**Documentation**

**Jakob Farrow**

**Date | 14 – 09 – 18**

# **Table of contents**

Contents

[**Table of contents** 2](#_Toc522900904)

[**About the Application** 3](#_Toc522900905)

[Hardware & Software 3](#_Toc522900906)

[Processes 3](#_Toc522900907)

[Features 3](#_Toc522900908)

[**TOE charts** 4](#_Toc522900909)

[Buttons 5](#_Toc522900910)

[Labels 5](#_Toc522900911)

[Textboxes & Data Grid 5](#_Toc522900912)

[**User Manual** 6](#_Toc522900913)

[**Testing Performed** 8](#_Toc522900914)

[Searching 8](#_Toc522900915)

[Adding new person 8](#_Toc522900916)

[Editing existing person 9](#_Toc522900917)

[Saving to csv 9](#_Toc522900918)

[Deleting an existing person 9](#_Toc522900919)

[Clearing the application 10](#_Toc522900920)

[Birthday in month of search 10](#_Toc522900921)

[Next Button 10](#_Toc522900922)

[Last Button 11](#_Toc522900923)

[Exit Button 11](#_Toc522900924)

[Select Data Grid 11](#_Toc522900925)

[Sorting Data Grid 12](#_Toc522900926)

[**Source code** 12](#_Toc522900927)

[**Flowchart** 13](#_Toc522900928)

[**Debugging** 14](#_Toc522900929)

[Tracing 14](#_Toc522900930)

[Watches 14](#_Toc522900931)

[Breakpoint 15](#_Toc522900932)

[**UML Diagram** 16](#_Toc522900933)

[**References** 17](#_Toc522900934)

# **About the Application**

The Network Game application was built to assist teachers and students in the teachings of mathematics by sending the student through a network questions while also allowing the teacher to easily see the results of the students answers as well as the questions asked.

## **Hardware & Software**

**Hardware:**

* Desktop computer OR sufficient laptop
* At least 4GB of RAM
* At least 2Ghz - 4 Core CPU
* At least 120 GB of Storage

**Software:**

* Visual Studio 2017
* Atomineer

## **Processes**

* To allow user to access the program connected to a network
* The user can enter numbers in the calculator and send them to the student
* The student can then receive the numbers and answer the result for them.
* The teacher can then see the results and questions asked

## **Features**

* Program able to connect to a network
* Send the information to the student form
* For the student to send back the information to the teacher form
* Teacher can then view results and questions asked in array, link lists and binary trees.

## **3rd Party use**

Newtonsoft.Json

Newtonsoft.Json was used for the 3rd Party library in the construction of the application.

Atomineer

Atomineer is a 3rd party auto documentation software which has been used throughout the development of the application.

