**Test Plan**

Calculator

Prepared by:

Dhiraj K. Shelke (COBB071)

Pranav N. Gavali (COBB046)

Komal P. Naphade (COBB018)

15/12/2021

TABLE OF CONTENTS

1.0 Introduction

2.0 Objectives and Tasks

2.1 Objectives

2.2 Tasks

3.0 Scope

4.0 Testing Strategy

4.1 Alpha Testing (Unit Testing)

4.2 Integration Testing

4.3 User Acceptance Testting

4.4 Regression Testing

4.5 Beta Testing

4.6 Compatibility Testing

5.0 Hardware Requirements

6.0 Environment Requirements

6.1 Main Frame

7.0 Test Schedule

8.0 Features To Be Tested

9.0 Features Not To Be Tested

10.0 Resources/Roles & Responsibilities

11.0 Schedules

12.0 Significantly Impacted Departments (Sids)

13.0 Dependencies

14.0 Risks/Assumptions

15.0 Tools

16.0 Approvals

**1.0 INTRODUCTION**

We created a small Web based calculator application by selecting Eclipse IDE as our environment and Java as our programming language. We tested the application with the help of Selenium Web driver and Chrome webdriver extension.

**2.0 OBJECTIVES AND TASKS**

**2.1 Objectives of Test Plan**

The test objectives are to verify the Functionality of Calculator Web Application, the project should focus on testing the Calculating operation such as Addition, negative addition, button presses…etc. to guarantee all these operation can work normally in real business environment.

**2.2 Tasks of Test Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **#Business Requirements**  **(BR)** | **#Functional Requirements**  **(FR)** | **#Test Scenarios (TS)** | **#Testing Approaches /Strategies**  **(TA)** |
| 01 | Reduce Error in Calculations | Perform Calculations | Test for Addition, Subtraction, Division, Multiplication,  And other operations | 1.Functional Testing  (Standard Testing)  2.  Non-Functional Testing |
| 02 | Calculator Application Available | Calculator Application must run in Browser | Check if Executable file available or not |

**3.0 SCOPE**

**General**

In this project we tested all possible combinations for the calculator like addition of two digits, three digits, subtraction of positive integers, and subtraction of negative numbers, addition of one positive and one negative number, positive multiplication, negative multiplication. This is performed along with opening of chrome browser as well as closing of chrome browser. All these tasks is performed with help of Selenium webdriver and installing additional TestNG Library.

**4.0 TESTING STRATEGY**

In this project we tested all possible combinations for the calculator like addition of two digits, three digits, subtraction of positive integers, and subtraction of negative numbers, addition of one positive and one negative number, positive multiplication, negative multiplication. This is performed along with opening of chrome browser as well as closing of chrome browser. All these tasks is performed with help of Selenium webdriver and installing additional TestNG Library.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approach** | **Type of Testing** | **Manual Testing** | **Automated Testing on Device** | **Tools/APIs/Libraries** |
| **Using Environment** |
| Standard Testing  (Functional Testing) | Unit Testing | Yes | Yes | 1. Junit   (unit testing framework)   1. Selenium WebDriver/IDE   (web application automation testing framework)   1. TestNG Library |
| Integration Testing | Yes | Yes |
| Regression Testing | No | No |
| Compatibility Testing | No | No |

**4.1 Unit Testing**

**Definition:**

A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system.

**Participants:**

Dhiraj Shelke, Pranav Gavali, Komal Naphade

**Methodology:**

This Test will be conducated by testing each and every method in the code. Project members wrote test scripts for the unit testing. Every method would be executed one by one and this will be the sequence of events of Unit Testing. The testing activity take place with the help of Eclipse IDE and Selenium.

|  |  |
| --- | --- |
| **MODULE/FUNCTIONALITY NAME:** | Set Up Browser |
| **UNIT/CLASS:** | setUpBrowser |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST UNIT/CLASS** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| 1 | setUpBrowser() | Chrome Browser Opens up | Chrome Browser installed | Gather chromedriver location path | chromedriver\_win32\\chromedriver.exe | Chrome Browser Opens | Following Code Executes | Chrome Browser Opens | Pass |

|  |  |
| --- | --- |
| **MODULE/FUNCTIONALITY NAME:** | Two Digit Addition |
| **UNIT/CLASS:** | twoDigitAddition |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST UNIT/CLASS** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| 2 | twoDigitAddition | Addition Performed of two digits | Need valid integers to perform operation | 1.Find element “1”  2. Find element “+”  3. Find element “2”  4. Find element “eval” | 1+2= | 3 | Perform addition | 3 | Pass |

|  |  |
| --- | --- |
| **MODULE/FUNCTIONALITY NAME:** | Three Digit Addition |
| **UNIT/CLASS:** | threeDigitAddition |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST UNIT/CLASS** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| 3 | threeDigitAddition | Addition Performed of three digits | Need valid integers to perform operation | 1.Find element “1”  2. Find element “+”  3. Find element “9”  4. Find element “+”  5. Find element “4”  6. Find element “eval” | 1+9+4= | 14 | Perform addition | 14 | Pass |

