Brayden Cantrill

Abstract

This document explains the appropriate design and technical documentation of the NetworkArithmeticGame Application.

documentation

**NetworkArithmeticGame Application**

Table of Contents

[**Requirements:** 3](#_Toc524796883)

[**Purpose:** 3](#_Toc524796884)

[**Processes:** 3](#_Toc524796885)

[**Features:** 4](#_Toc524796886)

[**UML Diagrams:** 4](#_Toc524796887)

[**FlowCharts:** 5](#_Toc524796888)

[**TOE Charts:** 6](#_Toc524796889)

[**INSTRUCTOR FORM:** 6](#_Toc524796890)

[**BUTTONS:** 6](#_Toc524796891)

[**LABLES:** 7](#_Toc524796892)

[**TEXTBOXES, LISTBOXES & DATAGRIDVIEW:** 7](#_Toc524796893)

[**STUDENT FORM:** 8](#_Toc524796894)

[**BUTTONS:** 8](#_Toc524796895)

[**LABLES:** 8](#_Toc524796896)

[**TEXTBOX & LISTBOX:** 8](#_Toc524796897)

[**Source Code:** 9](#_Toc524796898)

[**Testing:** 10](#_Toc524796899)

[**Testing Performed:** 10](#_Toc524796900)

[**APPLICATION STARTS:** 10](#_Toc524796901)

[**SENDING:** 11](#_Toc524796902)

[**STUDENT SUBMIT CORRECT:** 11](#_Toc524796903)

[**STUDENT SUBMIT INCORRECT:** 12](#_Toc524796904)

[**ARRAY OF QUESTIONS SORTING:** 12](#_Toc524796905)

[**BINARY TREE DISPLAY/SAVE:** 14](#_Toc524796906)

[**User Manual:** 16](#_Toc524796907)

[**Introduction:** 16](#_Toc524796908)

[**System requirements:** 16](#_Toc524796909)

[**Application Functionality:** 17](#_Toc524796910)

[**MAINFORM:** 17](#_Toc524796911)

[**1.** **OPEN FILE BUTTON:** 17](#_Toc524796912)

[**2.** **CLASSROOM DATAGRIDVIEW:** 17](#_Toc524796913)

[**3.** **MINIMIZE & EXIT BUTTONS:** 18](#_Toc524796914)

[**4.** **SEARCH FUNCTIONALITY:** 18](#_Toc524796915)

[**5.** **SORT BUTTON:** 18](#_Toc524796916)

[**6.** **CLEAR BUTTON:** 18](#_Toc524796917)

[**7.** **SAVE BUTTON:** 18](#_Toc524796918)

[**8.** **RAF BUTTON:** 19](#_Toc524796919)

[**SORTFORM:** 20](#_Toc524796920)

[**1.** **SORTLIST DATAGRIDVIEW:** 20](#_Toc524796921)

[**2. MINIMIZE & EXIT BUTTONS:** 20](#_Toc524796922)

[**Sorting Algorithms:** 21](#_Toc524796923)

[**BUBBLE:** 21](#_Toc524796924)

[**INSERTION:** 21](#_Toc524796925)

[**SELECTION:** 21](#_Toc524796926)

[**Third-Party Reference:** 22](#_Toc524796927)

[**ATOMINEER:** 22](#_Toc524796928)

[**NEWTONSOFT.JSON**: 22](#_Toc524796929)

[**Communication:** 23](#_Toc524796930)

[**MANAGER COMMUNICATION EMAIL:** 23](#_Toc524796931)

[**Debugging Facilities:** 24](#_Toc524796932)

[**Breakpoints:** 24](#_Toc524796933)

[**Watches:** 24](#_Toc524796934)

[**Tracing:** 25](#_Toc524796935)

[**References:** 27](#_Toc524796936)

# **Requirements:**

## **Purpose:**

The purpose of this application is designed in assisting teachers and students within the area of mathematics and the creation of simple mathematical equations. Teachers are able to send questions to the student through network connections that allow both the Teacher and Student to send and receive specific responses while also permitting the teacher to view the results of the student’s answers as well as the questions that were asked.

## **Processes:**

The application must be able to allow users to interact with the user interface in a quick and efficient approach while also having the ability in sending and receiving results from both the Teacher and the Student as well as saving results displayed in list boxes to text files. Having the ability of sorting an array of questions and display results in certain orders is another process.

The primary processes within the application include:

* Allow the user to access the program connected to the network.
* Enter numbers into textboxes and send them to the student.
* Student receives the numbers and answers the question.
* Teacher has the ability in viewing the answers and the questions asked.

## **Features:**

* Application connects to the network connections.
* Send questions to the student.
* Student send either correct or incorrect answers back to the teacher.
* Teacher views the answers and question in an array of questions, linked list and binary tree.

# **UML Diagrams:**

## **MAIN FORM:**

Figure : Displays Main Form UML Diagram

The Main Form UML Diagram uses Node and NodeList classes to create and get the values from the main form linked list. The Main Form also uses the client and listener classes to receive the values from the Student Form. The Main Form UML Diagram uses the Form1 Class to link all the fields, properties, methods and events together.

## **STUDENT FORM:**



Figure : Displays Student Form UML Diagram

The Student UML Diagram uses the client and listener classes to receive the values from the Main Form. The Student UML Diagram uses the Form2 Class to link all the fields, properties, methods and events together.

# **TOE Charts:**

## **MAIN FORM:**

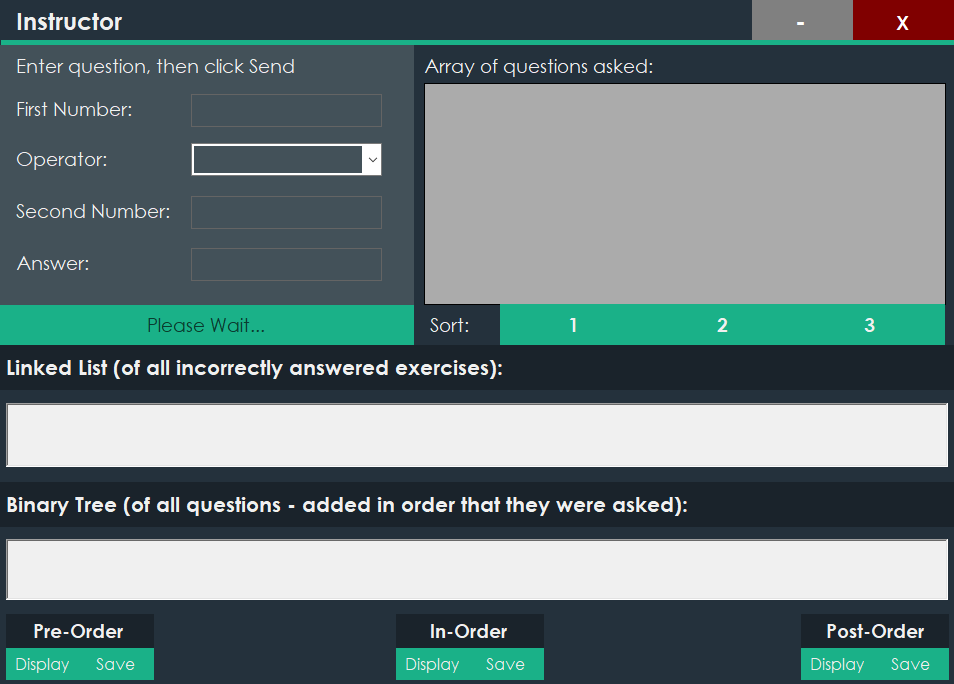


Figure : Overview of the Instructor Form

### **BUTTONS:**

|  |  |  |
| --- | --- | --- |
| Task | Object | Event |
| Minimize Application | btnMinimize | btnMinimize\_Click |
| Exit Application | btnExit | btnExit\_Click |
| Send | btnSend | btnSend\_Click |
| Sort 1 | btnSort1 | btnSort1\_Click |
| Sort 2 | btnSort2 | btnSort2\_Click |
| Sort 3 | btnSort3 | btnSort3\_Click |
| Pre Order Display | btnPreDisplay | btnPreDisplay\_Click |
| In Order Display | btnInDisplay | btnInDisplay\_Click |
| Post Order Display | btnPostDisplay | btnPostDisplay\_Click |
| Pre Order Save | btnPreSave | btnPreSave \_Click |
| In Order Save | btnInSave | btnInSave\_Click |
| Post Order Save | btnPostSave | btnPostSave\_Click |

