	

Student's declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Student's signature: 

Submission Date: 16/10/23

Assignment 1

ORIGINALITY REPORT

1%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
2	tdr.lib.ntu.edu.tw Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Table of Contents

1.0 Brief Report on Application Building

1.1 Components

The components used in building the application, Simple Discount Calculator App included 1 Main Activity (MainActivity.kt), which is the user interface controller for the user to interact with the application and do the calculation for discount. Furthermore, the project included 1 data class (DiscountData.kt) to represent the data layer, which in responsible to store the user input (Sales Price, Discount, Tax) and output, which is the discount information (Save Amount, Tax Amount, Total Price). In addition, the project also included 1 view model (DiscountModel.kt) for data persistence as in corresponds to the project requirement, which enable the user to retrieve back the discount information even though there happens a configuration change such as rotating phone. Moreover, the project also having an object class (Calculation.kt) to handle business logic, which is calculating the discount information in this case. Furthermore, the project included a class (AboutDialog.kt) for showing application information such as the application version and the programmer's name. In addition, the project also has a class for (RangeInputFilters) limiting the range of input. Moreover, it corresponds to the requirements of the project, two layouts, which is portrait and landscape layout is included for the user to rotate the application according to their preference. Lastly, the project components included 4 resources files (menu.xml, colours.xml, strings.xml, themes.xml) which used to apply for design themes as well as holding for each piece of text used in the application.

1.2 Design Themes

Figure 1.1 Colour Palette of User Interface Design

As shown in *Figure 1.1*, the design theme of the application is basically using the main colour, pink and brown which is very aesthetically combination as a palette. There are also other colours that closed to the main theme colour as a minor colour to make the application to seems livelier and more entertaining. The design is more towards the minimalistic idea, which keeping the user interface to be as simple as possible, clean and without images that will distract the user from the main task, which is calculating the discount information, as well as reducing the system performance.

Moreover, the design of the application also focuses on user friendliness, which reduce the user's input as well as speed up the discount calculation. To illustrate that, the application allows users to directly switch upon Percent Off mode or Fixed Amount mode without the need to jump into other view which make the user convenient to perform task. Furthermore, users are allowed to slide the slider (SeekBar) for the tax input to the desired amount instead of typing by themselves. The application also remains the availability of text input for user to type the value themselves and make the slider (SeekBar) as an option for the user to choose according to their preferences. It is also same for the Discount Percentage in Percent Off mode. In addition, the

user also allowed to enable or disable the calculation for including Tax Rate, by only single touch on the With Tax Switch which serves convenience to the user. On top of that, if user choose to disable Tax Rate, the Tax Rate input will automatically fill in with 0 as 0% without the need for users to type in themselves. Moreover, in Fixed Amount mode, the slider (SeekBar) is removed from the user interface and replaced with 6 number button (-10, -5, -1, +1, +5, +10), which allows the user to press on it and modify the Discount Amount without the need to type in themselves. As identical to Discount Percentage and Tax Rate, the input field is also remained available for user to type in themselves to fulfil different preferences of all users. Last but not least, the application also provided a feature to clear all including the input by user as well as the previous discount information by one click, which is on the toolbar option. It makes the user convenient as they do not have to clear the input field one by one themselves.

1.3 GUI Widgets

The GUI Widgets that have been applied in the application included TextView, EditText, Button, Switch, RadioGroup and RadioButton, SeekBar, toolbar and CardView. Some of the widgets such as toolbar, Radio Button, SeekBar, Switch and Button have applied themes as in themes resourse to make it aesthetic, minimalistic and follow the ideal user interface design. To further illustrate, TextView have been used for the header (Select Discount Type), the symbol at the end of input field ($, %) as well as for displaying the discount information (Your Save:, Tax:, Total Price:…). Next, EditText is used for user input (Sales Price, Discount, Tax). Following that, Button is used for the user to perform calculation (Calculate). Then, RadioGroup is used as a container for 2 RadioButton (Percent Off, Fixed Amount) which serve for users to select discount option. Next, SeekBar is applied for Percent Off Discount and Tax Rate for user to drag the value. Following that, toolbar is the one on the top of the user interface, which showing the application title, application information after pressing the about icon and clear all option after pressing the triple dot icon. Lastly, CardView is used as a container to place the discount information in a card view which make the application cleaner and more aesthetic.

2.0 Screenshots of Application

2.1 Portrait Mode

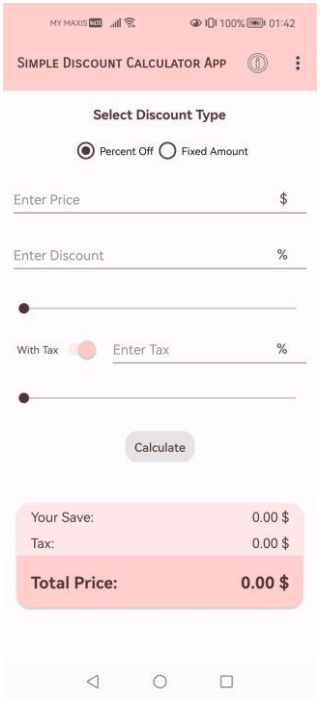

Figure 2.1 Portrait Mode on Percent Off and With Tax On (Calculate Button Disabled as no input)

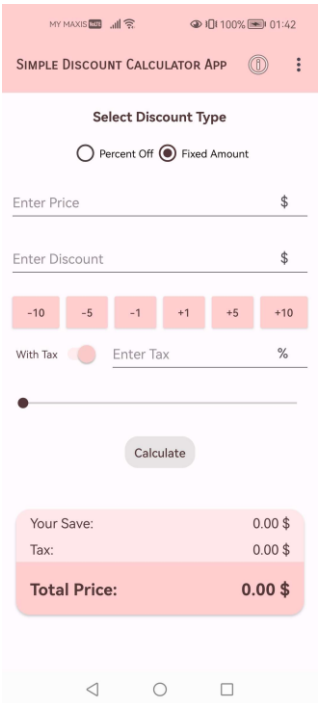

Figure 2.2 Portrait Mode on Fixed Amount and With Tax On (Calculate Button Disabled as no input)

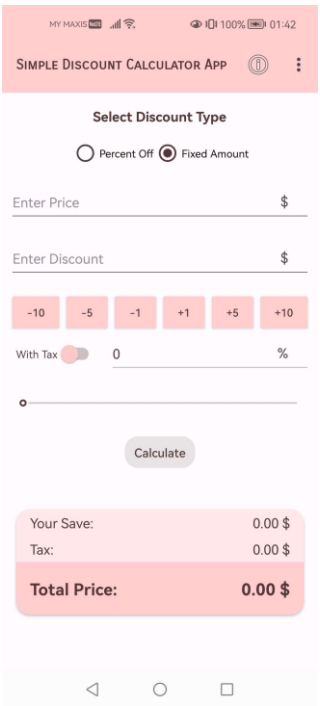

Figure 2.3 Portrait Mode on Fixed Amount and With Tax Off (Calculate Button Disabled as no input)

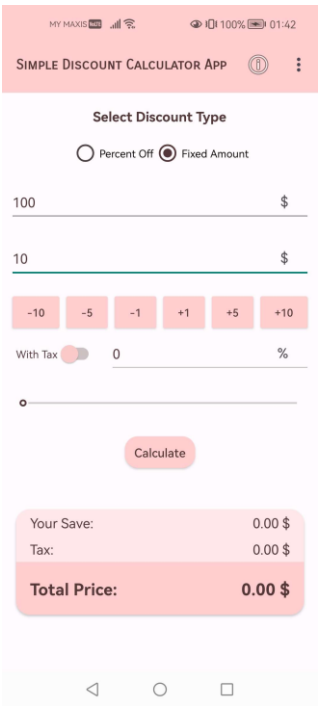

Figure 2.4 Portrait Mode on Fixed Amount and With Tax Off (Calculate Button is enabled as input is valid)

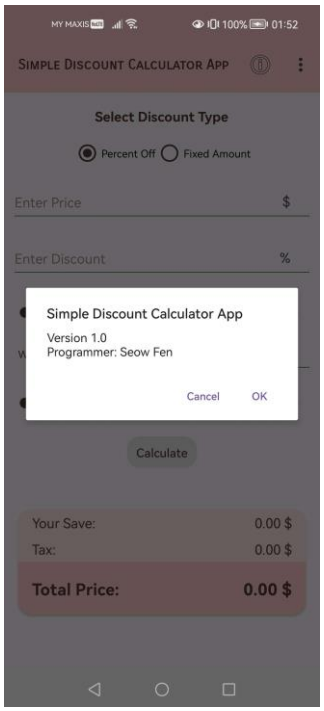

Figure 2.5 Portrait Mode About Message

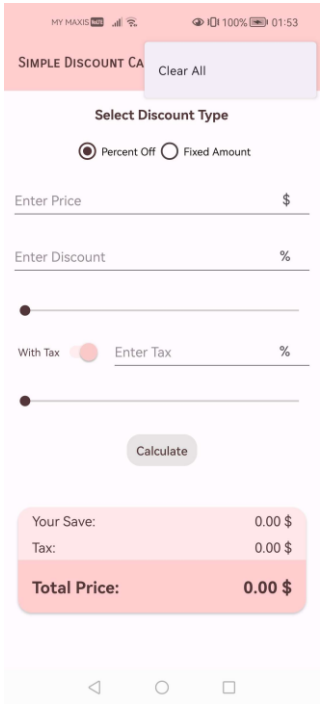

Figure 2.6 Portrait Mode Tool Bar

2.2 Landscape Mode

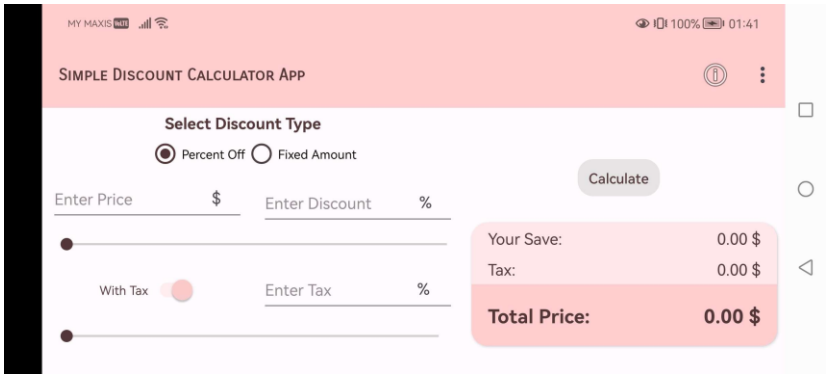

Figure 2.7 Landscape Mode on Percent off

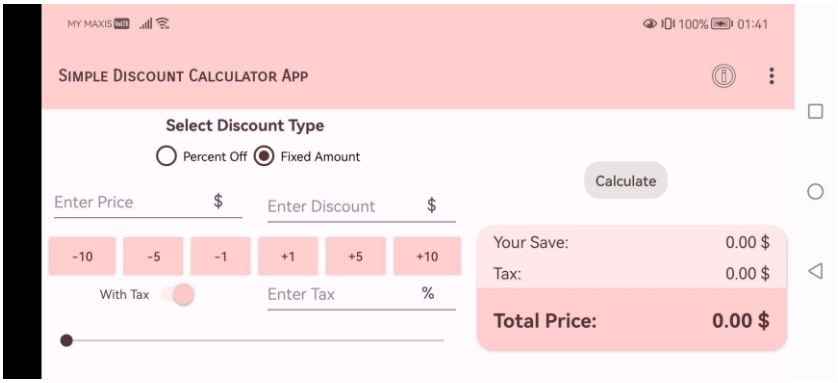

Figure 2.8 Landscape Mode on Fixed Amount

3.0 Test Cases

3.1 Percent Off Option

3.1.1 Test Case 1

	Input		Expected Output
Sales Price ($)	45	Your Save:	4.50 $
Percent Off (%)	10	Tax:	0.00 $
Tax (%)	0	Total Price:	40.50 $

