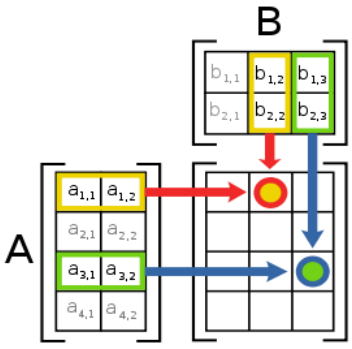



# Matrix Calculator Documentation

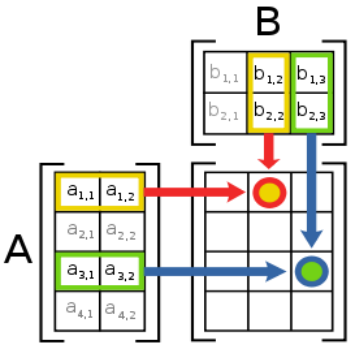
Ka Son Chan

# Introduction

This Matrix Calculator Multi-threaded Android Application allows the user to enter 2 matrices row and column sizes, the number data and performs matrix operation – Multiplication. This application is utilizing multi-threads in the operation, which enhance the speed of the application comparing to single thread app.

This app is written in Scala with Eclipse, Android SDK, ADT ad AndroidProguardScala. This documentation is divided into two main sections - Developer Guide and User Guide.

## Table of Contents




# Matrix Calculator Developer Guide

Ka Son Chan

## D1. Overview

This Developer Guide presents the architecture, class and sequence diagrams of the Matrix Calculator Multi-threaded Android Application in three sections along with explanations.

## D2. Architecture

Figure D-1 below shows the Android Architecture. The Android Operating System (OS) can be referred to as a software stack of different layers, where each layer is a group of several program components. It includes operating system, middleware and important applications. Each layer in the architecture provides different services to the layer just above it.

