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 1: 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 2: 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:**

## **INSTRUCTOR FORM:**

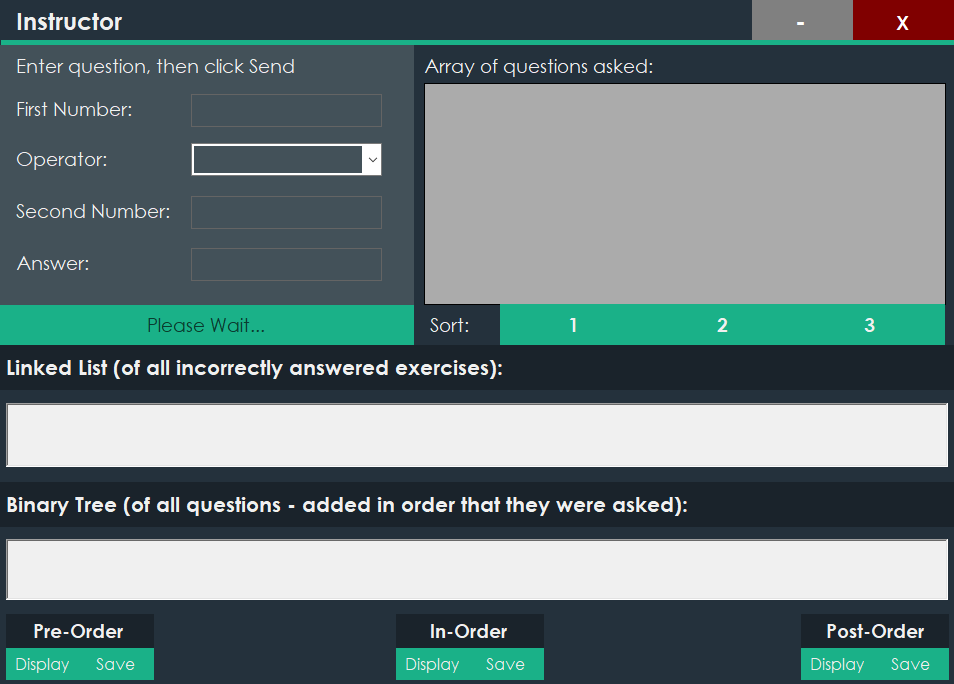


Figure 3: Overview of the Instructor Form

### **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:**

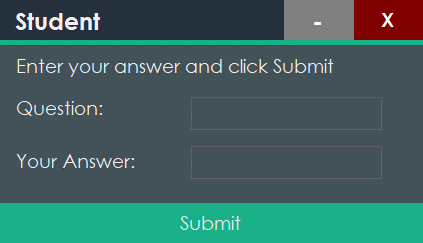


Figure 4: Overview of the Student Form

### **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 5: Display that the application starts with no complications

### **SENDING:**

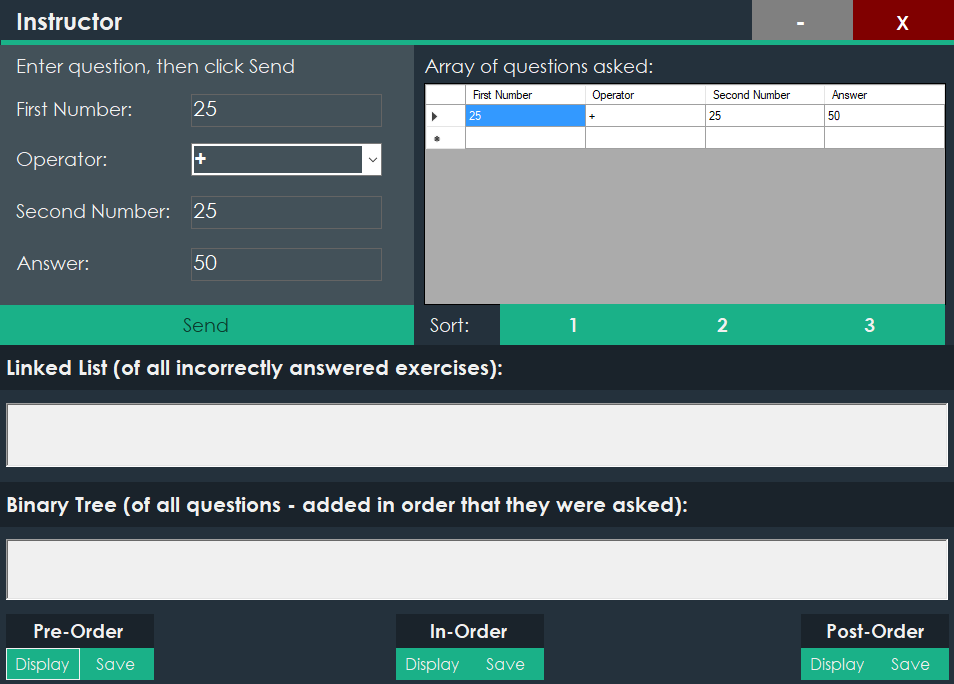
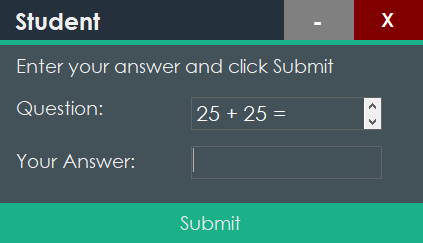


Figure 6: Successfully send and receives the values inputted across forms

### **STUDENT SUBMIT CORRECT:**

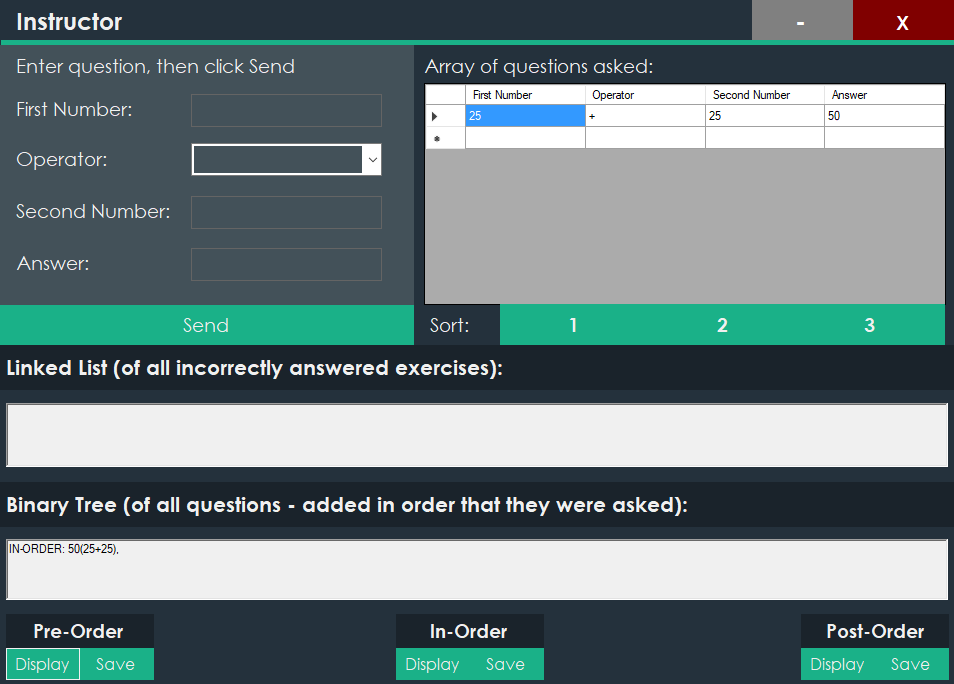
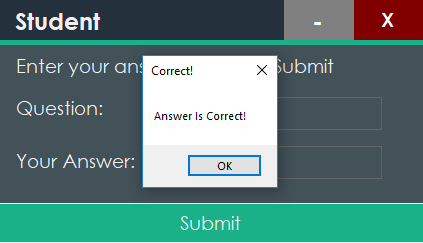