### **LABLES:**

|  |  |  |
| --- | --- | --- |
| Task | Object | Event |
| Instructor | lblInstructor |  |
| Enter | lblEnter |  |
| First Number | lblFirstNumber |  |
| Operator | lblOperator |  |
| Second Number | lblSecondNumber |  |
| Answer | lblAnswer |  |
| Linked List | lblList |  |
| Binary Tree | lblBinaryTree |  |
| Pre-Order | lblPreOrder |  |
| In-Order | lblInOrder |  |
| Post-Order | lblPostOrder |  |

### **TEXTBOXES, LISTBOXES & DATAGRIDVIEW:**

|  |  |  |
| --- | --- | --- |
| Task | Object | Event |
| First Number TextBox | txtFirstNumber |  |
| Operator ComboBox | comboBoxOperator |  |
| Second Number TextBox | txtSecondNumber |  |
| Answer TextBox | txtAnswer |  |
| Linked List ListBox | lstLinkedList |  |
| Binary Tree ListBox | lstBinaryTree |  |
| Array Questions DataGridView | dgdArrayQuestions |  |

## **STUDENT FORM:**

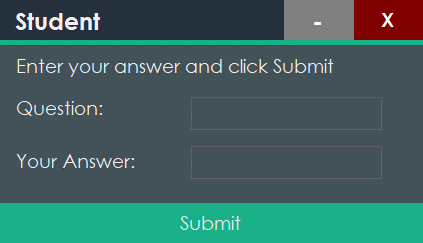


Figure : Overview of the Student Form

### **BUTTONS:**

|  |  |  |
| --- | --- | --- |
| Task | Object | Event |
| Minimize Application | btnMinimizeStudent | btnMinimizeStudent \_Click |
| Exit Application | btnExitStudent | btnExitStudent \_Click |

### **LABLES:**

|  |  |  |
| --- | --- | --- |
| Task | Object | Event |
| Student | lblStudent |  |
| Question | lblStudentQuestion |  |
| Your Answer | lblStudentAnswer |  |

### **TEXTBOX & LISTBOX:**

|  |  |  |
| --- | --- | --- |
| Task | Object | Event |
| Question ListBox | lstQuestion |  |
| Answer TextBox | txtStudentAnswer |  |

# **Source Code:**

# **Testing:**

## **Testing Performed:**

### **APPLICATION STARTS:**

Figure : Display that the application starts with no complications

### **SENDING:**

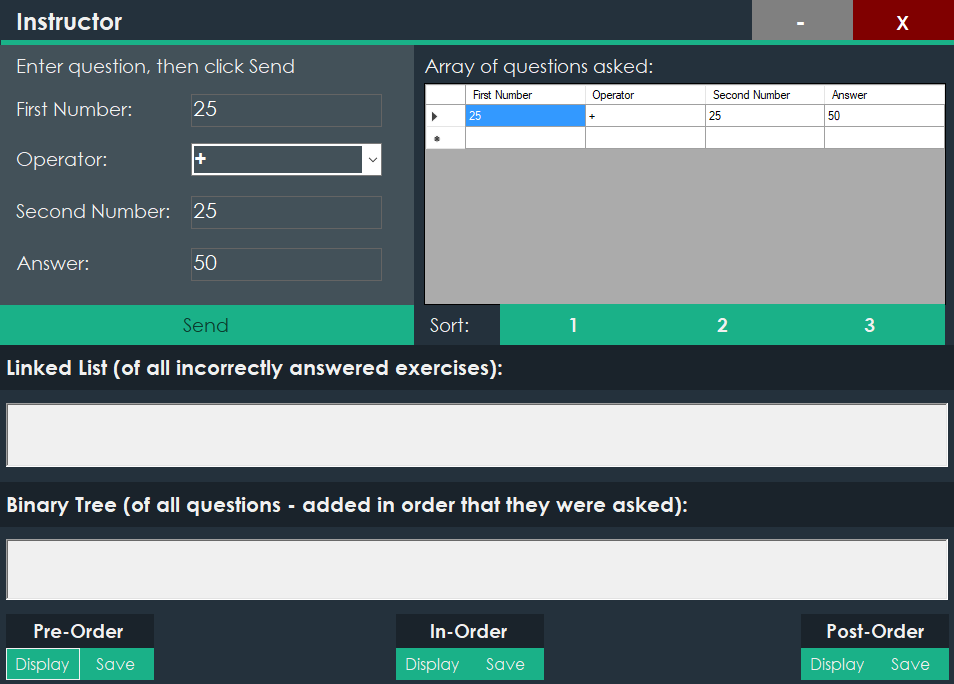
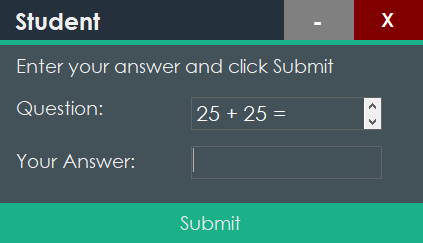


Figure : Successfully send and receives the values inputted across forms

### **STUDENT SUBMIT CORRECT:**

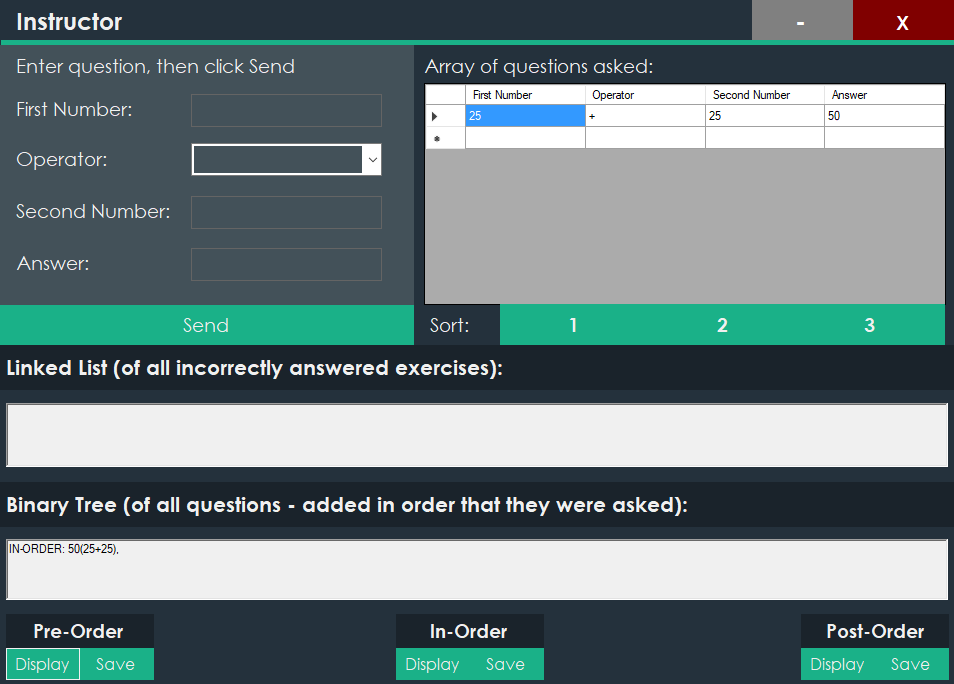
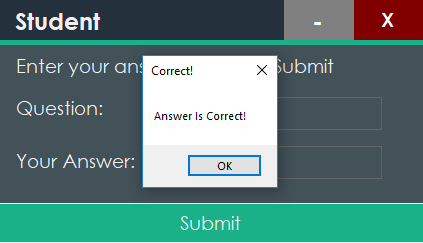


Figure : After getting the answer correct, the question and answer are added to the array of questions and the binary tree

