**GROUP 12 MINI PROJECT REPORT**

**DEMONSTRATION OF SORTING ALGORITHMS ON ARRAY**

***(BUBBLE SORT, QUICK SORT, INSERTION SORT)***

**I. Assignment of members.**

|  |  |
| --- | --- |
| Nguyễn Huy Hoàng – 20194433 | Lê Đức Anh - 20194416 |
| - Feasibility Study (50%)  - Requirement Analysis (50%)  - Design:  + Component Design:   * Use case diagram (50%) * Class diagram (50%)   + UI Design:  SortingDemoScreen.java  + Implementation:   * BubbleSort.java * InsertionSort.java * QuickSort.java * ISortAlgorithm.java * SortArray.java   + Testing (50%)  - Writing Report. | - Feasibility Study (50%)  - Requirement Analysis (50%)  - Design:  + Component Design:   * Use case diagram (50%) * Class diagram (50%)   + UI Design:  MainMenuScreen.java  + Implementation:   * Input.java * Modify SortingDemoScreen.java to fit MainMenuScreen.java   + Testing (50%)  - Preparing Presentation Slide. |

**II. Mini-project description.**

- Description:

Array is the most basic structure of computer science. Most operations as well as other data structures are built and performed on array. In this project, we will make an application in order to explain three sorting algorithms on array: bubble sort, quick sort and insertion sort.

- Requirements:

+ Must use pure array for the main data structure.

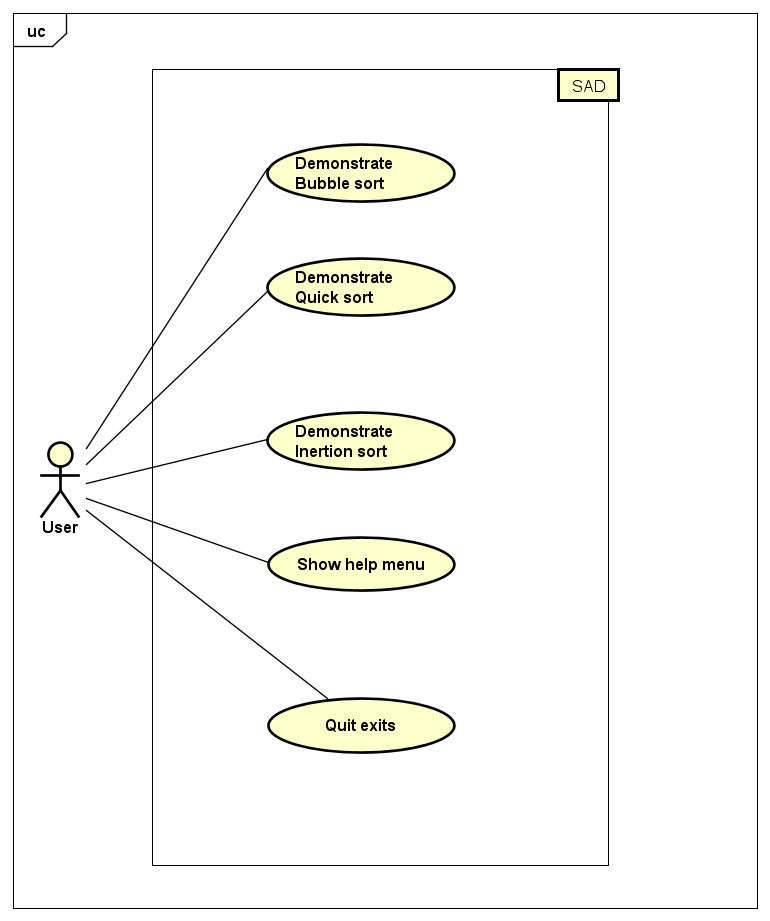
+ On the main menu:

* 3 types of sort algorithms to select.
* Help menu show the basic usage and aim of the program.
* Quit exits the program (must ask for confirmation).

+ In the demonstration:

* User can choose to input an array or create an array randomly.
* A back button to return to the main menu at any time.