Figure 7: After getting the answer correct, the question and answer are added to the array of questions and the binary tree

### **STUDENT SUBMIT INCORRECT:**

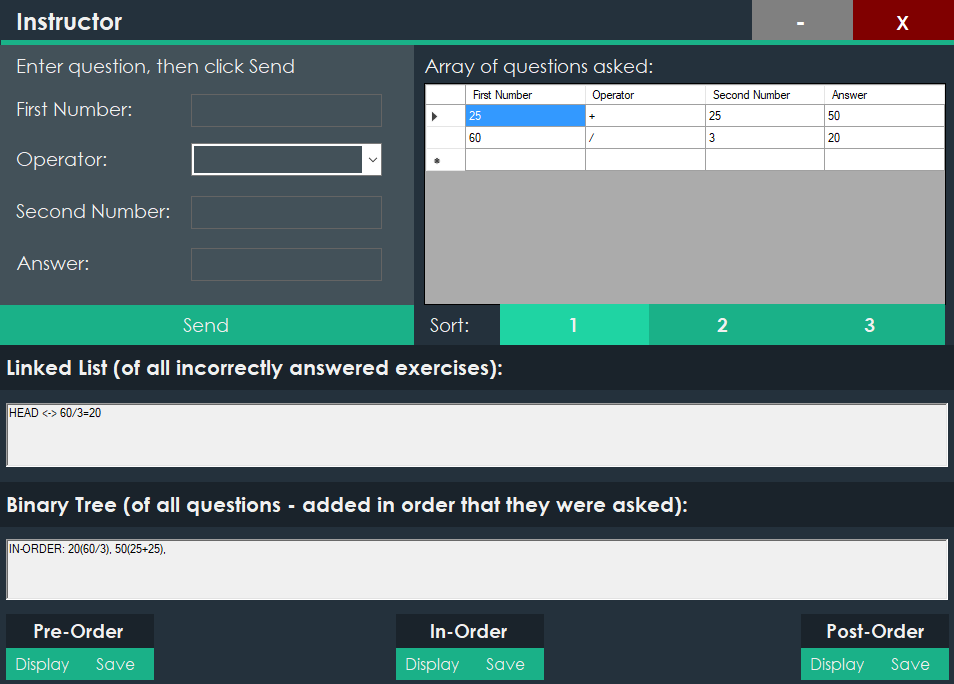
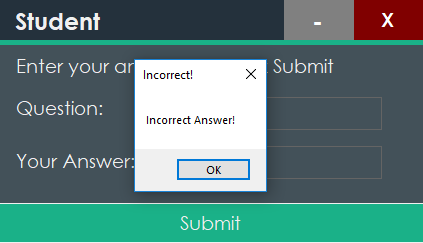


Figure 8: After getting answer incorrect, the question and answer are added to the array of questions, the linked list of all incorrectly answered questions and the binary tree