# **TOE charts**

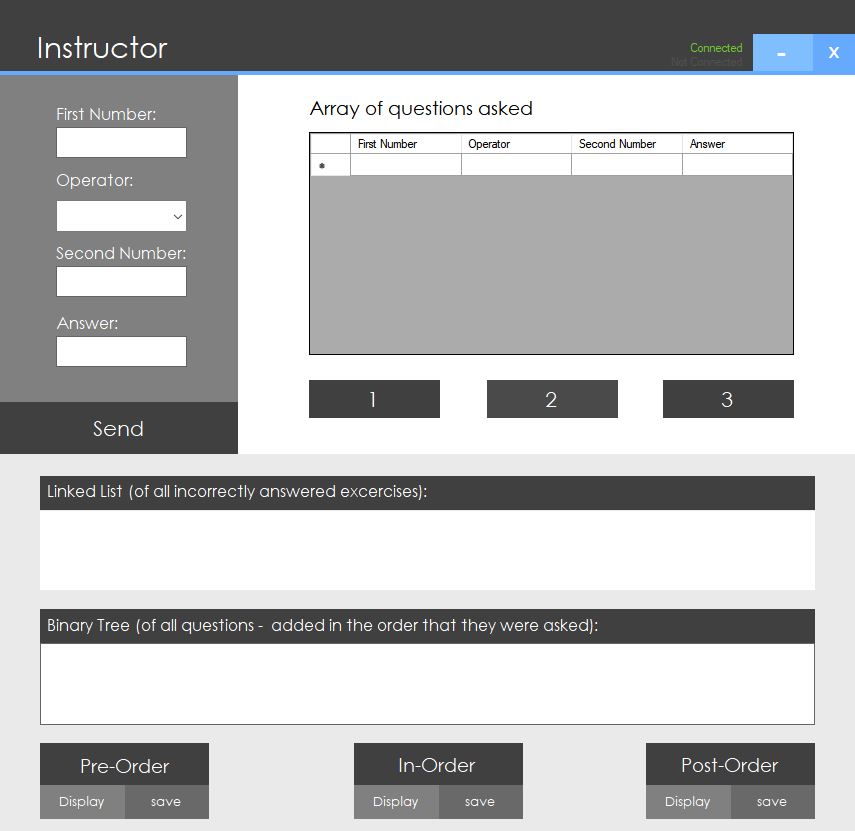
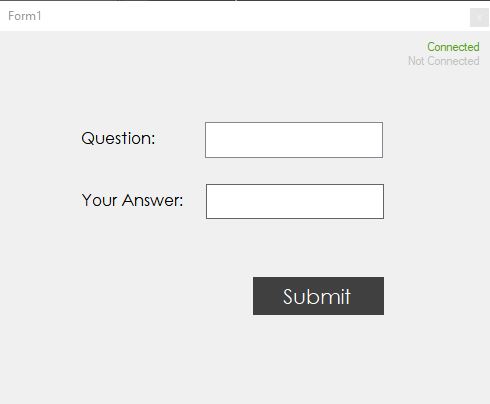
****

Figure 2: Layout and Design of SortGUI

Figure 1: Layout and Design of GUI



|  |  |  |
| --- | --- | --- |
| **Task** | **Object** | **Event** |
| Exit application | btnExit | btnExit\_Click |
| Minimize | btnMinimize | btnMinimize\_Click |
| Send | btnSend | btnSend\_Click |
| 1 | btnSort1 | btnSort1\_Click |
| 2 | btnSort2 | btnSort2\_Click |
| 3 | btnSort3 | btnSort3\_Click |
| PreOrder display | btnDisplayPreorder | btnDisplayPreorder\_Click |
| PreOrder Save | btnSavePreorder | btnSavePreorder\_Click |
| InOrder display | btnDisplayInorder | btnDisplayInorder\_Click |
| InOrder save | btnSaveInorder | btnSaveInorder\_Click |
| PostOrder Display | btnDisplayPostOrder | btnDisplayPostOrder\_Click |
| PostOrder Save | btnSavePostOrder | btnSavePostOrder\_CLick |
| Student submit | btnSubmit | btnSubmit\_Click |

## **Buttons**

|  |  |  |
| --- | --- | --- |
| **Task** | **Object** | **Event** |
| Instructor | lbInstructor |  |
| First Number | lbFirstNumber |  |
| Operator | lbOperator |  |
| Second Number | lbSecondNumber |  |
| Answer | lbAnswer |  |
| Array of questions asked | lbQuestionsAsked |  |
| Linked List | lbLinkedList |  |
| Binary Tree | lbBinaryTree |  |

## **Labels**

|  |  |  |
| --- | --- | --- |
| **Task** | **Object** | **Event** |
| First Number | txtFirstNumber |  |
| Operator | txtOperator |  |
| Second Number | txtSecondNumber |  |
| Answer | txtAnswer |  |
| Linked List | txtRichLinkedList |  |
| Binary Tree | txtLinkedList |  |
| Teacher datagrid | dataGrid |  |
| Student questions | lstQuestion |  |
| Student Answer | txtAnswer |  |

## **Textboxes & Data Grid**

# **User Manual**

**5**

**4**

**3**

**2**

**1**

Figure 2: Button and textbox properties and functions

This is the section containing the teacher calculator, which she would use to input values and send the math question to the student.