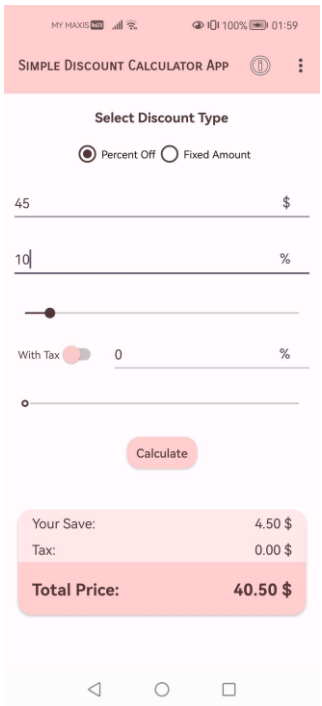

Figure 3.1 Percent Off Test Case 1 Output

3.1.2 Test Case 2

	Input		Expected Output
Sales Price ($)	60	Your Save:	15.00 $
Percent Off (%)	25	Tax:	6.00 $
Tax (%)	10	Total Price:	51.00 $

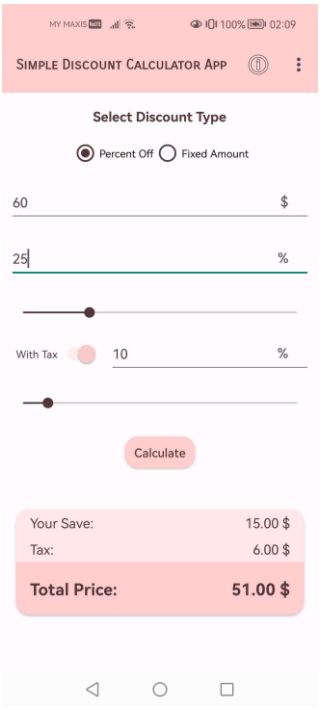

Figure 3.2 Percent Off Test Case 2 Output

3.2 Fixed Amount Option

3.2.1 Test Case 1

	Input		Expected Output
Sales Price ($)	50	Your Save:	10.00$
Fixed Amount ($)	10	Tax:	0.00 $
Tax (%)	0	Total Price:	40.00$

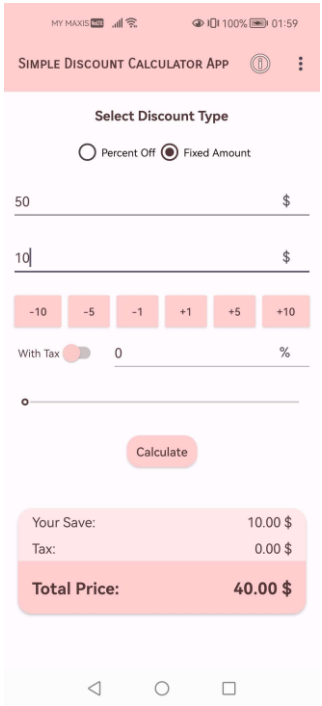

Figure 3.3 Fixed Amount Test Case 1 Output

3.2.2 Test Case 2

	Input		Expected Output
Sales Price ($)	1016	Your Save:	210.00$
Fixed Amount ($)	210	Tax:	60.96 $
Tax (%)	6	Total Price:	866.96

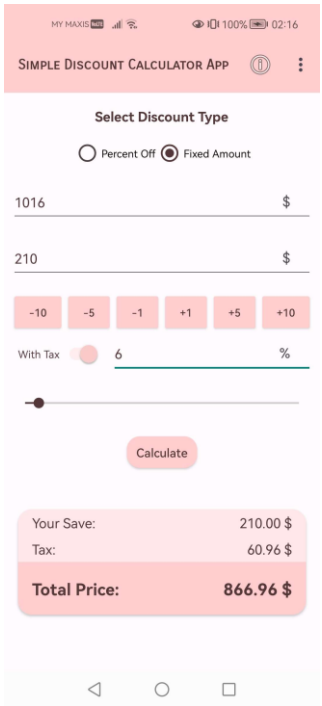

Figure 3.4 Fixed Amount Test Case 2 Output

3.3 Input Validation

3.3.1 Sales Price Validation

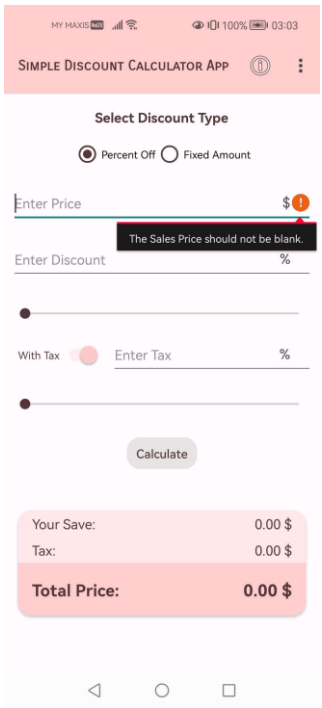

Figure 4.1 Invalid Sales Price – input is blank

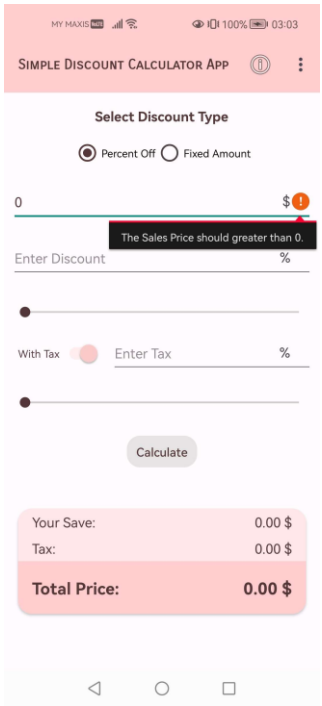

Figure 4.2 Invalid Sales Price – input is not greater than 0

3.3.2 Percent Off Discount Validation

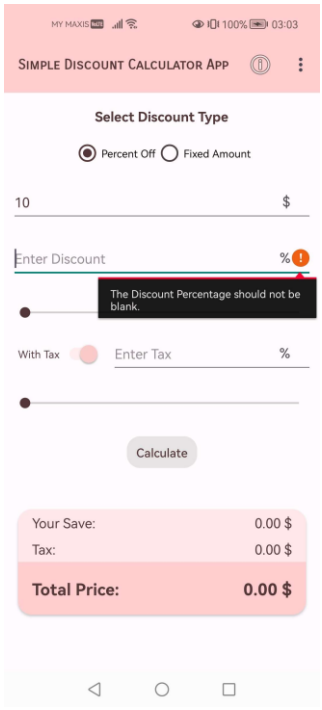

Figure 4.3 Invalid Discount (Percent Off) – input is blank

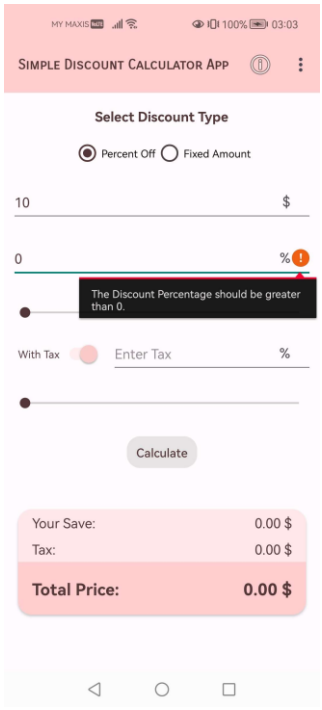

Figure 4.4 Invalid Discount (Percent Off) – input is not greater than 0

3.3.3 Fixed Amount Discount Validation

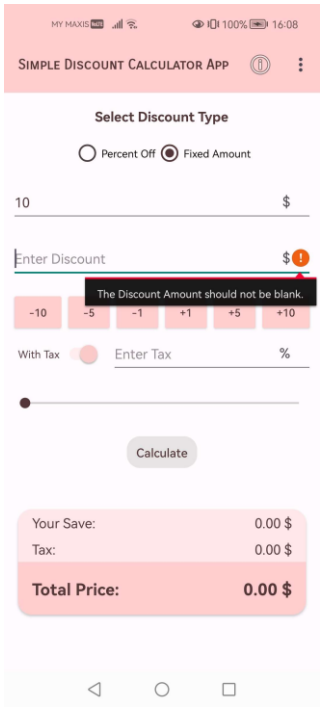

Figure 4.5 Invalid Discount (Fixed Amount) – input is blank

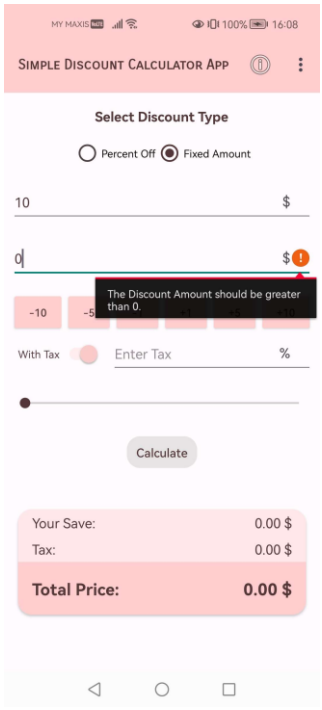

Figure 4.6 Invalid Discount (Fixed Amount) – input is not greater than 0

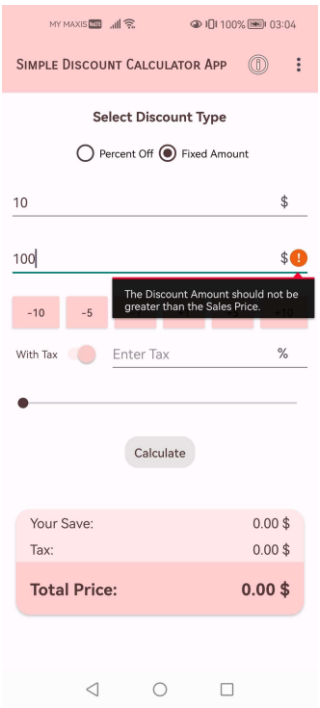

Figure 4.7 Invalid Discount (Fixed Amount) – input is greater than Sales Price

3.3.4 Tax Validation

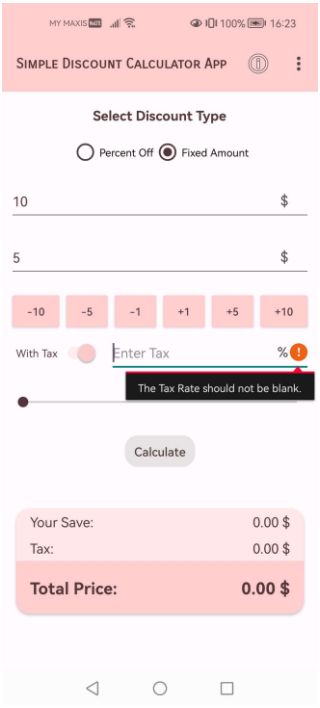

Figure 4.8 Invalid Tax – input is blank

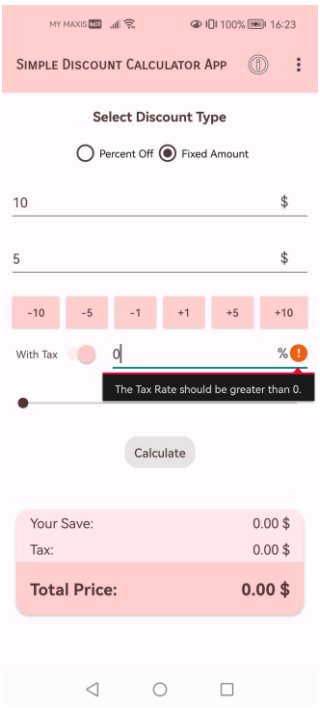

Figure 4.9 Invalid Tax – input is not greater than 0

3.3.5 Validation Handling Details

As range filter function is implemented for Sales Price [0.00-10000000.00], Discount (Percent Off) [0-100], Discount Fixed Amount [0-10000000] and Tax [0-100], hence it will prevent user from entering input other than that range, thus reducing the error handling of input validation. Moreover, the slider (SeekBar) for Percent Off discount and Tax, as well as the button for Fixed Amount have set to ensure that the input field aligns with their respective range.

