



DDWC 3723 SOFTWARE DEVELOPMENT

SOFTWARE DESIGN DOCUMENT (SDD) REPORT

CALTACTOE APP DEVELOPMENT

NAME	ASHIF ABDUL MALIK BIN KAMARUDIN
I/C NUMBER	010116-03-0033
PROGRAMME	DIPLOMA IN COMPUTER SCIENCE (DDWD)
SEMESTER	V

TABLE OF CONTENT

1.0 Overview.....	1
2.0 Context.....	1
3.0 Goals.....	2
4.0 Architecture and Components.....	4
5.0 User Interfaces (UI) with XML Codes	6
5.1 activity_main.xml.....	6
5.2 activity_add_players.xml.....	14
5.3 activity_tic_tac_toe.xml.....	16
5.4 activity_result_dialog.xml.....	21
6.0 Coding (Java Codes)	23
6.1 MainActivity.java.....	23
6.2 AddPlayers.java.....	29
6.3 TicTacToe.java.....	30
6.4 ResultDialog.java.....	34
7.0 Testing Strategy	35
8.0 Deployment and Rollout.....	37
9.0 Security Considerations.....	37

1.0 Overview

CalTacToe (calculator + tic tac toe) is a simple calculator app that perform regular arithmetic operations like addition, subtraction, division and multiplication with additional operation which is percentage. That's not all, CalTacToe also embedded with Tic Tc Toe for fun factor. A two-players strategic games with one player marked one of a 3x3 grid with noughts (O's) while other marking with crosses (X's) every turn. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.

A software design document (SDD) is a crucial artifact that outlines how we plan to solve a problem throughout our software solution / development. It serves as a blueprint for our project and helps ensure that the right work gets done.

2.0 Context

The app development or creation is necessary because, while we know that most phones come with built-in calculator apps, creating our own calculator app can be a valuable learning experience. First of all, building a calculator app allows us to practice fundamental programming concepts especially mobile programming. We will gain hands-on experience with user interfaces (UI), event handling, and logic implementation. It is also a chance for us to reinforce our understanding of data types, operators, and control structures when developing this app.

Next, we can customize this calculator to our (and our client) preferences. We can design it exactly the way we like. We can design it with a specific color scheme / theme (mix between lavender and orange color is our choice for soothing aura) and choose our preferred layout (minimalist approach). Plus, we can add extra features beyond basic arithmetic if we are feeling adventurous. That leads to our third points:

Creativity! We add Tic Tac Toe game to our calculator so that if user is bored after using calculator non-stop (doing assignment that requires mass calculation or doing mathematics homework or any calculating stuffs), they can open tic tac toe using the same app for a little break fun game.

3.0 Goals

- **Calculator Goals:**

- **Functionality:**

- Ensure that the calculator performs basic arithmetic operations (addition, subtraction, multiplication, and division) accurately.

- **User Interface (UI):**

- Create an intuitive and user-friendly calculator interface with clear buttons and a display for showing calculation results.

- **Error Handling:**

- Handle edge cases (like division by zero) with no sweat.

- **Tic-Tac-Toe Goals:**

- **Game Logic:**

- Implement the rules of tic-tac-toe (3x3 grid, X and O players, winning conditions).

- **User Experience (UX):**

- Design a visually appealing game board with X and O symbols.

- **Single Player vs. Multiplayer:**

- Allow two players to play.

- **Win/Lose Conditions:**

- Notify players when someone wins, loses, or the game ends in a draw.

- **Restart/Reset Feature:**

- Allow players to start a new game without reopening the app.

- **Integration Goals:**

- **Switching Between Modes:**

- Enable users to seamlessly switch between calculator and tic-tac-toe modes.

- **Shared UI Elements:**

- UI elements (buttons and layout) can serve both functionalities without confusion.

- **Testing and Quality Assurance Goals:**
 - **Unit Testing:**

Each component (calculator and game) being test independently (passed during Software Test).
 - **Integration Testing:**

The combined app works as expected.
 - **User Testing:**

Feedback from potential users gathered to improve the app.
- **Personalization Goals:**
 - **Custom Icons:**

Icon reflects both calculator and tic-tac-toe aspects.

4.0 Architecture and Components

In this section, we will detail up our app's high-level architecture and break down the system into components to explain their responsibilities.

- Overall System Architecture:

Our app will run on an Android platform. It consists of two main components: The Calculator and the Tic-Tac-Toe Game.

- Calculator Component:

Responsibilities:

- Handles arithmetic calculations (addition, subtraction, multiplication, division) plus percentage.
- Manages user input when button pressed.
- Displays the result on the clear and wide screen.

User Interface (UI):

- Buttons for digits (0-9), operators (+, -, *, /), and special functions (clear, equals and percentage (%)).
- Display area to show the current calculation.

Logic Flow:

- User presses buttons to input numbers and operators.
- The calculator processes the input and updates the display.
- When the user presses "equals," the result is shown.

- Tic-Tac-Toe Game Component:

Responsibilities:

- Implements the rules of tic-tac-toe (3x3 grid, X and O players).
- Manages user interactions (taps on the grid).
- Determines game outcomes (win, lose, draw).