### **STUDENT SUBMIT INCORRECT:**

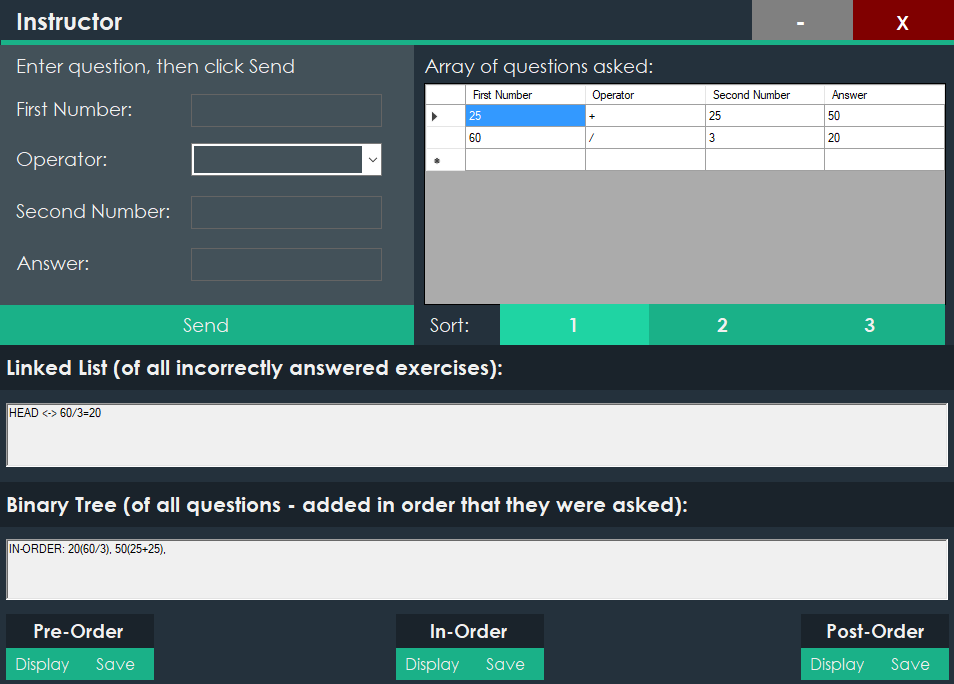
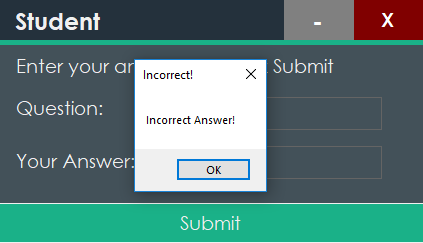


Figure : After getting answer incorrect, the question and answer are added to the array of questions, the linked list of all incorrectly answered questions and the binary tree

### **ARRAY OF QUESTIONS SORTING:**

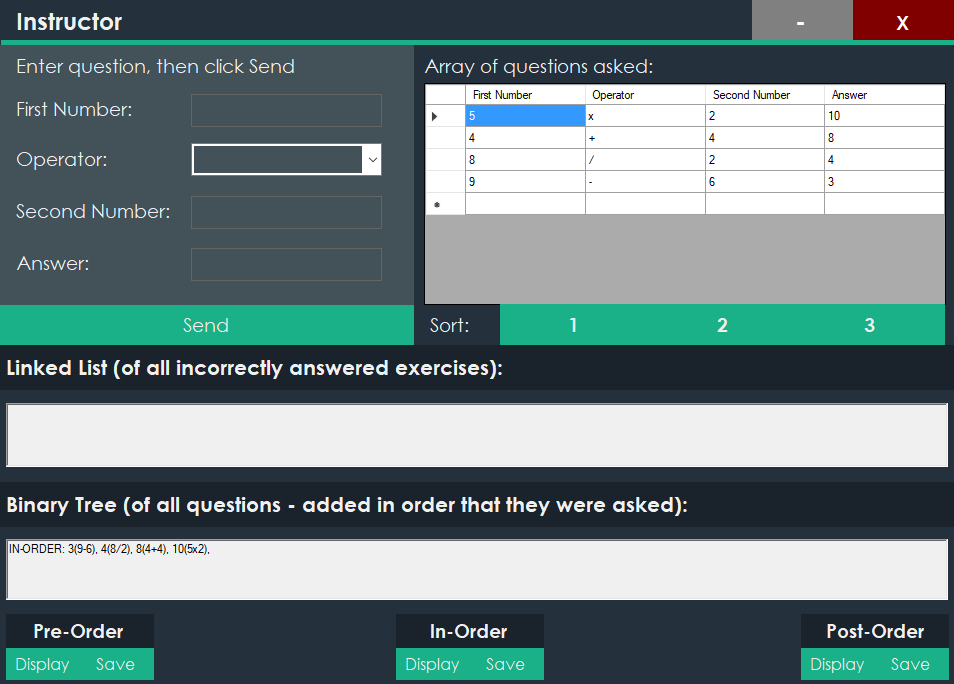


Figure : Overview of the array of questions with Sort 1 selected

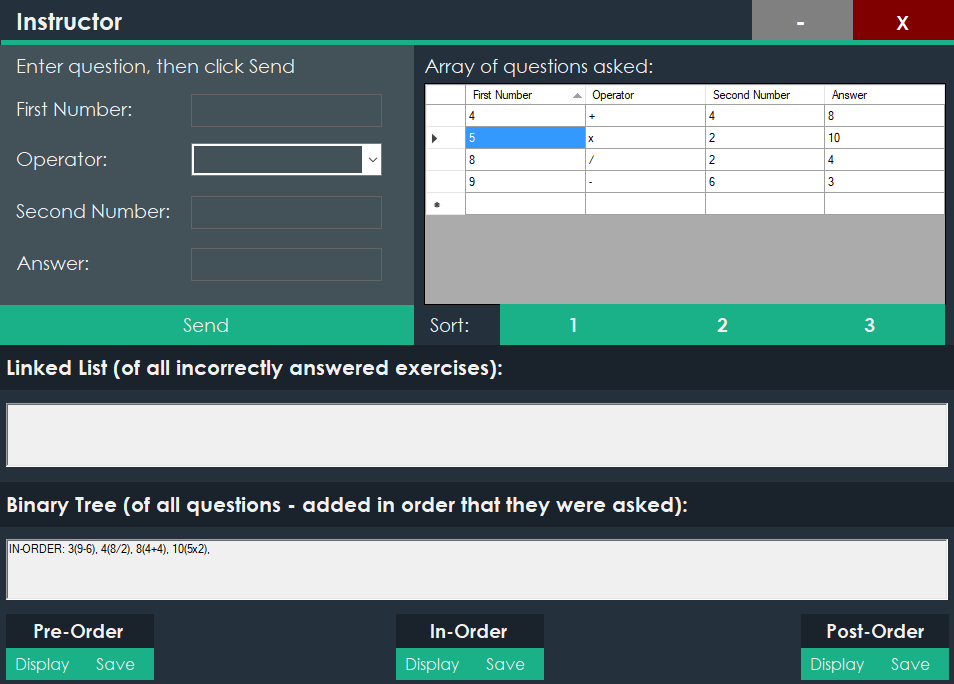
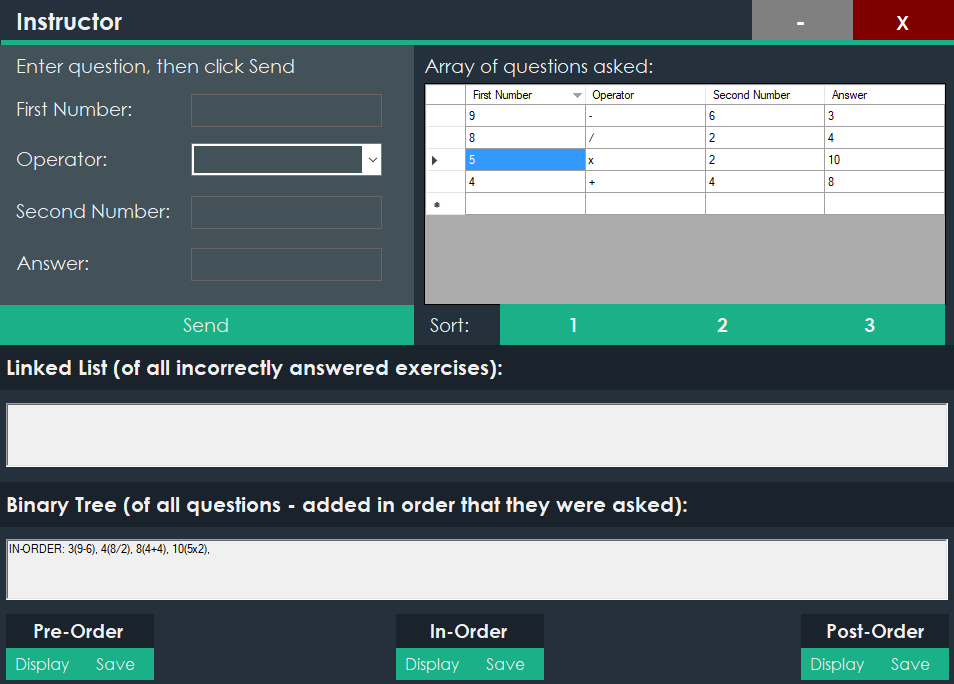


