**Chapter 1**

**Introduction**

Java is a general purpose computer programming language that is concurrent, class based, and object–oriented and architecturally neutral. It is a blend of the best elements of its rich heritage combined with the innovation concepts required by its unique mission.

This application simulates the sorting algorithms that generates random array size with respect to the button selected. On button press respective algorithm get selected and pressing the sort button selected algorithm starts to execute. This application uses the java concepts of swing applets, package, interfaces and event handling.

**1.1 Problem Statement**

To design and develop a java based GUI Application for theVisualization of Working of Sorting Algorithms

**Chapter 2**

**System Analysis**

System analysis is the process of observing system for troubleshooting or development purpose. The development of a computer based information system includes a system analysis phase. This helps to produce the data model, a precursor to creating or enhancing a database. When a computer based information system is developed, system analysis would constitute the steps such as development of feasibility study, the fact-finding measures for the requirements of end users and gauging how the end user would operate the system.

System Analysis for the present application:

1. The project is a java GUI based application.

2. It uses the concepts of java swings applets and event handling.

3. Demonstrates an interactive application for sorting algorithm.

4. This application generates the sorting demo when corresponding buttoms are clicked.

**2.1 System Requirements Specification**

System Requirements means a detailed description of a what system should do and it sets out the system’s functions, services and operational constraints in details. The System Requirements Specification should be precise. It should define exactly what is to be implement .

**2.1.1 Functional Requirements**

These are statements of services the system should provide, how the system should behave in particular situation.

1. The system shall provide the user to select the various Sorting Algorithms to Demonstrate or Visualize using Buttons

2. The system should enable the user to visualize various sorting algorithms.

**2.1.2 Non Functional Requirements**

These are constraints on the services or function offered by the system. They include timing constraints, constraints on the development proposes and standards. It includes hardware and software requirements.

**SOFTWARE REQUIREMENTS**

* + Operating system: Windows 7 or higher
  + Text Editor: Sublime Text Editor and Visual Studio code
  + IDE: Eclipse
  + Java Development kit 8
  + Java Runtime Environment 8

**HARDWARE REQUIREMENTS**

* A Personal Computer with Pentium Processor or above with 4GB or more RAM
* 250GB or more Hard Disk Drive (HDD)
* Color monitor
* Mouse
* Keyboard

**Chapter 3**

**System Design**

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements.

**3.1 Data Flow Diagram**

It describes the flow of data and the process that changes the data throughout a system. It is constructed using set of symbols that do not imply physical implementations.

**3.1.1 Level-0 DFD**

Level-0 DFD sees the whole system as a single process and emphasis the interaction between the system and external entities.

Figure 3.1.1 : Level-0 DFD

USER

Graphic visualization of sorting code

**3.1.2 Level- 1 DFD**

Figure 3.1.2: Level-1 DFD

Button selected

Bar graph

Sorting demo

**3.2 System Architecture**

Large systems are always decomposed into sub-systems that provide some related set of services. The initial design process of identifying these sub-systems and establishing a framework for sub-system control and communication is called architectural design. The output of this design process is a description of the software architecture.

**Chapter 4**

**Implementation**

It is an important stage where the defined procedures are transformed in to control specifications with the help of computer language.

Its primary goal is to write the source code and supporting documentation. A frame is initialized as the basic container for all other components added further.

An executable jar file is created to enable execution of the application from a desktop.

1. JLabel

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

1. JButton

The class JButton is an implementation of a push button. This component has a label and generates an event when pressed. It can also have an image.

1. JTextField

The object of a JTextField class is a text component that allows the editing of a single

line text. It inherits JTextComponent class.

**CODE**

Private void initialize () {

Frame = new JFrame ();

frame.getContentPane ().setForeground (Color.WHITE);

frame.setBounds (100, 100, 794, 429);

frame.setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane ().setLayout (null);

JButtonbtnNewButton = new JButton ("C");

btnNewButton.setForeground (Color.BLACK);

btnNewButton.setBackground (Color.WHITE);

btnNewButton.addActionListener (new ActionListener () {

**if**(sort)

{

String whichSort = "";

**for**(**int** i = 0; i<radioButtons[1].length;i++)

**if**(radioButtons[1][i].isSelected())

whichSort = radioButtons[1][i].getText();

//if selection sort has been chosen call it

**if**(whichSort.equals("Selection"))

{

Thread t = **new** selectionSortThread();

t.start();

