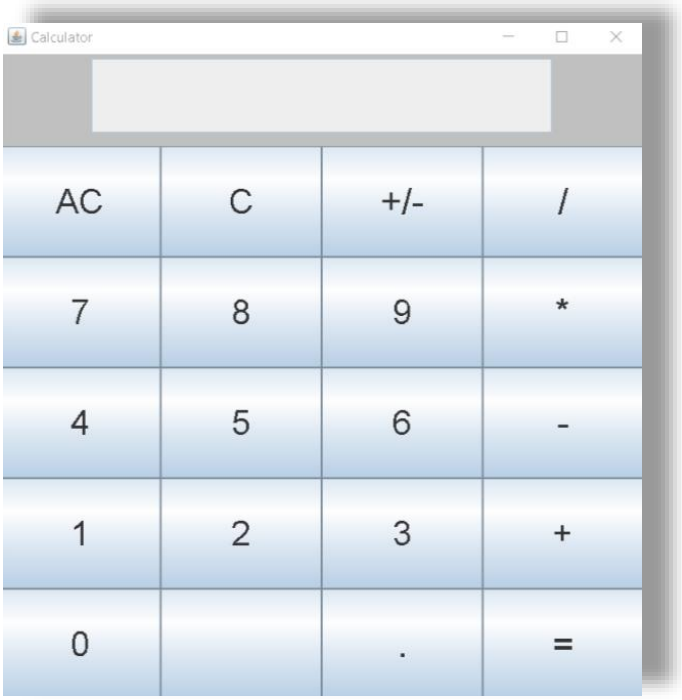

PROJECT REPORT

Calculator

Anthony Pena | Queens College Cs 370- Software Engineering | Sept 18, 2018

TABLE OF CONTENTS

Document Description

Mission Statement

"To be the most accurate and easy GUI for a four function calculator."

Who are we

We are all a group of 5 students who for the most part are beginning our senior year in Queens College. We consist of Asma Pasha, Jiening Li, Louis Deng, Christian Bonifacio and myself.

Goal

Our goal is to have one of the most accurate program, an application that would run fast in any system as long as they have the bare minimum but that functionality will remain as effective through out. A program that can be easily maintain and to work on the development of simplicity and not complexity.

About this Document

This document explains some of the theories, methods and process used in order to achieve the completion of the creation of the software for the Calculator Application. Developing on the initial organization of group, to ordering and execution of the project.

Debugging Procedures & Testing

Process Model

Having meet in the same classroom and being able to collaborate through WhatsApp messaging platform as our main communication method. We were planning to utilize GitHub for collaboration and version control but the lack of knowledge and previous experience with GitHub was not the most adequate for us considering our current timeframe to provide working application, therefor we decided to use google cloud services in order for the group to have access to files where any major correction to the code would be uploaded in own folder with name scheme to know version. Having everyone agreed to use Java as the only programming language to collaborate prevented issues of compatibility, and since the requirement of application in not intensive and complex with a basic java project structure would be enough for this time.

There are 3 models from waterfall model, incremental development, and reuse-oriented software model. Among this due to scalability, complexity and our current ability, we went with the incremental development approach were we would create a version with narrow amount of planning about structure, framework and maintenance but we would have a functioning application in fast manner by sacrificing aspects mention and work on the first task at hand. This is a method that I believe is convenient due to the fact that is of low complexity and little time to develop it. Considering time there was no need to do metric analysis of progress and we wouldn't have to worry about structure degrading overtime because of complexity of the project. This was best Process model to implement.

Requirement of Engineering Process

- *Feasibility Study*

  For Feasibility there are four factors we consider in order to evaluate the engineering process according to Sommerville from the book Software Engineering 9th edition, this are User satisfaction, Cost effective, Existing budget, study time spend. In our project the fact is that due to our current goal and mission the importance is the user satisfaction from an accurate and light app with a product that requires low maintenance in both labor and skill. Without budget, this is cost effective and the time spend in study was insignificant due to the fact that options were limited to free services as WhatsApp, Google Drive, and Java's build in repositories.