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

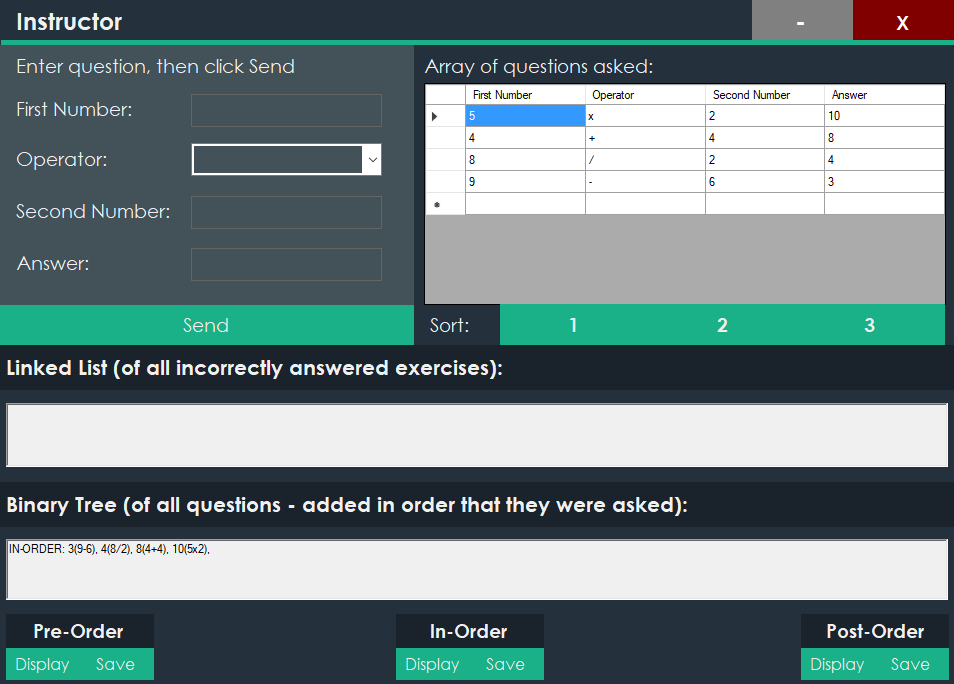


Figure : Overview of the array of questions with Sort 1 selected

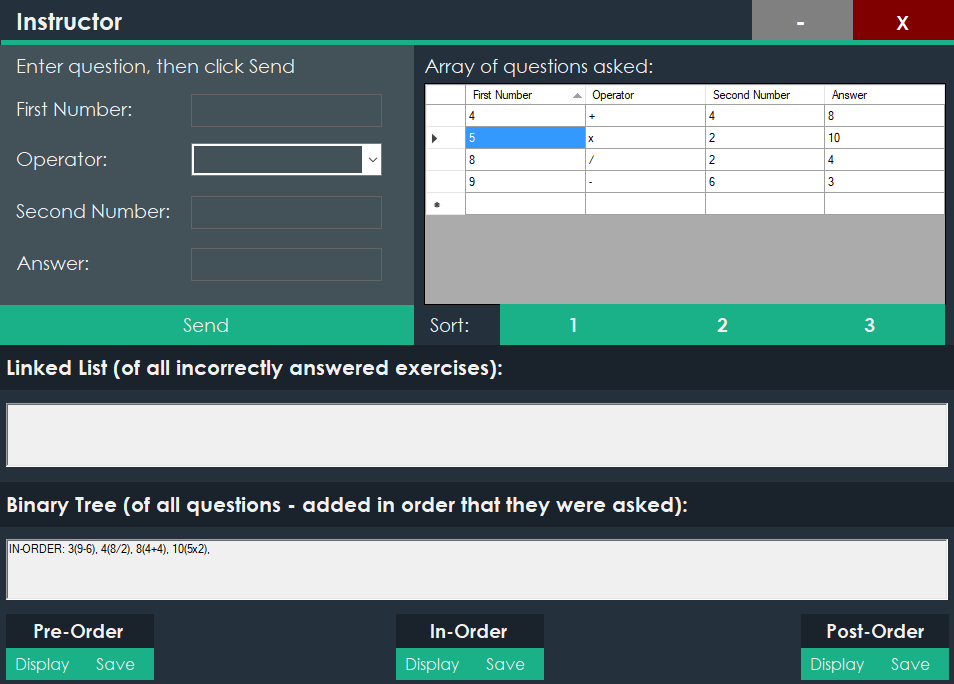
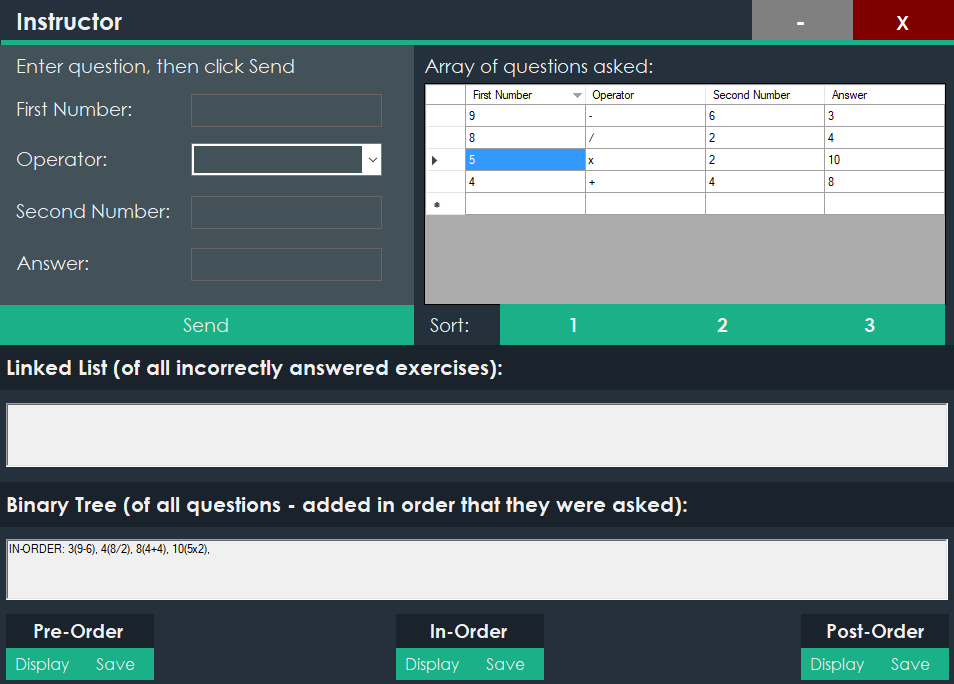