1. **Calc**

Here is the data grid for which the asked questions from the teacher get logged.

1. **Questions**

The edit button will exit the application and minimize will shrink the window.

1. **Exit**

The linked list is where the incorrect answers from the students will be logged.

1. **LinkList**

The binary tree will log the questions asked and their answers and give the teacher options on how it is displayed.

1. **Binary**

**Tree**

# **Testing Performed**

## **Sending**

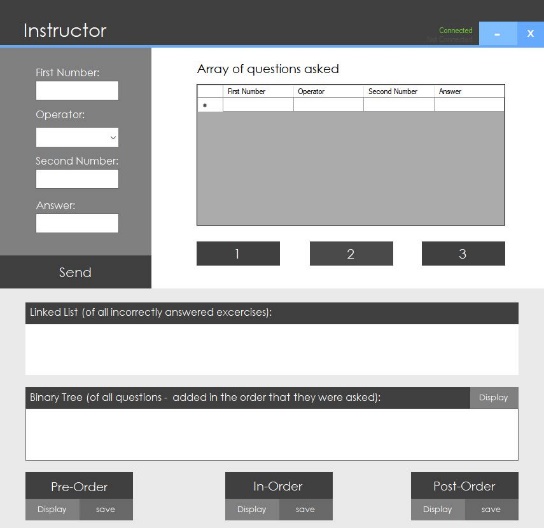
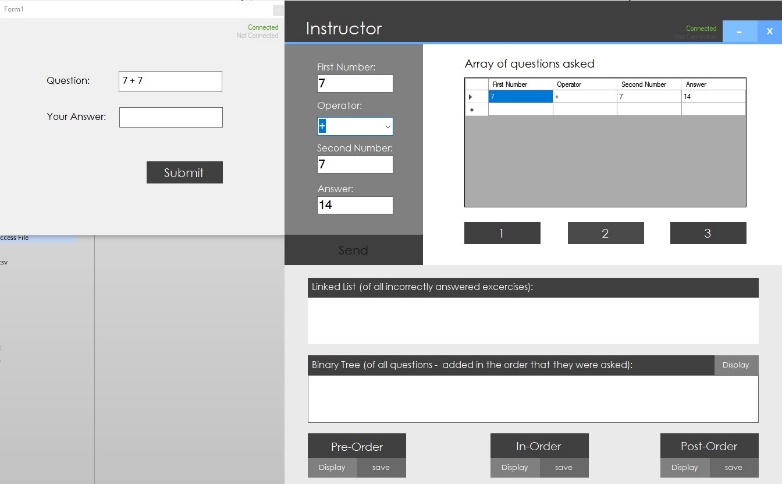
****

Figure 3: Before Seending

Figure 4: After sending

When the data has been entered into the calculator and the user clicks send the student from and array of questions asked will receive the questions.