//selectionSort(g);

}

//if bubble sort has been chosen call it

**else** **if**(whichSort.equals("Bubble"))

{

Thread t = **new** bubbleSortThread();

t.start();

//bubbleSort(g);

}

//if merge sort has been chose call it

**else** **if**(whichSort.equals("Merge"))

{

Thread t = **new** mergeSortThread();

t.start();

//numbers = mergeSort(numbers, 0, g);

}

**else** **if**(whichSort.equals("Insertion"))

{

Thread t = **new** insertionSortThread();

t.start();

}

sort = **false**;

} btnNewButton.setFont (new Font ("Tahoma", Font.BOLD, 22));

btnNewButton.setVerticalAlignment (SwingConstants.BOTTOM);

btnNewButton.setBounds (10, 202, 51, 177);

frame.getContentPane ().add (btnNewButton);

**Chapter 5**

**System Testing**

Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifying a software program or application or product.

Unit testing is a software development process in which the smallest testable parts of an application called units are individually and independently scrutinized for proper operation.

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group.

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

**5.1 Test Case and Result**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Case Title** | **Test Condition** | **System Behavior** | **Expected Result** |
| T01 | Click GET START | Use of Start key | Application should start running | Application runs |
| T02 | Select preferred algorithm button | Use of algorithm button | Application should sort the array in the selected algorithm. | Application sorts the array. |
| T03 | Click on stop button | Use of stop button | Application should stop the sorting process | Application stops |
| T04 | Click on step button | Use of step button | Application should single steps | Application shows single step. |

**Chapter 6**

**User Manual and Snapshots**

**6.1 User Manual**

1. Click the adh.jar executable file (App icon) on the desktop to open the application.

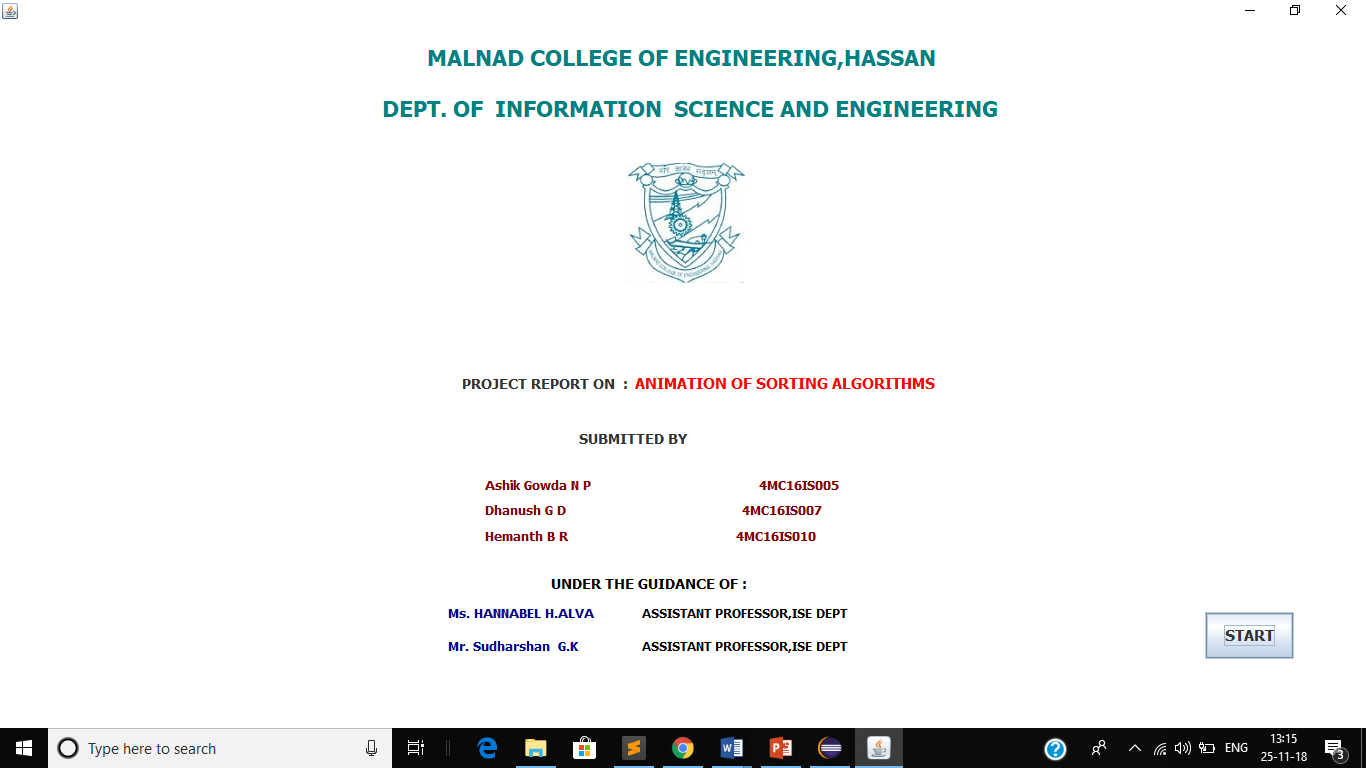
2. Click on “get start” button.