Figure 10: Sorts the numbers ascending

Figure 11: Sorts the numbers descending

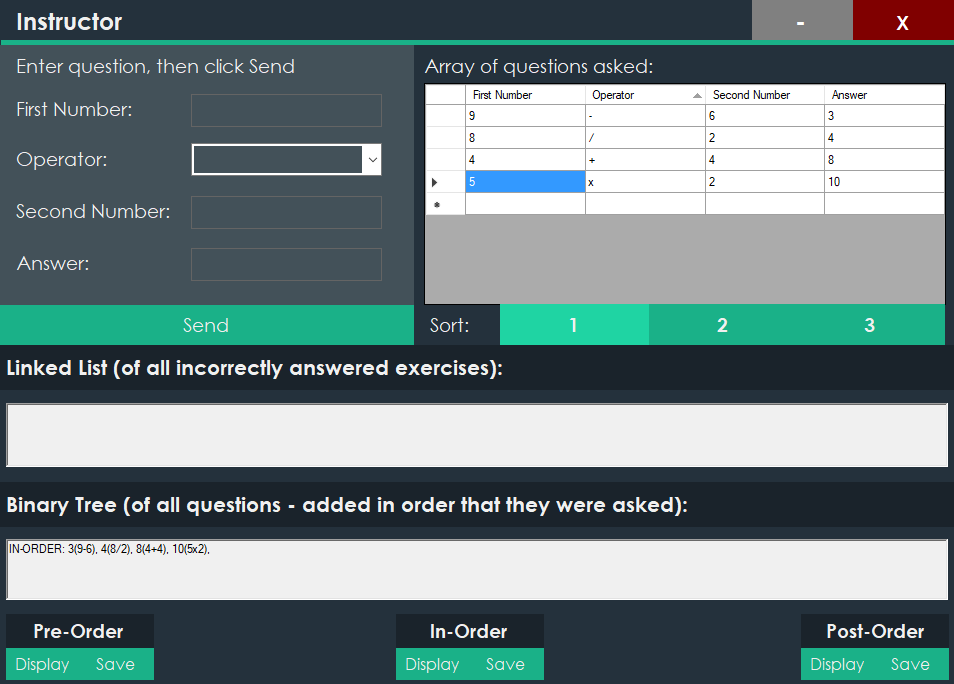


Figure 12: Sorts by Operator

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

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

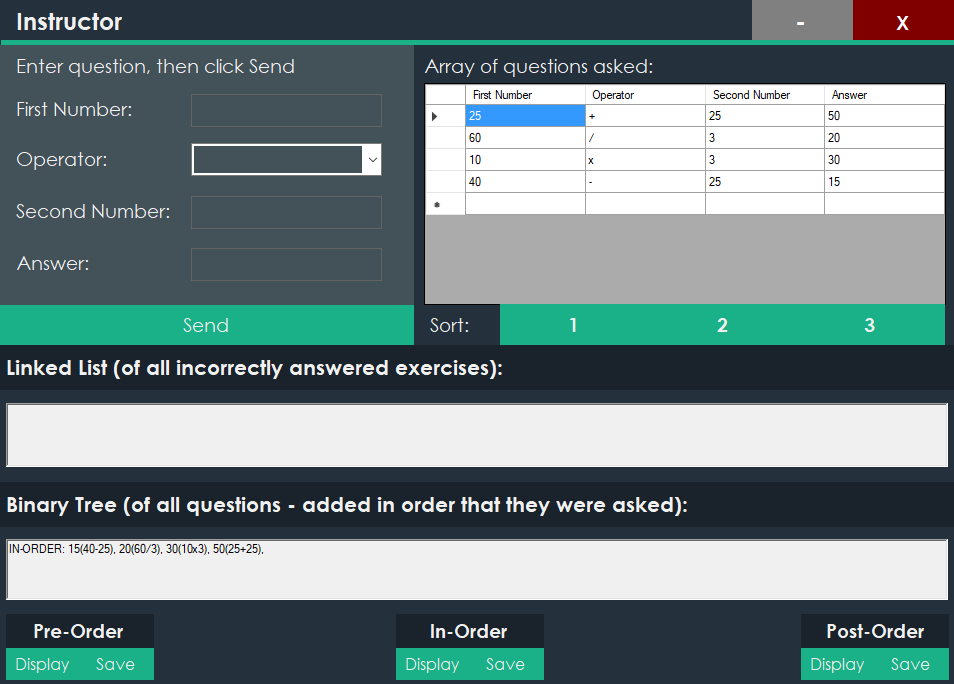
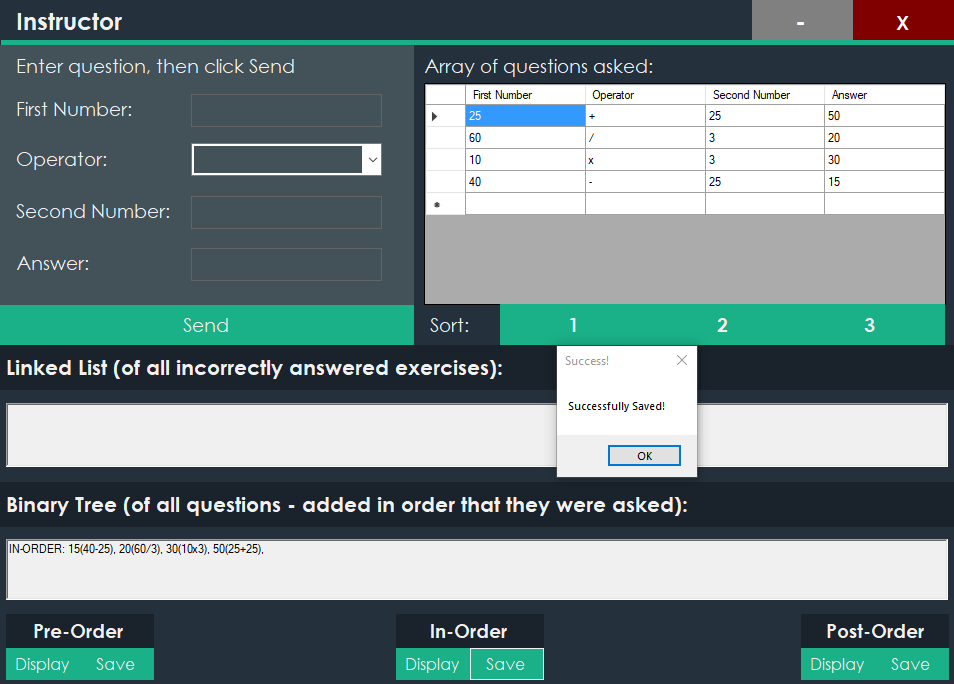


Figure 13: Displays In-Order Saved

Figure 14: Displays Binary Tree In-Order

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

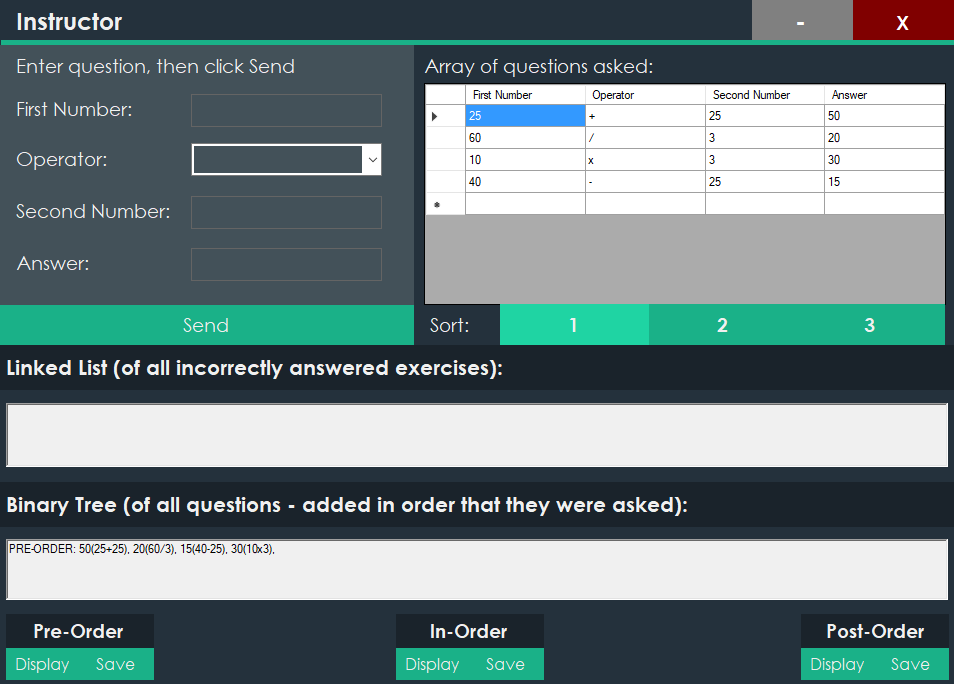
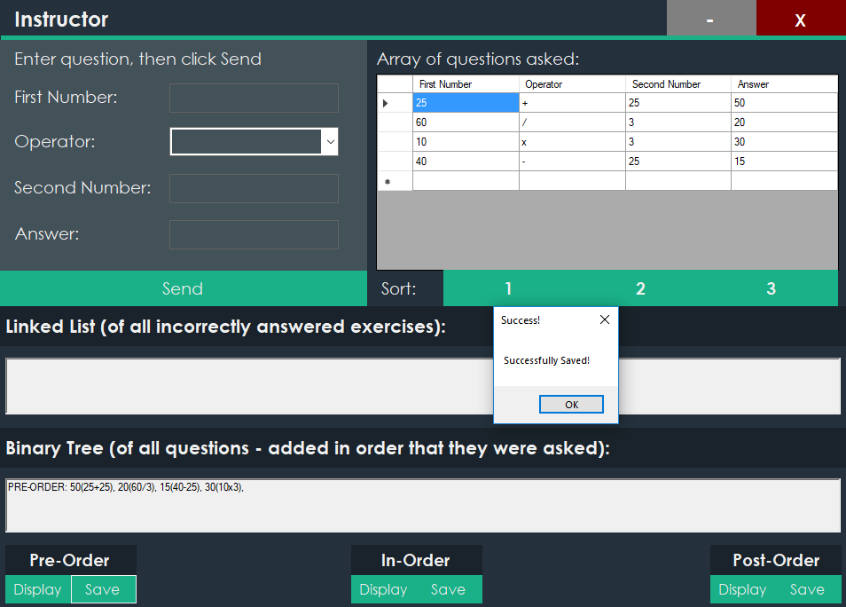