Figure : Sorts the numbers ascending

Figure : Sorts the numbers descending

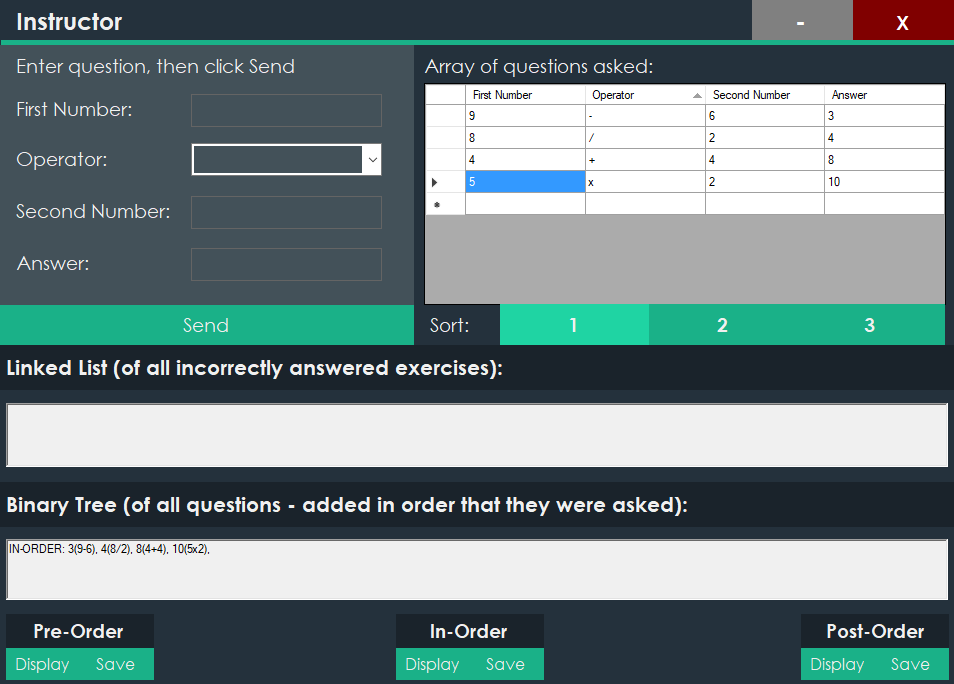


Figure : Sorts by Operator

### **BINARY TREE DISPLAY/SAVE:**

#### **IN-ORDER:**

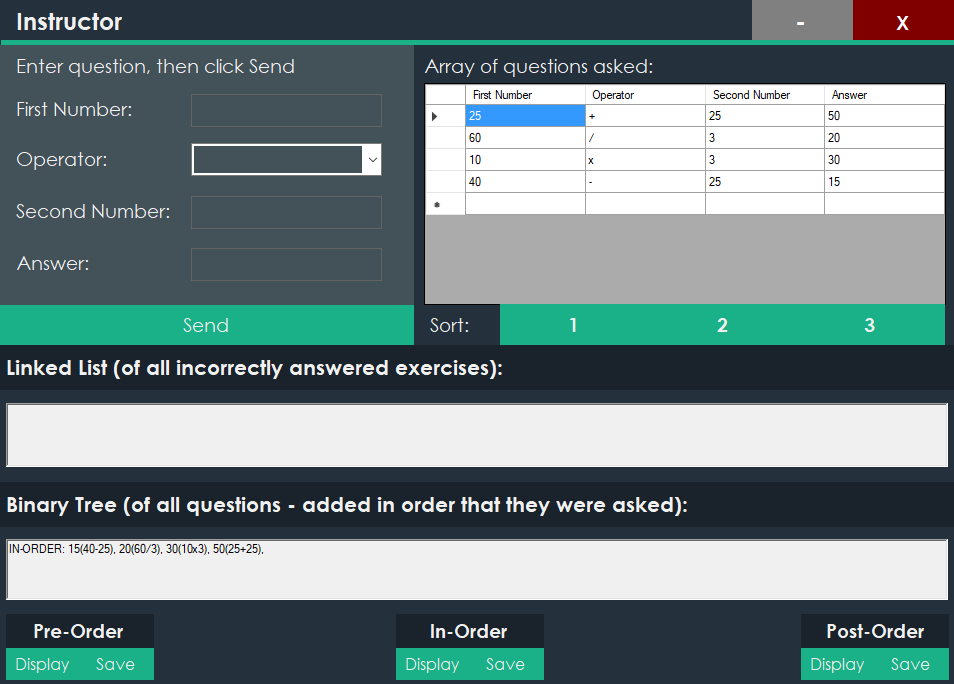
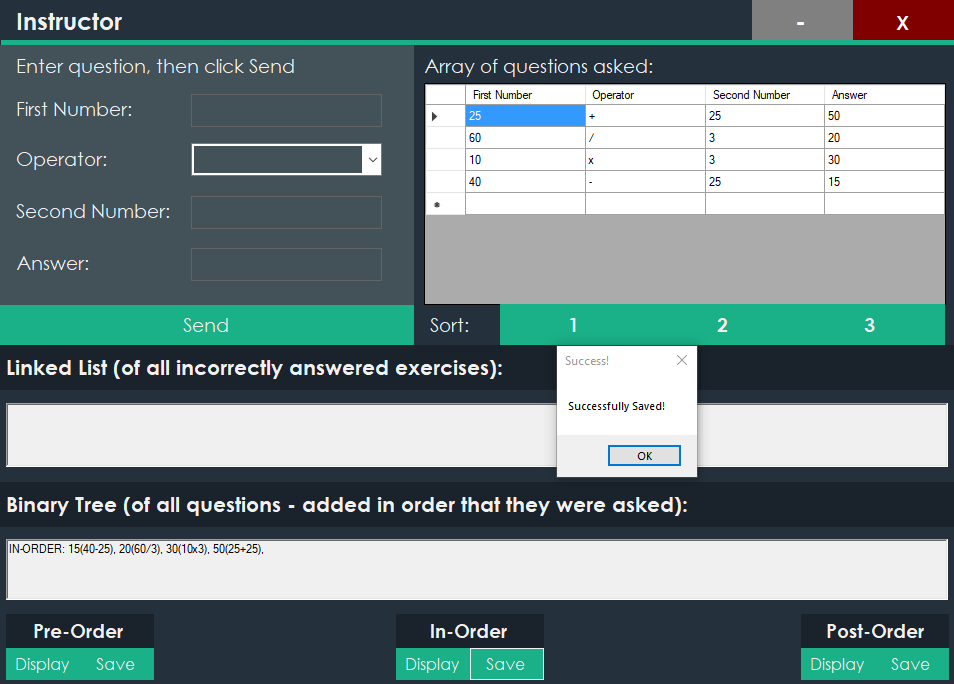