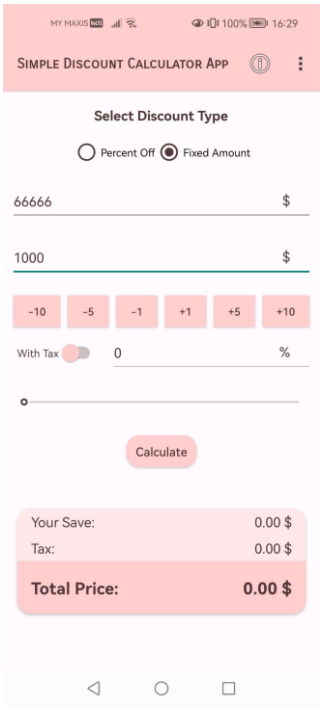

Figure 4.10 In Fixed Amount Mode

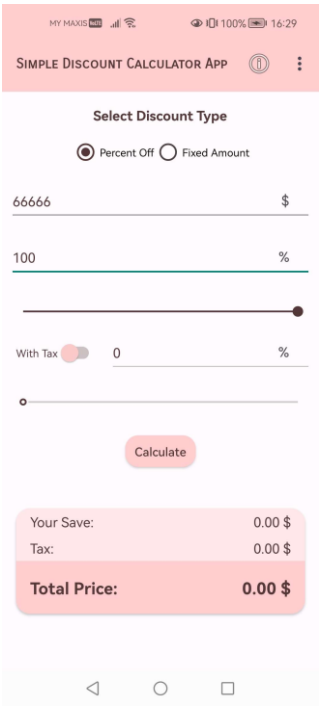

Figure 4.11 Switch to Percent Off Mode

Moreover, the validation handling also included when user entered number that is greater than 100 in Fixed Amount mode and switch to Percent Off, the program will automatically change the Discount Edit Text into 100 as 100 is the maximum for Percent Off mode as shown in *Figure 4.10*, *Figure 4.11*.

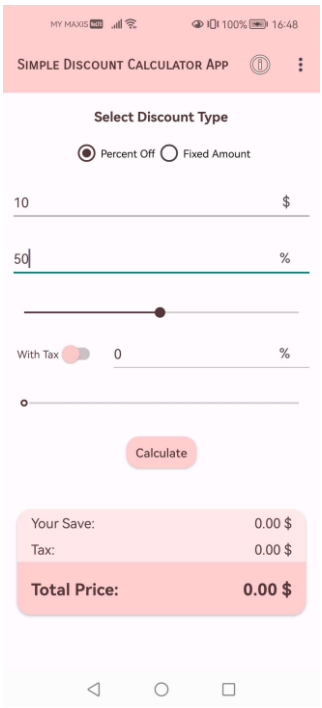

Figure 4.12 In Percent Off Mode

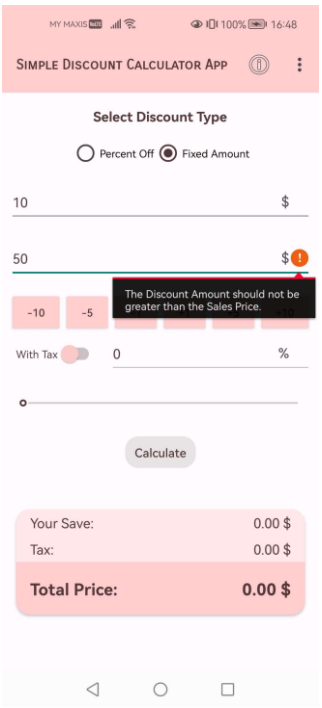

Figure 4.13 Switch to Fixed Amount Mode

Furthermore, there is also a validation handling that handles the checking of Discount (Fixed Amount) is valid or not after switching the mode from Percent Off to Fixed Amount even though user did not make changes to the input field as shown in *Figure 4.12*, *Figure 4.13*.

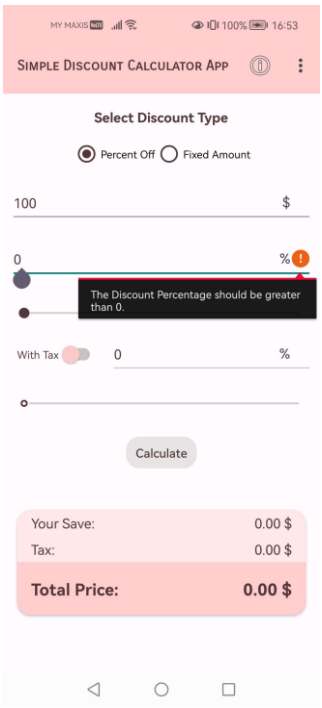

Figure 4.14 In Percent Off Mode

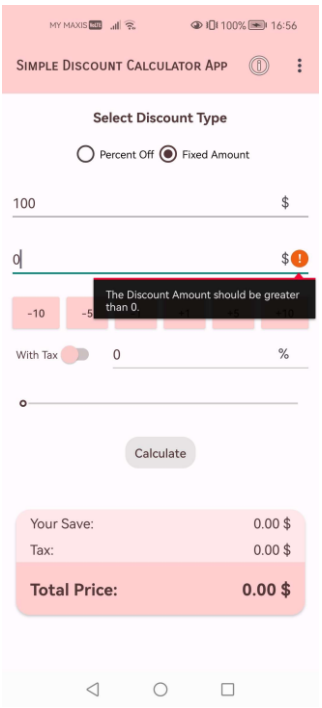

Figure 4.15 Switch to Fixed Amount Mode

In addition, the program also able to switch the error message *according* to the mode that user switch to as shown in *Figure 4.14*, *Figure 4.15* as one of the examples.

3.4 Enable of Calculate Button

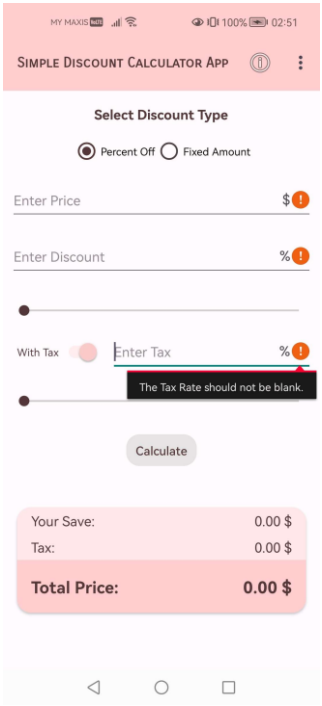

Figure 5.1 3 Invalid Input-- Calculate Button Disabled

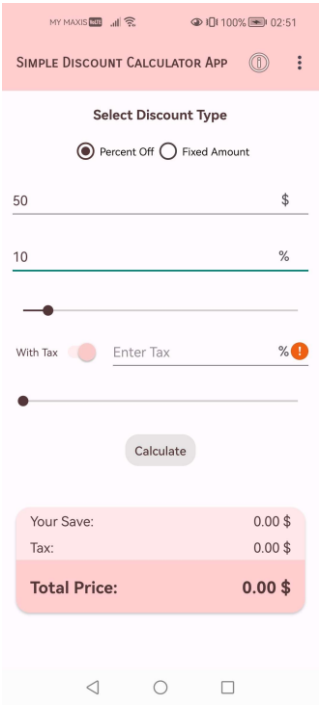

Figure 5.2 1 Invalid Input - Calculate Button Disabled

Figure 5.1 and Figure 5.2 shows that, the Calculate button is not enable as long as there is an invalid input.

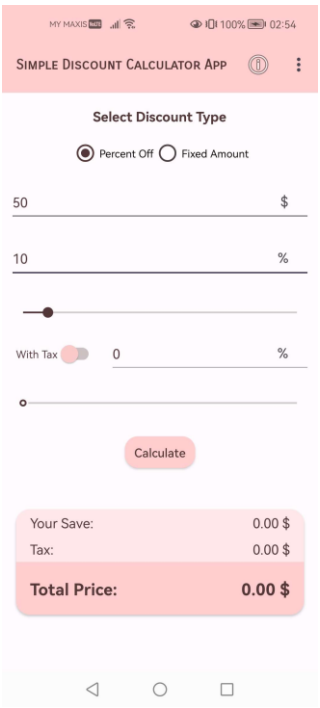

Figure 5.3 No Invalid Input – Calculate Button Enabled

Figure 5.3 shows that, after all the input is valid, the Calculate button is then only enabled.

3.5 Data Persistence

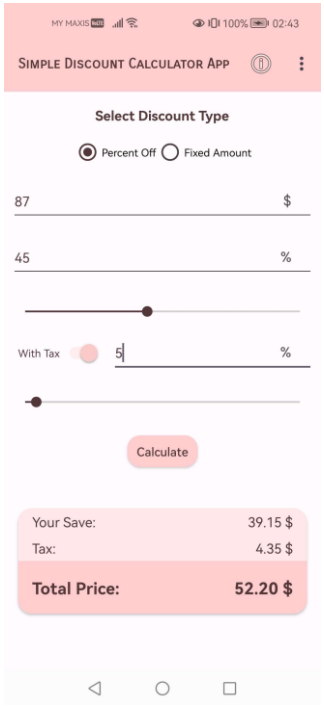

Figure 6.1 Before Configuration Changes

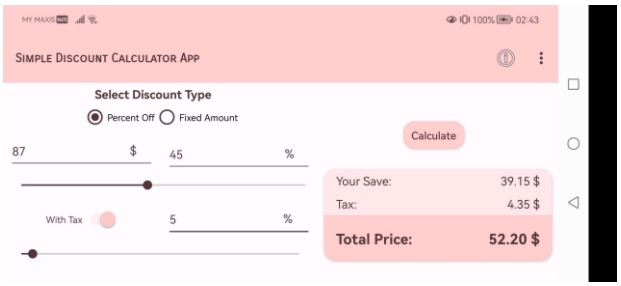

Figure 6.2 After Configuration Changes

As shown in *Figure 6.1* and *Figure 6.2*, the application is able to save and restore the discount information (save amount, tax amount, total price) after configuration changes, which is rotating the phone.

4.0 Weaknesses and Strengths

4.1 Weaknesses

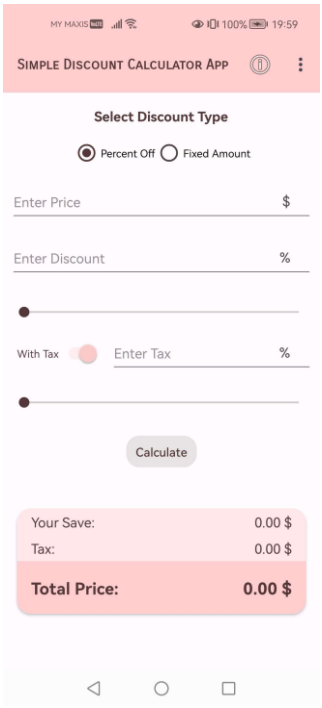

Figure 7.1 Before Rotating

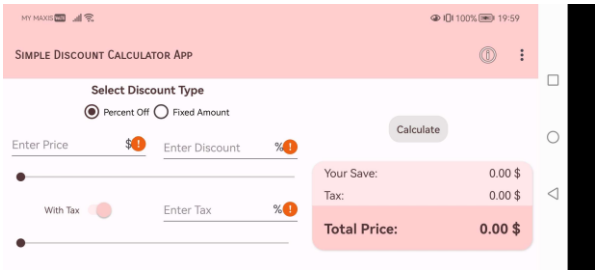

Figure 7.2 After Rotating

One of the weaknesses of the application is that, if the user rotates the phone which cause the configuration changes while have not entered any input yet, it will immediately show the error message for the input validation as shown in *Figure 7.1*, *Figure 7.2* which is undesired behaviour as error message should only show after user have modify the input field with invalid input. Due to the time constraints, have not get to figure out the solution to eliminate this weakness of the application.