Figure 15: Displays Pre-Order Saved

Figure 16: Displays Binary Tree Pre-Order

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

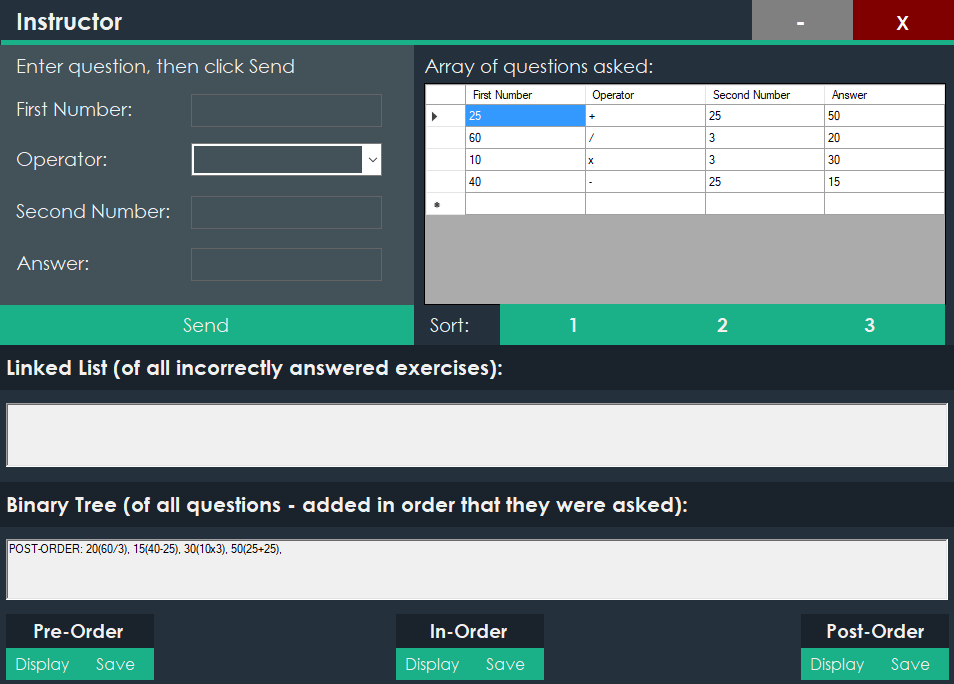
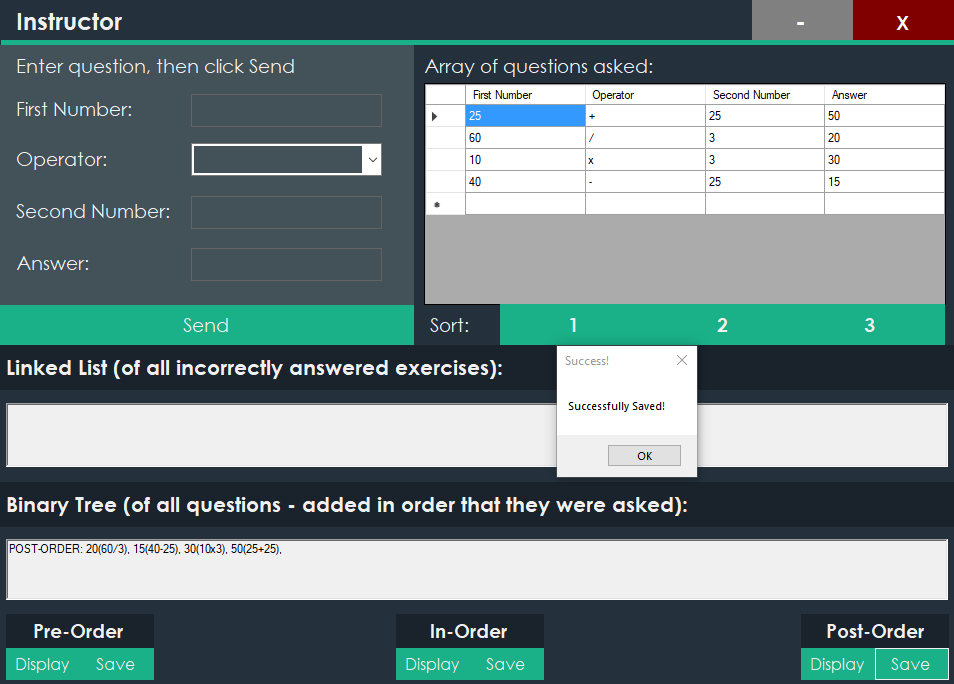


Figure 17: Displays Post-Order Saved

Figure 18: Displays Binary Tree Post-Order

# **User Manual:**

## **Introduction:**

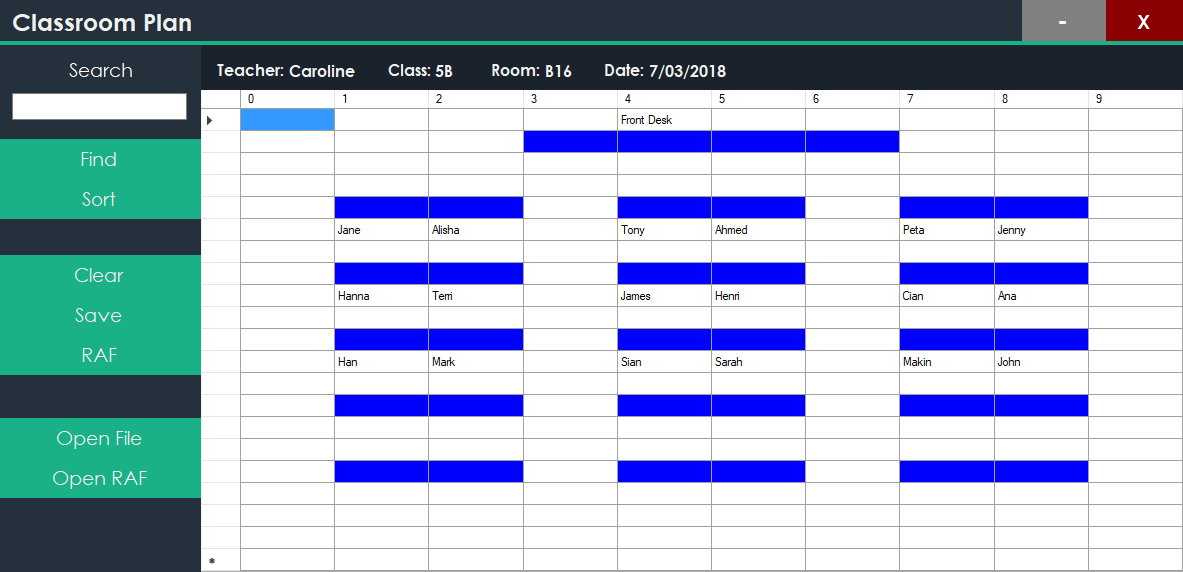
Welcome to the User Manual of the NetworkArithmeticGame application. This manual will provide vital information in regard to explaining the capability of the application and the functionality within the application that is to be utilized by the user. This manual is aimed specifically towards first time 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:**

### **MAINFORM:**



**9**

**4**

**5**

**8**

**7**

**6**

**3**

**2**

**1**

Figure 19: Displays the functionality of the mainform into individual parts

The open file button allows the user to open a csv file in the application to be imported towards the Classroom Datagridview (2).

### **OPEN FILE BUTTON:**



Figure 20: Displays the Open File Button functionality

### **CLASSROOM DATAGRIDVIEW:**

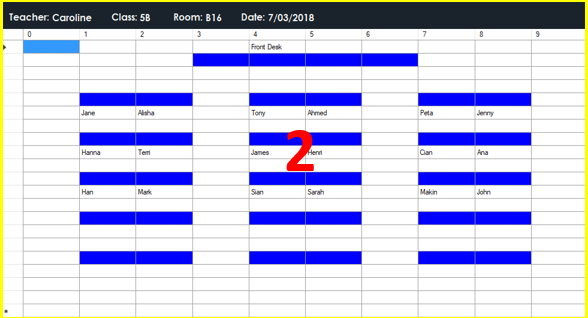


Figure 21: Displays the Classroom Datagridview functionality

The classroom datagridview displays each line & student from the csv file into a datagridview positioned specifically based on values with the csv file towards the user. It also allows the user to edit students within the datagridview and save them the updated information to a new csv file or overwrite the current one.

