Classroom Robots

Technical Documentation

Kaitlyn Parsons

2018

Contents

[User Manual 2](#_Toc523990261)

[Toe Chart 4](#_Toc523990262)

[UML Diagrams 5](#_Toc523990263)

[Debugging 7](#_Toc523990272)

## User Manual

This application was developed to help teachers keep track of students’ attendance and seating positions for each classroom.

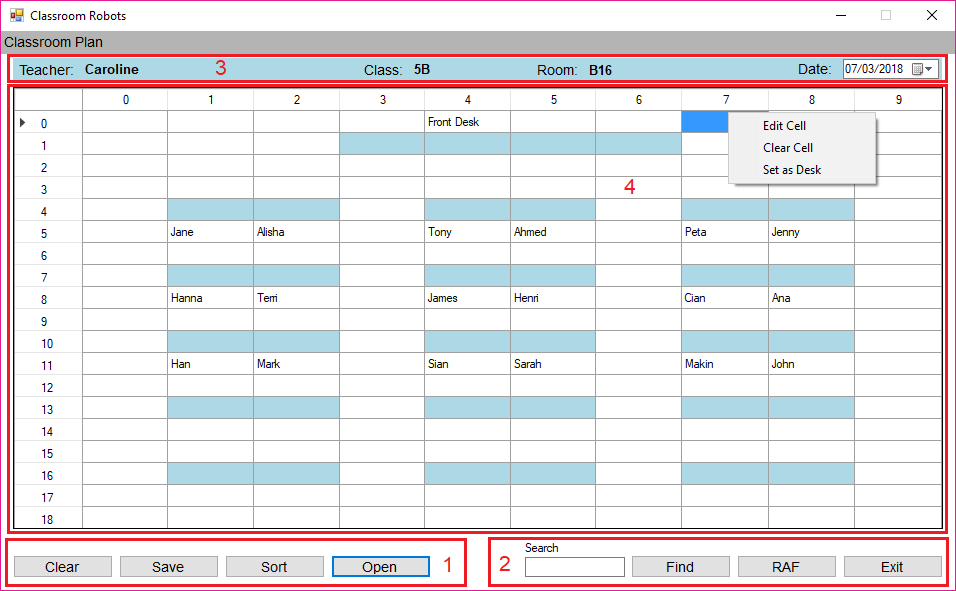


Image 1.0 - User Interface 1

* Image 1.0 – Section 1: This section contains buttons.

1. Clear – Clears the students and desks.
2. Save – Saves the classroom to a new csv file via a save file dialog.
3. Sort – Sorts the students’ names in the classroom and displays their names and locations in a new window.
4. Open – Opens a new csv file via an open file dialog.

* Image 1.0 – Section 2: This section contains buttons and a textbox to search.

1. Find – Finds the students name via a binary search of the sorted student list window and highlights the cell in the main application and row in the student list.
2. RAF – Reads the RAF file and find student at entered index, highlights cell in classroom and row in student list.
3. Exit – Exits the application.

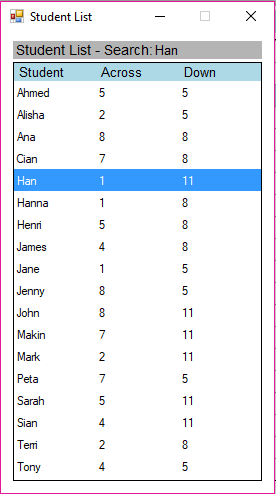


Image 2 - Student List Window 1

* Image 1.0 – Section 3: This section contains the classrooms details.

1. Teacher’s Name
2. Class ID
3. Room ID
4. Date Picker – The day the class was on.

* Image 1.0 – Section 4: This section contains the classroom layout and context menu.

1. Blue cells – Represents a desk. This can be altered using the right click context menu.
2. Cells that contain names – Students name. This can be altered using the right click context menu.
3. The context menu contains – Edit Cell, Clear Cell and Set Desk options.

## Toe Chart

|  |  |  |
| --- | --- | --- |
| **Task** | **Object** | **Event** |
| End the application. | buttonExit | Click |
| Reads the raf file that’s opened and searches for index entered in search box, displays sorted student list (popup), with index highlighted. | buttonRAF | Click |
| Find a student in the sorted list via a binary search.  Highlight the row in the sorted student list. | buttonFind | Click |
| Displays a sorted list (popup) of students’ names and locations. | buttonSort | Click |
| Saves the classroom details and students’ cell location and name to a csv file via a savefiledialog. | buttonSave | Click |
| Opens a csv file via an openfiledialog and populates the datagridview and textboxes. | buttonOpen | Click |
| Clear the students from the grid. | buttonClear | Click |
| Shows a context menu via right click on cell of dataGridClassroom. | dataGridClassroom | MouseClick |
| Sets a cell to a desk. | AddDesk | Click |
| Clears the cell. | ClearCell | Click |
| Edits the cell. | EditCell | Click |
| Loads an empty classroom. | Form1 | Load |
| Loads the students name and locations from the datagridview in form1 to a new datagridview in a list style. | Form2 | Load |