Moreover, another weakness of the application is unable to accept the Discount Amount for Fixed Amount Mode as Double type which allow decimal points. The reason is that both Discount Percentage and Discount Amount for Percent Off and Fixed Amount mode is using the same EditText widget, implying that if the input type of Discount changed into numberDecimal, the SeekBar progress for Discount Percentage will also be number with decimal point, which is inconvenient for user to drag as the sensitivity is too high, need to drag very accurately for whole number (Eg: 80.00). Hence, for better user experience, Discount Amount of Fixed Amount mode could only accept integer. It might be a solution to eliminate this weakness but however have not get to figure out due to time constraints.

4.2 Strengths

First of all, one of the strengths of the application is that it has a clean, minimalistic and user-friendly interface. Moreover, the application has applied custom styles to make the application visually appealing and align with Material Design Guidelines or Human Computer Interaction (HCI) standards.

In addition, the application come along with real-time validation and error handling for user inputs as soon as user have modified the input field. To further illustrate the application has included plenty of handling that cover most of the possible situation that may cause error. For example, as mentioned in 3.3.5 Validation Handling Details, even though a range has been set for Discount Percentage of Percent Off mode, it might happens the situation that user key in value larger than the range during Fixed Amount mode and switch to Percent Off mode, causing the error that the input is not flagged as invalid as the TextWatcher only work after user make changes on the EditText, hence set the value programmatically whenever user switch from Fixed Amount to Percent Off mode with Discount that greater than 100 to 100 will handle this situation.

Furthermore, flexibility and convenient to use is one of the highlights of this application. To illustrate that, as stated in **1.2 Design Themes,** the application allows user to choose between directly enter the input themselves or using SeekBar or Button to enter the input. Moreover, the "Clear All" tools option allows users to clear all the input instantly without the need to delete the input manually. In addition, users can also switch between Percent Off and Fixed Amount mode easily and quickly and perform task according to their needs. Moreover, users can choose to perform discount calculation with tax or without including tax rate easily by just turning on and off on the "With Tax" Switch, thus serving convenience and flexibility to users.

Last but not least, the pros of the application that lies on the code layer is that having a proper code structure with commenting that highly improve the readability and maintainability of the code. To illustrate that, most of the code have commented with their respective purpose and similar code have been placed together instead of distributed around everywhere. In addition, the code is well-structured and follows best practices for Android Applications Development. For instance, separating User Interface logic from business logic.

5.0 Self Reflection

5.1 Course Impact and Accomplishments

After enrolling the course, Mobile Application Development (CET3013), I have developed a better understanding on programming which I never learnt such thoroughly and understood the theory. Unlike simply memorizing theory like what I used to do in the past, I am now able to understand the underlying principles and concepts deeply. This is crucial aspect of learning as it empowers me to apply my knowledge effectively. Moreover, I have developed the skill for developing a functional Android Application, which will be very valuable and inspirative, no matter on unlocking a new option for my final year project, or on potential career purpose in the future, or even as a hobby to develop the ideal mobile application that fits my personal requirements and preferences.

By working on this project, I have achieved on implementing functional Switch, SeekBar and TextWatcher that have not been covered in the lecture by researching the online materials. Moreover, I also accomplished the bidirectional synchronization on the EditText and SeekBar, which means the changes made on EditText will reflect on the SeekBar as progress and vice versa, which could provide an interactive user experience. In addition, I also accomplished success on the aspect of handling error of most of the possible situations according to my user interface design. Moreover, by building this application, I also get to understand more about mobile application development as building by my own can always learn more than simply following guidelines or instructions.

5.2 Successes and Problem-Solving in Design and Implementation

Things that went well during the design and implementation of the application includes the user interface, which I am very satisfied with the end product which is ideal and aligned with my aesthetic conception. Most of the design part went well when I am working on the project, there are only a few minor issues and can be addressed by adding theme. However, there is a issue that cost me a lot of time to resolve it, which is, the Fixed Amount Discount Button (-10, -5, -1, +1, +5, +10) appear to be different from what I see on the Android Studio layout design, which those button perfectly fit in the layout design but when I run the application on my smartphone, it turns out to be insufficient place for the button and ended up squeezing up the button as well as the text on it. I have tried tons of approach to fix it, including set smaller margin, set smaller button, set smaller text on the button, applying themes on the Internet, however, none of this approach address the issue. After consuming up plenty of time, finally figured out setting the inset smaller so that the text has larger space to fit in the button.

On the other hand, which is the implementation of functional parts, most of them went pretty well especially on the last part where implementing the listener on calculation button, perform the calculation, display the discount information and view model, the process went very smoothly. The major issue that I faced was the bug caused by the bidirectional synchronization of EditText and SeekBar. To illustrate that, since one of the changes on them will reflect to another, take Tax as example, if the user clear the input of Tax, then the SeekBar progress will also sync to its minimum value, however, I have set the minimum value of SeekBar as “1”, resulting the SeekBar reflect “1” back to the Tax input field and hence causing the user need to delete twice in order to make the tax input field blank again. I have tried plenty of ways to get rid of this problem, but none of them works, when I was actually going to give up as I have spent too much time on this, I have finally figured out a logic implementation way, which is initializing a isTaxClear Boolean variable as false and only reflect the changes on SeekBar to Tax input field when the isTaxClear

is not true and after that immediately set it back to false, and there is also a assignment on isTaxClear equal to Tax.isBlank() in TextWatcher. Hence, it makes it become when user just started the app without entering the input, thus the Tax input field is blank, however, as the initialization of isTaxClear is false, hence the SeekBar can reflect the progress value to the Tax input field. After that, isTaxClear is set to false. If the user clears the Tax input field, the isTaxClear assignment in TextWatcher will set it to true, thus the SeekBar would not reflect the progress “1” to the Tax input field as isTaxClear is not false and right after that isTaxClear is set back to false and hence resulting if the user drags the SeekBar it will reflect the progress to the Tax input field. Thus, resolving the issue and same goes to Discount SeekBar.

6.0 Appendix

6.1 System Functionalities Screencast Recording URL

https://uowmalaysia-my.sharepoint.com/:v:/g/personal/0207368_student_uow_edu_my/EcQN1Jf2IoxLicCzp2MLgdUBoqns94Ga-qRHaQ1S4iJy5Q?e=zjIaL0&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJTdHJlYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOiJTaGFyZURpYWxvZyIsInJlZmVycmFsQXBwUGxhdGZvcm0iOiJXZWIiLCJyZWZlcnJhbE1vZGUiOiJ2aWV3In19

6.2 layout\activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@color/pink"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/text_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="sans-serif-smallcaps"
            android:text="Simple Discount Calculator App"
            android:textColor="@color/brown"
            android:textSize="16dp"
            android:textStyle="bold"
            tools:layout_editor_absoluteX="16dp"
            tools:layout_editor_absoluteY="22dp" />

    </androidx.appcompat.widget.Toolbar>

    <RadioGroup
        android:id="@+id/group_mode"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_header">
```

```
        <RadioButton
            android:id="@+id/radio_percent_off"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/radio_percentOff"
            android:textAppearance="@style/TextAppearance.AppCompat.Small"
            android:theme="@style/RadioButton" />

        <RadioButton
            android:id="@+id/radio_fixed_amount"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:theme="@style/RadioButton"
            android:text="@string/radio_fixed_amount"
            android:textAppearance="@style/TextAppearance.AppCompat.Small" />
    </RadioGroup>

    <EditText
        android:id="@+id/text_sales_price"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:ems="10"
        android:hint="@string/text_sales_price"
        android:inputType="numberDecimal"
        android:textColor="@color/brown"
        android:textColorHint="@color/light_brown"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/group_mode" />

    <EditText
        android:id="@+id/text_discount"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:ems="10"
        android:hint="@string/text_discount"
        android:inputType="number"
        android:textColor="@color/brown"
        android:textColorHint="@color/light_brown"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_sales_price" />

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
```

```
        android:layout_marginEnd="8dp"
        android:text="@string/text_header"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        android:textColor="@color/brown"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/group_mode"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/toolbar"
        app:layout_constraintVertical_bias="0.0" />

    <TextView
        android:id="@+id/text_sales_price_symbol"
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_marginTop="12dp"
        android:layout_marginEnd="20dp"
        android:gravity="end"
        android:text="@string/text_sales_price_symbol"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintEnd_toEndOf="@+id/text_sales_price"
        app:layout_constraintHorizontal_bias="0.981"
        app:layout_constraintStart_toStartOf="@+id/text_sales_price"
        app:layout_constraintTop_toTopOf="@+id/text_sales_price" />

    <TextView
        android:id="@+id/text_discount_symbol"
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_marginTop="12dp"
        android:layout_marginEnd="20dp"
        android:gravity="end"
        android:text="@string/text_discount_symbol"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintEnd_toEndOf="@+id/text_discount"
        app:layout_constraintHorizontal_bias="0.981"
        app:layout_constraintStart_toStartOf="@+id/text_discount"
        app:layout_constraintTop_toTopOf="@+id/text_discount" />

    <SeekBar
        android:id="@+id/slider_discount_percent"
        style="@style/Widget.AppCompat.SeekBar"
        android:layout_width="0dp"
        android:layout_height="32dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:indeterminate="false"
        android:theme="@style/Slider"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.495"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/guideline1" />

    <SeekBar
        android:id="@+id/slider_tax_percent"
```

```
        android:layout_width="0dp"
        android:layout_height="32dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:theme="@style/Slider"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_tax" />

    <EditText
        android:id="@+id/text_tax"
        android:layout_width="230dp"
        android:layout_height="48dp"
        android:layout_marginEnd="8dp"
        android:ems="10"
        android:hint="@string/text_tax"
        android:inputType="number"
        android:textColor="@color/brown"
        android:textColorHint="@color/light_brown"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline2" />

    <androidx.appcompat.widget.SwitchCompat
        android:id="@+id/switch_tax"
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_marginStart="8dp"
        android:showText="false"
        android:splitTrack="false"
        android:text="@string/switch_tax"
        android:textColor="@color/brown"
        android:theme="@style/Switch"
        app:layout_constraintEnd_toStartOf="@+id/text_tax"
        app:layout_constraintHorizontal_bias="0.402"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/guideline2" />

    <TextView
        android:id="@+id/text_tax_symbol"
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_marginTop="12dp"
        android:layout_marginEnd="20dp"
        android:gravity="end"
        android:text="@string/text_tax_symbol"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintEnd_toEndOf="@+id/text_tax"
        app:layout_constraintHorizontal_bias="0.968"
        app:layout_constraintStart_toStartOf="@+id/text_tax"
        app:layout_constraintTop_toTopOf="@+id/text_tax" />

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/button_calculate"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:enabled="false"
        android:text="@string/button_calculate"
        android:textAllCaps="false"
        android:textSize="16sp"
        android:theme="@style/CalculateButton"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/slider_tax_percent" />

    <androidx.cardview.widget.CardView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:layout_marginBottom="32dp"
        app:cardBackgroundColor="#FFFFFF"
        app:cardCornerRadius="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button_calculate">
        <!--Discount Information Table-->

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/button_calculate">

            <!-- Save amount-->

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/light_pink"
                android:orientation="horizontal">