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

The minimize & exit button allows the user to minimize the application into the taskbar and to terminate the and save the application by clicking the exit button.



Figure 22: Displays the Minimize & Exit Buttons functionality

The search functionality utilizes both the user’s input and the find button. A user can input specifically a person’s name then proceed to press the find button. After this has occurred, cells are highlighted within the classroom datagridview to highlight the searched student (2).

### **SEARCH FUNCTIONALITY:**

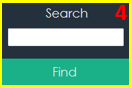


Figure 23: Displays the Search functionality

The sort button corresponds with both the classroom datagridview (2) and the search functionality (4). When the sort button is clicked a second form appears which is a sorted list alphabetically of all the students and their positions in the classroom, also the input gathered from the search functionality highlights that specific student.

### **SORT BUTTON:**



Figure 24: Displays the Sort Button functionality

The clear button corresponds with the classroom datagridview (2). It enables the user to clear all the students from the datagridview.

### **CLEAR BUTTON:**



Figure 25: Displays the Clear Button functionality

The save button corresponds with the classroom datagridview (2). It enables the user to save and update the imported datagridview to a new csv file or overwrite the current csv file imported.

### **SAVE BUTTON:**



Figure 26: Displays the Save Button functionality

The RAF button corresponds with the classroom datagridview (2). It includes writing the data from the datagridview to a text file enabling data to be read quickly and efficiently.

### **RAF BUTTON:**



Figure 27: Displays the RAF Button functionality

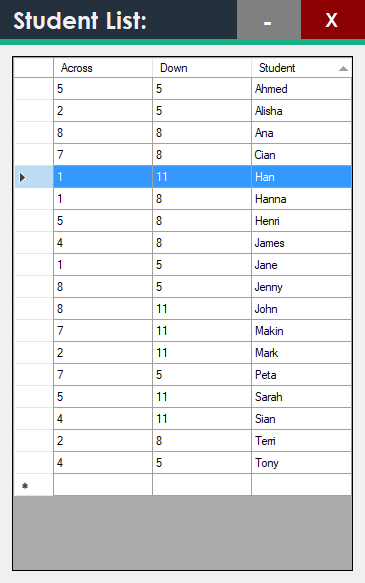
The Open RAF includes allowing the user to search for a specific entry within a Random-Access File then displaying the results into the classroom datagridview (2).

1. **OPEN RAF BUTTON:**



Figure 28: Displays the Open RAF Button functionality

## **SORTFORM:**



**2**

Figure 29: Displays the functionality of the sortform into individual parts

### **SORTLIST DATAGRIDVIEW:**

The sort list datagridview corresponds with the sort button (5). The sort list datagridview includes a sorted list of all the students in alphabetical order and their desk positions noted.

**1**

### **2. MINIMIZE & EXIT BUTTONS:**

The minimize & exit button allows the user to minimize the application into the taskbar and to terminate the and save the application by clicking the exit button.