User Interface (UI):

- 3x3 grid for the game board.
- X and O symbols displayed in the grid but with a funny twist (X icon (formerly known as Twitter) as cross and red doughnut as circle).
- Restart when a round of game is finished to start another new game.

Logic Flow:

- Users take turns tapping empty cells on the grid.
- The game checks for winning conditions (3 in a row horizontally, vertically, or diagonally).
- If a player wins or the grid is full (draw), the game ends and automatically start a new game so a restart button is not need.

- Integration Points:

- Switching Between Modes:

- Users can switch between the calculator and tic-tac-toe modes.
 - The UI updates to show the relevant components.

- Shared UI Elements:

Buttons and layout for both modes (calculator and tic tac toe) is distinct from each other to avoid confusion among users.

- Error Handling:

- In the calculator:

- Handle division by zero.
 - Display error messages for invalid input.

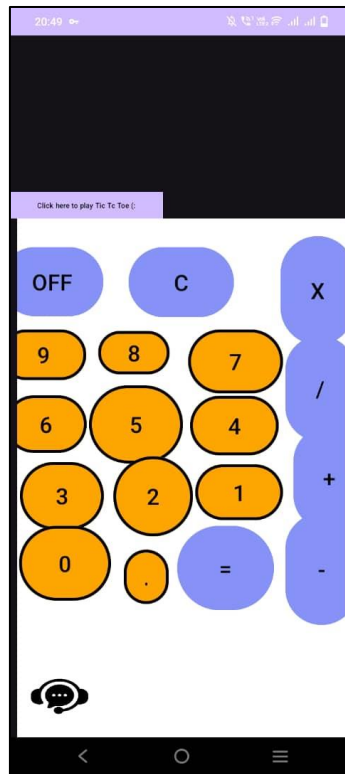
- In tic-tac-toe:

- Prevent users from making invalid moves (like selecting an already occupied cell).

5.0 User Interfaces (UI) with XML Codes

5.1 activity_main.xml

Below is the user interface of main activity which is the calculator. The user will be greeted with calculator firstly first when they opened the app.



XML (eXtensible Markup Language) is a markup language similar to HTML, used to describe data. In Android Studio, XML is commonly used for implementing user interface (UI) related data like what layout to use (linear or constraint layout or both) and what widgets to hold (like buttons, containers, texts and more).

To come out a UI like above (main activity), simply create a layout file using Android Studio's Layout Editor, copy the code below and paste it under that file.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="40sp"
        android:textColor="@color/Lavender"
        android:padding="10dp"
        android:layout_gravity="end"
        android:id="@+id/input" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="60sp"
        android:textColor="@color/Lavender"
        android:padding="10dp"
        android:layout_gravity="end"
        android:id="@+id/output" />

    <Button
        android:id="@+id/btnGame"
        android:layout_width="171dp"
        android:layout_height="30dp"
        android:background="#E91E63"
        android:backgroundTint="#E91E63"
        android:drawableTint="#E91E63"
        android:foregroundTint="#E91E63"
        android:text="Click here to play Tic Tc Toe (:)"
        android:textColor="#0C0C0C"
        android:textSize="8sp" />

    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/relativeLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:background="@color/white"
        android:gravity="bottom">

        <include
            android:id="@+id/include"
            layout="@layout/buttons_layout"
            android:layout_width="420dp"
            android:layout_height="549dp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent" />

    </LinearLayout>
```

```

        app:layout_constraintTop_toTopOf="parent" />

        <ImageButton
            android:id="@+id/btnHelp"
            android:layout_width="66dp"
            android:layout_height="49dp"
            android:background="@drawable/contact"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.048"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.96" />

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

```

This is the ‘sub-file’ or child to the above file, specially created to make organized calculator buttons.

buttons_layout.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:id="@+id/relativeLayout2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:layout_editor_absoluteY="25dp">

    <com.google.android.material.button.MaterialButton
        android:id="@+id/off"
        android:layout_width="115dp"
        android:layout_height="85dp"
        android:backgroundTint="@color/Lavender"
        android:text="OFF"
        android:textColor="@color/black"
        android:textSize="25sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.023"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.027" />

    <com.google.android.material.button.MaterialButton
        android:id="@+id/clear"
        android:layout_width="118dp"
        android:layout_height="86dp"
        android:backgroundTint="@color/Lavender"
        android:text="C"
        android:textColor="@color/black"

```

```

        android:textSize="25sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="@+id/multiply"
        app:layout_constraintHorizontal_bias="0.527"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.027" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/percent"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:backgroundTint="@color/Lavender"
    android:text="%"
    android:textColor="@color/black"
    android:textSize="25sp"
    tools:layout_editor_absoluteX="3dp"
    tools:layout_editor_absoluteY="2dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/division"
    android:layout_width="78dp"
    android:layout_height="126dp"
    android:backgroundTint="@color/Lavender"
    android:text="/"
    android:textColor="@color/black"
    android:textSize="25sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.951"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.264" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn7"
    android:layout_width="106dp"
    android:layout_height="79dp"
    android:backgroundTint="@color/Orange"
    android:text="7"
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.691"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.222"
    app:strokeColor="@color/black"
    app:strokeWidth="3dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn8"
    android:layout_width="80dp"
    android:layout_height="56dp"
    android:layout_marginBottom="1dp"
    android:backgroundTint="@color/Orange"
    android:text="8"
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toTopOf="@+id/subtract"