                <TextView
                    android:id="@+id/text_save_label"
                    android:layout_width="0dp"
                    android:layout_height="30dp"
                    android:layout_marginStart="16dp"
                    android:layout_weight="1"
                    android:gravity="start|center_vertical"
                    android:text="@string/text_save_label"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                    android:textColor="@color/brown" />

                <TextView
```

```
                    android:id="@+id/text_save_amount"
                    android:layout_width="0dp"
                    android:layout_height="30dp"
                    android:layout_marginEnd="16dp"
                    android:layout_weight="1"
                    android:gravity="center_vertical|end"
                    android:text="@string/text_save_amount"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                    android:textColor="@color/brown" />
            </LinearLayout>

            <!-- Tax amount -->
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/light_pink"
                android:orientation="horizontal">

                <TextView
                    android:id="@+id/text_tax_label"
                    android:layout_width="0dp"
                    android:layout_height="30dp"
                    android:layout_marginStart="16dp"
                    android:layout_weight="1"
                    android:gravity="start|center_vertical"
                    android:text="@string/text_tax_label"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                    android:textColor="@color/brown" />

                <TextView
                    android:id="@+id/text_tax_amount"
                    android:layout_width="0dp"
                    android:layout_height="30dp"
                    android:layout_marginEnd="16dp"
                    android:layout_weight="1"
                    android:gravity="center_vertical|end"
                    android:text="@string/text_tax_amount"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                    android:textColor="@color/brown" />

            </LinearLayout>

            <!-- Tax amount -->
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/pink"
                android:orientation="horizontal">

                <TextView
                    android:id="@+id/text_total_label"
                    android:layout_width="0dp"
                    android:layout_height="60dp"
                    android:layout_marginStart="16dp"
```

```
                    android:layout_weight="1"
                    android:gravity="start|center_vertical"
                    android:text="@string/text_total_label"

android:textAppearance="@style/TextAppearance.AppCompat.Large"
                    android:textColor="@color/brown"
                    android:textStyle="bold" />

                <TextView
                    android:id="@+id/text_total_amount"
                    android:layout_width="0dp"
                    android:layout_height="60dp"
                    android:layout_marginEnd="16dp"
                    android:layout_weight="1"
                    android:gravity="center_vertical|end"
                    android:text="@string/text_total_amount"

android:textAppearance="@style/TextAppearance.AppCompat.Large"
                    android:textColor="@color/brown"
                    android:textStyle="bold" />

            </LinearLayout>
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <LinearLayout
        android:id="@+id/buttonGroup"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:gravity="center_horizontal|center_vertical"
        android:orientation="horizontal"
        android:visibility="invisible"
        app:layout_constraintBottom_toTopOf="@id/guideline2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/guideline1"
        tools:context=".MainActivity">

        <Button
            android:id="@+id/button_minus_ten"
            style="@style/Widget.AppCompat.Button.Small"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:insetLeft="-10dp"
            android:insetTop="4dp"
            android:insetRight="-10dp"
            android:insetBottom="4dp"
            android:minWidth="48dp"
            android:text="@string/button_minus_ten"
            android:textAllCaps="false"
            android:textColor="@color/brown"
            android:textSize="14sp"
            app:backgroundTint="@color/pink" />
```

```
<Button
    android:id="@+id/button_minus_five"
    style="@style/Widget.AppCompat.Button.Small"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.9"
    android:insetLeft="-10dp"
    android:insetRight="-10dp"
    android:text="@string/button_minus_five"
    android:textColor="@color/brown"
    android:textSize="14sp"
    app:backgroundTint="@color/pink" />

<Button
    android:id="@+id/button_minus_one"
    style="@style/Widget.AppCompat.Button.Small"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.9"
    android:insetLeft="-10dp"
    android:insetRight="-10dp"
    android:text="@string/button_minus_one"
    android:textColor="@color/brown"
    android:textSize="14sp"
    app:backgroundTint="@color/pink" />

<Button
    android:id="@+id/button_plus_one"
    style="@style/Widget.AppCompat.Button.Small"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.9"
    android:insetLeft="-10dp"
    android:insetRight="-10dp"
    android:text="@string/button_plus_one"
    android:textColor="@color/brown"
    android:textSize="14sp"
    app:backgroundTint="@color/pink" />

<Button
    android:id="@+id/button_plus_five"
    style="@style/Widget.AppCompat.Button.Small"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.9"
    android:insetLeft="-10dp"
    android:insetRight="-10dp"
    android:text="@string/button_plus_five"
    android:textColor="@color/brown"
    android:textSize="14sp"
    app:backgroundTint="@color/pink" />

<Button
    android:id="@+id/button_plus_ten"
    style="@style/Widget.AppCompat.Button.Small"
    android:layout_width="0dp"
```

```
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:insetLeft="-10dp"
            android:insetRight="-10dp"
            android:text="@string/button_plus_ten"
            android:textColor="@color/brown"
            android:textSize="14sp"
            app:backgroundTint="@color/pink" />
    </LinearLayout>

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="282dp" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="337dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

6.3 land\activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@color/pink"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/text_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="sans-serif-smallcaps"
            android:text="Simple Discount Calculator App"
            android:textColor="@color/brown"
            android:textSize="16dp"
            android:textStyle="bold"
```

```
            tools:layout_editor_absoluteX="16dp"
            tools:layout_editor_absoluteY="22dp" />

    </androidx.appcompat.widget.Toolbar>

    <RadioGroup
        android:id="@+id/group_mode"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:orientation="horizontal"
        app:layout_constraintEnd_toStartOf="@+id/guideline"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_header">

        <RadioButton
            android:id="@+id/radio_percent_off"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/radio_percentOff"
            android:textAppearance="@style/TextAppearance.AppCompat.Small"
            android:theme="@style/RadioButton" />

        <RadioButton
            android:id="@+id/radio_fixed_amount"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/radio_fixed_amount"
            android:textAppearance="@style/TextAppearance.AppCompat.Small"
            android:theme="@style/RadioButton" />
    </RadioGroup>

    <EditText
        android:id="@+id/text_sales_price"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="4dp"
        android:layout_marginEnd="8dp"
        android:ems="10"
        android:hint="@string/text_sales_price"
        android:inputType="numberDecimal"
        android:textColor="@color/brown"
        android:textColorHint="@color/light_brown"
        app:layout_constraintEnd_toStartOf="@+id/guideline3"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/group_mode" />

    <EditText
        android:id="@+id/text_discount"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:ems="10"
```

```
        android:hint="@string/text_discount"
        android:inputType="number"
        android:textColor="@color/brown"
        android:textColorHint="@color/light_brown"
        app:layout_constraintEnd_toStartOf="@+id/guideline"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="@+id/guideline3"
        app:layout_constraintTop_toBottomOf="@+id/group_mode" />

    <TextView
        android:id="@+id/text_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:layout_marginEnd="8dp"
        android:gravity="center"
        android:text="@string/text_header"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        android:textColor="@color/brown"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@+id/group_mode"
        app:layout_constraintEnd_toStartOf="@+id/guideline"
        app:layout_constraintHorizontal_bias="0.478"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/toolbar"
        app:layout_constraintVertical_bias="0.0" />

    <TextView
        android:id="@+id/text_sales_price_symbol"
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_marginTop="12dp"
        android:layout_marginEnd="20dp"
        android:gravity="end"
        android:text="@string/text_sales_price_symbol"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintEnd_toEndOf="@+id/text_sales_price"
        app:layout_constraintHorizontal_bias="0.981"
        app:layout_constraintStart_toStartOf="@+id/text_sales_price"
        app:layout_constraintTop_toTopOf="@+id/text_sales_price" />

    <TextView
        android:id="@+id/text_discount_symbol"
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_marginTop="12dp"
        android:layout_marginEnd="20dp"
        android:gravity="end"
        android:text="@string/text_discount_symbol"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintEnd_toEndOf="@+id/text_discount"
        app:layout_constraintHorizontal_bias="0.981"
        app:layout_constraintStart_toStartOf="@+id/text_discount"
        app:layout_constraintTop_toTopOf="@+id/text_discount" />

    <SeekBar
        android:id="@+id/slider_discount_percent"
```

```
        style="@style/Widget.AppCompat.SeekBar"
        android:layout_width="0dp"
        android:layout_height="32dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="4dp"
        android:indeterminate="false"
        android:theme="@style/Slider"
        app:layout_constraintEnd_toStartOf="@+id/guideline"
        app:layout_constraintHorizontal_bias="0.495"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_sales_price" />

    <SeekBar
        android:id="@+id/slider_tax_percent"
        android:layout_width="0dp"
        android:layout_height="32dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="4dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="16dp"
        android:theme="@style/Slider"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/guideline"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_tax"
        app:layout_constraintVertical_bias="0.144" />

    <EditText
        android:id="@+id/text_tax"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginStart="8dp"
        android:layout_marginEnd="8dp"
        android:ems="10"
        android:hint="@string/text_tax"
        android:inputType="number"
        android:textColor="@color/brown"
        android:textColorHint="@color/light_brown"
        app:layout_constraintEnd_toStartOf="@+id/guideline"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="@+id/guideline3"
        app:layout_constraintTop_toTopOf="@+id/guideline1" />

    <androidx.appcompat.widget.SwitchCompat
        android:id="@+id/switch_tax"
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_marginStart="8dp"
        android:showText="false"
        android:splitTrack="false"
        android:text="@string/switch_tax"
        android:textColor="@color/brown"
        android:theme="@style/Switch"
        app:layout_constraintEnd_toStartOf="@+id/text_tax"
        app:layout_constraintHorizontal_bias="0.431"
        app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintTop_toTopOf="@+id/guideline1" />

    <TextView
        android:id="@+id/text_tax_symbol"
        android:layout_width="wrap_content"
        android:layout_height="48dp"
        android:layout_marginTop="12dp"
        android:layout_marginEnd="20dp"
        android:gravity="end"
        android:text="@string/text_tax_symbol"
        android:textAppearance="@style/TextAppearance.AppCompat.Medium"
        app:layout_constraintEnd_toEndOf="@+id/text_tax"
        app:layout_constraintHorizontal_bias="0.968"
        app:layout_constraintStart_toStartOf="@+id/text_tax"
        app:layout_constraintTop_toTopOf="@+id/text_tax" />

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/button_calculate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:enabled="false"
        android:text="@string/button_calculate"
        android:textAllCaps="false"
        android:textSize="16sp"
        android:theme="@style/CalculateButton"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.477"
        app:layout_constraintStart_toStartOf="@+id/guideline"
        app:layout_constraintTop_toTopOf="@+id/guideline2" />

    <androidx.cardview.widget.CardView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginBottom="16dp"
        app:cardBackgroundColor="#FFFFFF"
        app:cardCornerRadius="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="@+id/guideline"
        app:layout_constraintTop_toBottomOf="@+id/button_calculate">
        <!--Discount Information Table-->

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/button_calculate">

            <!-- Save amount-->
```

```
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/light_pink"
                android:orientation="horizontal">

                <TextView
                    android:id="@+id/text_save_label"
                    android:layout_width="0dp"
                    android:layout_height="30dp"
                    android:layout_marginStart="16dp"
                    android:layout_weight="1"
                    android:gravity="start|center_vertical"
                    android:text="@string/text_save_label"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                    android:textColor="@color/brown" />

                <TextView
                    android:id="@+id/text_save_amount"
                    android:layout_width="0dp"
                    android:layout_height="30dp"
                    android:layout_marginEnd="16dp"
                    android:layout_weight="1"
                    android:gravity="center_vertical|end"
                    android:text="@string/text_save_amount"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                    android:textColor="@color/brown" />
            </LinearLayout>

            <!-- Tax amount -->
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/light_pink"
                android:orientation="horizontal">