Figure : Displays the Minimize & Exit Buttons functionality

# **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.

# **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:**

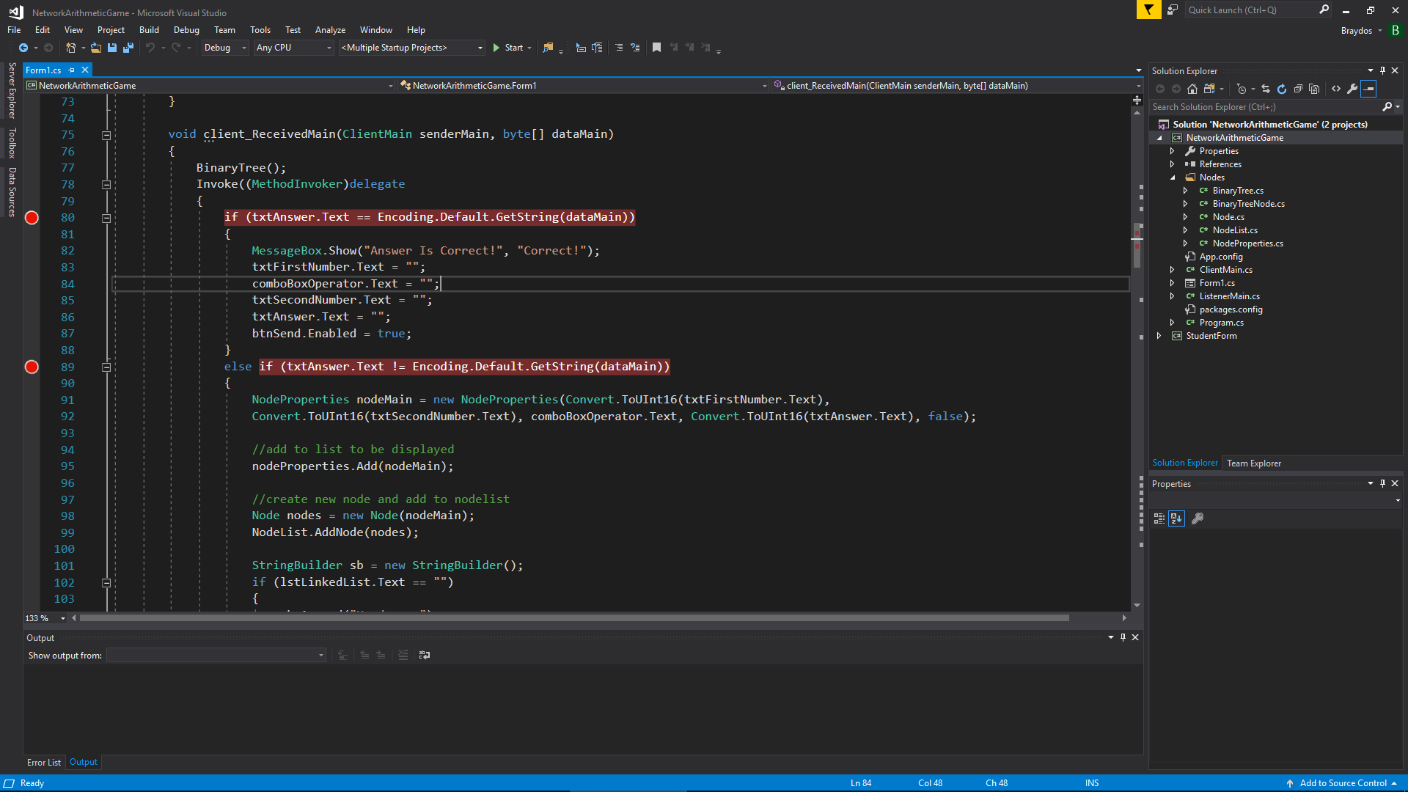


Figure 31: Show’s the breakpoints used in debugging the application

## **Watches:**

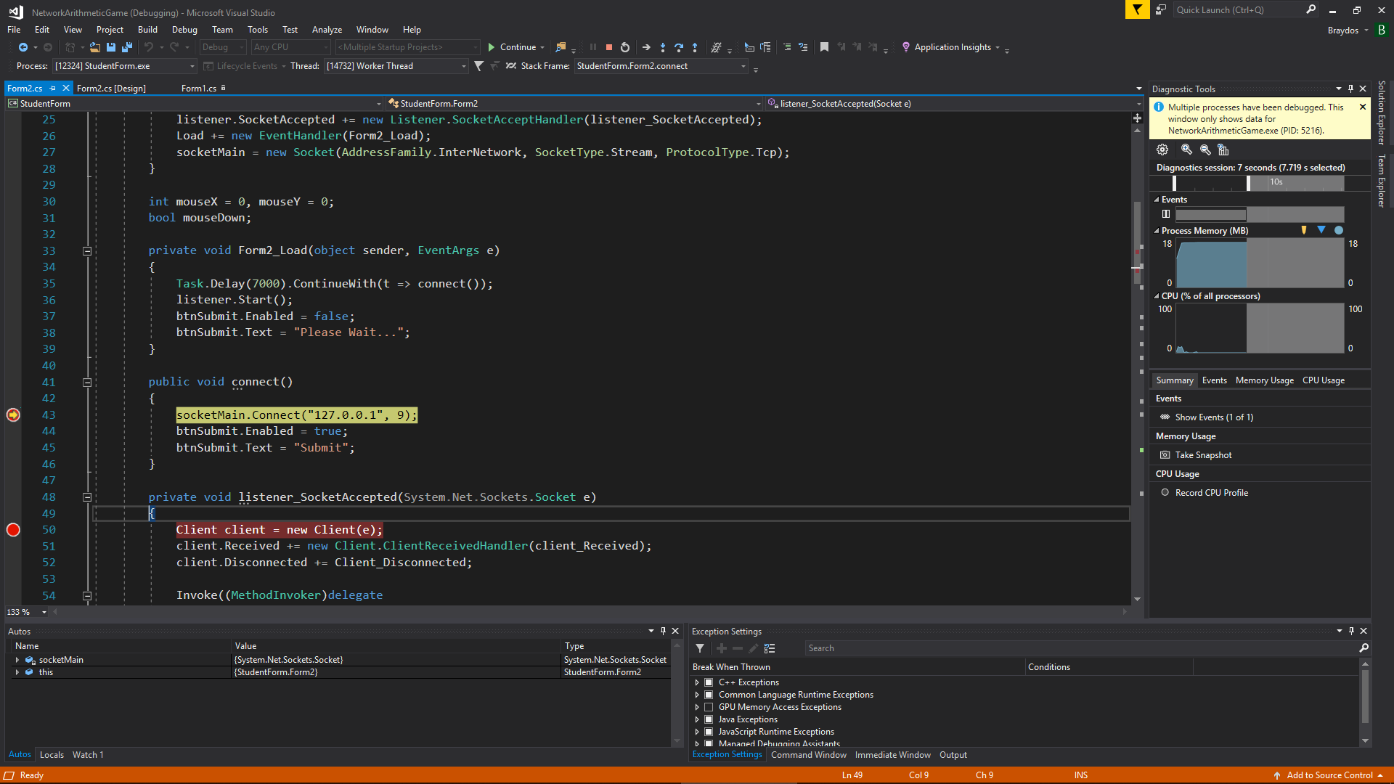


Figure 32: Show’s the watches used in debugging the application

## **Tracing:**

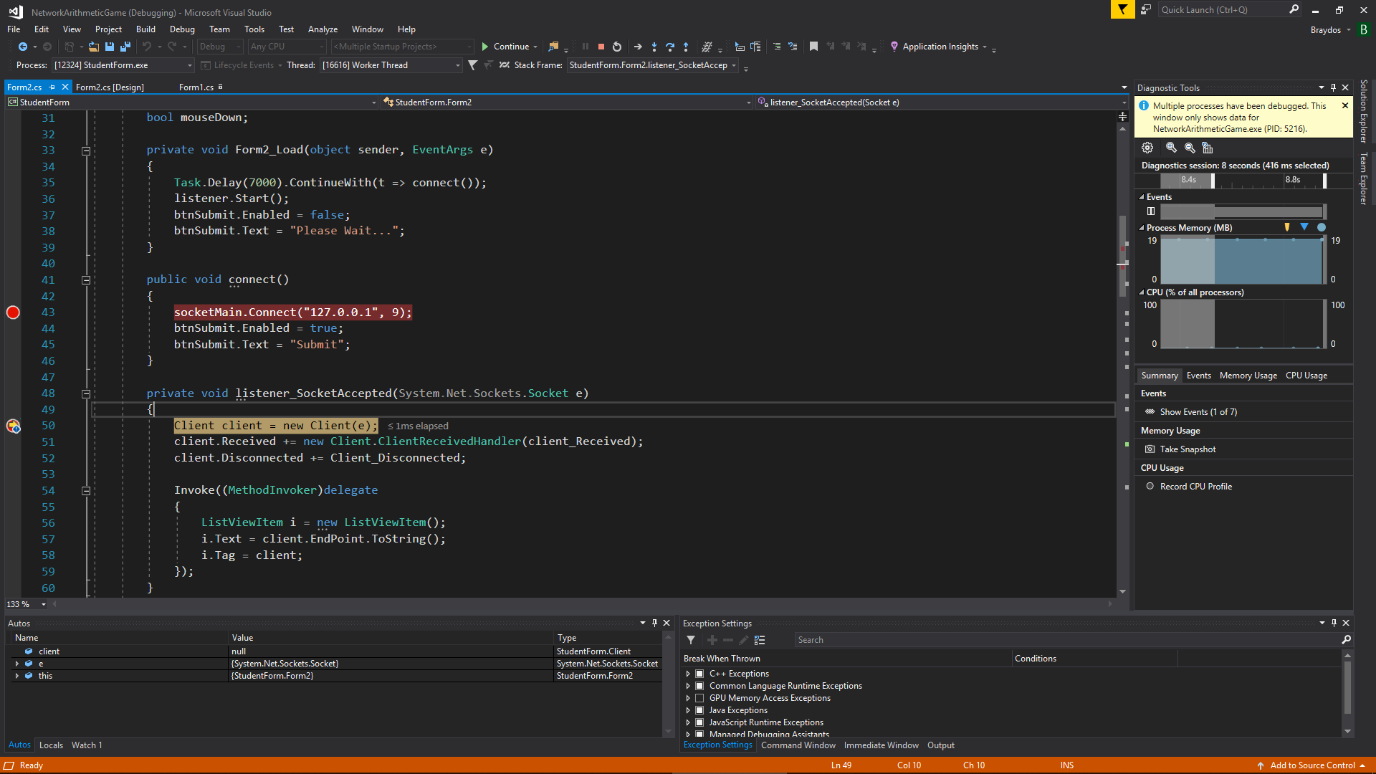