```

```

        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.341"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.419"
        app:strokeColor="@color/black"
        app:strokeWidth="3dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn9"
    android:layout_width="96dp"
    android:layout_height="64dp"
    android:backgroundTint="@color/Orange"
    android:text="9"
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.022"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.216"
    app:strokeColor="@color/black"
    app:strokeWidth="3dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/multiply"
    android:layout_width="83dp"
    android:layout_height="129dp"
    android:backgroundTint="@color/Lavender"
    android:text="X"
    android:textColor="@color/black"
    android:textSize="25sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.951"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn4"
    android:layout_width="99dp"
    android:layout_height="74dp"
    android:backgroundTint="@color/Orange"
    android:text="4"
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.682"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.376"
    app:strokeColor="@color/black"
    app:strokeWidth="3dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn5"
    android:layout_width="105dp"
    android:layout_height="95dp"

```

```

        android:backgroundTint="@color/Orange"
        android:text="5"
        android:textColor="@color/black"
        android:textSize="25dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.336"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.368"
        app:strokeColor="@color/black"
        app:strokeWidth="3dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn6"
    android:layout_width="91dp"
    android:layout_height="72dp"
    android:backgroundTint="@color/Orange"
    android:text="6"
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.037"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.374"
    app:strokeColor="@color/black"
    app:strokeWidth="3dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/substract"
    android:layout_width="81dp"
    android:layout_height="133dp"
    android:backgroundTint="@color/Lavender"
    android:text="-"
    android:textColor="@color/black"
    android:textSize="25sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.96"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.745" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn1"
    android:layout_width="98dp"
    android:layout_height="70dp"
    android:backgroundTint="@color/Orange"
    android:text="1"
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.699"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.533"
    app:strokeColor="@color/black"
    app:strokeWidth="3dp" />

```

```

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn2"
    android:layout_width="89dp"
    android:layout_height="97dp"
    android:backgroundTint="@color/Orange"
    android:text="2"
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.403"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.546"
    app:strokeColor="@color/black"
    app:strokeWidth="3dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn3"
    android:layout_width="94dp"
    android:layout_height="83dp"
    android:backgroundTint="@color/Orange"
    android:text="3"
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.088"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.543"
    app:strokeColor="@color/black"
    app:strokeWidth="3dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/add"
    android:layout_width="80dp"
    android:layout_height="125dp"
    android:backgroundTint="@color/Lavender"
    android:text="+"
    android:textColor="@color/black"
    android:textSize="25sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.984"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn0"
    android:layout_width="103dp"
    android:layout_height="91dp"
    android:backgroundTint="@color/Orange"
    android:text="0"
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.087"
    app:layout_constraintStart_toStartOf="parent"