                <TextView
                    android:id="@+id/text_tax_label"
                    android:layout_width="0dp"
                    android:layout_height="30dp"
                    android:layout_marginStart="16dp"
                    android:layout_weight="1"
                    android:gravity="start|center_vertical"
                    android:text="@string/text_tax_label"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                    android:textColor="@color/brown" />

                <TextView
                    android:id="@+id/text_tax_amount"
                    android:layout_width="0dp"
                    android:layout_height="30dp"
                    android:layout_marginEnd="16dp"
                    android:layout_weight="1"
                    android:gravity="center_vertical|end"
```

```
                    android:text="@string/text_tax_amount"

android:textAppearance="@style/TextAppearance.AppCompat.Medium"
                    android:textColor="@color/brown" />

            </LinearLayout>

            <!-- Tax amount -->
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:background="@color/pink"
                android:orientation="horizontal">

                <TextView
                    android:id="@+id/text_total_label"
                    android:layout_width="0dp"
                    android:layout_height="60dp"
                    android:layout_marginStart="16dp"
                    android:layout_weight="1"
                    android:gravity="start|center_vertical"
                    android:text="@string/text_total_label"

android:textAppearance="@style/TextAppearance.AppCompat.Large"
                    android:textColor="@color/brown"
                    android:textStyle="bold" />

                <TextView
                    android:id="@+id/text_total_amount"
                    android:layout_width="0dp"
                    android:layout_height="60dp"
                    android:layout_marginEnd="16dp"
                    android:layout_weight="1"
                    android:gravity="center_vertical|end"
                    android:text="@string/text_total_amount"

android:textAppearance="@style/TextAppearance.AppCompat.Large"
                    android:textColor="@color/brown"
                    android:textStyle="bold" />

            </LinearLayout>
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <LinearLayout
        android:id="@+id/buttonGroup"
        android:layout_width="403dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="4dp"
        android:layout_marginEnd="8dp"
        android:gravity="center_horizontal|center_vertical"
        android:orientation="horizontal"
        android:visibility="invisible"
        app:layout_constraintEnd_toStartOf="@+id/guideline"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/text_sales_price"
```

```
        tools:context=".MainActivity">

        <Button
            android:id="@+id/button_minus_ten"
            style="@style/Widget.AppCompat.Button.Small"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:insetLeft="-10dp"
            android:insetTop="4dp"
            android:insetRight="-10dp"
            android:insetBottom="4dp"
            android:minWidth="48dp"
            android:text="@string/button_minus_ten"
            android:textAllCaps="false"
            android:textColor="@color/brown"
            android:textSize="14sp"
            app:backgroundTint="@color/pink" />

        <Button
            android:id="@+id/button_minus_five"
            style="@style/Widget.AppCompat.Button.Small"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.9"
            android:insetLeft="-10dp"
            android:insetRight="-10dp"
            android:text="@string/button_minus_five"
            android:textColor="@color/brown"
            android:textSize="14sp"
            app:backgroundTint="@color/pink" />

        <Button
            android:id="@+id/button_minus_one"
            style="@style/Widget.AppCompat.Button.Small"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.9"
            android:insetLeft="-10dp"
            android:insetRight="-10dp"
            android:text="@string/button_minus_one"
            android:textColor="@color/brown"
            android:textSize="14sp"
            app:backgroundTint="@color/pink" />

        <Button
            android:id="@+id/button_plus_one"
            style="@style/Widget.AppCompat.Button.Small"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.9"
            android:insetLeft="-10dp"
            android:insetRight="-10dp"
            android:text="@string/button_plus_one"
            android:textColor="@color/brown"
            android:textSize="14sp"
            app:backgroundTint="@color/pink" />
```

```
        <Button
            android:id="@+id/button_plus_five"
            style="@style/Widget.AppCompat.Button.Small"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.9"
            android:insetLeft="-10dp"
            android:insetRight="-10dp"
            android:text="@string/button_plus_five"
            android:textColor="@color/brown"
            android:textSize="14sp"
            app:backgroundTint="@color/pink" />

        <Button
            android:id="@+id/button_plus_ten"
            style="@style/Widget.AppCompat.Button.Small"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:insetLeft="-10dp"
            android:insetRight="-10dp"
            android:text="@string/button_plus_ten"
            android:textColor="@color/brown"
            android:textSize="14sp"
            app:backgroundTint="@color/pink" />
    </LinearLayout>

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="217dp" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="96dp" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_begin="410dp" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_begin="205dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

6.4 themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Base.Theme.SimpleDiscountCalculator"
parent="Theme.Material3.DayNight.NoActionBar">
        <!-- Customize your light theme here. -->
        <!-- <item name="colorPrimary">@color/my_light_primary</item> -->
        <item name="android:windowLightStatusBar">true</item>
        <item name="android:statusBarColor">@color/pink</item>
    </style>

    <style name="Theme.SimpleDiscountCalculator"
parent="Base.Theme.SimpleDiscountCalculator" />

    <style name="ToolbarTitle"
parent="TextAppearance.AppCompat.Widget.ActionBar.Title">
        <item name="android:textSize">16sp</item>
        <item name="android:textStyle">bold</item>
        <item name="android:textColor">@color/brown</item>
    </style>

    <style name="RadioButton" parent="Theme.AppCompat.Light">
        <item name="android:textColor">@color/brown</item>
        <item name="colorControlNormal">@color/brown_grey</item>
        <item name="colorControlActivated">@color/brown</item>
    </style>

    <style name="Slider" parent="Theme.AppCompat.Light">
        <!-- progress background color -->
        <item name="android:progressBackgroundTint">@color/brown_grey</item>
        <!-- thumb color -->
        <item name="android:thumbTint">@color/brown</item>
        <!-- progress foreground color -->
        <item name="android:progressTint">@color/brown</item>
    </style>

    <style name="Switch" parent="Theme.AppCompat.Light">
        <!-- active thumb and track color  -->
        <item name="colorControlActivated">@color/pink</item>
        <!-- inactive thumb color -->
        <item name="colorSwitchThumbNormal">@color/pink</item>
        <!-- inactive track color -->
        <item name="android:colorForeground">@color/brown</item>
    </style>

    <style name="CalculateButton" parent="ThemeOverlay.AppCompat.Light">
        <item name="colorButtonNormal">@color/light_grey</item>
        <item name="colorControlHighlight">@color/light_pink</item>
        <item name="android:buttonCornerRadius">15dp</item>
        <item name="colorAccent">@color/pink</item>
        <item name="android:textColor">@color/brown</item>
        <item
name="android:buttonStyle">@style/Widget.AppCompat.Button.Colored</item>
    </style>
```

```
</resources>
```

6.5 strings.xml

```
<resources>
    <string name="app_name">Simple Discount Calculator App</string>

    //discount types select
    <string name="text_header">Select Discount Type</string>
    <string name="radio_percentOff">Percent Off</string>
    <string name="radio_fixed_amount">Fixed Amount</string>
    //user input
    <string name="text_sales_price">Enter Price</string>
    <string name="text_discount">Enter Discount</string>
    <string name="text_sales_price_symbol">$</string>
    <string name="text_discount_symbol">%</string>
    <string name="text_tax_symbol">%</string>
    <string name="switch_tax">With Tax</string>
    <string name="text_tax">Enter Tax</string>
    <string name="button_calculate">Calculate</string>
    <string name="button_minus_ten">-10</string>
    <string name="button_minus_five">-5</string>
    <string name="button_minus_one">-1</string>
    <string name="button_plus_one">+1</string>
    <string name="button_plus_five">+5</string>
    <string name="button_plus_ten">+10</string>
    //user input validation
    <string name="sales_price_error1">The Sales Price should not be
blank.</string>
    <string name="sales_price_error2">The Sales Price should greater than
0.</string>
    <string name="fixed_amount_error1">The Discount Amount should not be
blank.</string>
    <string name="fixed_amount_error2">The Discount Amount should be greater
than 0.</string>
    <string name="fixed_amount_error3">The Discount Amount should not be
greater than the Sales Price.</string>
    <string name="percent_off_error1">The Discount Percentage should not be
blank.</string>
    <string name="percent_off_error2">The Discount Percentage should be
greater than 0.</string>
    <string name="tax_error1">The Tax Rate should not be blank.</string>
    <string name="tax_error2">The Tax Rate should be greater than 0.</string>
    //output
    <string name="text_save_label">Your Save:</string>
    <string name="text_save_amount">0.00 $</string>
    <string name="text_tax_label">Tax:</string>
    <string name="text_tax_amount">0.00 $</string>
    <string name="text_total_label">Total Price:</string>
    <string name="text_total_amount">0.00 $</string>
    //menu item
    <string name="menu_clear">Clear All</string>
    <string name="about">About</string>
    <string name="about_message">Version 1.0\nProgrammer: Seow Fen</string>
```

```
    <string name="about_title">Simple Discount Calculator App</string>
</resources>
```

6.6 colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="pink">#FFCDCD</color>
    <color name="light_pink">#FFE6E8</color>
    <color name="brown">#533638</color>
    <color name="light_brown">#974A4043</color>
    <color name="brown_grey">#4A4043</color>
    <color name="grey">#42CACACA</color>
    <color name="light_grey">#9F9F9F</color>
</resources>
```

6.7 menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/menu_clear"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:title="@string/menu_clear" />
    <item
        android:id="@+id/menu_about"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:icon="@android:drawable/ic_menu_info_details"
        android:title="@string/about"
        app:showAsAction="ifRoom" />
</menu>
```

6.8 MainActivity.kt

