Software Requirement Specification

1. Provide a few sentences describing how you think the application would work in your own words.

This application is a calculator program that is intended to help students test their knowledge of simple math equations by allowing them to guess the result rather than giving them the answer.

The program will have a user interface with fields for entering numbers, selecting an operation, and entering an expected result. The student will be able to perform addition, subtraction, multiplication, and division. The program will then display if the student had the correct answer. The program is designed to be simple and easy to use for elementary students. The application will be running on Windows and will have no tracking or usage logging of any kind.

1. Provide a drawing of a guess of what the user interface would look like in Paint or Word or even scanned in from a napkin.

The user interface would consist of a simple layout with 2 fields for entering numbers, a drop-down menu for selecting the operation (addition, subtraction, multiplication, and division), one field for entering an expected result and a "calculate" button to execute the calculation test. There would also be a message box to display if the student had the correct answer or not.

[Number 1] [Number 2]

[Drop-down menu for operation]

[Expected Result]

[Calculate] [Message box]

1. Provide a use case diagram indicating the features required. The Use Case Diagram Template is supplied if you don't have software other then word for this.

A use case diagram is a type of behavioral diagram defined in the Unified Modeling Language (UML) that shows the different interactions between users (actors) and a system (use cases). Here is an example of a use case diagram representing the features required for the calculator program

Actor: The student who will be using the calculator program.

Use Case: Perform Calculation: This represents the main functionality of the program, which is to allow the student to perform mathematical calculations by providing input numbers and choosing the operation.

Use Case: Input Data: This represents the functionality of the program which allows the student to enter numbers and expected result.

Text

Description automatically generated with medium confidence

1. Provide a use case detail for the first feature, include the actor/steps (A template has also been provided in Word)
2. Provide a user story for the second feature, include the 3-part user story sentence and acceptance criteria. Scenario's will be looked at later

A user story is a simple, brief statement that describes a specific feature or functionality from the perspective of an end user. It's typically written in the form of a three-part sentence: "As a [user], I want [functionality], so that [goal]."

Here is an example of a user story for the second feature of the calculator program:

User Story: As a student, I want to be able to input numbers and expected result, so that I can perform calculations and check if my expected result is correct.

Acceptance Criteria:

* The student can input numbers in the designated fields.
* The student can input an expected result in the designated field.
* The program displays a message indicating whether the student's expected result is correct or not.
* Invalid input (non-numeric characters) are handled and the student is prompted to re-enter the numbers.
* The input fields are limited to accept only numeric inputs
* The expected result field is limited to accept only numeric inputs

The acceptance criteria are a set of specific, measurable, testable and verifiable requirements that define what the user story should do. They are used to ensure that the user story is complete and that the developer understands the requirements of the feature.