Figure 33: Displays tracing through lines of code using the debugger

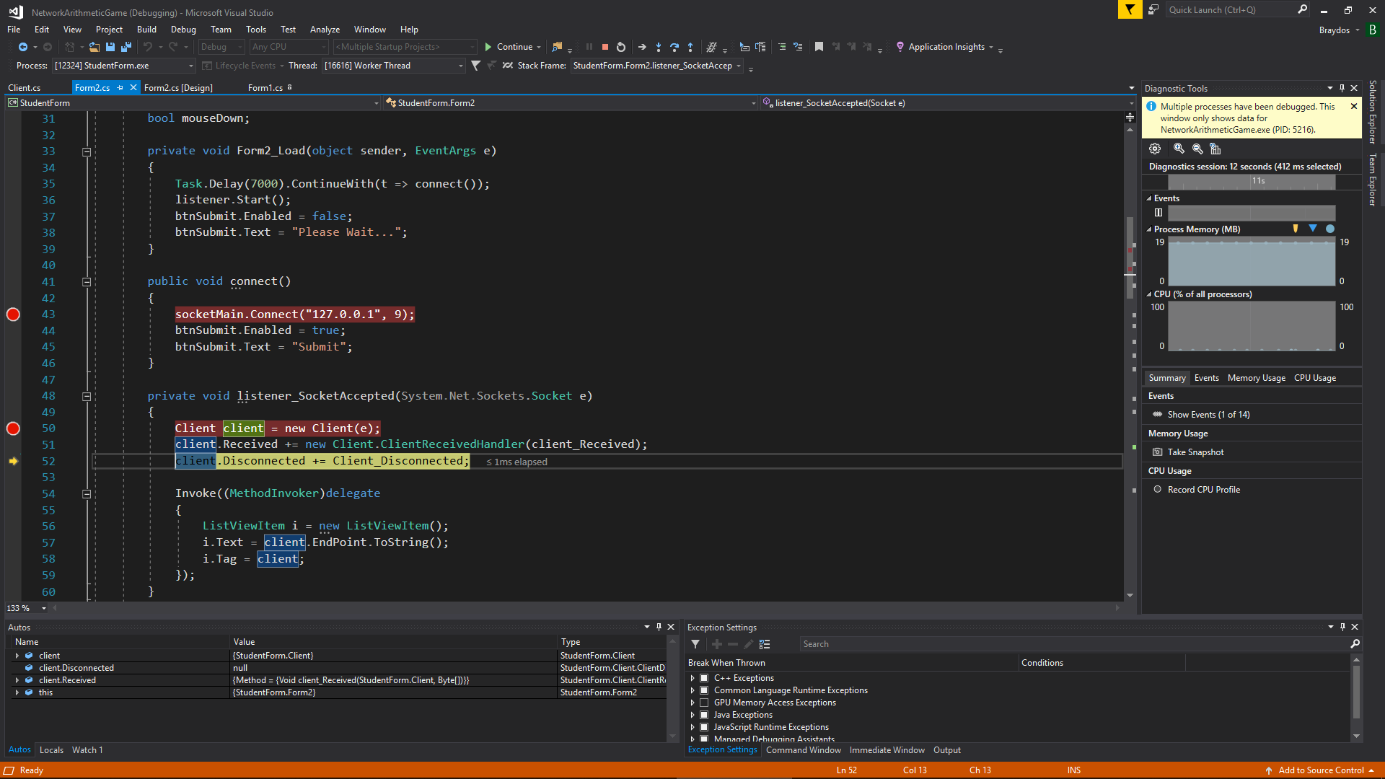


Figure 34: Displays tracing through lines of code using the debugger

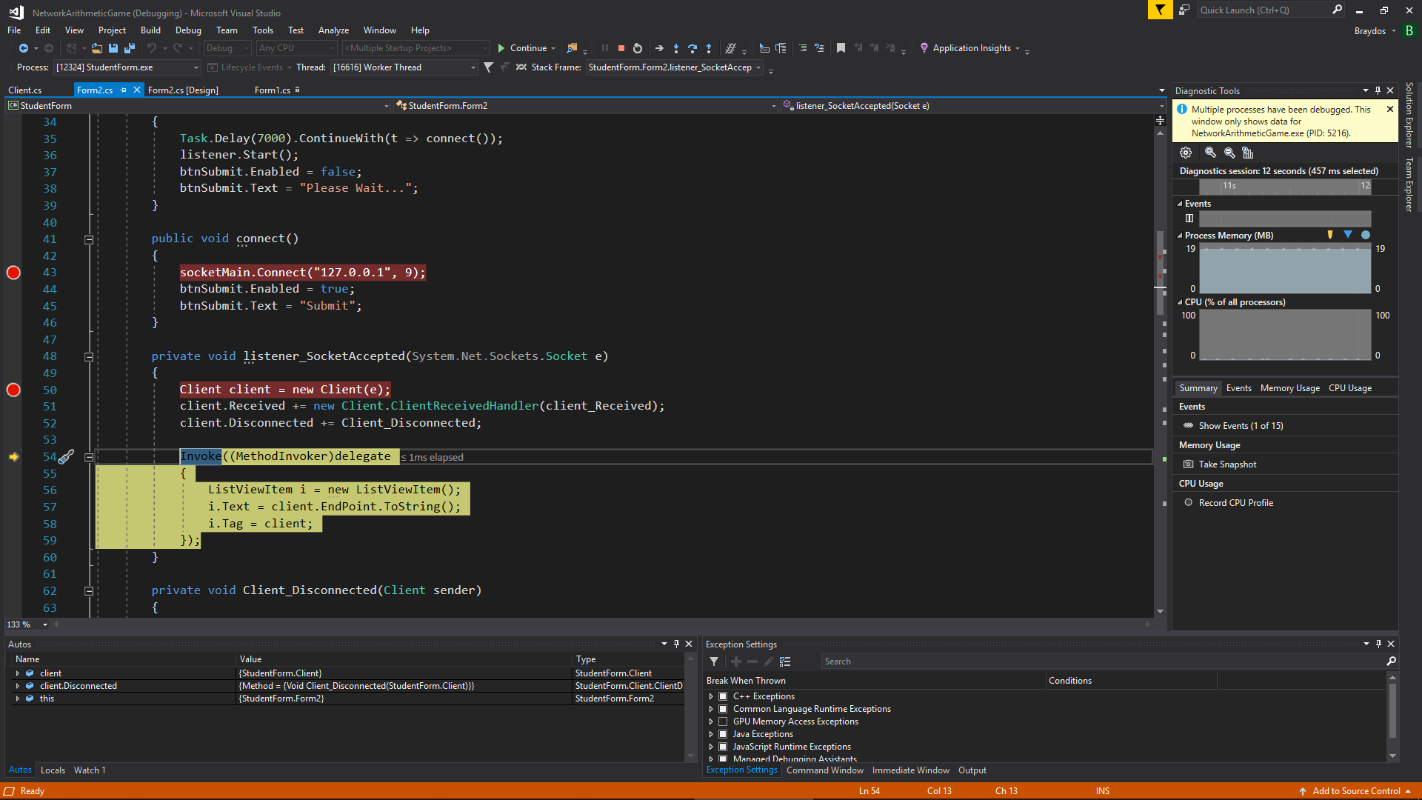


Figure 35: Displays tracing through lines of code using the debugger

# **References:**

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

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