## **Student submit**

Figure 5: Before Submit

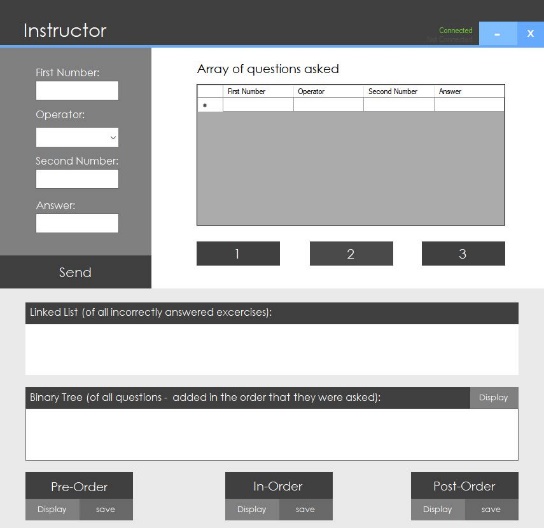
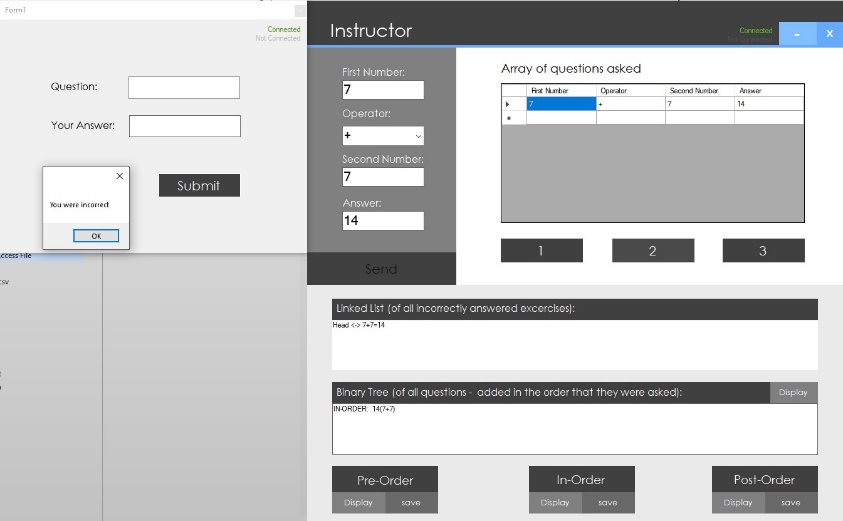
****

Figure 6: After submit

When the student clicks submit to answer the question sent by the teacher the results are inputted in to the link list and binary tree

## **Sort**

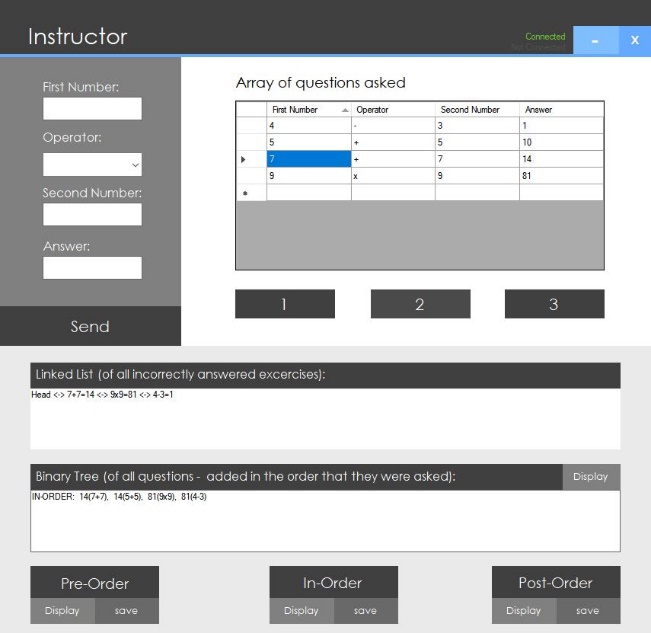
****

Figure 7: Before sort

Figure 8: After sort

When the 1, 2 and 3 buttons are clicked the array of questions will be sorted by ascending or descending order

