Use Case Document Example

Introduction

This Use Case document has been developed to allow grade students to find an easy program to test their arithmetic skills. This new program will not replace their existing handwritten methods, but allow them to easy cross check their answers against a known calculation. This Math Calculator program was fully developed using the business requirements gathered via a survey of the top 100 grade teachers in the country.

Use Case

Name of Use Case:	Math Calculations		
Created By:	Robbie Dodenbier	**Last Updated By:**	R.Dodenbier
Date Created:	11/8/2017	**Last Revision Date:**	11/8/2017
Description:	Student uses the calculator program to test their arithmetic skills using the addition, subtraction, multiplication, division, and power math operations. After the student uses each math operator, the system will output the correct answer of their desired calculation.		
Actors:	Student		
Preconditions:	1. Student must launch program 2. Student must enter a valid integer value 3. Student must enter a valid math operator		
Postconditions:	1. On successful completion, a single output string will show both integer values, the math operator, and the result of the calculation. 2. Student will land on the main menu screen		
Flow:	1. Student will launch main program 2. Student will be presented with a main menu 3. Student must select calculator program from menu 4. Student will be asked to enter their first integer value 5. Student will enter in any integer value 6. Student will be asked to enter in a math operator based on a presented list of valid operators 7. Student will enter in a valid math operator 8. Student will be asked to enter their second integer value 9. System will output the entire math equation including the calculated result		

	10. Main menu will be presented 11. Student can start a new calculation or exit program
Alternative Flows:	1. No other flows exists
Exceptions:	1. In Step #4 and Step #8, if student enters in a non-integer value 1. Error message will appear and ask for a valid number 2. Program will exit if second non-integer value is entered 2. In Step #6, if student enters in a non-math operator 1. Error message will appear and ask for a valid math operator 2. Program will exit if second non-math operator is entered
Requirements:	The following requirements must be met before execution of the use case: 1. Execute latest version of the program 2. Valid integer entered on Step #4 and Step #8 3. Valid math operator entered on Step #6