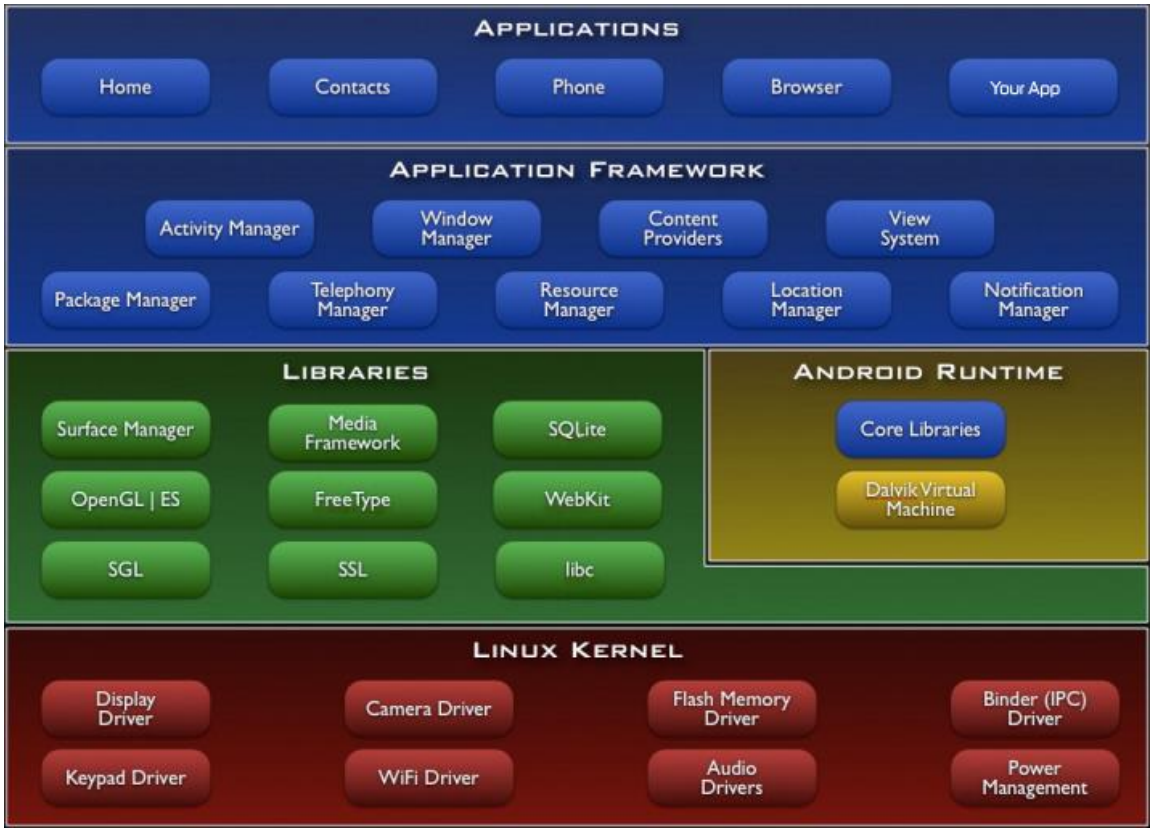



Figure D-1

In this project, I mostly use the Core libraries in Android Runtime layer. In addition, the Application Framework layer is the block that my application directly interacts with as follow. These programs manage the basic functions of phone like resource management. I used the following to build my basic application - Activity Manager, Window Manager, Content Providers, View System, Package Manager, Resource Manager and Notification Manger in Application Framework layer.

## D3. Class Diagram

This application contains three classes – Main, Matrix and Result; and one trait – Functions. Figure D-2 below shows the class diagram of the application.