```

```

        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.709"
        app:strokeColor="@color/black"
        app:strokeWidth="3dp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/btnPoint"
    android:layout_width="50dp"
    android:layout_height="68dp"
    android:backgroundTint="@color/Orange"
    android:text="."
    android:textColor="@color/black"
    android:textSize="25dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.392"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.73"
    app:strokeColor="@color/black"
    app:strokeWidth="3dp" />

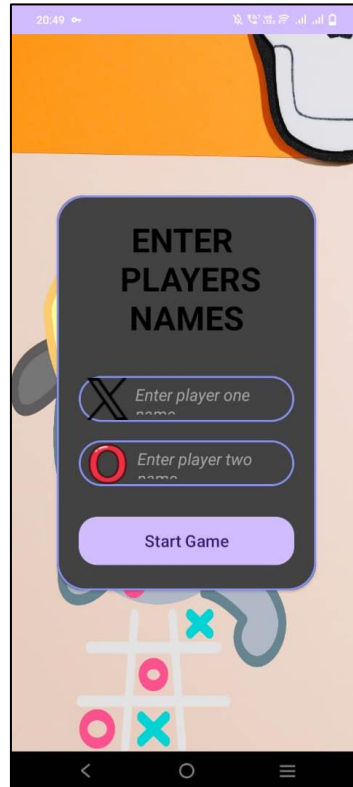
<com.google.android.material.button.MaterialButton
    android:id="@+id/equal"
    android:layout_width="108dp"
    android:layout_height="102dp"
    android:backgroundTint="@color/Lavender"
    android:text="="
    android:textColor="@color/black"
    android:textSize="25sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.656"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.725" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

5.2 activity_add_players.xml

This is a UI that firstly prompted when user changing mode from calculator to tic tac toe game. It will ask user to enter player 1 and 2's name.



The code to produce layout or UI such above is as below:

activity_add_players.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/catto"
    tools:context=".AddPlayers">

    <android.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="50dp"
        app:cardCornerRadius="30dp"
        android:elevation="20dp" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
```



```

        android:orientation="vertical"
        android:gravity="center_horizontal"
        android:padding="24dp"
        android:background="@drawable/lavender_border">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="ENTER \n PLAYERS NAMES"
            android:textAlignment="center"
            android:textSize="36sp"
            android:textStyle="bold"
            android:textColor="@color/black" />

        <EditText
            android:layout_width="match_parent"
            android:layout_height="50dp"
            android:id="@+id/playerOne"
            android:background="@drawable/lavender_border"
            android:layout_marginTop="40dp"
            android:padding="8dp"
            android:maxLines="1"
            android:textStyle="italic"
            android:hint="Enter player one name"
            android:textColor="@color/Lavender"
            android:drawableLeft="@drawable/elonmusk"
            android:drawablePadding="8dp" />

        <EditText
            android:layout_width="match_parent"
            android:layout_height="50dp"
            android:id="@+id/playerTwo"
            android:background="@drawable/lavender_border"
            android:layout_marginTop="20dp"
            android:padding="8dp"
            android:maxLines="1"
            android:textStyle="italic"
            android:hint="Enter player two name"
            android:textColor="@color/Lavender"
            android:drawableLeft="@drawable/reddonut"
            android:drawablePadding="8dp"
            />

        <Button
            android:layout_width="match_parent"
            android:layout_height="60dp"
            android:layout_marginTop="30dp"
            android:id="@+id/startGameButton"
            android:text="Start Game"
            android:textSize="18sp"
            app:cornerRadius="20dp" />

    </LinearLayout>

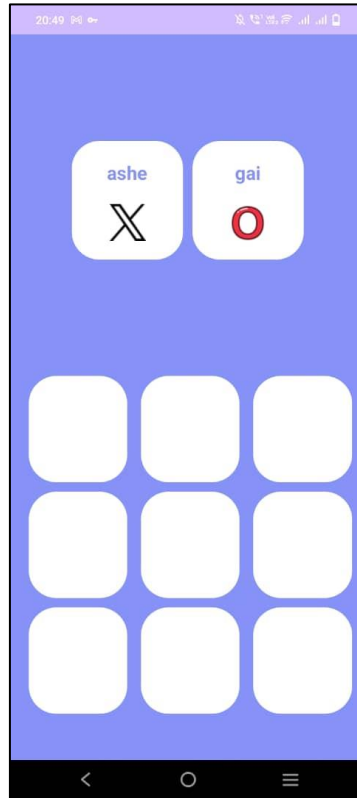
</17ndroid.cardview.widget.CardView>

</RelativeLayout>

```

5.3 activity_tic_tac_toe.xml

This UI is the board, consist of 3x3 grid for two players compete with each other.



Below, is the .xml code to get the similar UI like above:

activity_tic_tac_toe.xml

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout
    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"
    android:background="@color/Lavender"
    tools:context=".TicTacToe">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:layout_above="@id/container"
        android:layout_alignParentTop="true"
        android:orientation="horizontal">

        <LinearLayout
            android:layout_width="120dp"
            android:layout_height="wrap_content"
```

```

        android:id="@+id/playerOneLayout"
        android:orientation="vertical"
        android:gravity="center"
        android:background="@drawable/white_box">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Player One"
            android:id="@+id/playerOneName"
            android:textSize="20sp"
            android:gravity="center"
            android:textStyle="bold"
            android:textColor="@color/Lavender"
            android:maxLines="1" />

        <ImageView
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:src="@drawable/elonmusk"
            android:layout_marginTop="20dp"
            android:layout_marginBottom="20dp"
            android:layout_gravity="center" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="120dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="10dp"
        android:id="@+id/playerTwoLayout"
        android:orientation="vertical"
        android:gravity="center"
        android:background="@drawable/white_box">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Player Two"
            android:id="@+id/playerTwoName"
            android:textSize="20sp"
            android:gravity="center"
            android:textStyle="bold"
            android:textColor="@color/Lavender"
            android:maxLines="1" />

        <ImageView
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:src="@drawable/reddonut"
            android:layout_marginTop="20dp"
            android:layout_marginBottom="20dp"
            android:layout_gravity="center" />

    </LinearLayout>

</LinearLayout>

<LinearLayout

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/container"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="10dp"
        android:layout_marginEnd="10dp"
        android:layout_marginBottom="50dp"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:weightSum="3">

            <ImageView
                android:layout_width="0dp"
                android:layout_height="115dp"
                tools:ignore="Suspicious0dp"
                android:id="@+id/image1"
                android:layout_weight="1"
                android:background="@drawable/white_box"
                android:layout_marginTop="10dp"
                android:layout_marginEnd="5dp"
                android:layout_marginStart="10dp"
                android:adjustViewBounds="true"
                android:padding="20dp"/>

            <ImageView
                android:layout_width="0dp"
                android:layout_height="115dp"
                tools:ignore="Suspicious0dp"
                android:id="@+id/image2"
                android:layout_weight="1"
                android:background="@drawable/white_box"
                android:layout_marginTop="10dp"
                android:layout_marginEnd="5dp"
                android:layout_marginStart="10dp"
                android:adjustViewBounds="true"
                android:padding="20dp"/>

