GENESIS - Learning Outcome & Mini-project Summary Report

LTTS
GLOBAL
ENGINEERING
ACADEMY

L&T Technology Services

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By	To be Approved	Remarks/Revision Details
1		Vinay Kumar			Requirements
2		Pavan S L			Design
3		Afrid Shariff			Test Plan

Table of Contents

List of Figures

List of Tables

Miniproject -2 JavaScript Jasmine and HTML [Team]

Modules

1. Modules linked to project: JavaScript and Jasmine. JavaScript is used for building and styling of web pages.
2. Jasmine is used for testing the built web page.

Topic and Subtopics

1. Basic tags, CSS, different attributes and their usage in styling, viewing the content in default browser.
2. For testing different functions used in web page, Jasmine framework is used.
3. Describing a test suite, Spec file creation. Writing test cases in Jasmine.

Objectives & Requirements

1. Creation of user interactive Calculator in web page and performing arithmetic functions.
2. Testing the created Calculator page using Jasmine framework.

Requirements:

High level

ID	Description
HL_01	Android Studio Code to perform programming.
HL_02	Launch the Chrome browser to display webpage.
HL_03	User Inputs
HL_04	Operation buttons
HL_05	Display bar to display the result.

Table 1 High Level Requirement Analysis for Calculator application

Low level

ID	Description
HL_LL_02_01	Open a Calculator webpage in chrome.
HL_LL_03_01	User input_1
HL_LL_03_02	User input_2

HL_LL_04_01	Arithmetic Operations (+, _, *, /)
HL_LL_04_02	Clear button to clear the inputs
HL_LL_04_03	Operations (Sqrt and %)

Table 2 Low Level Requirement analysis for Calculator application

Design:

Structural diagram:

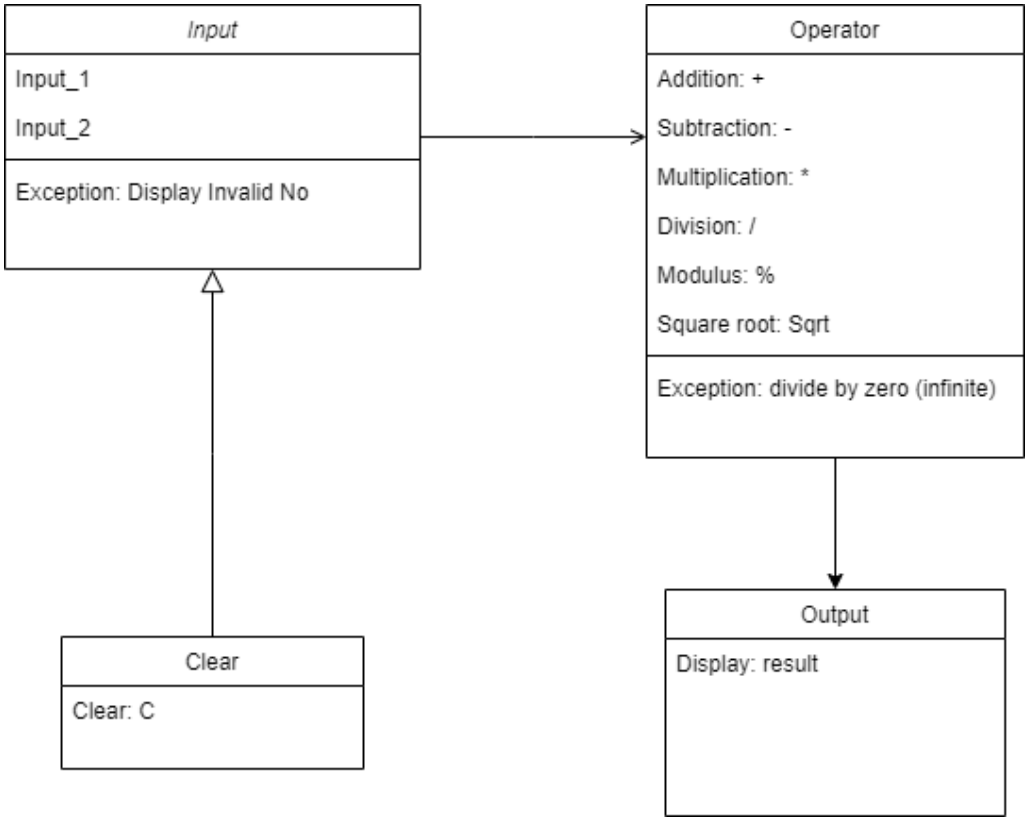
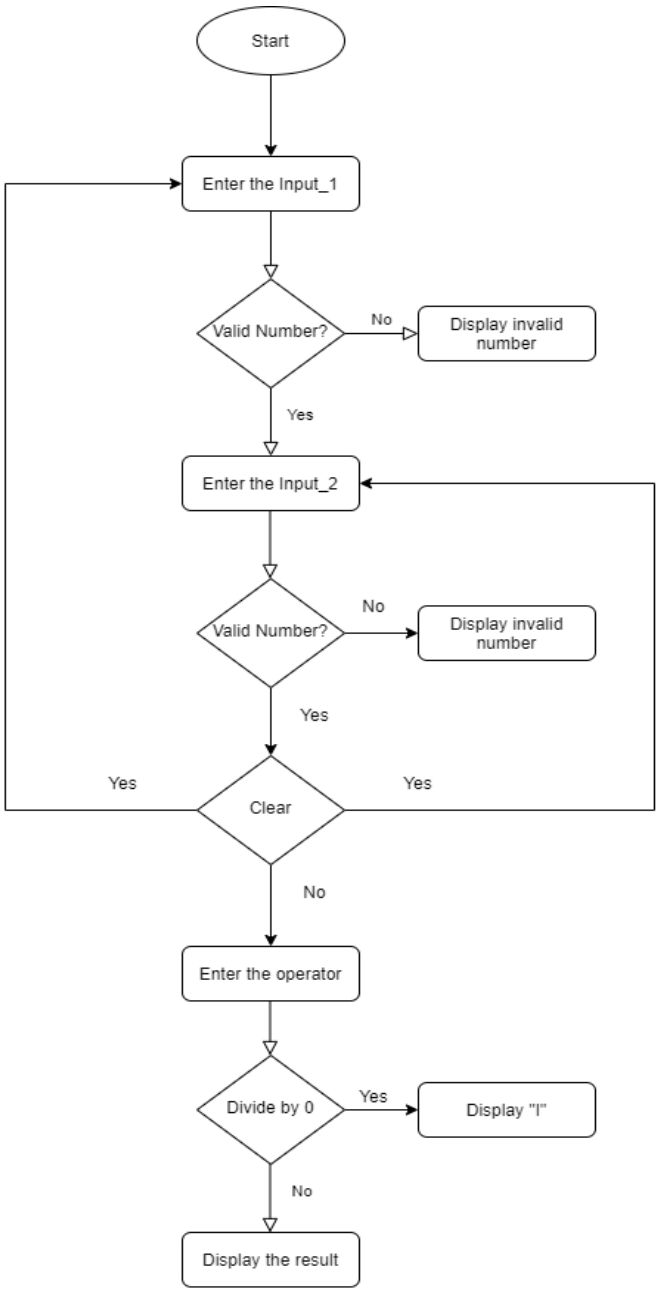