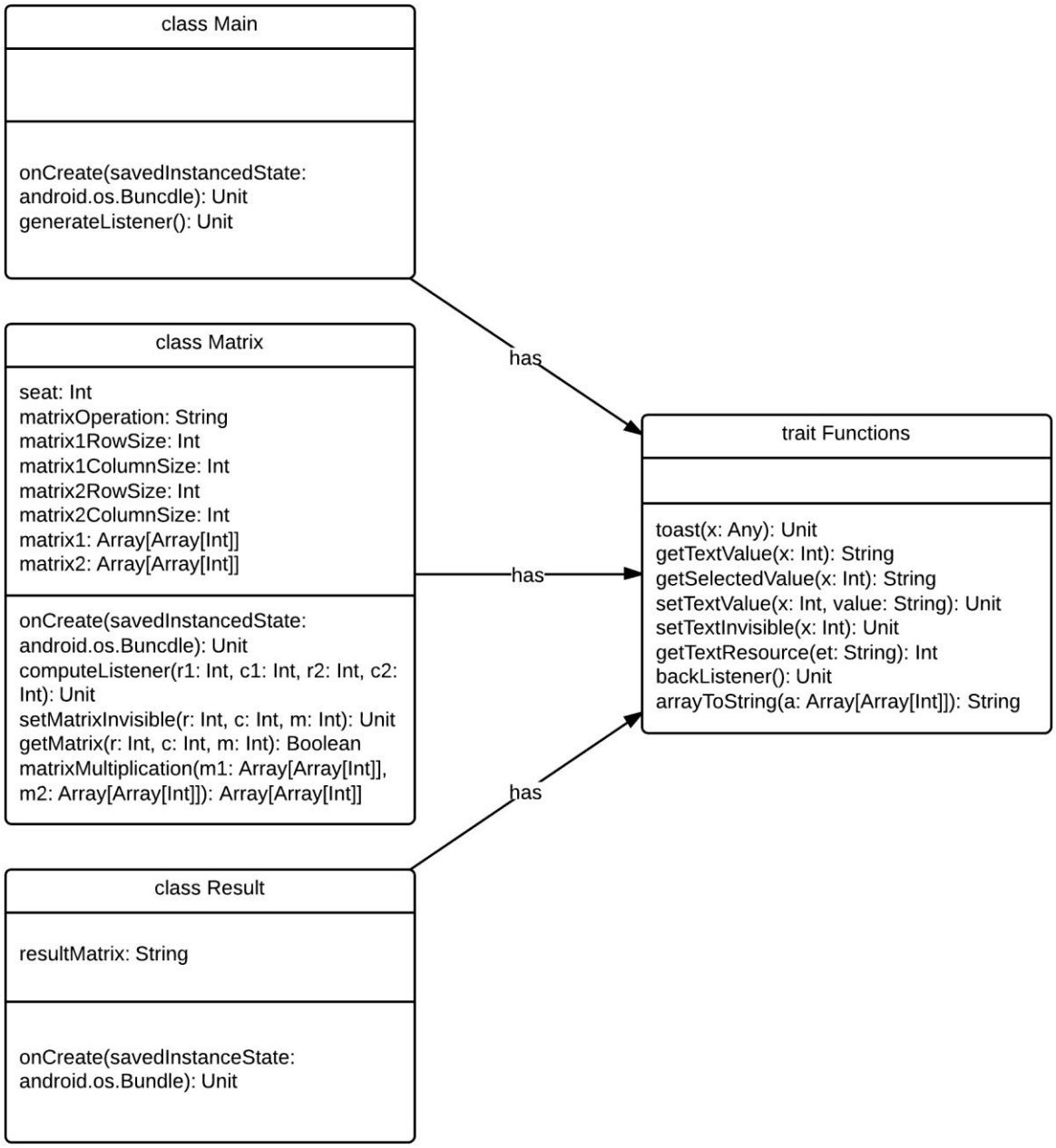



Figure D-2

### D3.1. Main

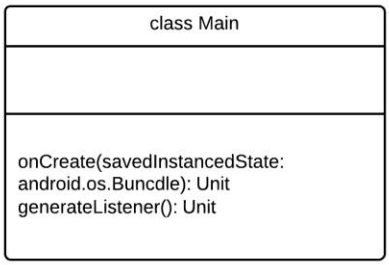



Figure D-3

Figure D-3 shows the class Main that contains two functions:

- onCreate(savedInstancedState: android.os.Bundle): Unit: show the main.xml layout and calls the generateListener() functions
- generateListener(): Unit: let the user to generate the matrices with the input row and column sizes

### D3.2. Matrix

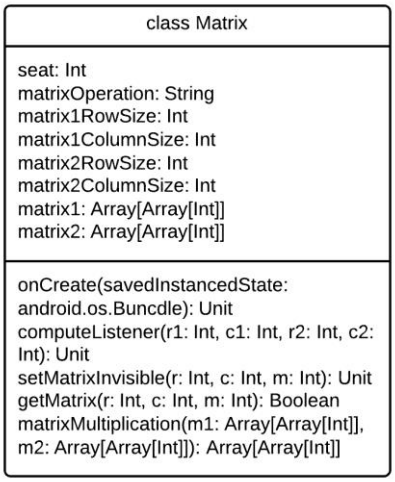



Figure D-4

Figure D-4 shows the class Matrix that contains the following functions:

- onCreate(savedInstancedState: android.os.Bundle): Unit: show the matrix.xml layout and calls the computeListener() and setMatrixInvisible()
- computeListener(r1: Int, c1: Int, r2: Int, c2: Int): Unit: call the getMatrix() functions to set up the arrays of matrix1 and matrix2, and calls the matrixMultiplication() to compute the matrix multiplication
- setMatrixInvisible(r: Int, c: Int, m: Int): Unit: set the matrix to be invisible to the user
- getMatrix(r: Int, c: Int, m: Int): Boolean: return true if there is no input error and false otherwise
- matrixMultiplication(m1: Array[Array[Int]], m2: Array[Array[Int]]): Array[Array[Int]]: compute the matrix multiplication and return as 2-dimensional array

### D3.3. Result

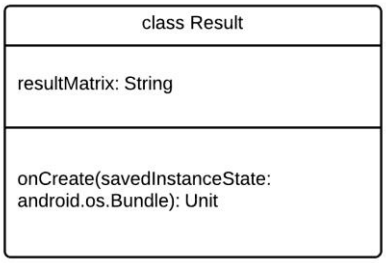



Figure D-5

Figure D-5 shows the class Result that contains only one function:

- onCreate(savedInstanceState: android.os.Bundle): Unit: show the result.xml layout and wait for the user to click on the back button

### D3.4. Functions

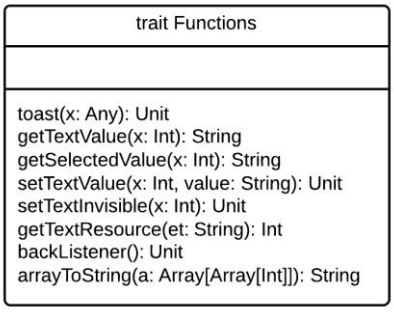



Figure D 6

Figure D-6 shows the trait Functions contains all the help functions that are used in more than one classes mentioned previously:

- toast(x: Any): Unit: prompt the error message
- getTextValue(x: Int): String: return the string value of the given edit text id x
- getSelectedValue(x: Int): String: return the string value of the given spinner id x
- setTextValue(x: Int, value: String): Unit: set the given edit text id x to string value
- setTextInvisible(x: Int): Unit: set the edit text id x to be invisible
- getTextResource(et: String): Int: get the resource id with the edit text id
- backListener(): Unit: finish the current activity and go back to the previous one
- arrayToString(a: Array[Array[Int]]): String: convert array a to string

## D4. Sequence Diagram

Figure D-7 below shows the sequence diagram of the app.

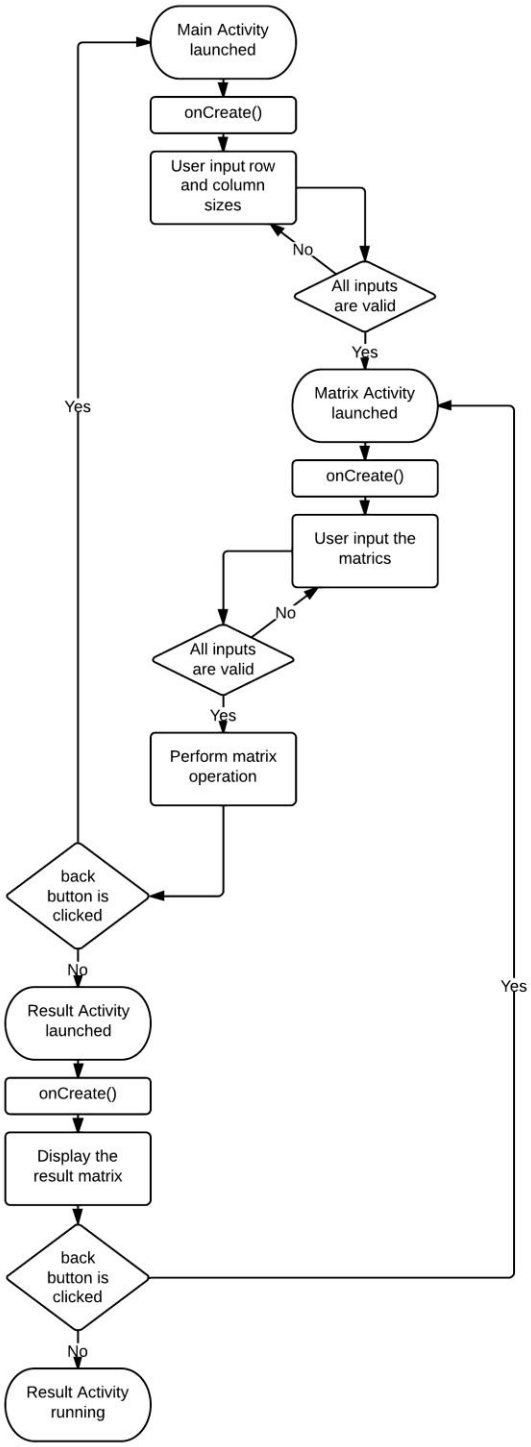
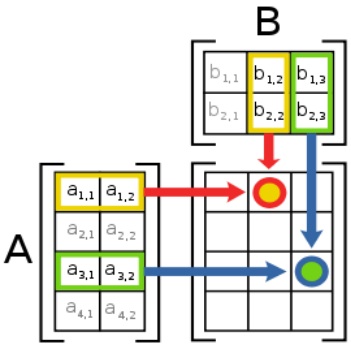



Figure D-7

## D5. References

*Android Architecture – The Key Concepts of Android OS*. (n.d.). Retrieved September 18, 2013, from Android-App-Market.com: Android Development Tutorial: http://www.android-app-market.com/android-architecture.html




# Matrix Calculator User Guide

Ka Son Chan

## U1. Overview

The Multi-threaded Android Application – Matrix Calculator allows the Android System to operate matrix operation – Multiplication more efficiently and quickly. The purpose of this user guide is to present a description of how to use this application. The application is divided into three main layout pages – Main, Matrix and Result. This user guide outlines each feature, in detail, that the application provides and would help the user to have a better understanding of the application. The user guide displays each picture of the interface with steps on how to proceed and get the result.