            <ImageView
                android:layout_width="0dp"
                android:layout_height="115dp"
                tools:ignore="Suspicious0dp"
                android:id="@+id/image3"
                android:layout_weight="1"
                android:background="@drawable/white_box"
                android:layout_marginTop="10dp"
                android:layout_marginEnd="5dp"
                android:layout_marginStart="10dp"
                android:adjustViewBounds="true"
                android:padding="20dp"/>

        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:weightSum="3">

```

```

<ImageView
    android:layout_width="0dp"
    android:layout_height="115dp"
    tools:ignore="Suspicious0dp"
    android:id="@+id/image4"
    android:layout_weight="1"
    android:background="@drawable/white_box"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="5dp"
    android:layout_marginStart="10dp"
    android:adjustViewBounds="true"
    android:padding="20dp"/>

<ImageView
    android:layout_width="0dp"
    android:layout_height="115dp"
    tools:ignore="Suspicious0dp"
    android:id="@+id/image5"
    android:layout_weight="1"
    android:background="@drawable/white_box"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="5dp"
    android:layout_marginStart="10dp"
    android:adjustViewBounds="true"
    android:padding="20dp"/>

<ImageView
    android:layout_width="0dp"
    android:layout_height="115dp"
    tools:ignore="Suspicious0dp"
    android:id="@+id/image6"
    android:layout_weight="1"
    android:background="@drawable/white_box"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="5dp"
    android:layout_marginStart="10dp"
    android:adjustViewBounds="true"
    android:padding="20dp"/>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:weightSum="3">

    <ImageView
        android:layout_width="0dp"
        android:layout_height="115dp"
        tools:ignore="Suspicious0dp"
        android:id="@+id/image7"
        android:layout_weight="1"
        android:background="@drawable/white_box"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="5dp"
        android:layout_marginStart="10dp"
        android:adjustViewBounds="true"
        android:padding="20dp"/>

```

```

<ImageView
    android:layout_width="0dp"
    android:layout_height="115dp"
    tools:ignore="Suspicious0dp"
    android:id="@+id/image8"
    android:layout_weight="1"
    android:adjustViewBounds="true"
    android:background="@drawable/white_box"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="5dp"
    android:layout_marginStart="10dp"
    android:padding="20dp"/>

<ImageView
    android:layout_width="0dp"
    android:layout_height="115dp"
    tools:ignore="Suspicious0dp"
    android:id="@+id/image9"
    android:layout_weight="1"
    android:background="@drawable/white_box"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="5dp"
    android:layout_marginStart="10dp"
    android:padding="20dp"
    android:adjustViewBounds="true"/>

</LinearLayout>

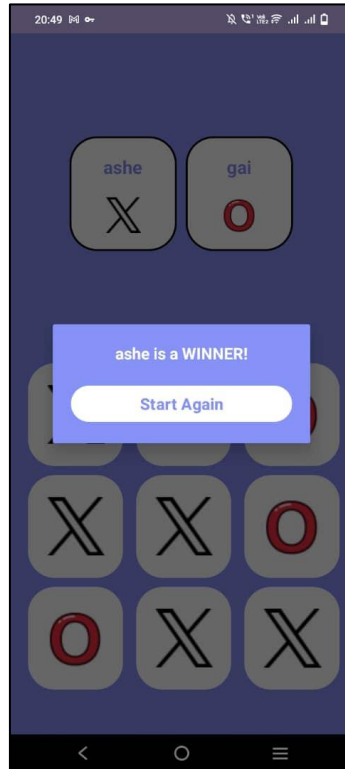
</LinearLayout>

</RelativeLayout>

```

5.4 activity_result_dialog.xml

The UI or rather a dialog box interface that will pop up when one of the players wins or the game is tie with no one is a winner.



The code to produce such UI is such below code:

activity_result_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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=".ResultDialog">

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

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/messageText"
            android:gravity="center"
            android:text="Results"
            android:textColor="@color/white"
            android:layout_marginTop="20dp"
```

```
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:textSize="18sp"
        android:textStyle="bold" />

        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/startAgainButton"
            android:textSize="18sp"
            android:textStyle="bold"
            android:backgroundTint="@color/white"
            android:text="Start Again"
            android:layout_margin="20dp"
            android:textColor="@color/Lavender"
            app:cornerRadius="20dp" />

    </LinearLayout>
</RelativeLayout>
```


6.0 Coding (Java Code)

6.1 MainActivity.java

In Android Studio, a Java file is a source code file written in the Java programming language. It contains the logic and functionality for our Android app. When we create a new Android project, Android Studio generates a default Java file (usually named MainActivity.java) that serves as the entry point for our app.

These files define how our app behaves, interacts with the user, and handles data. Our MainActivity.java is our calculator where it defines how the calculation works, define value for operand buttons, define function for special buttons (OFF and CLEAR button), intent to tic tac toe activity via an image button and so on.