## **Sending to binary tree**

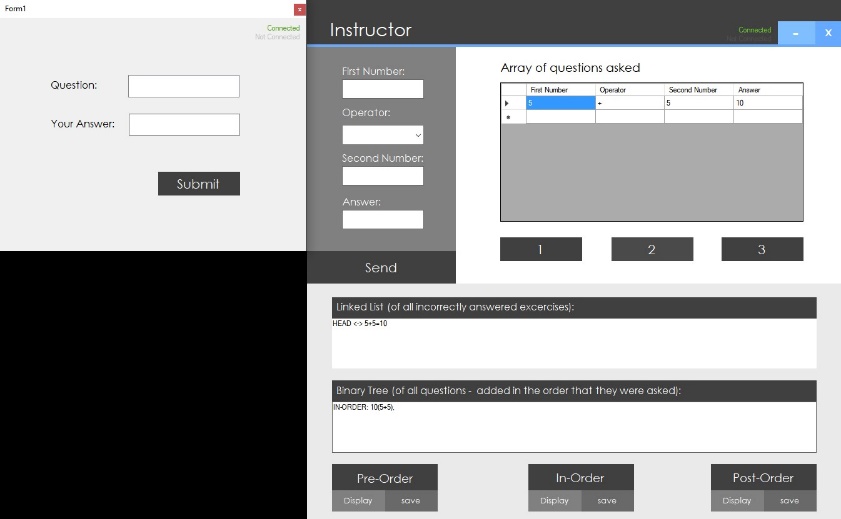
****

Figure 9: Before sending to teacher

Figure 10: After sending to teacher

When the student receives the math question and answers it correctly or incorrectly it will be sent to the teachers binary tree.

## **Using Order Buttons**

Figure 11: Before clicking order button

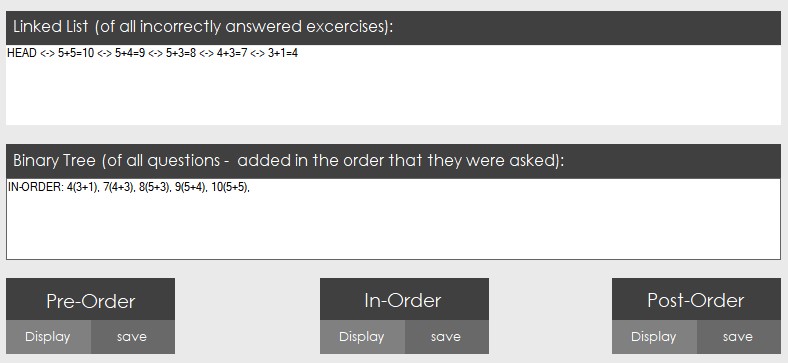
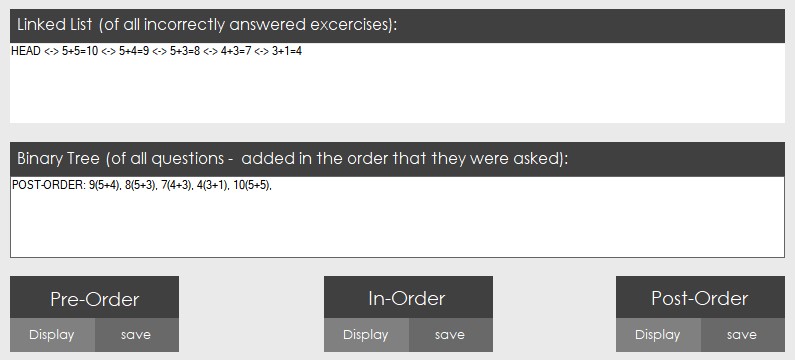
****