Figure : Displays In-Order Saved

Figure : Displays Binary Tree In-Order

#### **PRE-ORDER:**

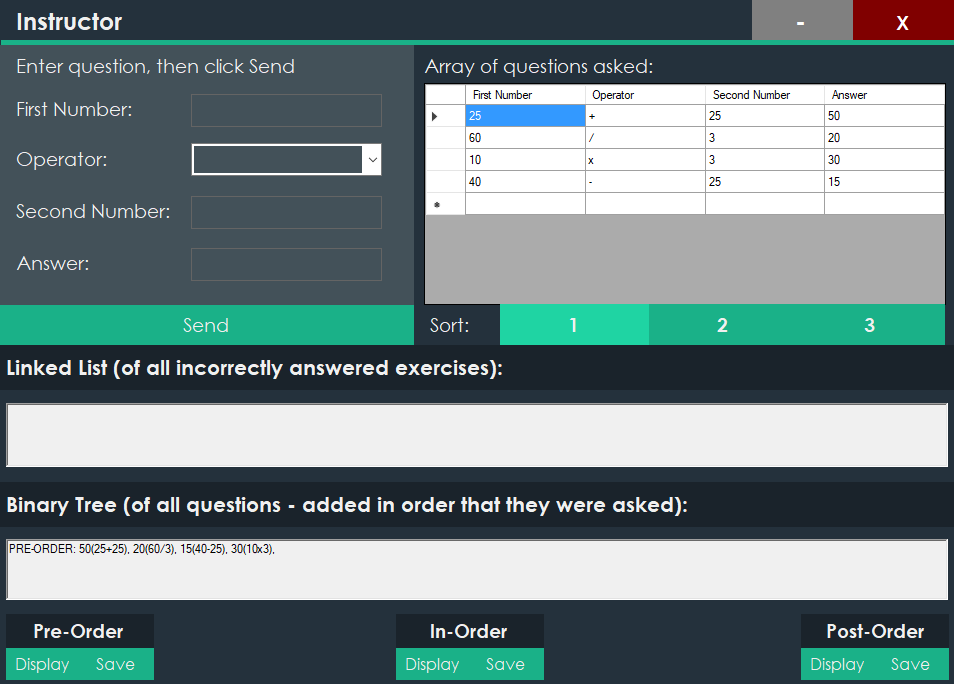
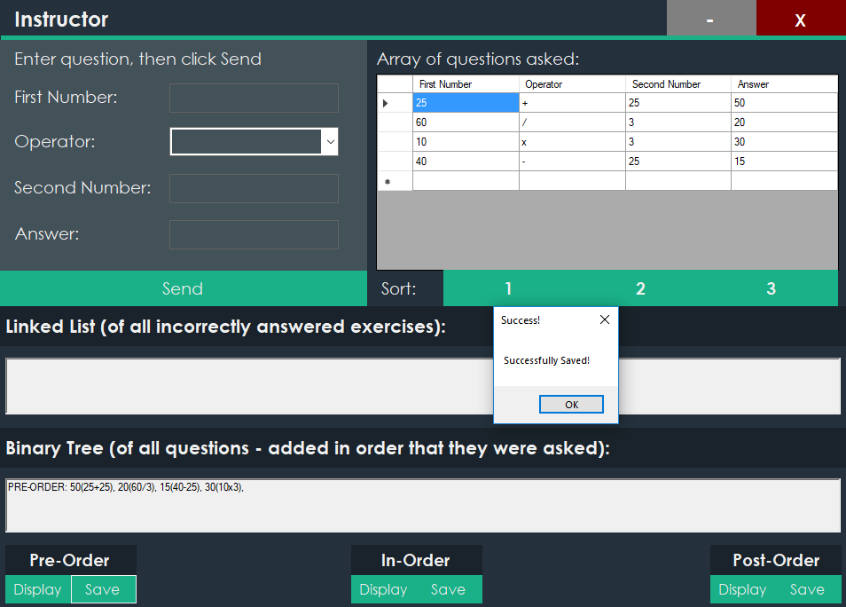


Figure : Displays Pre-Order Saved

Figure : Displays Binary Tree Pre-Order

#### **POST-ORDER:**

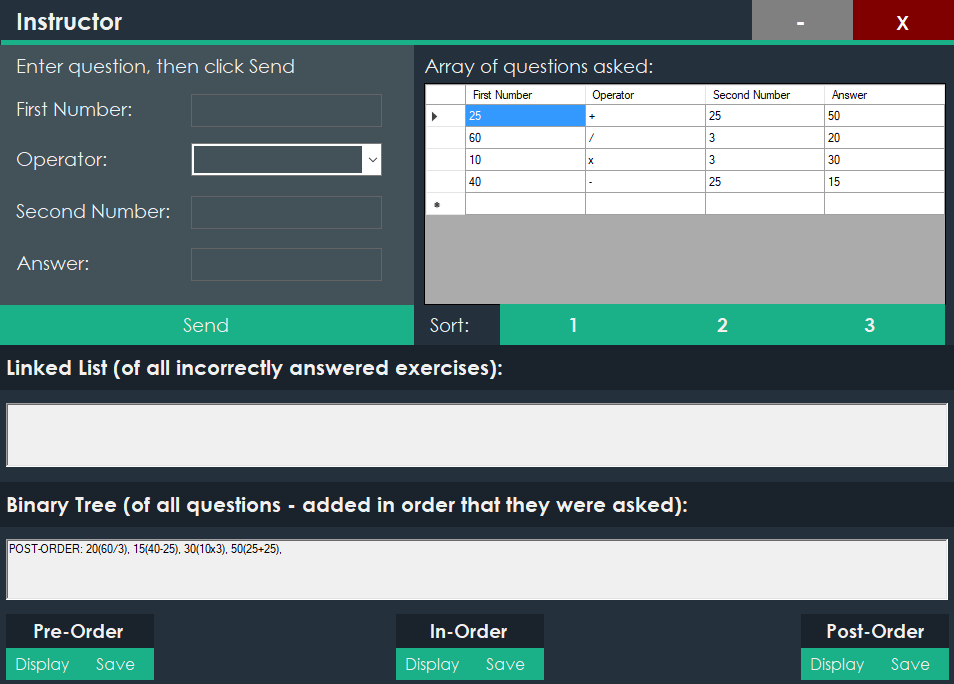
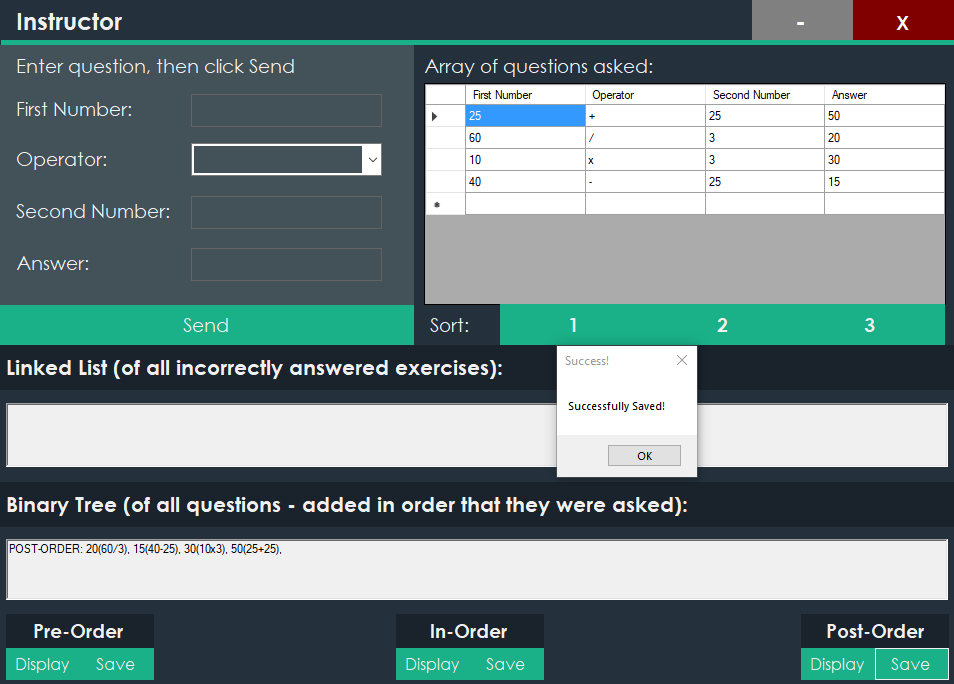


