Simple RPN Calculator Design Specification

by: Tai Le

July 11, 2021

# Introduction:

Another calculator app? You asked. There are hundreds if not thousands of calculator apps on Google Play Store. Simple RPN Calculator app is different. It uses Reverse Polish Notation (RPN). It provides many advantages over typical calculators. There are not many RPN calculators on Google Play Store. Of those, many requires internet, contact, location, etc. permissions. Why? Simple RPN Calculator protects your privacy. It does not require any permissions to operate; except when user requests support. User must sign into Google account to request support.

# Purpose:

This purpose of this app is provide users a better way to calculate algebraic equations. It has a stack that stores intermediate results for later use. It also has couple of percentage functions that are useful in everyday life.

For example, how do you calculate the following:

(3+5) / (7+6)

In typical calculator, you type in 3, +, 5. Write down the result. Type in 7, +, 6. Write down the result. Now type in first result, /, second result.

With RPN calculator, you type in 3, *ENTER* key, 5, +. This adds 3 and 5 and stores the result to stack. Type in 7, *ENTER* key, 6, +. This adds 7 and 6 and stores the result to stack. Type in /. This divides the 2 numbers in the stack.

This calculator also has +% and -% functions. They add/subtract the percentage of a value. For example: if an item cost $17.89 and 30% off. How much it is?

With typical calculators, you need to calculate 30% of $17.89 then subtract it from $17.89.

With Simple RPN Calculator, you enter 17.89, 30, then -%.

# Scope

This app is for all audiences and those who care about their privacy. There will be some learning curve for people who do not know Reverse Polish Notation. See <https://www.thecalculatorstore.com/Manuals/why-RPN> on how to use RPN calculator.

# Design Overview

Android Studio will be used to design/develop this app. It is based on Java language. The app contains a stack, input field, and buttons.

* Stack – Use RecyclerView to display entered numbers and intermediate results
* Input Field – Use EditText view to allow users to enter the numbers
* Buttons – Allow users to enter numbers and operations. Users also can enter the numbers from the keyboard.
* Menu items
  + Contact support – Submit a support ticket
  + EULA – Display End-User License Agreement

##### Test Cases:

* Numbers and decimal point are correctly entered and displayed
* Math operations (divide, multiply, add, subtract, percentages) are done correctly
* Extreme values – upper and lower ends operations (overflow/underflow)
* Navigation buttons – Clear, erase, back buttons function as designed
* Texts and numbers format are correctly displayed in localized languages

# Use Cases

Case 1 - Find the total of the following items: 5 apples at $0.27 each, 8 oranges at $0.39 each3 banana at $0.16 each:

To calculate this, type in:

* 1. 5, 0.27, X: This calculates the cost of apples and stores it in the stack.
  2. 8, 0.39, X: This calculates the cost of oranges and stores it in the stack.
  3. 3, 0.16, X: This calculates the cost of bananas and stores it in the stack.
  4. +, + : This adds cost of apples, oranges, and bananas (values in stack)

Case 2 – Now add 8.9% tax to the total in case 1:

To calculate this, type in:

* + 8.9, +%

Case 3 – A Shirt, which cost 25.38, is 35% off. What is the price?

To calculate this, type in:

* 1. 25.38, *ENTER:* This put 25.38 to the stack.
  2. 35, -%: This subtracts 35% from the value in stack.

# Mock-Up

