Figure 12: After clicking order button

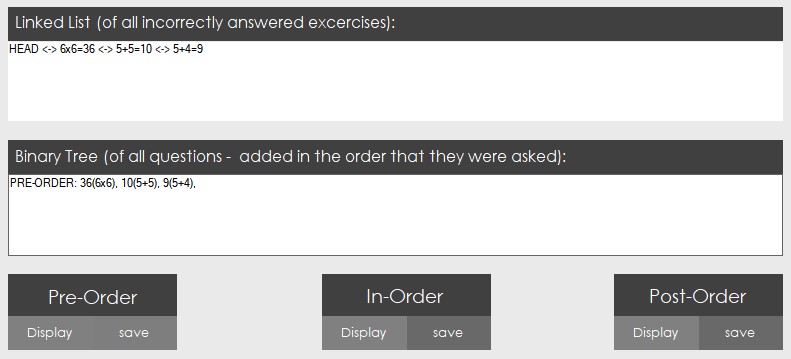
****

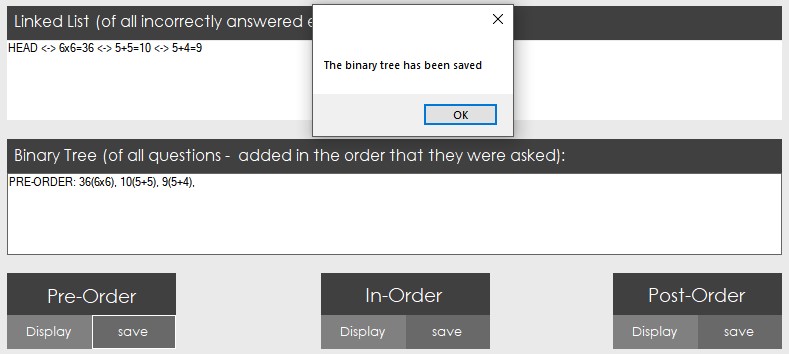
When the teacher receives the data in the binary tree she can then use the order buttons to display the values differently

## **Saving**

Figure 13: After clicking save

Figure 14: Before clicking save

****

****

When the teacher has the data in the binary tree they can then save that data to a text file by clicking save on any order.

# **Communication**

Manager Communication Email

Hey Dave,

With the documentation, GUI, and TOE charts now complete I will now be commencing development on the NetworkArithmeticGame application which will require multiple applications to communicate between each other.

Please contact me if you have any other questions about the documentation or development of the project.

Regards,

Jakob Farrow

# **Sorting Algorithms**

The advantages and disadvantages of three sorting algorithms include:

## Bubble

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

**Advantages**: Easy to implement

**Disadvantages**: Fails to fully operate with a list with a large number of items.

## Insertion

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

**Advantages**: Popular for its simplicity

**Disadvantages**: Bad precision for large lists

## Selection

The selection sort sifts through a list of items and makes a selection based on its ordering