Figure 1 Structural class diagram

Behavioral diagram:



Yes

Figure 2 Behavioral flow chart

Test Plan

No.	Test Case -ID	Test case Objective	Prerequisite	Steps	Input data	Expect ed Result	Actual Result	Re marks/Sta tus
1	TC1	To add two integer and display the result on calculator	Calculator App is launched	1.First Operand a valid integer 2.second operand, a valid integer 3. operator +	135 + 100	235(addition, above twenty-one digits will be expressed in exponential form)	235	Pass
2	TC-2	To subtract two integer and display the result on calculator	Calculator App is launched	1.First Operand a valid integer 2. second operand, a valid integer 3. operator -	135-100	35(abstractio n n,above twenty-one digits will be expressed in exponential form)	35	Pass
3.	TC-3	To multiply two integer and display the result on calculator	Calculator App is launched	1.First Operand a valid integer 2. second operand, a valid integer 3. operator x	100 x 400	40000(multi plication, above twenty-one digits will be expressed in exponential form)	40000	Pass
4.	TC4	To divide two integer and display the result on calculator	Calculator App is launched	1.First Operand a valid integer 2. second operand, a valid integer	10 0/ 25	40	40	Pass

No.	Test Case -ID	Test case Objective	Prerequisite	Steps	Input data	Expect ed Result	Actual Result	Re marks/Sta tus
				3. operator /				
5	TC5	To clear the screen	Calculator App is launched	Press C		Input1 and Input2 Fields should be cleared and placeholder should display	Input 1 and Input2 Fields are cleared and placehol der are displayed	Pass
6	TC6	To Find a Square-root of a number	Calculator App is launched	1.Operand a valid integer 2. Operator Sqrt	25	5	5	Pass
7	TC7	To Find a Modulo of a given numbers	Calculator App is launched	1. First Operand a valid integer 2. second operand, a valid integer 3. operator %	5%6	5	5	Pass

No.	Test Case -ID	Test case Objective	Prerequisite	Steps	Input data	Expect ed Result	Actual Result	Re marks/Sta tus
8	TC7	To check for NaN(Not a number)	Calculator App is launched	Any one or both of the operands are empty		NaN	NaN	Pass

Table 3 Test Plan for Calculator

Implementation Summary

JavaScript is used for building web pages .Different basic tags and styling tags were used to design the web page. Functions for performing different arithmetic operations are written in .js files and are included using script tags in HTML pages. An user interactive calculator is developed using JavaScript. Testing functions are written in .js file and test cases are written in Spec file. All the files are included in Spec Runner.HTML file and executed in default browser to view the test cases validation.