|  |  |
| --- | --- |
| **MODULE/FUNCTIONALITY NAME:** | Subtraction of positive integers |
| **UNIT/CLASS:** | SubtractionOfPositiveIntegers |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST UNIT/CLASS** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| 4 | SubtractionOfPositiveIntegers() | Subtraction Performed of positive integers | Need valid integers to perform operation | 1.Find element “1”  2. Find element “0”  3. Find element “0”  4. Find element “-”  5. Find element “5”  6. Find element “0”  7. Find element “eval” | 100-50= | 50 | Perform subtraction | 50 | Pass |

|  |  |
| --- | --- |
| **MODULE/FUNCTIONALITY NAME:** | Subtraction of negative integers |
| **UNIT/CLASS:** | SubtractionOfNegativeIntegers |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST UNIT/CLASS** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| 5 | SubtractionOfNegativeIntegers() | Subtraction Performed of negative integers | Need valid integers to perform operation | 1.Find element “-”  2. Find element “1”  3. Find element “0”  4. Find element “-”  5. Find element “5”  6. Find element “0”  7. Find element “eval” | -10-50= | -60 | Perform subtraction | 50 | Pass |

|  |  |
| --- | --- |
| **MODULE/FUNCTIONALITY NAME:** | Positive Negative addition |
| **UNIT/CLASS:** | PositiveNegativeAddition() |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST UNIT/CLASS** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| 6 | PositiveNegativeAddition() | Addition Performed of positive and negative integers | Need valid integers to perform operation | 1.Find element “-”  2. Find element “1”  3. Find element “0”  4. Find element “+”  5. Find element “1”  6. Find element “0”  7. Find element “0”  8. Find element “eval” | -10+100= | 90 | Perform Addition | 90 | Pass |

|  |  |
| --- | --- |
| **MODULE/FUNCTIONALITY NAME:** | Positive multiplication |
| **UNIT/CLASS:** | PositiveMultiplication() |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST UNIT/CLASS** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| 7 | PositiveMultiplication() | Multiplication Performed of positive integers | Need valid integers to perform operation | 1.Find element “1”  2. Find element “0”  3. Find element “\*”  4. Find element “5”  5. Find element “0”  6. Find element “eval” | 10\*50= | 500 | Perform multiplication | 500 | Pass |

|  |  |
| --- | --- |
| **MODULE/FUNCTIONALITY NAME:** | Negative multiplication |
| **UNIT/CLASS:** | NegativeMultiplication() |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST UNIT/CLASS** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| 8 | NegativeMultiplication() | Multiplication Performed of Negative integers | Need valid integers to perform operation | 1.Find element “-”  2. Find element “1”  3. Find element “0”  4. Find element “\*”  5. Find element “5”  6. Find element “0”  7. Find element “eval” | -10\*50= | -500 | Perform multiplication | -500 | Pass |

**4.2 Integration Testing**

**Definition**: Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group.

**Participants:**

Dhiraj Shelke, Pranav Gavali, Komal Naphade

**Methodology:**

This Test will be conducated by testing whole code. Project members wrote test scripts for the Integration testing. Whole code would be executed at once and this will be the sequence of events of Integration Testing. The testing activity take place with the help of Eclipse IDE and Selenium.

|  |  |
| --- | --- |
| **PROJECT NAME:** | BrowserLaunch\_TestNG |
| **MODULE/FUNCTIONALITY:** | Open Calculator Application |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/2021 |
| **DATE OF REVIEW:** | 15/12/2021 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST SCENARIO** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| Open Calculator | VERIFY THE Opening of Calculator | Enter valid inputs | Need valid integers to perform operation | 1. Launch Chrome Browser  2. Open Calculator  3. Perform Calculations | Given inputs from Program | Successful Calculations | Calculations Successful and Chrome closes | Successful Calculations | Passed |

**4.3 User Acceptance Testing**

**Definition:**

The purpose of acceptance test is to confirm that the system is ready for operational use. During acceptance test, end-users (customers) of the system compare the system to its initial requirements.

**Participants:** Dhiraj Shelke, Pranav Gavali, Komal Naphade

**Methodology:** This Test will be conducated by testing whole code. Project members wrote test scripts for the user acceptance testing. Whole code would be executed at once and this will be the sequence of events of User Acceptance Testing. The testing activity take place with the help of Eclipse IDE and Selenium and User will tell the Satisfaction level.

|  |  |
| --- | --- |
| **PROJECT NAME:** | BrowserLaunch\_TestNG |
| **MODULE/FUNCTIONALITY:** | Calculator Web Application |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Description** | **Step #** | **Test Steps** | **Exp. Results** | **#Business Req. Covered** | **#Functional Req. Covered** |
| 1 | User Tries to open the application | 1 | Run the code in eclipse IDE | Code Runs | Calculator Opens | Code Executes |
| 2 | User performs calculation | 2 | Give Input by clicking icons | Gets Result in Textbox | Calculator Works | Calculator methods Executes without error |

**4.4 Regression Testing**

**Definition:**

Regression testing is the selective retesting of a system or component to verify that

Modifications have not caused unintended effects and that the system or component stillworks as specified in the requirements.