Figure : Displays Post-Order Saved

Figure : Displays Binary Tree Post-Order

# **User Manual:**

## **Introduction:**

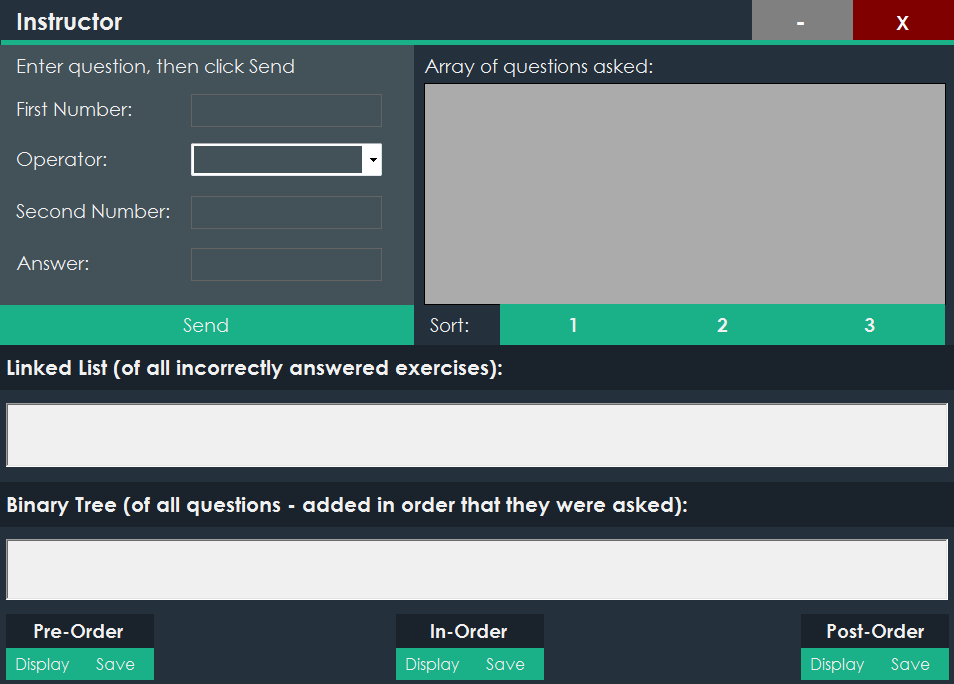
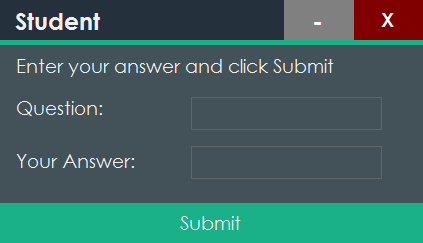
Welcome to the User Manual of the NetworkArithmeticGame application. This manual will provide vital information in regard to explaining the capability of the application and the functionality within the application that is to be utilized by the user. This manual is aimed specifically towards first time users, although knowledgeable users may find useful information in here as well.

## **System requirements:**

|  |  |
| --- | --- |
| Minimum | Recommended |
| **Operating System:** Windows 7 (x64)  **Processor:** 1.6 GHz Dual-Core 64-bit CPU  **Memory:** 2 GB RAM  **Graphics:** DirectX9 Compatible GPU with 2 GB Video RAM  **Hard Drive:**  5 GB Available Space | **Operating System:** Windows 7 (x64) or higher  **Processor:** 2.2 GHz Dual-Core 64-bit CPU or higher  **Memory:** 4 GB RAM or higher  **Graphics:** DirectX9 Compatible GPU with 2 GB Video RAM  **Hard Drive:**  10 GB Available Space |

## **Application Functionality:**

### **MAIN FORM & STUDENT FORM:**



**5**

Figure : Displays an overview of the Student Form & Main Form

**6**

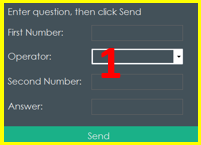
**2**

**1**

**3**

**4**

### **CALCULATION INPUT:**



The calculation input contains where the teacher is to input values equivalent to a mathematical equation towards the student form (6).

Figure : Displays the Calculation Input

### **ARRAY OF QUESTIONS:**

The array of questions is a DataGridView that contains questions and answers that have been asked towards the student.

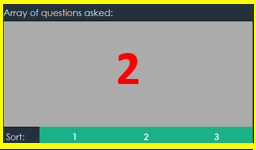


Figure : Displays the Array of Questions

### **LINKED LIST:**

The Linked List corresponds with student form (6). And contains all the incorrect answers from the question have been incorrectly answered by the students.



Figure : Displays the Linked List

### **BINARY TREE:**



The Binary Tree corresponds with the student from (6). The binary tree stores all of the questions and answers asked by the teacher

Figure : Displays the Binary Tree

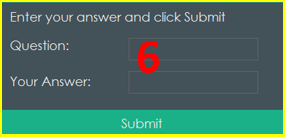
The minimize and exit buttons allow you to minimize the application and exit the application on click.

### **MINIMIZE & EXIT BUTTONS:**



Figure : Displays the Minimize & Exit Buttons

### **STUDENT FORM:**



The student form corresponds with most functionality in the main form including the Binary Tree (4) & Linked List (3). The student from also the receives the questions that have been asked by the teacher from the main form to answer.

# **Sorting Algorithms:**

## **BUBBLE:**

The bubble sort changes elements that are out of order until the entire list of times is in sequence.

The advantages & disadvantages of the bubble sort include the following:

**Advantages** – Simple Implementation Nature.

**Disadvantages** – Fails to fully operate with a list with an extensive number of items.

## **INSERTION:**

The insertion sort scans a list of items and inserts each item with an unordered sequence into its correct place.

The advantages & disadvantages of the insertion sort include the following:

**Advantages** – Is popular for its simplicity.

**Disadvantages** – Failure to perform with the same precision as other sorting algorithms and does not deal particularly well with a large list.

## **SELECTION:**

The selection sort sifts through a list of items and makes a selection based on its ordering and allocates to its proper position.

The advantages & disadvantages of the insertion sort include the following:

**Advantages** – Works extremely well dealing with a small list of items.

**Disadvantages** – Inefficiency to function with a large list of items.

# **VERSION CONTROL:**

# **Third-Party Reference:**

## **ATOMINEER:**

Atomineer is a third-party tool that is used for automatically documenting projects in visual studio and has been used throughout each individual file of the application.

## **NEWTONSOFT.JSON**:

Newtonsoft.Json is a third-party library tool that contains references and was used in the construction of the application being implemented within the networking connections.

# **Communication:**

## **MANAGER COMMUNICATION EMAIL:**

Hello Dave,

I would like to inform you that the requirements documentation has been completed which means that we will be proceeding forward in the development of the NetworkArithmeticGame that has been requested by you. This application will require the provisions of communicating between numerous applications to meet the specifications that have been listed.

If you have any further enquiries or questions about the requirements documentation or the development of the project, be sure to contact me.