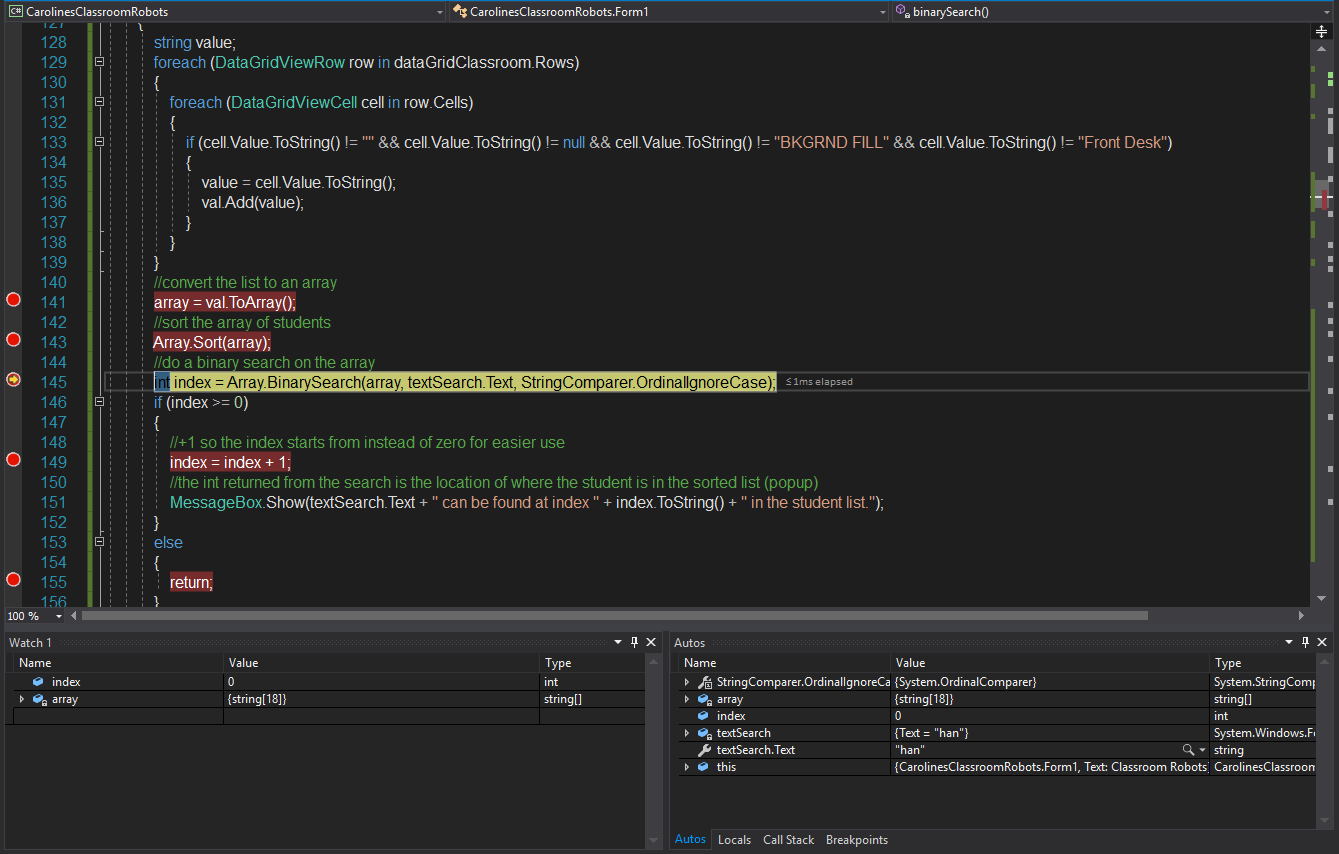
## UML Diagrams

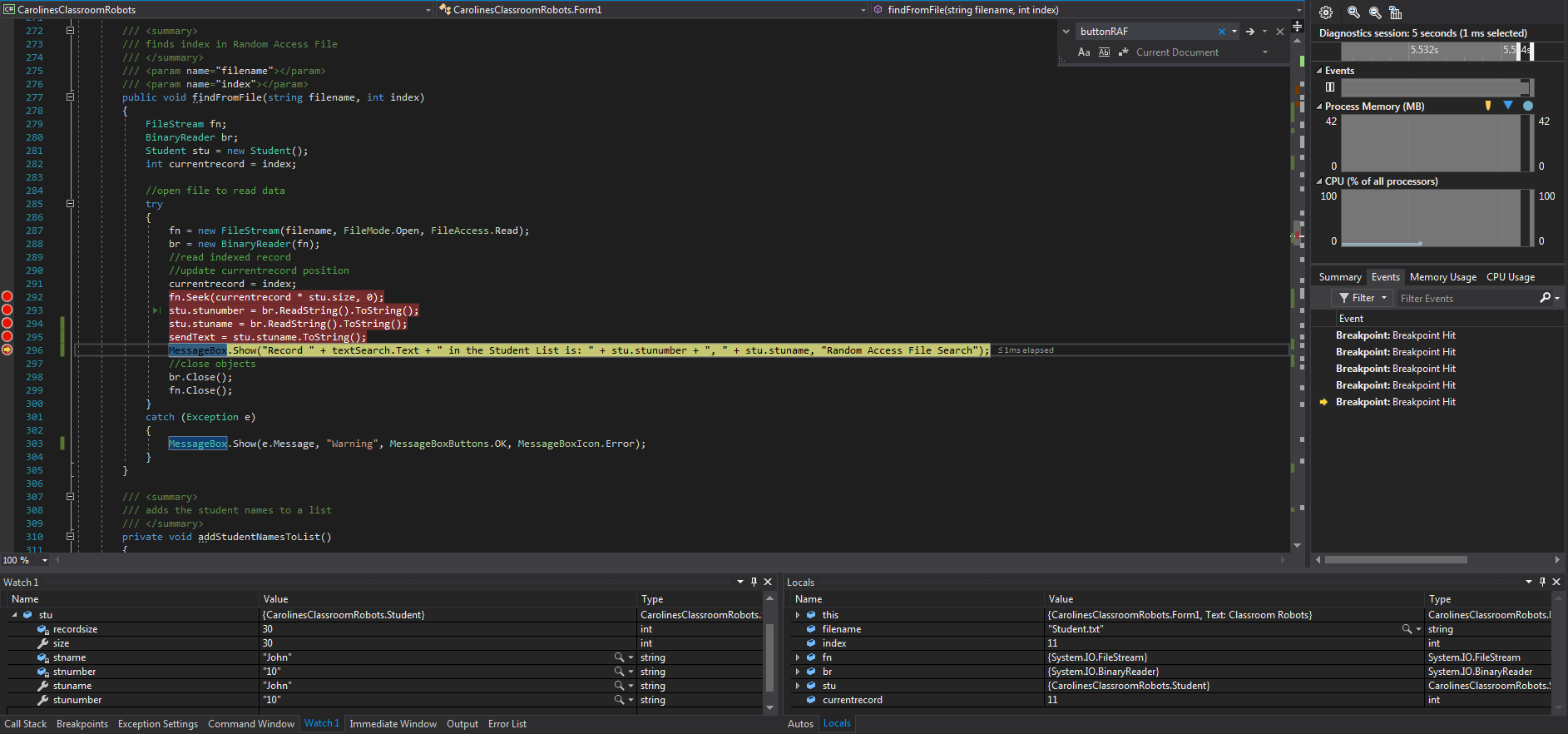
|  |
| --- |
| **Form 1** |
| * sendText: Public static string * pathToFile: Private string * stringVal: List<string> * studentNames: List<string> * fields: ArrayList * rowIndex: Private int * colIndex: Private int * array: String[] * RecSize: int * pos: int |
| * buttonExit\_Click(object sender, EventArgs e) * buttonRAF\_Click(object sender, EventArgs e) * buttonFind\_Click(object sender, EventArgs e) * buttonSort\_Click(object sender, EventArgs e) * buttonSave\_Click(object sender, EventArgs e) * buttonOpen\_Click(object sender, EventArgs e) * buttonClear\_Click(object sender, EventArgs e) * Form1\_Load(object sender, EventArgs e) * dataGridClassroom\_MouseClick(object sender, MouseEventArgs e) * AddDesk\_Click(object sender, EventArgs e) * ClearCell\_Click(object sender, EventArgs e) * EditCell\_Click(object sender, EventArgs e) * openFileDialog() * saveFileDialog() * binarySearch() * addClassroomToList() * classroomDetails() * search() * clearDetails() * clearStudents() * addStudentNamesToList() * displayAllData() * disableButtons() * addrecord() * writeToFile(string filename, Student obj, int pos, int size) * readFromFile(string filename) * findFromFile(string filename, int index) * highlightCell(string searchValue) * setRowNumber(DataGridView dgv) * populateGrid(string path) * displayLine(int lineNum) * textSearch\_KeyPress(object sender, KeyPressEventArgs e) |

|  |
| --- |
| **Form 2** |
| * students: list<string> |
| * Form2\_Load(object sender, EventArgs e) * addListToDataGrid() * highlightRow(string searchValue) |