MainActivity.java

```
{

    Button game;
    ImageButton tolong;

    private static final char ADDITION = '+';
    private static final char SUBTRACTION = '-';
    private static final char MULTIPLICATION = '*';
    private static final char DIVISION = '/';
    private static final char PERCENT = '%';

    private char currentSymbol;

    private double firstValue = Double.NaN;
    private double secondValue;
    private TextView inputDisplay, outputDisplay;
    private DecimalFormat decimalFormat;
    private MaterialButton button0, button1, button2, button3, button4,
button5, button6, button7, button8, button9,
        buttonDot, buttonAdd, buttonSub, buttonMultiply,
buttonDivide, buttonPercent, buttonClear, buttonOFF, buttonEqual;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        decimalFormat = new DecimalFormat("#.#####");

        inputDisplay = findViewById(R.id.input);
        outputDisplay = findViewById(R.id.output);

        button0 = findViewById(R.id.btn0);
        button1 = findViewById(R.id.btn1);
        button2 = findViewById(R.id.btn2);
        button3 = findViewById(R.id.btn3);
        button4 = findViewById(R.id.btn4);
```

```

        button5 = findViewById(R.id.btn5);
        button6 = findViewById(R.id.btn6);
        button7 = findViewById(R.id.btn7);
        button8 = findViewById(R.id.btn8);
        button9 = findViewById(R.id.btn9);

        buttonAdd = findViewById(R.id.add);
        buttonSub = findViewById(R.id.subtract);
        buttonDivide = findViewById(R.id.division);
        buttonDot = findViewById(R.id.btnPoint);
        buttonMultiply = findViewById(R.id.multiply);
        buttonClear = findViewById(R.id.clear);
        buttonOFF = findViewById(R.id.off);
        buttonEqual = findViewById(R.id.equal);
        buttonPercent = findViewById(R.id.percent);

        game = findViewById(R.id.btnGame);
        tolong = findViewById(R.id.btnHelp);

        game.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this,
AddPlayers.class);
                //intent.putExtra("Username", username);
                startActivity(intent);
            }
        });

        tolong.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Log.i("Send Email", "Email processing");
                String[] TO =
{"ashifabdulmalikbinkamarudin2001@gmail.com"};
                String[] CC = {"ali@gmail.com"};

                Intent emailIntent = new Intent(Intent.ACTION_SEND);

                emailIntent.setData(Uri.parse("mailto"));
                emailIntent.setType("text/Plain");
                emailIntent.putExtra(Intent.EXTRA_EMAIL, TO);
                emailIntent.putExtra(Intent.EXTRA_CC, CC);
                emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Per:
MAKLUMAN");
                emailIntent.putExtra(Intent.EXTRA_TEXT,
"Assalamualikum...");

                try {
                    startActivity(Intent.createChooser(emailIntent, "Send
Email..."));
                    finish();
                    Log.i("Finish sending Email...", "");

                } catch (android.content.ActivityNotFoundException ex) {
                    Toast.makeText(MainActivity.this, "There is no email
installation...",
                                Toast.LENGTH_SHORT).show();
                }
            }
        });

```

```

});

button0.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + "0");
    }
});

button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + "1");
    }
});

button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + "2");
    }
});

button3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + "3");
    }
});

button4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + "4");
    }
});

button5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + "5");
    }
});

button6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + "6");
    }
});

button7.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + "7");
    }
});

button8.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

```

```

        inputDisplay.setText(inputDisplay.getText() + "8");
    }
});

button9.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + "9");
    }
});

buttonAdd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        allCalculations();
        currentSymbol = ADDITION;
        outputDisplay.setText(decimalFormat.format(firstValue) +
"+"");
        inputDisplay.setText(null);
    }
});

buttonSub.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        allCalculations();
        currentSymbol = SUBTRACTION;
        outputDisplay.setText(decimalFormat.format(firstValue) +
"-");
        inputDisplay.setText(null);
    }
});

buttonMultiply.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        allCalculations();
        currentSymbol = MULTIPLICATION;
        outputDisplay.setText(decimalFormat.format(firstValue) +
"*");
        inputDisplay.setText(null);
    }
});

buttonDivide.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        allCalculations();
        currentSymbol = DIVISION;
        outputDisplay.setText(decimalFormat.format(firstValue) +
"/");
        inputDisplay.setText(null);
    }
});

buttonPercent.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        allCalculations();
        currentSymbol = PERCENT;
        outputDisplay.setText(decimalFormat.format(firstValue) +

```

```

"%");
        inputDisplay.setText(null);
    }
});

buttonDot.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        inputDisplay.setText(inputDisplay.getText() + ".");
    }
});

buttonClear.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (inputDisplay.getText().length() > 0) {
            CharSequence currentText = inputDisplay.getText();
            inputDisplay.setText(currentText.subSequence(0,
currentText.length() - 1));
        } else {
            firstValue = Double.NaN;
            secondValue = Double.NaN;
            inputDisplay.setText("");
            outputDisplay.setText("");
        }
    }
});

buttonOFF.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});

buttonEqual.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        allCalculations();
        outputDisplay.setText(decimalFormat.format(firstValue));
        firstValue = Double.NaN;
        currentSymbol = '0';
    }
});
}

private void allCalculations() {
    if (!Double.isNaN(firstValue)) {
        secondValue =
Double.parseDouble(inputDisplay.getText().toString());
        inputDisplay.setText(null);

        if (currentSymbol == ADDITION)
            firstValue = this.firstValue + secondValue;
        else if (currentSymbol == SUBTRACTION)
            firstValue = this.firstValue - secondValue;
        else if (currentSymbol == MULTIPLICATION)
            firstValue = this.firstValue * secondValue;
        else if (currentSymbol == DIVISION)
            firstValue = this.firstValue / secondValue;
        else if (currentSymbol == PERCENT)