Kind Regards,

Brayden Cantrill

# **Debugging Facilities:**

## **Breakpoints:**

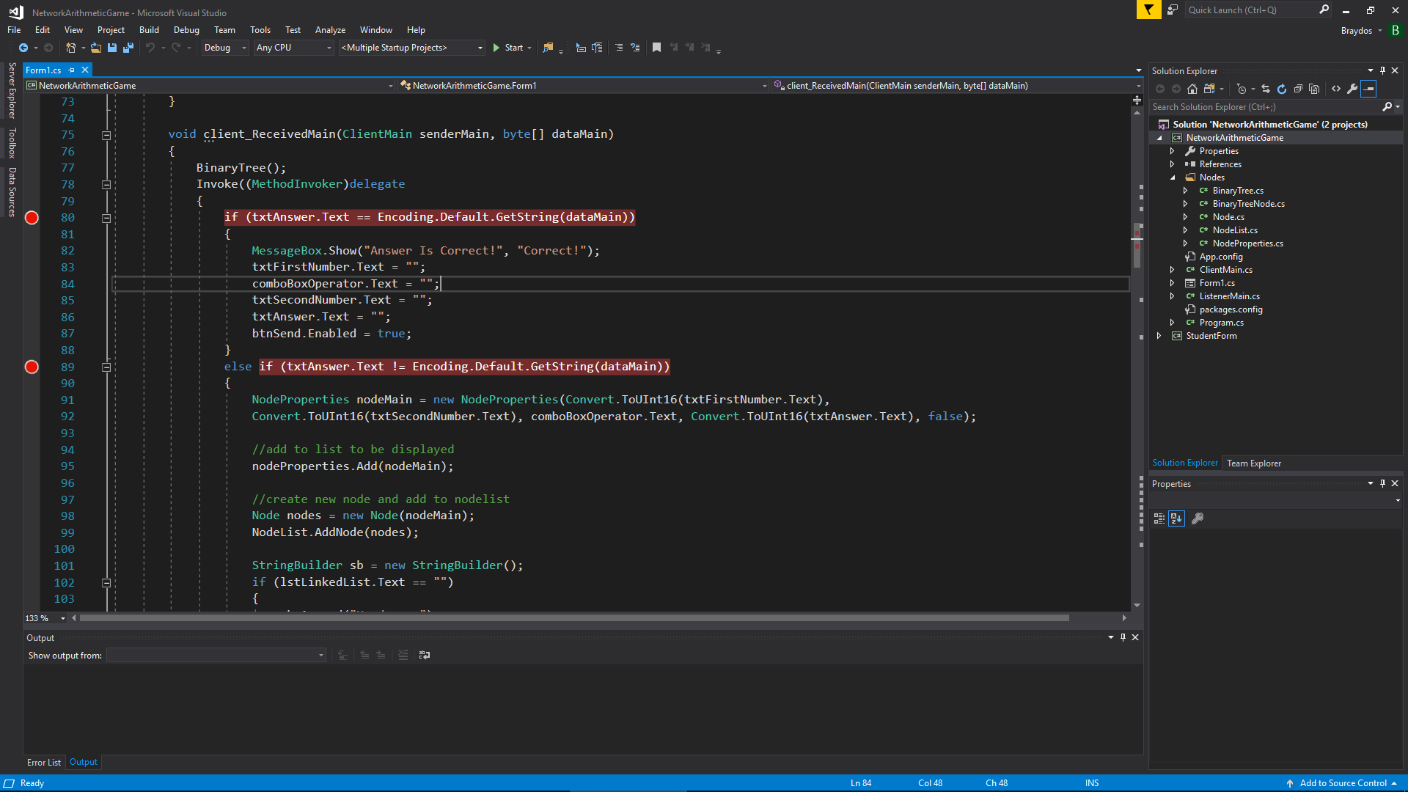


Figure : Show’s the breakpoints used in debugging the application

## **Watches:**

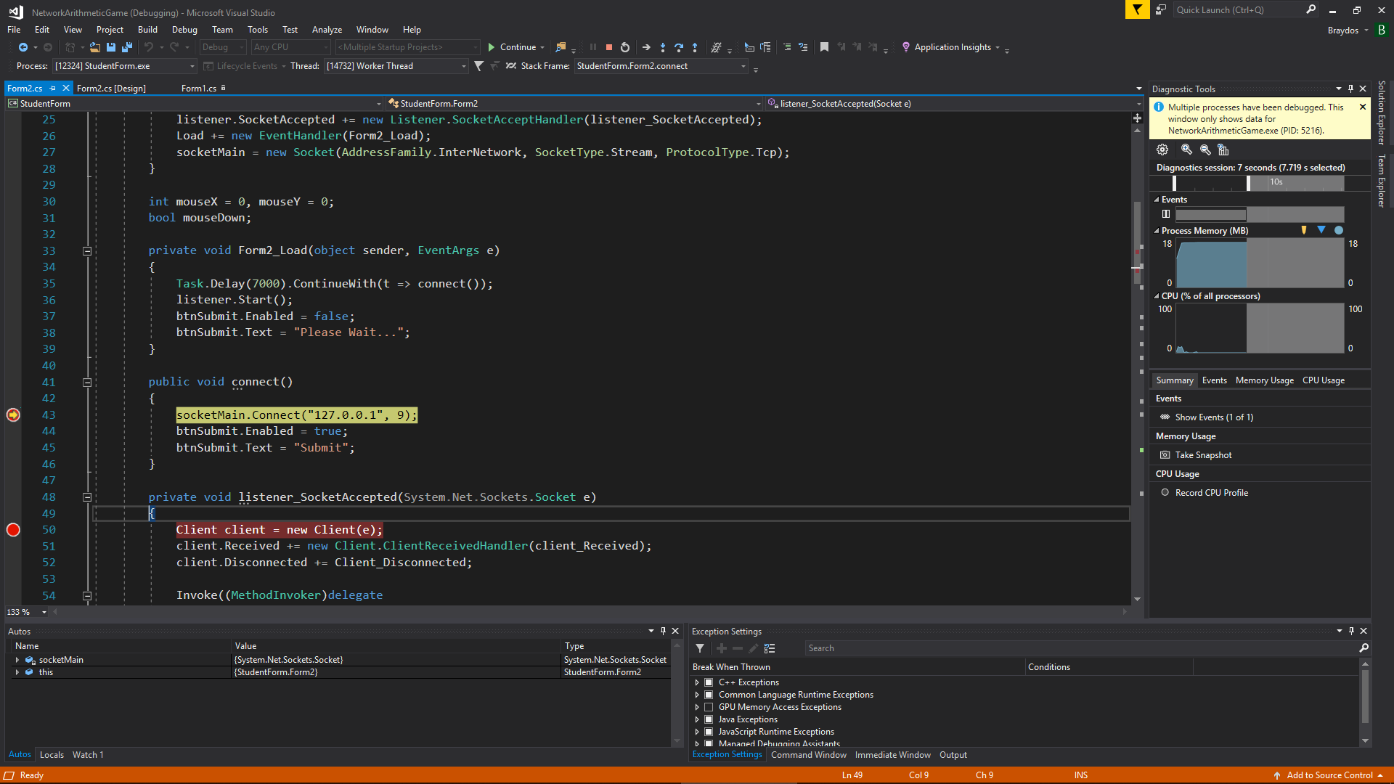


Figure : Show’s the watches used in debugging the application

## **Tracing:**

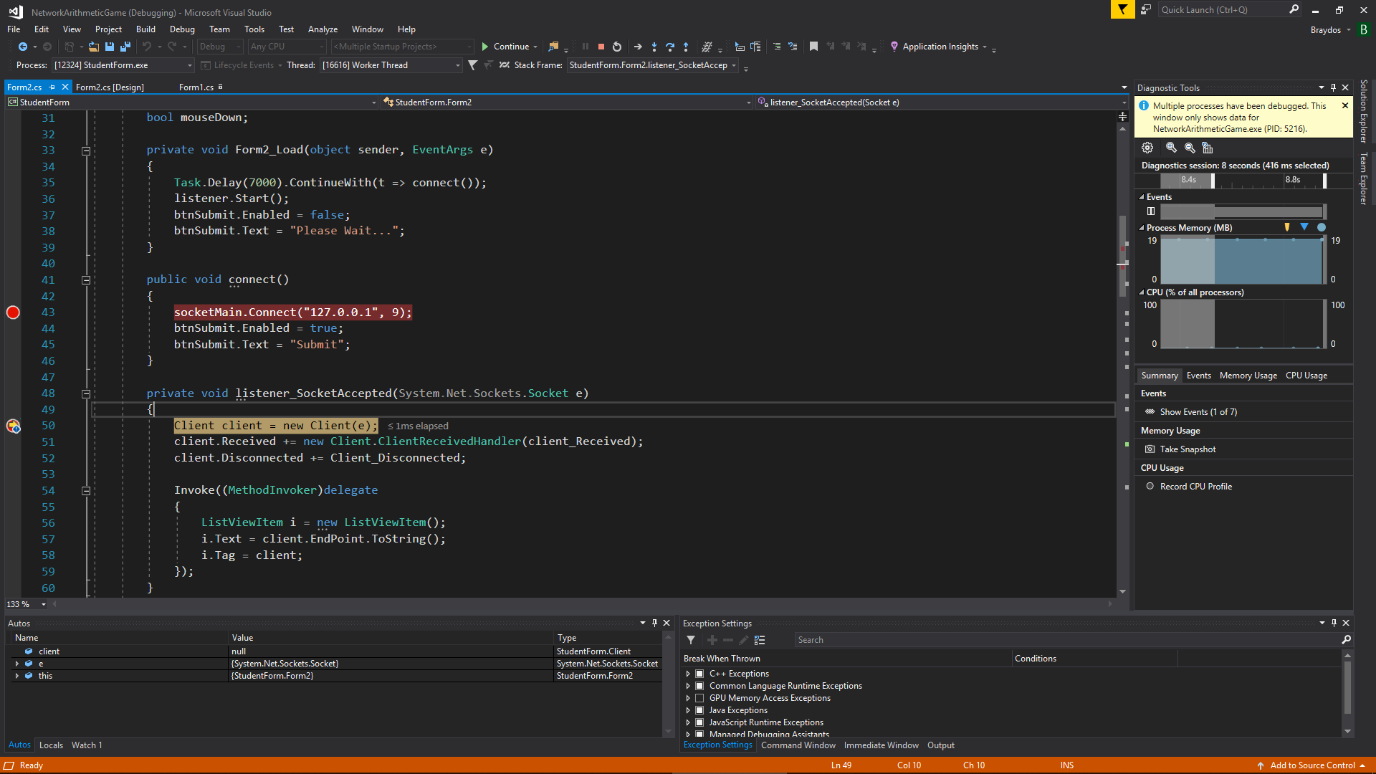