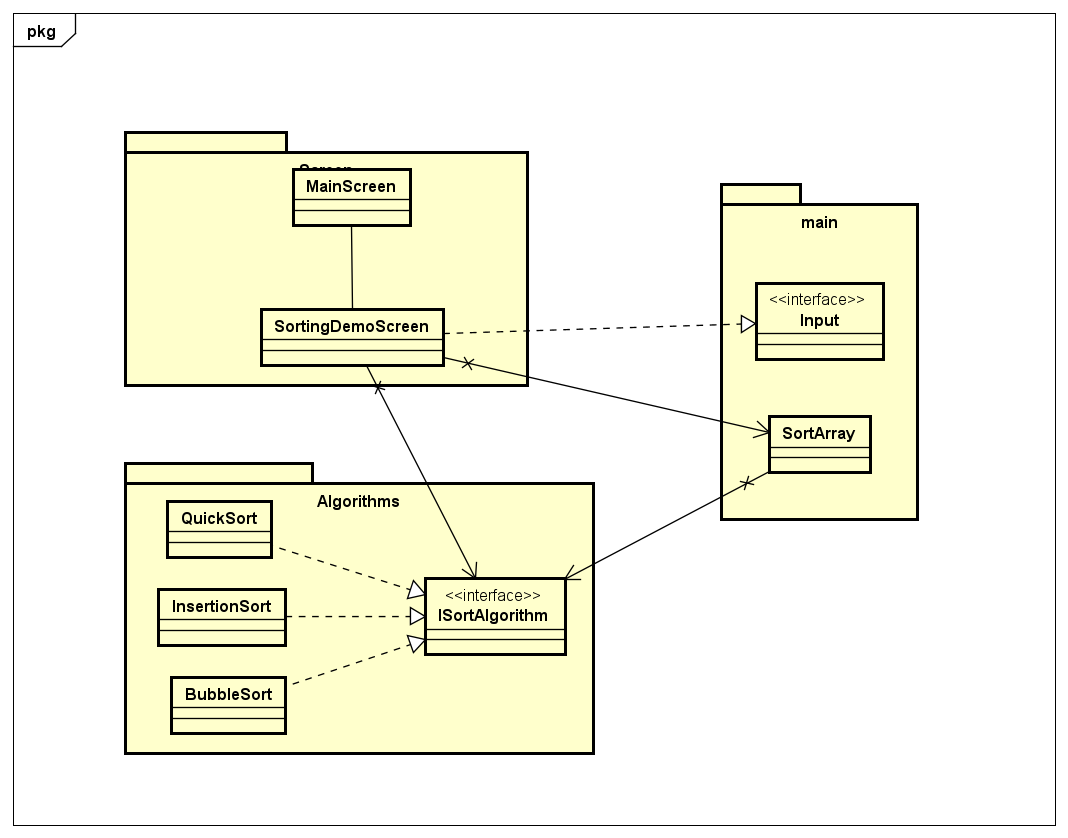
- Use case diagram:



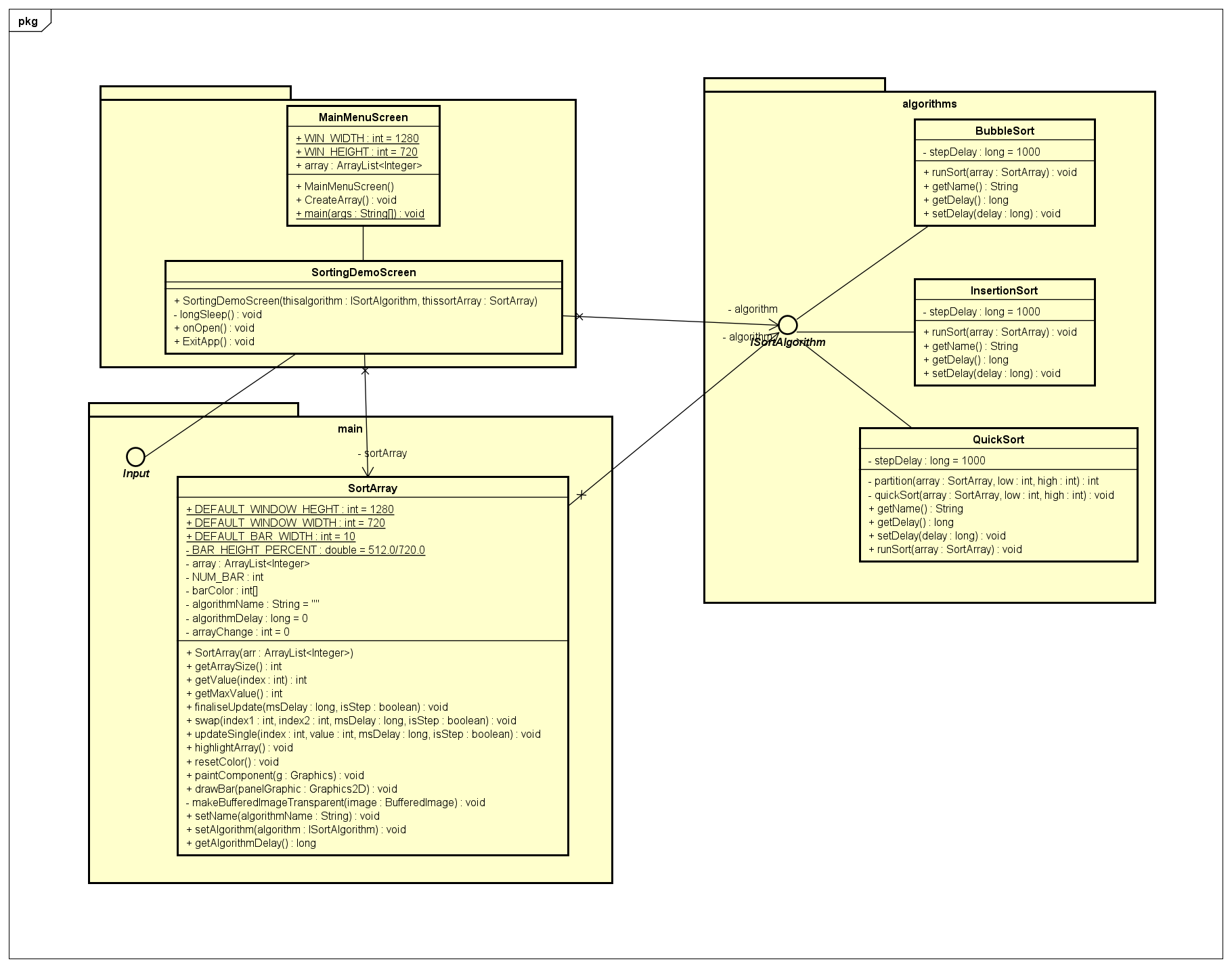
The user will have 5 options in the main menu. He/she can choose to see the information about the program and how to use it in the Help menu. They can also close the program by choosing Exit. After understand how the program works, user can pick the type of algorithm he/she want to have a clearer understanding. Then, they can choose from create an array randomly or input an array manually. Finally, the demonstration will start.

**III. Design.**

1. General class diagram:



2. Detailed class diagram:



Relationship between classes:

BubleSort, InsertionSort, QuickSortSort implements ISortAlgorithm

SortArray associate to ISortAlgorithm (SortArray: Non-navigable, ISortAlgorithm: Navigable)

SortingDemoScreen associate to ISortAlgorithm (SortingDemoScreen: Non-navigable, ISortAlgorithm: Navigable)

SortingDemoScreen associate to SortArray (SortingDemoScreen: Non-navigable, SortArray: Navigable)

SortingDemoScreen associate to Input (Unspecified)

MainMenuScreen associate to SortingDemoScreen (Unspecified)

Explanation:

The BubbleSort, InsertionSort and QuickSort implements the ISortAlgorithm interface and the runSort() method will take the instance of the SortArray class as an parameter.

In the SortArray class, there will be some the method like swap(), updateSingle() so that the above algorithms can use in the sorting process. Moreover, this class also override the paintComponent() method with the drawBar() and drawBufferedImageTransparent() to visualize the array. We use Thread.sleep() method at each time there is a change in the array and repaint the array.

The SortingDemoScreen constructor will have two parameter, they are the instance of ISortingAlgorithm (to know which algorithm it will demonstrate) and the instance of SortArray (to paint the array). It also connects to the Input interface to take the array (whether it is created manually or randomly).

MainMenuScreen will display the main menu to choose including choose type of algorithm to visualize, help and exits.

**IV. References:**

|  |  |  |
| --- | --- | --- |
| Name | Purpose | Link |
| VisualAlgo (website) | Design idea, graphic idea | https://www.notion.so/VisualAlgo-website-918de1237ff948ddb2eebf20b3881914 |
| Animation (Zetcode) | Animation implementation | https://zetcode.com/javagames/animation/ |
| Sort-Algorithm-Visualizer (Hopson97) | Graphic design, algorithms and animation (modify source code) | https://github.com/Hopson97/Sort-Algorithm-Visualiser |
| Class BufferedImage | Graphic deisgn | <https://docs.oracle.com/javase/7/docs/api/java/>  awt/image/BufferedImage.html |
| AWT Graphics2D Class | Graphic design | <https://www.tutorialspoint.com/awt/>  awt\_graphics2d\_class.htm |