**Advantages**: Works really well for a small list of items

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

# **Debugging**

## **Tracing**

Figure 26: Example of tracing being used during debugging

## **Watches**

Figure 27: Example of watches being use during debugging

## **Breakpoint**

Figure 28: Example of breakpoints being use during debugging

# **UML Diagram**

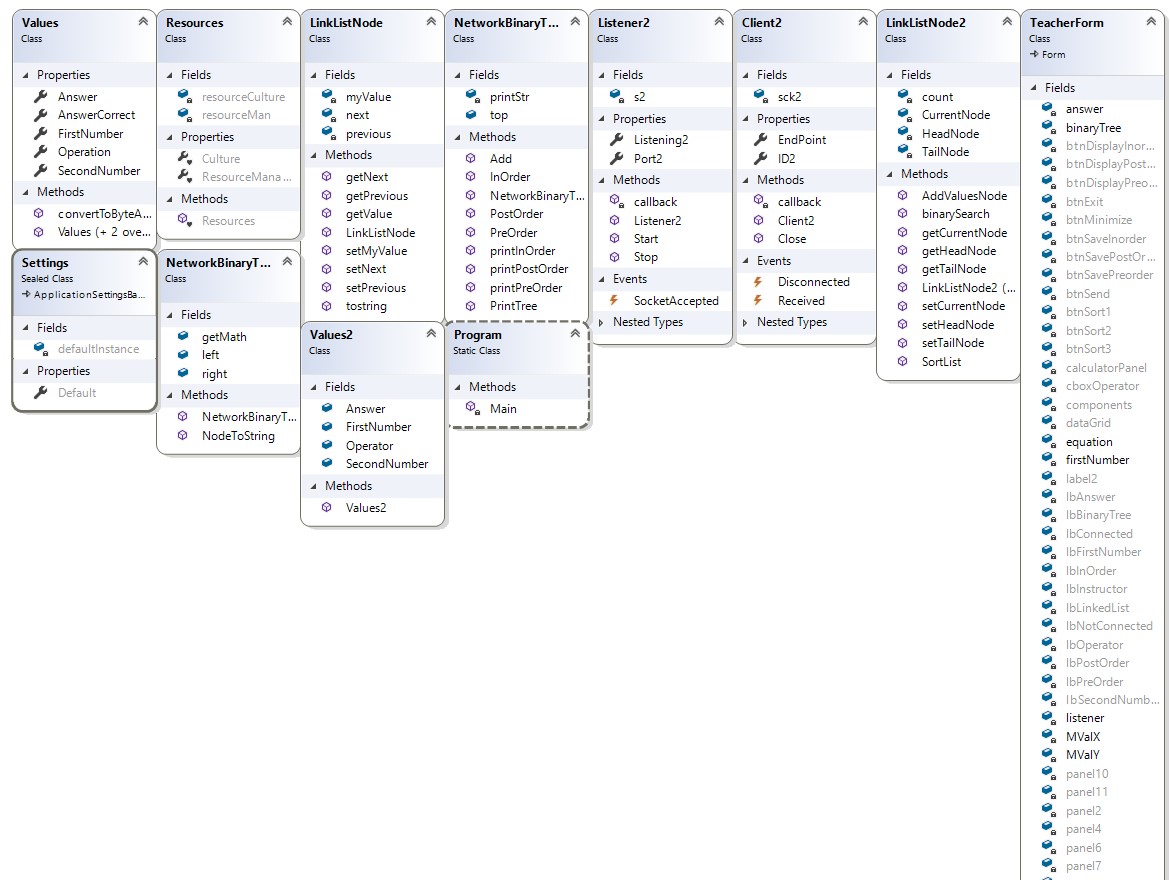


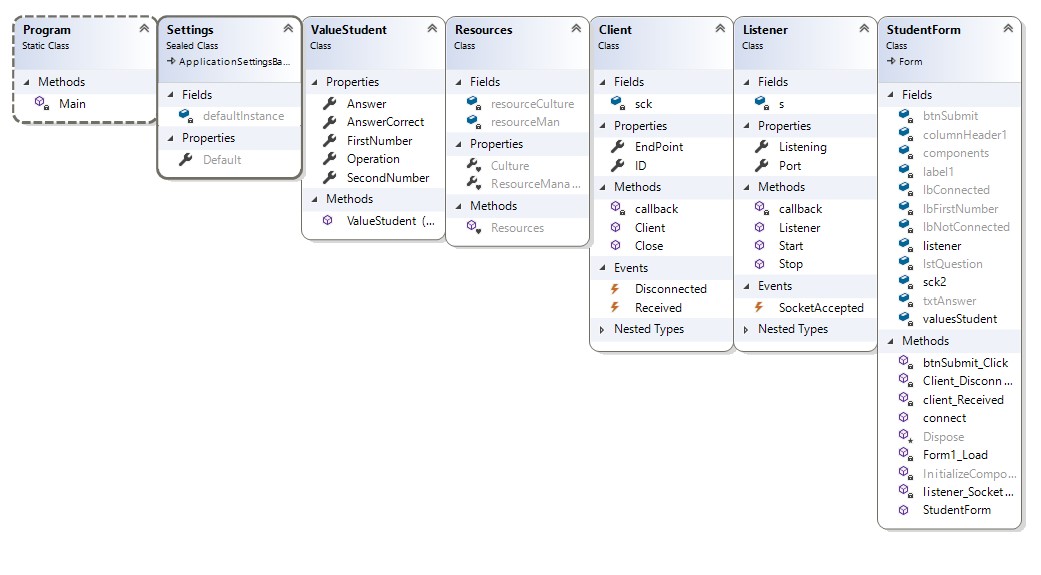
Figure 29: Final UML Class diagram

**Teacher UML**

The teacher UML uses LinkListNode, and LinkListNode 2 classes to create and get the values for the main form linked list. The teacher uml also uses the client and Listener classes to get the values sent from the student. The Teacher UML uses the TeacherForm class to link all the buttons, functions and GUI together.

**Student UML**

The Student UML uses the client and listener classes to get the values sent from the teacher form. The student UML uses the StudentForm class to link all the buttons, functions and GUI together.



# **References**

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

Atomineer: <https://www.atomineerutils.com/>