```

```
        firstValue = this.firstValue % secondValue;
    }else {
        try{
            firstValue =
Double.parseDouble(inputDisplay.getText().toString());
        }catch (Exception e) {

        }
    }
}
```

6.2 AddPlayers.java

This code is where it holds player 1 and 2's name before it brings player to the game board. It also defines player 1 as the one with X's mark while player two with O's mark.

```
AddPlayers.java
package com.example.calculator;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class AddPlayers extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_players);

        EditText playerOne = findViewById(R.id.playerOne);
        EditText playerTwo = findViewById(R.id.playerTwo);
        Button startGameButton = findViewById(R.id.startGameButton);

        startGameButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String getPlayerOneName = playerOne.getText().toString();
                String getPlayerTwoName = playerTwo.getText().toString();

                if (getPlayerOneName.isEmpty() ||
                    getPlayerTwoName.isEmpty()) {
                    Toast.makeText(AddPlayers.this, "Please enter player
name", Toast.LENGTH_SHORT).show();
                } else {
                    Intent intent = new Intent(AddPlayers.this,
TicTacToe.class);
                    intent.putExtra("playerOne", getPlayerOneName);
                    intent.putExtra("playerTwo", getPlayerTwoName);
                    startActivity(intent);
                }
            }
        });
    }
}
```

6.3 TicTacToe.java

This is where it handles the logics for 3x3 grid game board. It makes the player 1 move first followed by player 2 until one of them win or there is no more room to mark (draw).

TicTacToe.java

```
package com.example.calculator;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;

import com.example.calculator.databinding.ActivityTicTacToeBinding;

import java.util.ArrayList;
import java.util.List;

public class TicTacToe extends AppCompatActivity {

    ActivityTicTacToeBinding binding;
    private final List<int[]> combinationList = new ArrayList<>();
    private int[] boxPositions = {0, 0, 0, 0, 0, 0, 0, 0, 0}; //9 zero
    private int playerTurn = 1;
    private int totalSelectedBoxes = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityTicTacToeBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        combinationList.add(new int[] {0,1,2});
        combinationList.add(new int[] {3,4,5});
        combinationList.add(new int[] {6,7,8});
        combinationList.add(new int[] {0,3,6});
        combinationList.add(new int[] {1,4,7});
        combinationList.add(new int[] {2,5,8});
        combinationList.add(new int[] {2,4,6});
        combinationList.add(new int[] {0,4,8});

        String getPlayerOneName =
        getIntent().getStringExtra("playerOne");
        String getPlayerTwoName =
        getIntent().getStringExtra("playerTwo");

        binding.playerOneName.setText(getPlayerOneName);
        binding.playerTwoName.setText(getPlayerTwoName);

        binding.image1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (isBoxSelectable(0)){
                    performAction((ImageView) v, 0);
                }
            }
        });
    }
}
```



```

binding.image2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isBoxSelectable(1)){
            performAction((ImageView) v, 1);
        }
    }
});

binding.image3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isBoxSelectable(2)){
            performAction((ImageView) v, 2);
        }
    }
});

binding.image4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isBoxSelectable(3)){
            performAction((ImageView) v, 3);
        }
    }
});

binding.image5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isBoxSelectable(4)){
            performAction((ImageView) v, 4);
        }
    }
});

binding.image6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isBoxSelectable(5)){
            performAction((ImageView) v, 5);
        }
    }
});

binding.image7.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isBoxSelectable(6)){
            performAction((ImageView) v, 6);
        }
    }
});

binding.image8.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (isBoxSelectable(7)){
            performAction((ImageView) v, 7);
        }
    }
});

```

```

    });

    binding.image9.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (isBoxSelectable(8)){
                performAction((ImageView) v, 8);
            }
        }
    });
}

private void performAction(ImageView imageView, int
selectedBoxPosition){
    boxPositions[selectedBoxPosition] = playerTurn;

    if (playerTurn == 1){
        imageView.setImageResource(R.drawable.elonmusk);
        if (checkResults()) {
            ResultDialog resultDialog = new
ResultDialog(TicTacToe.this, binding.playerOneName.getText().toString()
+ " is a WINNER!", TicTacToe.this);
            resultDialog.setCancelable(false);
            resultDialog.show();
        } else if (totalSelectedBoxes == 9) {
            ResultDialog resultDialog = new
ResultDialog(TicTacToe.this, "Match DRAW!", TicTacToe.this);
            resultDialog.setCancelable(false);
            resultDialog.show();
        } else {
            changePlayerTurn(2);
            totalSelectedBoxes ++;
        }
    } else {
        imageView.setImageResource(R.drawable.reddonut);
        if (checkResults()) {
            ResultDialog resultDialog = new
ResultDialog(TicTacToe.this, binding.playerTwoName.getText().toString()
+ " is a WINNER!", TicTacToe.this);
            resultDialog.setCancelable(false);
            resultDialog.show();
        } else if (totalSelectedBoxes == 9) {
            ResultDialog resultDialog = new
ResultDialog(TicTacToe.this, "Match DRAW!", TicTacToe.this);
            resultDialog.setCancelable(false);
            resultDialog.show();
        } else {
            changePlayerTurn(1);
            totalSelectedBoxes ++;
        }
    }
}

private void changePlayerTurn(int currentPlayerTurn) {
    playerTurn = currentPlayerTurn;
    if (playerTurn == 1) {
binding.playerOneLayout.setBackgroundResource(R.drawable.black_border);
binding.playerTwoLayout.setBackgroundResource(R.drawable.black_border);
    } else {

