

**Department of Electrical & Computer Engineering**

**North South University**

**Project Report**

**Project Name:** Calculator App

**Submitted by: Team - 03**

|  |  |  |
| --- | --- | --- |
| **Team Details** | | |
| **1** | Kazi Sakib Ahmad | ID: 1510702042 |
| **2** | Sohanur Rahman | ID: 1510464642 |

**GitHub Link:** <https://github.com/nsuspring2019cse427/Group03>

**Course:** CSE427 (Software Quality Assurance & Testing)

**Section:** 01

**Semester:** Spring 2019

**Submitted to:** Shaikh Shawon Arefin Shimon (SAS3)

Lecturer, Department of ECE

# **Project Description**

## **Introduction**

The project is mainly emphasized on implementing the learning outcomes of the academic course **‘Software Quality Assurance & Testing (CSE427)’.** In order to meet project checkpoints we selected a simple Calculator App, which was developed in Android platform and further tested it following software testing methodologies. Initially we have implemented the basic and obvious features of a Calculator. Our primary focus was on unit testing the basic features and methods of the app.

## **Background and Product Context**

This calculator app we choose to work with was developed about 5-6 months ago as a test project. This app is based on Android platform and has the basic functionalities (i.e. Addition, Subtraction, Multiplication, and Division) of a calculator. As an Android app we choose to implement its full backend using Java. It has a user interface which seems very simple to its users.

## **Testing Aspects We Implemented**

We have successfully implemented following tasks as testing aspects:

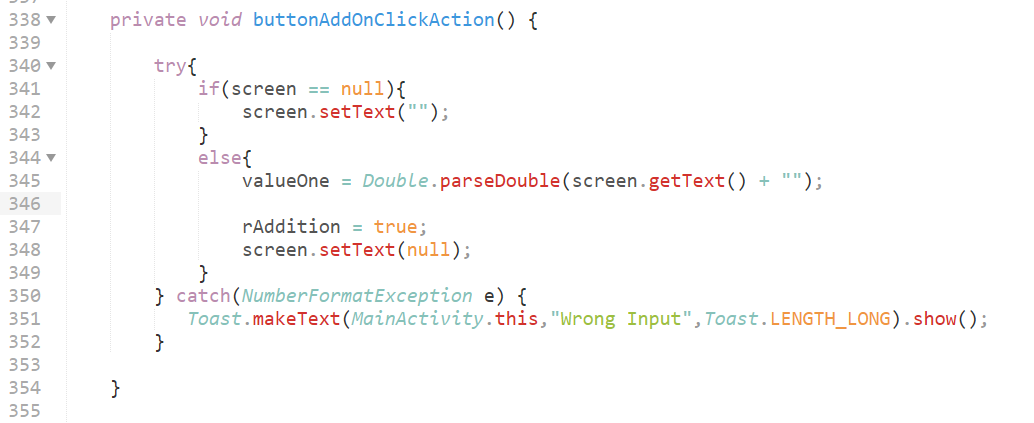
* Unit testing each JAVA methods implemented in the existing project using JUnit.
* Catching the uncaught exception.
* Integration testing.
* Functionality testing.
* Input space partitioning.
* Graph partitioning.
* Fixing out the existing bugs after unit testing.

## **Tools/Frameworks Used:**

* JUnit 4
* Android Studio (IDE)
* Eclipse (IDE)

# **Graph Partitioning**

Graph partitioning of the piece of code under the method buttonAddOnClickAction() [line 338 – 354 of MainActivity.java ]



**Figure 1: Snapshot of buttonAddOnClickAction()**

**B**

**A**

**C**

**D**

**E**

Figure 2: Method buttonAddOnClickAction() with node defined

**E**

**D**

**C**

**B**

**!B**

**A**

**!A**

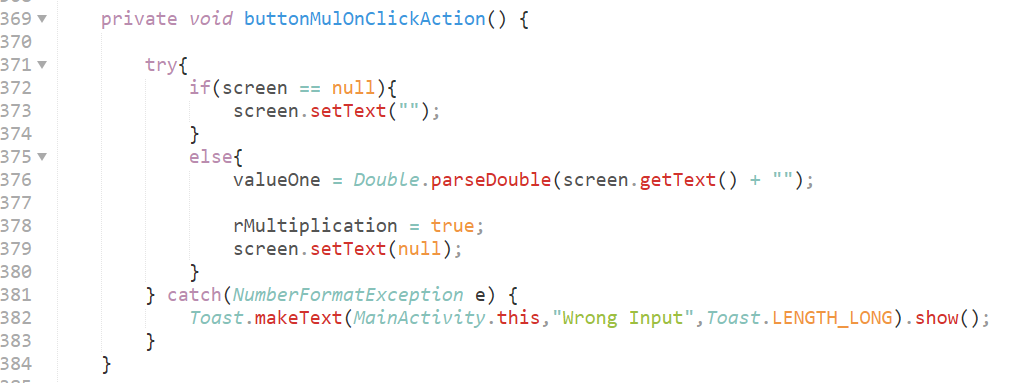
 Graph partitioning of the piece of code under the method buttonMulOnClickAction()

Figure 3: Method buttonMulOnClickAction() with node defined

**A**

**B**

**C**

**D**

**E**

**E**

**D**

**C**

**B**

**!B**

**A**

**!A**

Graph partitioning of the piece of code under the method buttonDivOnClickAction()

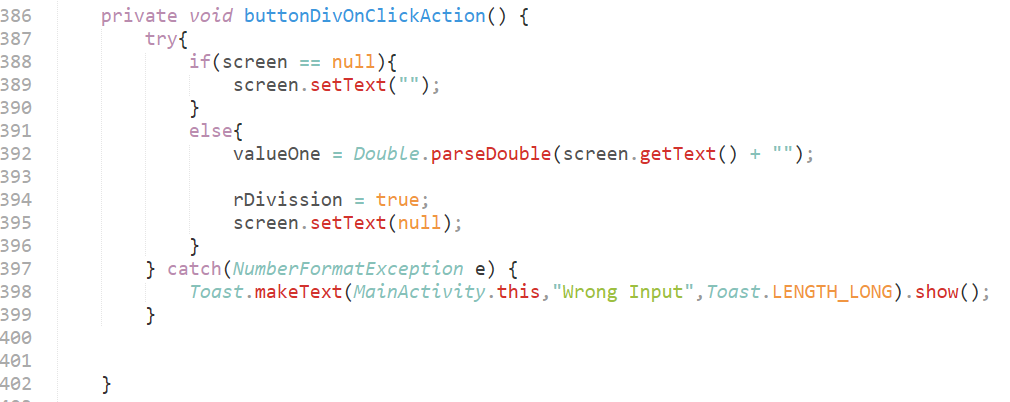


Figure 4: Method buttonDivOnClickAction() with node defined

**A**

**B**

**C**

**D**

**E**

**E**

**D**

**C**

**B**

**!B**

**A**

**!A**