## U2. Main

This is the Main layout page that the user sees when he or she first accesses the application as shown in Figure U-1 below.

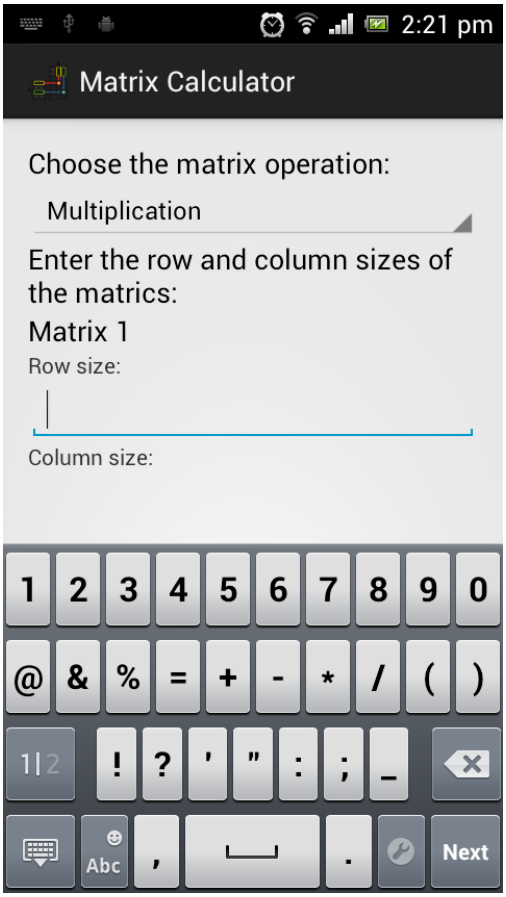



Figure U-1

- This page allows the user to choose the matrix operation.
- This page allows the user to enter two matrices row and column sizes.
- The user is allowed to enter the row and column sizes in the range between 1 and 5 inclusively.
- The user can only enter keypad 1, 2, 3, 4, or 5 and others are locked.

After the user entered all the sizes, he or she can press the Generate button to let the application generate the row and column fields. Figure U-2 shows the user did not entered all the size and press the Generate button. The application will prompt an error message "Empty matrix row or column size. Enter size range between 1 to 5."

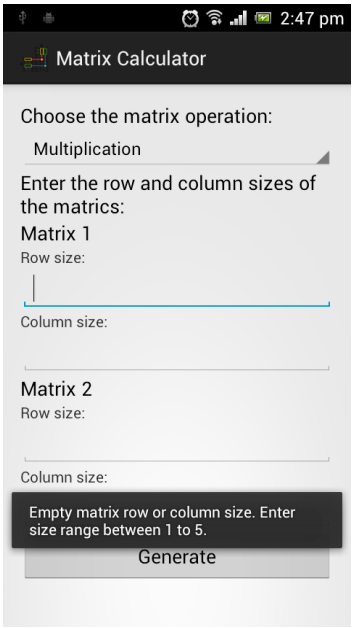



Figure U-2

Figure U-3 below shows the user entered the matrix 1 column size 2 and matrix 2 row size 3. The matrix multiplication requires matrix 1 column size should be equals to matrix 2 row size. Therefore error message "Matrix Multiplication: Matrix 1 column size should be equals to Matrix 2 row size." will be prompted.

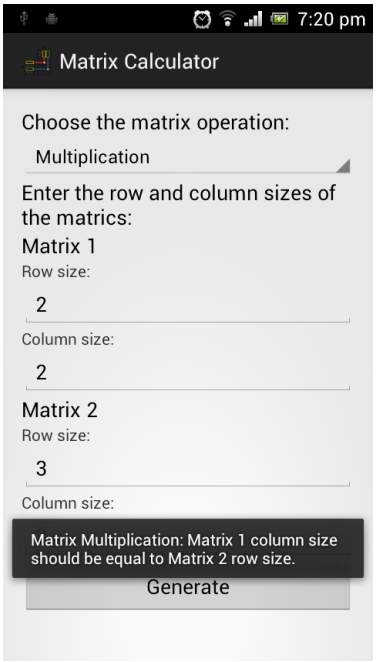



Figure U-3

Figure U-4 shows the user correctly entered the row and column sizes of the matrices below. The user can press the Generate button. The application will show the generated matrices according to the user's input as shown in Figure U-5.