Figure : Displays tracing through lines of code using the debugger

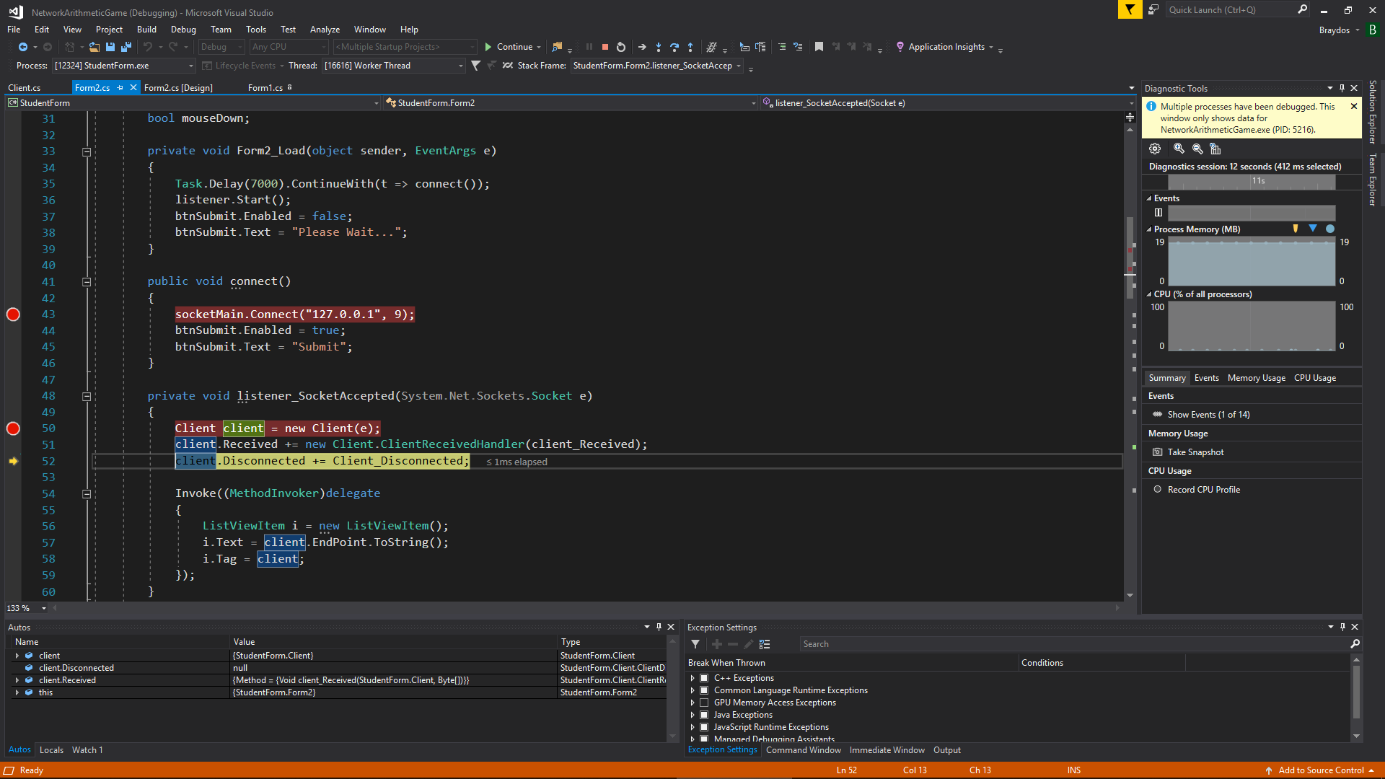


Figure : Displays tracing through lines of code using the debugger

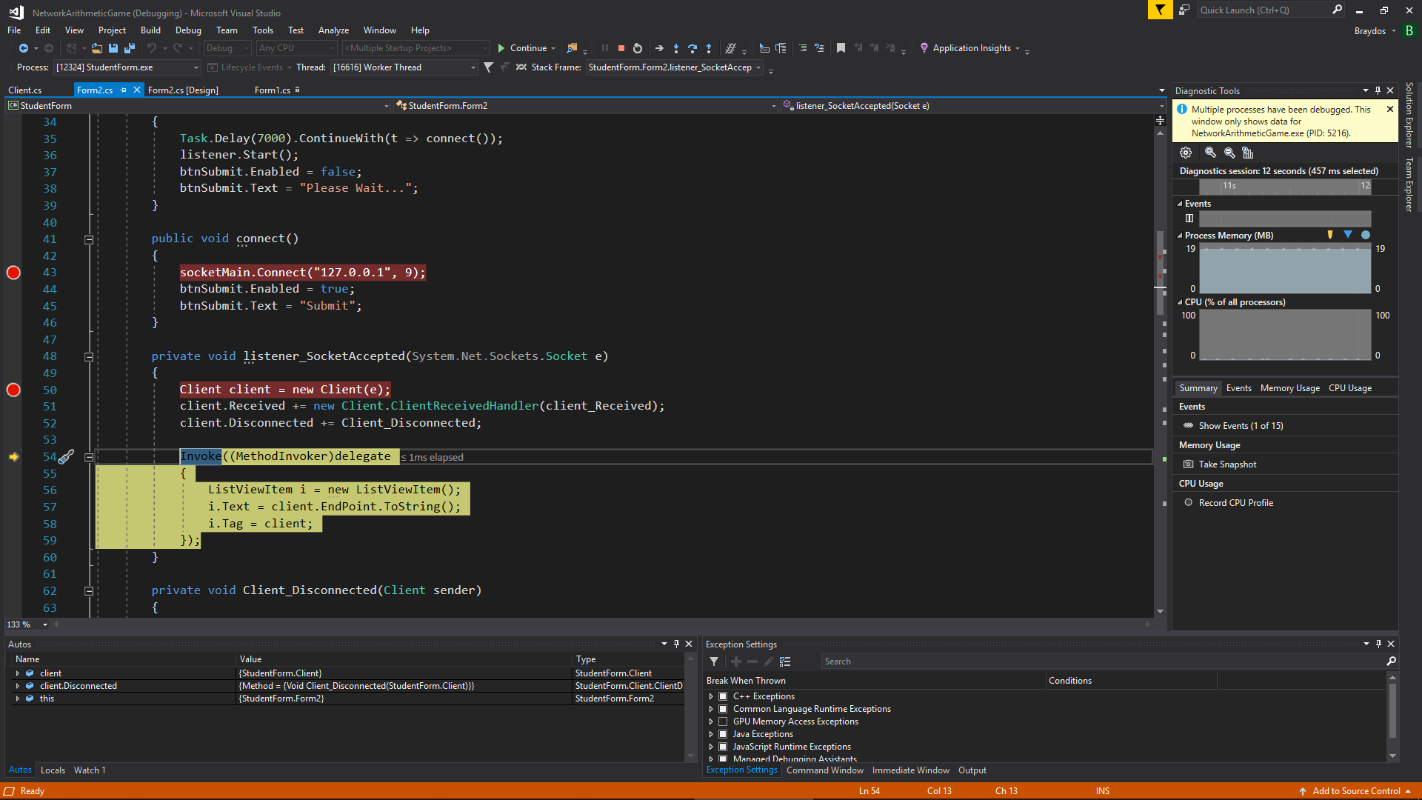


Figure : Displays tracing through lines of code using the debugger

# **References:**

Visual Studio 2017 - <https://visualstudio.microsoft.com/downloads/>

Atomineer Pro Documentation - <https://www.atomineerutils.com/>