```

```

binding.playerTwoLayout.setBackgroundResource(R.drawable.black_border);
binding.playerOneLayout.setBackgroundResource(R.drawable.black_border);
    }

    private boolean checkResults() {
        boolean response = false;
        for (int i = 0; i < combinationList.size(); i++) {
            final int[] combination = combinationList.get(i);

            if (boxPositions[combination[0]] == playerTurn &&
                boxPositions[combination[1]] == playerTurn &&
                boxPositions[combination[2]] == playerTurn) {
                response = true;
            }
        }
        return response;
    }

    private boolean isBoxSelectable(int boxPosition) {
        boolean response = false;
        if (boxPositions[boxPosition] == 0) {
            response = true;
        }
        return response;
    }

    public void restartMatch() {
        boxPositions = new int[] {0, 0, 0, 0, 0, 0, 0, 0, 0}; //9 zero
        playerTurn = 1;
        totalSelectedBoxes = 1;

        binding.image1.setImageResource(R.drawable.white_box);
        binding.image2.setImageResource(R.drawable.white_box);
        binding.image3.setImageResource(R.drawable.white_box);
        binding.image4.setImageResource(R.drawable.white_box);
        binding.image5.setImageResource(R.drawable.white_box);
        binding.image6.setImageResource(R.drawable.white_box);
        binding.image7.setImageResource(R.drawable.white_box);
        binding.image8.setImageResource(R.drawable.white_box);
        binding.image9.setImageResource(R.drawable.white_box);
    }
}

```

6.4 ResultDialog.java

This is where it will pop up the result box that will foreground the TicTacToe.java when the winning or draw conditions met. Within this dialogue box too, there is a restart button too to refresh the game.

ResultDialog.java

```
package com.example.calculator;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Dialog;
import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class ResultDialog extends Dialog {

    private final String message;
    private final TicTacToe ticTacToe;

    public ResultDialog(@NonNull Context context, String message,
TicTacToe ticTacToe) {
        super(context);
        this.message = message;
        this.ticTacToe = ticTacToe;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_result_dialog);

        TextView messageText = findViewById(R.id.messageText);
        Button startAgainButton = findViewById(R.id.startAgainButton);

        messageText.setText(message);

        startAgainButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ticTacToe.restartMatch();
                dismiss();
            }
        });
    }
}
```

7.0 Testing Strategy

During each sprint, we will actively involve multiple stakeholders in testing the progress of our app. This collaborative approach ensures that we receive valuable feedback from different perspectives. The key participants include:

1. Development Team (Us):

Our team of developers will thoroughly test the app to identify any issues, validate functionality, and ensure adherence to requirements.

We'll focus on unit testing, integration testing, and other relevant testing techniques.

2. Customers (End Users):

Our app's intended users—whether they are potential customers, students, or other stakeholders—will participate in testing.

Their feedback is crucial for understanding real-world usage scenarios and uncovering usability issues.

3. Lecturers:

Involving lecturers or domain experts provides an additional layer of scrutiny.

They can evaluate the app against academic or industry standards, ensuring its quality and alignment with best practices.

Testing Process:

a) Daily Testing Sessions:

At the end of each day during the sprint, we'll conduct testing sessions.

These sessions will be collaborative, involving developers, users, and lecturers.

We'll explore various features, scenarios, and use cases.

b) Feedback Collection:

Participants will provide feedback on:

- App functionality.
- User experience (UX).
- Any issues encountered.
- Suggestions for improvement.

c) Feedback Utilization:

We'll actively use the collected feedback to enhance the app:

- Address reported issues promptly.
- Incorporate usability suggestions.
- Iteratively improve the app based on the feedback loop.

By involving all stakeholders, we aim to create a robust and user-friendly app that meets both technical and user-centric requirements. Continuous testing and feedback will drive our development process, ensuring a high-quality end product.

8.0 Deployment and Rollout

We are planning to deploy our app in Google Play Store after polished and add more games like checkers or even chess in the future. Who knows, with ads integrated to it we can generate a ton of money in a second.

Or we will upload our code in GitHub so any developers can take it and enhance it more with more powerful arithmetic operations like power (**), log, sin, cos, tan, modulus and more.

9.0 Security Considerations

We are aware our app lack security because there is no login feature to begin with. Its is because, when using calculator, we want to open the app as quickly as possible so we scrapped the idea to add login page in this app. It will be a hassle when we calculate let say, event budget but we need to login first into the app. Plus, our team is still lacking in applying database or SQLite to be precise so making a login will take a longer time than expected.