```
package com.example.simplediscountcalculator

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.text.Editable
import android.text.TextWatcher
import android.view.Menu
import android.view.MenuItem
import android.view.View
import android.widget.SeekBar
import androidx.core.view.isVisible
import androidx.lifecycle.ViewModelProvider
import com.example.simplediscountcalculator.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity(), View.OnClickListener {
    lateinit var binding: ActivityMainBinding
    private lateinit var viewModel:DiscountModel
    var isSalesPriceValid = false // For Sales Price Validation
    var isDiscountValid = false // For Discount Validation
    var isTaxValid = false // For Tax Validation
    var isTaxClear = false // To avoid the syncing problem between TaxWatcher and Seekbar Listener
    var isDiscountClear = false // To avoid the syncing problem between TaxWatcher and Seekbar Listener
```

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)

    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)

    // Add the Action Bar at here
    setSupportActionBar(binding.toolbar)
    // Hide the default title as wanted to show customized title
    supportActionBar?.setDisplayShowTitleEnabled(false)

    viewModel = ViewModelProvider(this)[DiscountModel::class.java]

    // Set the default mode as Percent Off
    binding.radioPercentOff.isChecked = true
    // Set the default mode as With Tax
    binding.switchTax.isChecked = true

    // Set up Discount Slider max to allow sliding from 1 to 100
    binding.sliderDiscountPercent.min = 1
    binding.sliderDiscountPercent.max = 100
    // Set up Tax Slider max to allow sliding from 1 to 100
    binding.sliderTaxPercent.min = 1
    binding.sliderTaxPercent.max = 100

    // Set up the range for Sales Price input
    binding.textSalesPrice.filters = arrayOf(DoubleRangeInputFilter(0.0, 10000000.0))
    // Set up the range for Percent Off Discount for the first time as default mode = Percent Off
    binding.textDiscount.filters = arrayOf(RangeInputFilter(0, 100))
    // Set up the range for Tax input
    binding.textTax.filters = arrayOf(RangeInputFilter(0, 100))

    // Setup SeekBarChange Listener for DiscountPercent (Syncing the slider progress according to text)
    val discountSliderListener = object : SeekBar.OnSeekBarChangeListener {
        override fun onProgressChanged(seekBar: SeekBar?, progress: Int, fromUser: Boolean) {
            // Update the EditText with the Slider progress
            if (!isDiscountClear) {
                binding.textDiscount.setText(progress.toString())
                binding.textDiscount.setSelection(binding.textDiscount.text.length)
            }
            /* Use isDiscountClear to fix the bug of syncing SeekerBar and EditTax so that
            * it will not appear 1 when user clear input as it is the min value of SeekerBar*/
            isDiscountClear = false
        }

        override fun onStartTrackingTouch(seekBar: SeekBar?) {
        }

        override fun onStopTrackingTouch(seekBar: SeekBar?) {
        }
    }

    // Setup TextWatcher for Discount
    binding.textDiscount.addTextChangedListener(object: TextWatcher {
        override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {
        }

        override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
        }

        override fun afterTextChanged(s: Editable?) {
            // Validate the discount input
            val discountInput = s.toString().trim()
            // If in Fixed Amount Mode
            if (binding.radioFixedAmount.isChecked) {
                if (discountInput.toIntOrNull() == null) {
                    binding.textDiscount.error = getString(R.string.fixed_amount_error1)
                }
                else if (discountInput.toInt() <= 0) {
                    binding.textDiscount.error = getString(R.string.fixed_amount_error2)
                }
                else if ((isSalesPriceValid) && (discountInput.toDouble()
                            > binding.textSalesPrice.text.toString().toDoubleOrNull()!!)) {
                    binding.textDiscount.error = getString(R.string.fixed_amount_error3)
                }
                else {
                    binding.textDiscount.error = null
                }
            }
            // If in Percent Off Mode
```

```
            else {
                if (discountInput.toIntOrNull() == null) {
                    binding.textDiscount.error = getString(R.string.percent_off_error1)
                }
                else if (discountInput.toInt() <= 0) {
                    binding.textDiscount.error = getString(R.string.percent_off_error2)
                }
                else {
                    binding.textDiscount.error = null
                }
            }
            // If no error = validate
            isDiscountValid = binding.textDiscount.error == null
            // Update the state of the calculate button
            updateCalculateButtonState()
        }
    })

    // Setup SeekBarChange Listener for Tax
    val taxSliderListener = object : SeekBar.OnSeekBarChangeListener {
        override fun onProgressChanged(seekBar: SeekBar?, progress: Int, fromUser: Boolean) {
            // Update the EditText with the Slider progress
            if (!isTaxClear) {
                binding.textTax.setText(progress.toString())
                binding.textTax.setSelection(binding.textTax.text.length)
            }
            /* Use isTaxClear to fix the bug of syncing SeekerBar and EditTax so that
            * it will not appear 1 when user clear input as it is the min value of SeekerBar*/
            isTaxClear = false
        }

        override fun onStartTrackingTouch(seekBar: SeekBar?) {
        }

        override fun onStopTrackingTouch(seekBar: SeekBar?) {
        }
    }

    // Setup TextWatcher for Tax
    val taxTextWatcher = object: TextWatcher {
        override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {
        }

        override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
        }

        override fun afterTextChanged(s: Editable?) {
                // Validate the tax input
                val taxInput = s.toString().trim()
                if (taxInput.toIntOrNull() == null) {
                    binding.textTax.error = getString(R.string.tax_error1)
                }
                else if (taxInput.toInt() <= 0) {
                    binding.textTax.error = getString(R.string.tax_error2)
                }
                else {
                    binding.textTax.error = null
                }
                // If no error = validate
                isTaxValid = binding.textTax.error == null
                // Update the state of the calculate button
                updateCalculateButtonState()
        }
    }

    // Add the taxTextWatcher for the first time
    binding.textTax.addTextChangedListener(taxTextWatcher)

    // Set up an OnCheckedChangeListener for the Radio Button
    binding.groupMode.setOnCheckedChangeListener { _, checkedId ->
        // If is in Percent Off mode
        if (checkedId == R.id.radio_percent_off) {
            // Set up the range for Percent Off Discount input
            binding.textDiscount.filters = arrayOf(RangeInputFilter(0, 100))
            if ((binding.textDiscount.text.isNotEmpty())
                && binding.textDiscount.text.toString().toIntOrNull()!! > 100) {
                // If user entered > 100 in fixed amount and switch to percent off, set to 100
                binding.textDiscount.setText("100")
                binding.textDiscount.setSelection(binding.textDiscount.length())
            }
            if (binding.textDiscount.error != null) {
```

```
                // Clear or change the error in case user switch mode when having error
                if (binding.textDiscount.error.toString()
                    == getString(R.string.fixed_amount_error1)) {
                        binding.textDiscount.error = getString(R.string.percent_off_error1)
                    }
                    else if (binding.textDiscount.error.toString()
                    == getString(R.string.fixed_amount_error2)) {
                        binding.textDiscount.error = getString(R.string.percent_off_error2)
                    }
                    else {
                        binding.textDiscount.error = null
                    }
                }

            if (binding.textDiscount.text.isNotEmpty() && binding.textDiscount.error == null) {
                // If no error = validate
                isDiscountValid = true
                // Update the state of the calculate button
                updateCalculateButtonState()
            }

            binding.textDiscountSymbol.text = "%"
            binding.sliderDiscountPercent.isVisible = true
            binding.buttonGroup.isVisible = false

            // Add the SeekBarChange Listener for discount input when percent off is checked
            binding.sliderDiscountPercent.setOnSeekBarChangeListener(discountSliderListener)

        } else { // If is in Fixed Amount mode
            // Set up the range for Fixed Amount Discount input
            binding.textDiscount.filters = arrayOf(RangeInputFilter(0, 10000000))
            binding.textDiscountSymbol.text = "$"
            binding.sliderDiscountPercent.isVisible = false
            binding.buttonGroup.isVisible = true
            // Remove SeekBarChange Listener when percent off is unchecked
            binding.sliderDiscountPercent.setOnSeekBarChangeListener(null)
            // Clear or change the error in case user switch mode when having error
            if (binding.textDiscount.error != null) {
                if (binding.textDiscount.error.toString()
                    == getString(R.string.percent_off_error1)) {
                    binding.textDiscount.error = getString(R.string.fixed_amount_error1)
                }
                else if (binding.textDiscount.error.toString()
                    == getString(R.string.percent_off_error2)) {
                    binding.textDiscount.error = getString(R.string.fixed_amount_error2)
                }
                else {
                    binding.textDiscount.error = null
                }
            }
            validateFixedAmountDiscount()
        }
    }

    // Add the discount SeekBarChange Listener for the first time
    binding.sliderDiscountPercent.setOnSeekBarChangeListener(discountSliderListener)
    // Add the tax SeekBarChange Listener for the first time
    binding.sliderTaxPercent.setOnSeekBarChangeListener(taxSliderListener)
    // Set up an OnCheckedChangeListener for the Tax Switch
    binding.switchTax.setOnCheckedChangeListener { _, isChecked ->
        if (isChecked) {
            // If the switch is checked (with tax), enable the tax views
            binding.textTax.isEnabled = true
            binding.sliderTaxPercent.isEnabled = true
            // To clear the "0" after unchecking tax to show hint
            binding.textTax.text.clear()
            // Set the tax invalid back in case user did not input and switch from unchecked to checked
            isTaxValid = false
            // Update the state of the calculate button
            updateCalculateButtonState()
            // Add the TextWatcher and SeekBarChange Listener for tax input when tax is enabled
            binding.textTax.addTextChangedListener(taxTextWatcher)
            binding.sliderTaxPercent.setOnSeekBarChangeListener(taxSliderListener)

        } else {
            // If the switch is unchecked (without tax), disable the slider and edit text
            binding.textTax.isEnabled = false
            binding.sliderTaxPercent.isEnabled = false

            // Remove the TextWatcher and SeekBarChange Listener when tax is disabled
            binding.textTax.removeTextChangedListener(taxTextWatcher)
```

```
            binding.sliderTaxPercent.setOnSeekBarChangeListener(null)
            // Clear the tax input
            binding.textTax.setText("0")
            // Set the error to null
            binding.textTax.error = null
            // Set the Tax to valid since it is the value we set
            isTaxValid = true

            // Update the state of the calculate button
            updateCalculateButtonState()
        }
    }

    // Set up an OnClickedListener for the FixedAmount Button
    //implement 7 buttons clicking
    binding.buttonMinusTen.setOnClickListener(this)
    binding.buttonMinusFive.setOnClickListener(this)
    binding.buttonMinusOne.setOnClickListener(this)
    binding.buttonPlusOne.setOnClickListener(this)
    binding.buttonPlusFive.setOnClickListener(this)
    binding.buttonPlusTen.setOnClickListener(this)


    binding.textDiscount.addTextChangedListener(object : TextWatcher {
        override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {
        }

        override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
        }

        override fun afterTextChanged(s: Editable?) {
            val inputText = s.toString()
            isDiscountClear = inputText.isBlank()
                // Try to parse the value as an integer
                val intValue = inputText.toIntOrNull() ?: 0
                // Clamp the value between 0 and 100
                val clampedValue = intValue.coerceIn(1, 100)
                // Update the Slider progress
            binding.sliderDiscountPercent.progress = clampedValue
        }
    })

    binding.textTax.addTextChangedListener(object : TextWatcher {
        override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {
        }

        override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
        }

        override fun afterTextChanged(s: Editable?) {
            val inputText = s.toString()
            isTaxClear = inputText.isBlank()
                // Try to parse the value as an integer
                val intValue = inputText.toIntOrNull() ?: 0
                // Clamp the value between 1 and 100
                val clampedValue = intValue.coerceIn(1, 100)
                // Update the Slider progress
                binding.sliderTaxPercent.progress = clampedValue
        }
    })

    binding.textSalesPrice.addTextChangedListener(object : TextWatcher {
        override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int, after: Int) {
        }

        override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count: Int) {
        }

        override fun afterTextChanged(s: Editable?) {
            // Validate the price input
                val priceInput = s.toString().trim()
                if (priceInput.toDoubleOrNull() == null) {
                    binding.textSalesPrice.error = getString(R.string.sales_price_error1)
                }
                else if (priceInput.toDouble() <= 0)
                {
                    binding.textSalesPrice.error = getString(R.string.sales_price_error2)
                }
                else {
                    binding.textSalesPrice.error = null
                }
```

```
                    // If no error = validate
                    isSalesPriceValid = binding.textSalesPrice.error == null
                    // Update the state of the calculate button
                    updateCalculateButtonState()
                    if ((isSalesPriceValid) && (binding.textDiscount.text.isNotEmpty()) &&
                        (binding.radioFixedAmount.isChecked) &&
                        (priceInput.toDouble() < binding.textDiscount.text.toString().toDouble())) {
                        binding.textDiscount.error = getString(R.string.fixed_amount_error3)
                    }
                    else {
                        if (binding.textDiscount.error == getString(R.string.fixed_amount_error3)) {
                            binding.textDiscount.error = null
                            // If no error = validate
                            isDiscountValid = binding.textDiscount.error == null
                        }
                    }
                    // If no error = validate
                    isSalesPriceValid = binding.textSalesPrice.error == null
                    // Update the state of the calculate button
                    updateCalculateButtonState()
                }
            })

            // Setup OnClickListener for calculate button
            binding.buttonCalculate.setOnClickListener {
                val salesPrice = binding.textSalesPrice.text.toString().toDoubleOrNull() ?: 0.0
                val discount = binding.textDiscount.text.toString().toIntOrNull() ?: 0
                val tax = binding.textTax.text.toString().toIntOrNull() ?: 0

                val discountData = DiscountData(
                    salesPrice = salesPrice,
                    discount = discount,
                    tax = tax
                )
                // If in Percent Off Mode
                if (binding.radioPercentOff.isChecked) {
                    Calculation.calculatePercentOffDiscountedPrice(discountData)
                }
                // If in FixedAmount Mode
                else {
                    Calculation.calculateFixedAmountDiscountedPrice(discountData)
                }
                // Calculate Tax Amount and Total Amount
                Calculation.calculateTaxAndTotalAmount(discountData)

                // Display the output in 2 decimal places with money symbol
                val saveAmount =  String.format("%.2f $", discountData.saveAmount)
                val taxAmount = String.format("%.2f $", discountData.taxAmount)
                val totalAmount = String.format("%.2f $", discountData.totalAmount)
                binding.textSaveAmount.text = saveAmount
                binding.textTaxAmount.text = taxAmount
                binding.textTotalAmount.text = totalAmount
            }
        }

        private fun updateCalculateButtonState() {
            // Set the button to be enabled only when the input is valid
            binding.buttonCalculate.isEnabled = isSalesPriceValid && isDiscountValid && isTaxValid
        }

        private fun validateFixedAmountDiscount() {
            if ((isSalesPriceValid) && (binding.textDiscount.text.isNotEmpty())
                && ((binding.textDiscount.text.toString().toDoubleOrNull() ?: 0.0)
                        > (binding.textSalesPrice.text.toString().toDoubleOrNull() ?: 0.0))
            ) {
                binding.textDiscount.error = getString(R.string.fixed_amount_error3)
                // If no error = validate
                isDiscountValid = binding.textDiscount.error == null
            }
            // Update the state of the calculate button
            updateCalculateButtonState()
        }

        override fun onCreateOptionsMenu(menu: Menu?): Boolean {
            //create the menu bar
            menuInflater.inflate(R.menu.menu,menu)
            return super.onCreateOptionsMenu(menu)
        }

        // Toolbar item
        override fun onOptionsItemSelected(item: MenuItem): Boolean {
```

```
        when (item.itemId) {
            R.id.menu_clear -> {
                clearAll()
            }

            R.id.menu_about -> {
                showDialog()
            }
            else -> {
            }
        }
        return super.onOptionsItemSelected(item)
    }

    companion object {
        const val SAVED_AMOUNT = "save_amount"
        const val TAX_AMOUNT = "tax_amount"
        const val TOTAL_PRICE = "total_amount"
    }

    // Clear all the input and output
    private fun clearAll() {
        binding.textSalesPrice.setText("")
        binding.textDiscount.setText("")
        binding.textSaveAmount.text = "0.00 $"
        binding.textTaxAmount.text = "0.00 $"
        binding.textTotalAmount.text = "0.00 $"
        // If with tax is unchecked, won't clear the "0"
        if(binding.switchTax.isChecked) {
            binding.textTax.setText("")
        }
    }

    // Save the discount information
    override fun onPause() {
        super.onPause()

        viewModel.saveAmount = binding.textSaveAmount.text.toString()
        viewModel.taxAmount = binding.textTaxAmount.text.toString()
        viewModel.totalAmount = binding.textTotalAmount.text.toString()
    }

    // Retrieve the discount information
    override fun onResume() {
        super.onResume()
        // Validate the tax input
        if (binding.textTax.text.isNotEmpty()) {
            val taxInput = binding.textTax.text.toString().trim()
            if (taxInput.toIntOrNull() == null || taxInput.toInt() <= 0) {
                binding.textTax.error = getString(R.string.tax_error1)
            } else {
                binding.textTax.error = null
            }
        }
        //recover the data for any configuration changes
        binding.textSaveAmount.text = viewModel.saveAmount
        binding.textTaxAmount.text = viewModel.taxAmount
        binding.textTotalAmount.text = viewModel.totalAmount
    }

    // Show App Information
    private fun showDialog() {
        val dialog = AboutDialog()
        dialog.show(supportFragmentManager, "123")
    }

    // Fixed Amount Button
    override fun onClick(v: View?) {
        val currentValue = binding.textDiscount.text.toString().toIntOrNull() ?: 0
        var newValue: Int
        val maxValue = 10000000
        val isMaxValue: Boolean = currentValue >= maxValue

        when (v?.id) {
            R.id.button_minus_ten -> {
                // Handle the -10 button click
                newValue = (currentValue - 10).coerceAtLeast(0)
                binding.textDiscount.setText(newValue.toString())
            }

            R.id.button_minus_five -> {
```

```
                // Handle the -5 button click
                newValue = (currentValue - 5).coerceAtLeast(0)
                binding.textDiscount.setText(newValue.toString())
            }

            R.id.button_minus_one -> {
                // Handle the -1 button click
                newValue = (currentValue - 1).coerceAtLeast(0)
                binding.textDiscount.setText(newValue.toString())
            }

            R.id.button_plus_one -> {
                // Handle the +1 button click
                if (!isMaxValue) {
                    newValue = currentValue + 1
                    if (newValue >= maxValue)
                    {
                        binding.textDiscount.setText(maxValue.toString())
                    }
                    else {
                        binding.textDiscount.setText(newValue.toString())
                    }
                }
            }

            R.id.button_plus_five -> {
                // Handle the +5 button click
                if (!isMaxValue) {
                    newValue = currentValue + 5
                    if (newValue >= maxValue)
                    {
                        binding.textDiscount.setText(maxValue.toString())
                    }
                    else {
                        binding.textDiscount.setText(newValue.toString())
                    }
                }
            }

            R.id.button_plus_ten -> {
                // Handle the +10 button click
                if(!isMaxValue) {
                    newValue = currentValue + 10
                    if (newValue >= maxValue)
                    {
                        binding.textDiscount.setText(maxValue.toString())
                    }
                    else {
                        binding.textDiscount.setText(newValue.toString())
                    }
                }
            }
        }
        validateFixedAmountDiscount()
        binding.textDiscount.setSelection(binding.textDiscount.text.length)
    }
}
```