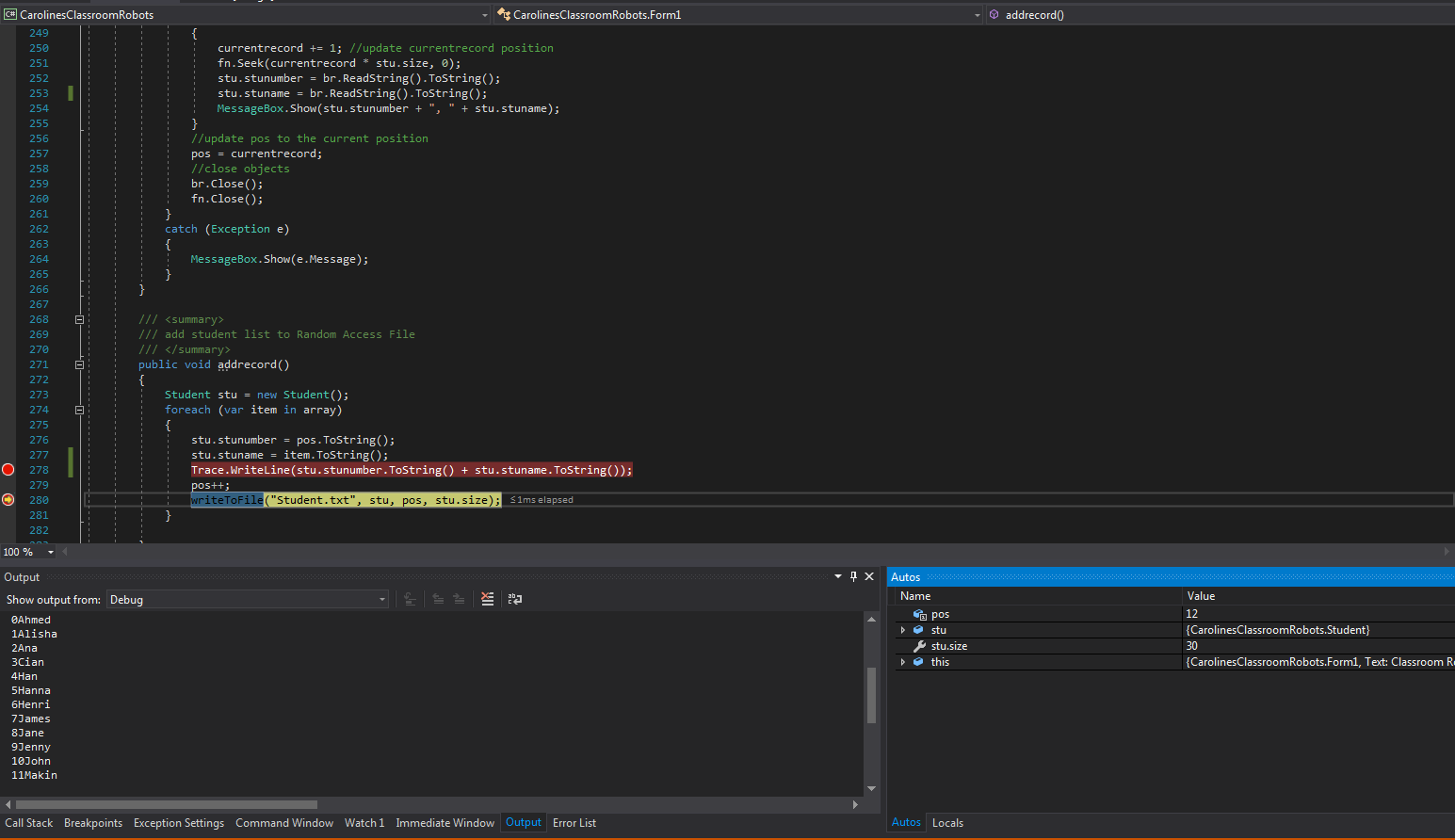
|  |
| --- |
| Student |
| stnumber: private string  stname: private string  recordsize: private int  stnumber: public string  stname: public string  size: public int |
| calcSize() |

Debugging**Figure 1 – Debugging on Binary Search**



****

**Figure 2 - Debugging on finding a record in RAF**

****

**Figure 3 - Tracing through adding record to RAF**