1. GitHub user - **99002562** as a contributor to the repo
2. GitHub user - **99002690** as a contributor to the repo
3. GitHub user - **99002675** as a contributor to the repo

Git Link: LINK

Git Dashboard

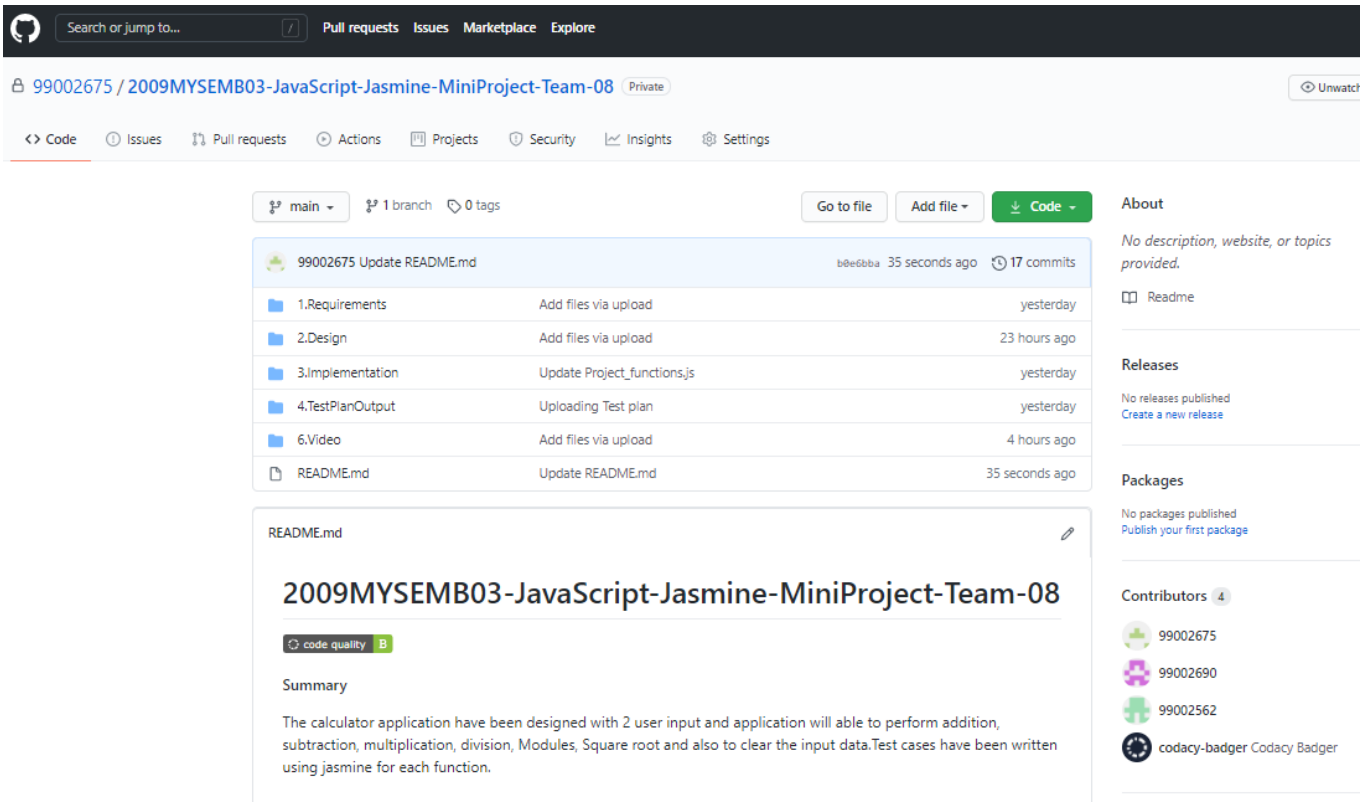


Figure 3 GitHub Dashboard

GIT Summary:

The GitHub repository has the following files which explains documentation of the project:

1. Requirements: This contains the high level and low-level requirements of the project
2. Architecture: This contains the necessary UML diagrams present in the project
3. Implementation: This contains the source code of the following project (HTML, CSS, JavaScript, Jasmine, and Selenium).
4. Test Plan and output: This contains the test plan and the obtained outputs of the project
5. Report: This contains the report of the project

6. Images and videos: This contains the images and the working of project

Video Summary

Team Members: Afrid Shariff, Pavan SL, Vinay Kumar V

Video link: Git Link

Summary

The calculator application have been designed with 2 user input and application will able to perform addition, subtraction, multiplication, division, Modules, Square root and also to clear the input data.Test cases have been written using jasmine for each function.

Code quality and Issues or Bug Tracking

Code quality is done for source code and 'B' badge is obtained.Issues related to certain functions were raised and those were solved by the teammates actively.

Unit Testing

The individual functions/modules used in our project are tested using Jasmine framework.

Individual Contribution & Highlights

Web page was developed by Pavan SL and Afrid.

Jasmine Testing of operations/functions was done by Vinay.

Requirement Analysis was done by Vinay, Design was done by Pavan, Test Plan was done by Afrid.

Most of the work related to documentation and GitHub were done as a team.

Future Scope:

Fallowing feature can be included to enhance the functionality

- Functions can be written to perform more mathematical and trigonometrical operations.
- Further new operations can be implemented so as to make this a scientific calculator.