- *Requirement Elicitation and Analysis*

  Now as for this analysis we will be looking at existing system, user potential procedures and tasks. The current system we had in mind to develop and test were laptops in both windows and Mac. There would be different IDE's for development therefore a constant and typical

structure for the coding project had to be simple which in our case Eclipse and NetBeans would be able understand interchangeable the JAVA Project structure with JAVA version 8. Current project remains under 75KB without being compressed yet therefore it can be run in any modern equipment. Now for some of the potential user inputs we need to keep in mind that even though there are only 4 operators, we need to make sure to create functions for error handling in both mathematics like infinities, and undefined arithmetic operations like 0/0. Besides this the other challenge is to determine how to handle button representation in display. Such action required exception handlers in order to properly display numbers in proper order and to know how to represent in GUI the period symbol. In our case we passed every test during demo but the display didn't show the "0.0" but ".0". Nonetheless we need to also consider some of the use by user is to do computation with large numbers where it might overflow but as described in project requirements we wont need to worry about this.

- *Requirement Specifications*

  Some of the specifications of the requirements for both the user and system were provided by the professor. Under the description for system requirement there is only the requirement to create the file with program and for further transition of software to be set in ZIP file.

- *Requirement Validation*

  We checking for realism, consistency, completeness, and further more on future modification. Currently it was realistic to provide software under a week based on the current mission of the team due to the simplicity, there will be consistency of services but we don't know about the GUI as of now. For a conclusion about the completeness of the project we need to verify its full operation and representation in the display, including the color scheme and dimensions of application. Now for the further future modifications will be minimum due to the fact that we will remain on only 4 arithmetic operations, but what we will be beneficial to consider is to implement a data structure in order to contain previous history as well as a better scheme for version control which most likely will be transition to GitHub. As for the code structure we currently have action listeners for each bottom, and it might be better to have one function as a listener functionality to reduce length of code.

Presentation of Demo

Testing process

- *Development Testing*

  Prior to the developing there had to be a clear understanding of the components that would be making the system, this were Eclipse, the system would be on a workstation not mobile, and it would have need to have JAVA. This conclusion was talked and set by the first day. Chris and Jiening Li were the two of the most contributors in the beginning by setting the pace and creating the structure of GUI in order for team to develop on the methods and corrections. Initially even though we had most of the application look, there was still much needed to have a fully functioning calculator. The second phase was to configure functions needed in order to do the arithmetic operations. We were planning on dividing team into two groups, one who focus on Calculator and other to continue the Email client but thanks to extension of EMAIL project we decided to put all our energy into the development of the calculator. Having the Calculator GUI and operations set we began the testing and debugging. One of the first issues found in 9/13/2018 was the sign +/- button was only changing sign once. For this we had to modify our function class “ListenToPlusMins” action listener in order to make sure that no matter how many times we select it, it will change the symbol and send character to the display. On the following day before having this issue fixed we also found the decimal point implementation incorrectly and the trailing zero problem by modifying “ListenToPoint” action listener and creating an extra if statement plus on the same day we realize that the addition was not displaying correctly therefore among Louis and Jiening we able to modify method and found way to show it on the display by creating a TEMP variable and concatenating “jtfResult” variable string in the “ListenToAdd” method. This version was uploaded as version 3. There was an email send that the zero in the front remained and also we were not able to keep correct result of previous computation to continue using it to do more arithmetic computations ex. 1+2=3 *.2 would equal 6 which fix became version 4 and after some modifications the final version for presentation.

- *Acceptance Testing*

  On the day of the demo, we were making sure to use Chris windows computer and have other laptops with program open in case problem to be able to continue the presentation. We did some final testing before class and we were happy with functionality of our software. After testing the only issue found was that when selecting zero, the number did not appear even though the calculations were working correctly. A regroup to know how to coordinate to finish this application and to continue on the Email application was made after the demo.

Additional Test

Some of the additional testing we continue were to fix the issue with the zero button not displaying which by now this bug in the display is fixed. As of now we have no further testing but in developing for better use, the aspect of being able to modify previous results in the display is not required but would be beneficial but it would be developing beyond of our current mission statement.