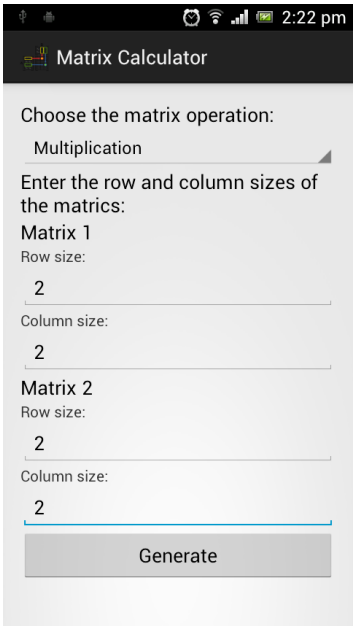



Figure U-4

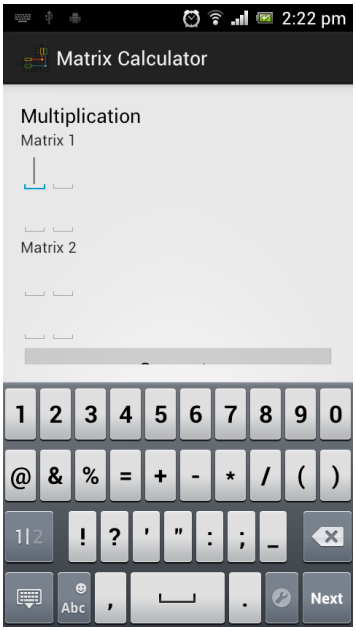



Figure U-5

Figure U-5 shows the second layout page – Matrix that allows the user to enter the matrix field by field.

## U3. Matrix

Figure U-6 shows the Matrix layout page as shown in Figure U-5.


Figure U-6

- This Matrix layout page allows the user to enter the matrix with his/her choice of row and column sizes between 1 to 5.
- The Matrix layout page allows the user to go back to the previous Main layout page by pressing the Back button.
- The Matrix layout page allows the user to compute the matrix operation by pressing on the Compute button.
- The Matrix layout page allows the user to enter digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and symbol “-”.

Figure U-7 on the next page shows the user did not enter the matrices and press the Compute button. A error message “Empty matrix.” will be prompted.

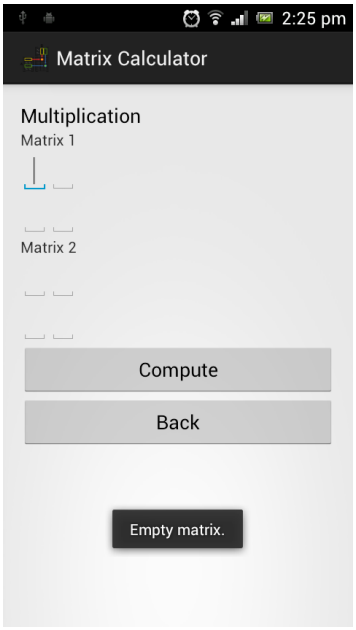



Figure U-7

As mention in the previous page, Figure U-7 shows the prompt of the error message.

Since Matrix allows negative integers, therefore, the user is allowed to enter "-" but not by itself. Figure U-8 shows the prompt of the error message "Invalid matrix data "-"."

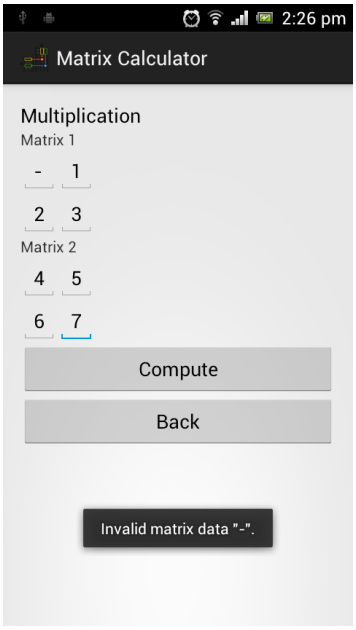



Figure U-8

Any symbol "-" by itself is not allowed in this Matrix layout page. Error message will be prompted.