6.9 AboutDialog.kt

```
package com.example.simplediscountcalculator

import android.app.Dialog
import android.os.Bundle
import androidx.appcompat.app.AlertDialog
import androidx.fragment.app.DialogFragment

class AboutDialog:DialogFragment() {
    override fun onCreateDialog(savedInstanceState: Bundle?): Dialog {
        val dialog = AlertDialog.Builder(requireContext())
            .setMessage(getString(R.string.about_message))
            .setTitle(getString(R.string.about_title))
            .setPositiveButton("OK") {_,_->}
```

```
            .setNegativeButton("Cancel") {_,_->}
            .create()
        return dialog
    }

}
```

6.10 DiscountData.kt

```
package com.example.simplediscountcalculator

data class DiscountData(
    var salesPrice:Double = 0.0,
    var discount:Int = 0,
    var tax:Int = 0,
    var totalAmount:Double = 0.0,
    var saveAmount:Double = 0.0,
    var taxAmount:Double = 0.0,
)
```

6.11 Calculation.kt

```
package com.example.simplediscountcalculator

object Calculation {
    fun calculatePercentOffDiscountedPrice(data: DiscountData) {
        data.saveAmount = data.salesPrice * (data.discount / 100.0)
    }
    fun calculateFixedAmountDiscountedPrice(data: DiscountData) {
        data.saveAmount = data.discount.toDouble()
    }
    fun calculateTaxAndTotalAmount (data: DiscountData) {
        data.taxAmount = data.salesPrice * (data.tax / 100.0)
        data.totalAmount = data.salesPrice - data.saveAmount + data.taxAmount
    }
}
```

6.12 DiscountModel.kt

```
package com.example.simplediscountcalculator

import androidx.lifecycle.ViewModel

class DiscountModel: ViewModel() {
    var saveAmount = "0.00 $"
    var taxAmount = "0.00 $"
    var totalAmount = "0.00 $"
}
```

6.13 RangeInputFilters.kt

```
package com.example.simplediscountcalculator

import android.text.InputFilter
import android.text.Spanned
```

```
class RangeInputFilter(private val minValue: Int, private val maxValue: Int) :
InputFilter {
    override fun filter(
        source: CharSequence?,
        start: Int,
        end: Int,
        dest: Spanned?,
        dstart: Int,
        dend: Int
    ): CharSequence? {
        try {
            val input = (dest.toString() + source.toString()).toInt()
            if (isInRange(input, minValue, maxValue)) {
                return null  // Input is between 0 and 100, then accept
            }
        } catch (nfe: NumberFormatException) {
            // Input is not a valid integer
        }
        return ""  // Input is not between 0 and 100, then reject
    }

    private fun isInRange(value: Int, minValue: Int, maxValue: Int): Boolean {
        return value in minValue..maxValue
    }
}

class DoubleRangeInputFilter(private val minValue: Double, private val maxValue:
Double) : InputFilter {
    override fun filter(
        source: CharSequence?,
        start: Int,
        end: Int,
        dest: Spanned?,
        dstart: Int,
        dend: Int
    ): CharSequence? {
        try {
            val input = (dest.toString() + source.toString()).toDouble()
            if (isInRange(input, minValue, maxValue)) {
                return null  // Input is between 0.0 and 10000000.0, then accept
            }
        } catch (nfe: NumberFormatException) {
            // Input is not a valid double
        }
        return ""  // Input is not between 0.0 and 10000000.0, then reject
    }

    private fun isInRange(value: Double, minValue: Double, maxValue: Double): Boolean
{
        return value in minValue..maxValue
    }
}
```

6.14 Build.gradle.kts

```
plugins {
    id("com.android.application")
    id("org.jetbrains.kotlin.android")
}
```

```
android {
    namespace = "com.example.simplediscountcalculator"
    compileSdk = 34

    defaultConfig {
        applicationId = "com.example.simplediscountcalculator"
        minSdk = 28
        targetSdk = 33
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = "1.8"
    }
    buildFeatures {
        viewBinding = true
    }
}

dependencies {

    implementation("androidx.core:core-ktx:1.9.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.10.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.cardview:cardview:1.0.0")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
}
```

CET3013, Mobile Application Development
Assignment [1]
September 2023 semester

CET3013/MOBILE APPLICATION DEVELOPMENT

MARKING RUBRIC – ASSIGNMENT 1

Building App with Basic Android Views and View Model

Section 1 (20%)

LEARNING OUTCOME	MARKING CRITERIA	SCALE					MARKS/COMMENTS		
		Fail (0-49)	3rd Class (50-59)	2nd Lower Class (60-69)	2nd Upper Class (70-79)	1st Class (80-100)	100%	Weightage	Actual Marks
CLO 1	**Report (20%)**	A poor piece of documentation has been produced. The coverage is unclear, and/or there are significant omissions	A fair piece of documentation has been produced, although there are some weaknesses—either the coverage is not particularly clear, or some aspects have been omitted	A fair piece of documentation of the work undertaken but not in outstanding ways.	A good piece of design documentation has been produced, providing detailed and clear coverage of the aspect concerned, although there may be a number of minor flaws which prevent it being regarded as excellent	An excellent piece of documentation has been produced, providing full and clear coverage of the aspect concerned		0.2	
						TOTAL (20%)			

CET3013, Mobile Application Development
Assignment [1]
September 2023 semester

Section 2 (80%)									
LEARNING OUTCOME	MARKING CRITERIA	SCALE					MARKS/COMMENTS		
		Fail (0-49)	3rd Class (50-59)	2nd Lower Class (60-69)	2nd Upper Class (70-79)	1st Class (80-100)	100%	Weightage	Actual Marks
CLO 4	**Fitness of Purpose (50%)**	Little or no attempt to implement the feature correctly	A partial implementation of the feature, but some aspects are incorrect and not particularly well coded. May give rise to run-time errors.	A complete implementation of the features with some flaws and not in exceptional way.	A mostly complete implementation of the feature which works correctly but with minor flaws	An excellent implementation of all the feature and clearly coded		0.5	
	Build Quality (30%)	Poor build quality provided according to the Android and general coding standard	Partial implementation and some aspects did not conform with general coding standard	Fair build quality provided according to the Android and general coding standard	Good build quality provided according to the Android and general coding standard	Excellent build quality provided according to the Android and coding standard		0.3	
						Total (80%)			
						Overall Score (100%)			