**Participants:** Dhiraj Shelke, Pranav Gavali, Komal Naphade

**Methodology:** This Test will be conducated by changing a method in the code. Project members wrote test scripts for the unit testing. Whole code would be executed at once and this will be the sequence of events of Regression Testing. The testing activity take place with the help of Eclipse IDE and Selenium.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **TEST SCENARIO** | **TEST CASE** | **PRE-CONDITION** | **TEST STEPS** | **TEST DATA** | **EXPECTED RESULT** | **POST CONDITION** | **ACTUAL RESULT** | **STATUS**  **(PASS/FAIL)** |
| Open Calculator | Verify The Changes in PositiveMultiplication() Method | Enter valid inputs in PositiveMultiplication() | Need valid integers to perform operation | 1. Launch Chrome Browser  2. Open Calculator  3. Perform Calculations | Given inputs from Program  10\*60= | Successful Calculations and give 600 | Calculations Successful and Chrome closes | Successful Calculations | Passed |

**4.5 Beta Testing**

**Definition:** Beta testing is a type of user acceptance testing where the product team gives a nearly finished product to a group of target users to Test

**Participants:** Dhiraj Shelke, Pranav Gavali, Komal Naphade

**Methodology:** This Test will be conducated by testing whole code. Project members wrote test scripts for the user acceptance testing. Whole code would be executed at once and this will be the sequence of events of Beta Testing. The testing activity take place with the help of Eclipse IDE and Selenium and User will tell the Satisfaction level.

|  |  |
| --- | --- |
| **PROJECT NAME:** | BrowserLaunch\_TestNG |
| **MODULE/FUNCTIONALITY:** | Calculator Web Application |
| **CREATED BY:** | Dhiraj Shelke |
| **DATE OF CREATION:** | 15/12/21 |
| **DATE OF REVIEW:** | 15/12/21 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Test Description** | **Step #** | **Test Steps** | **Exp. Results** | **#Business Req. Covered** | **#Functional Req. Covered** |
| 1 | User Tries to open the application | 1 | Run the code in eclipse IDE | Code Runs | Calculator Opens | Code Executes |
| 2 | User performs calculation | 2 | Give Input by clicking icons | Gets Result in Textbox | Calculator Works | Calculator methods Executes without error |

**4.6 Compatibility Testing:**

**Definition:** Compatibility Testing is a type of Software testing to check whether your software is capable of running on different hardware, operating systems, applications , network environments or Mobile devices

**Participants:** Dhiraj Shelke, Pranav Gavali, Komal Naphade

**Methodology:** This Test will be conducated by running the code in other envirnments. Project members wrote test scripts for the unit testing. Whole code would be executed at once and this will be the sequence of events of Compatibility Testing. The testing activity take place with the help of Eclipse IDE and Selenium.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Hardware Specification**  **(Processor/Clock Speed/RAM)** | **Operating System** | **Telecom Network** | **Browsing Application** | **Interactive Testing**  **(PASS/FAIL)** | **Comments** |
| 1 | 2gb ram | Windows | Jio | Chrome | Pass | None |
| 2 | 4gb ram | Windows | Vi | Chrome | Pass | None |

**5.0 HARDWARE REQUIREMENTS**

Working System

**6.0 ENVIRONMENT REQUIREMENTS**

Eclipse IDE

Chrome Driver

Selenium Webdriver

Selenium Standalone Server

**7.0 TEST SCHEDULE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Start Date** | **Finish Date** | **Effort Estimation** | **Comments** |
| Test Planning | 14/12/21 | 15/12/21 | 1d |  |
| Review Requirements documents |  |  | 2 d |  |
| Create initial test estimates |  |  | 1 d |  |
| Functional testing – Iteration 2 | 15/12/21 | 15/12/21 | 1d |  |
| System testing | 15/12/21 | 15/12/21 | 1d |  |
| Regression testing | 15/12/21 | 15/12/21 | 1d |  |
| User Acceptance Testing | 15/12/21 | 15/12/21 | 1d |  |

**8.0 FEATURES TO BE TESTED**

Calculations

Button Clicks

**9.0 FEATURES NOT TO BE TESTED**

Chrome Driver

Eclipse IDE

**10.0 RESOURCES/ROLES & RESPONSIBILITIES**

Required Resources: Eclipse IDE

Responsibility: Create Test for each Operation

**11.0 SCHEDULES**

**Major Deliverables**

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **For** | **Date / Milestone** |
| Test Plan | Project Manager | 16/12/21 |
| Test Result Report | Project Manager | 16/12/21 |
| Test Execution Report | Project Manager | 16/12/21 |

**12.0 SIGNIFICANTLY IMPACTED DEPARTMENTS (SIDs)**

Testing, Web Development

**13.0 DEPENDENCIES**

TestNG Library should be installed.

**14.0 RISKS/ASSUMPTIONS**

The Program might give unexpected output which totallt depends on the user inputs

**15.0 TOOLS**

List the Automation tools you are going to use.

Selenium

**16.0 APPROVALS**

Specify the names and titles of all persons who must approve this plan. Provide space for the signatures and dates.

Name (In Capital Letters) Signature Date

1.

2.

3.

4.