# User Story

Create a .Net Console application that takes in a single number, n, and returns the n-th number in the Fibonacci sequence.

Confirmed requirements:

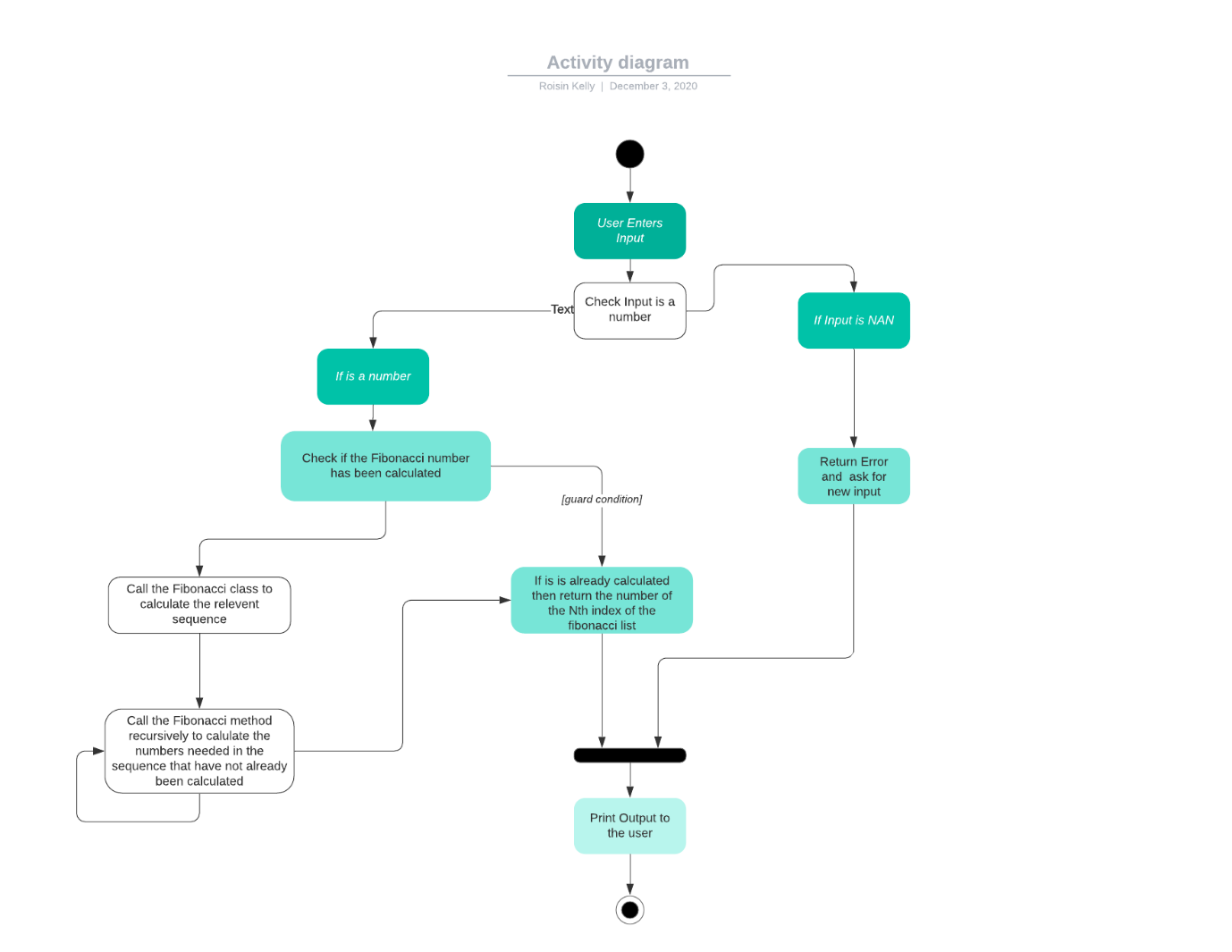
CMD input

Given that the first number of the Fibonacci Sequence is 0, the input is required to return 0 is 0

## Presumptions made

1. User may or may not enter a number eg Eight instead of 8
2. User may wish to use the process multiple times before closing the app

# Flow Diagram



# Design Decisions

Visual Studio 2019 was used to create the project as that is the version I have on my laptop.

Console App (.Net Core) was used to create the project Named Fibonacci.

In the Main Class the Main method Declares and Creates the Class Processing, the reason for this is to minimise the code in the main method.

UnitTestingClass was created to test the functionality of the Fibonacci Class

* The class has 1 public variable and 2 private Lists of ints
  + The public variable is a measure of if the test has passed (bool)
  + Private List of Ints fib is manually gets its values in the class and the values are from <https://www.mathsisfun.com/numbers/fibonacci-sequence.html>
  + Private list of ints fibReturned is used to get the values of the fibbonacci sequence calculated in the
* Constructor Method is the only Public method and calls all the other private methods
* The Private method Create list is used to manually add the values to the fib list with values from <https://www.mathsisfun.com/numbers/fibonacci-sequence.html>
* The Private method testClass calls the Fibonacci class and gets the Fibbonacci Sequence that was calculate
* The Private method compare lists compares the 2 lists to make sure the right values were returned
  + If the lists are not the same then the Public variable passed is set to false

The Processing Class has only one public method which is the constructor method, the reason for this is this class acts as a trigger for the program to interact with the user and the Fibonacci Class.

* The Class has 2 private variables,
  + A list of Ints to store the Fibonacci Sequence the reason for this is in the event the user would like to enter multiple numbers
  + An int to store the user input
* The Start Processing method,
  1. Adds the 1st two entries of the Fibonacci Sequence to the list
  2. Calls the private method getNum
  3. Checks if the nth number entered has not already been calculated
  4. calls the private method fibOutput if it has not been calculated
  5. Prints out the number to the user and asks if they would like to enter another number
  6. If the user enters a number or Yes or No the above steps 3-5 are repeated
  + The reason this is the only public method is to protect the other methods from outside interference and their individual functionality is listed below
* Private getNum method
  1. This method asks the user for a number input
  2. Checks if it is a numerical value
  3. If it is Not A Number it will repeat the previous steps until the user enters a valid number
  4. Where a negative number is entered it is inverted as the negative sequence of the Fibbonacci Sequence mirrors the positive sequence.
  + This method is private to protect it from outside interference
* Private fibOutput method
  + Declares & creates the Fibonacci Class
  + Gets the Fibonnacci Sequence to the nth number returned

Fibonacci Class

* The Class has 1 private List of integers FibonacciSequence for entering the Fibbonacci Sqeuence in order
* There are 3 Public methods as listed below
* Constructor method which gives the FibonacciSequence its initial value
* Public CalculateFibbonacci takes an input of num
  + This method is a recursive method which means it calls itself until the length of the FibbonacciSequence is equal to the num value
  + Each time the method calls itself it reduces the num value by 1
  + Once the FibbonacciSequence is equal to the num value the next number in the sequence is calculated by adding the last and second last numbers already in the sequence together and is then added to the list
  + The method then goes up a level and repeats the last step this continues until the original method call completes the above step
* The public method getFib returns the value of the FibbonacciSequence list to the class it was called from.

# Limitations of the program

The main limitation of the program is related to the max integer value which is achieved after the 55th number in the Fibbonacci Sequence in other words the 56th number of the sequence returns a negative number as its value is above the Max Int Value