Figure U-9 shows the user has entered valid data in all the row and column fields. The user can press Back button to go back to previous Main layout page or press the Compute button to let the application to compute the result.

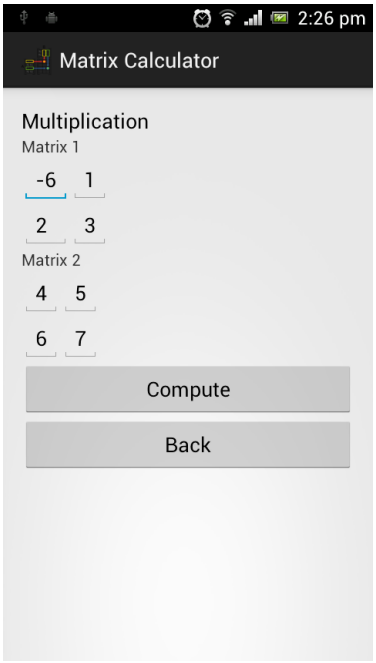



Figure U 9

Figure U-10 shows the user press on the Compute button after entering valid data in all the row and column fields. The matrix multiplication is computed and the result is displayed.

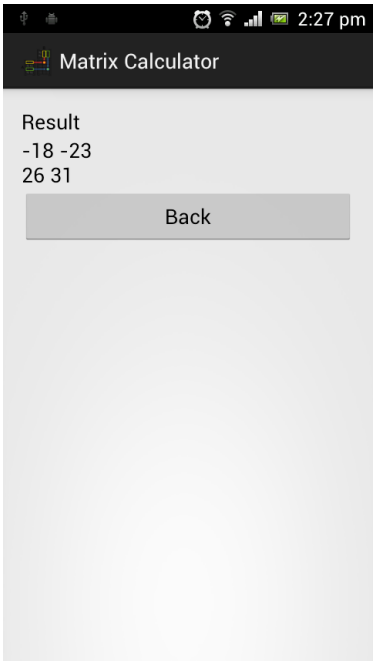



Figure U-10

## U4. Result

The Result layout page displays the result of the matrix operation as shown in Figure U-10 and Figure U-11.

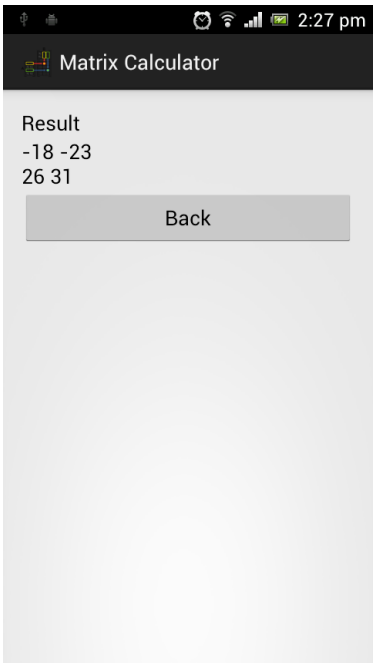



Figure U-11

- This Result layout page allows user to go back to Matrix layout page.

Figure U-12 below shows the Matrix layout page after the user pressed on the Back button in the Result layout page.

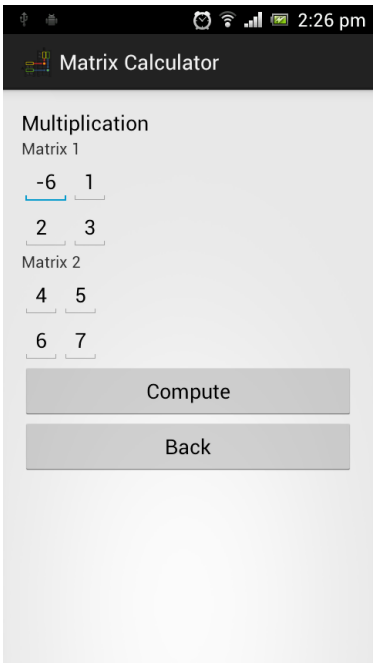



Figure U 12