3. Select any button corresponding to the type of sorting algorithms.

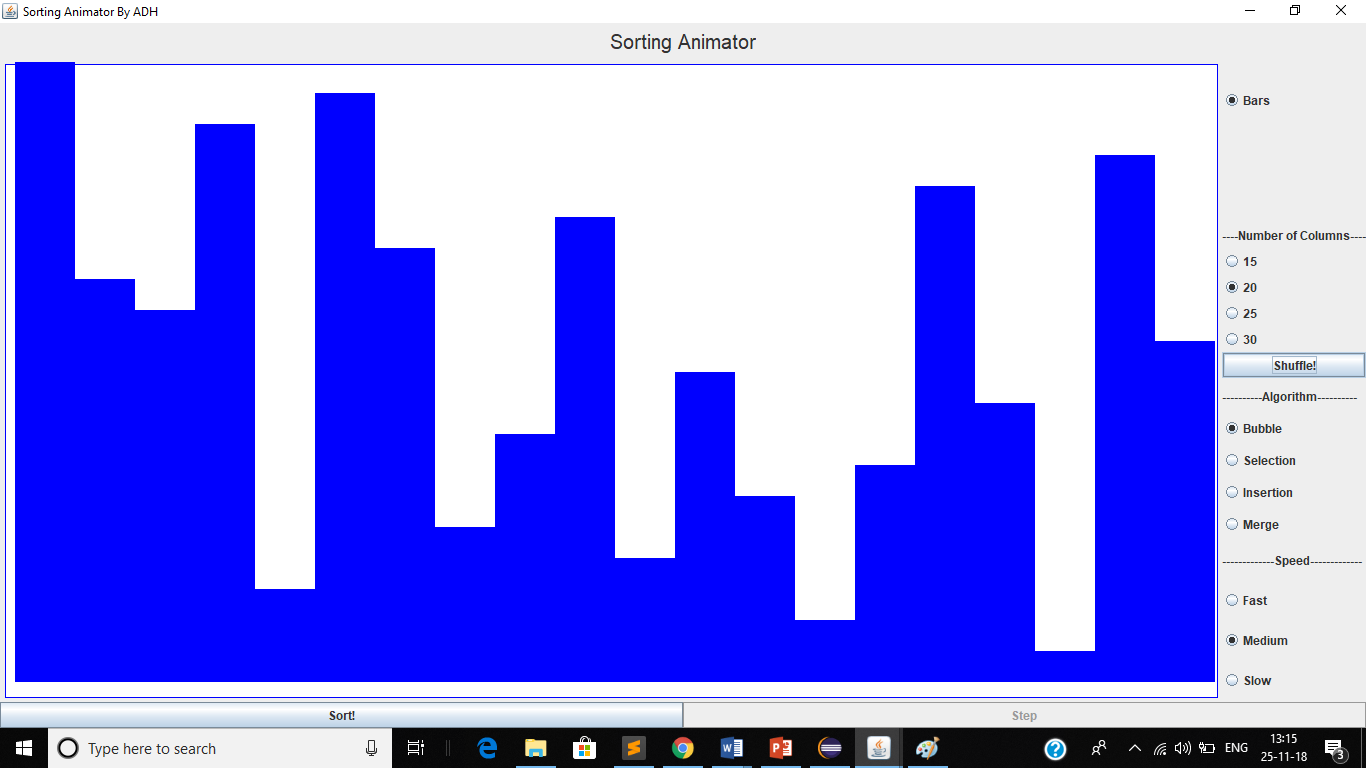
4. Select exit to close the application.

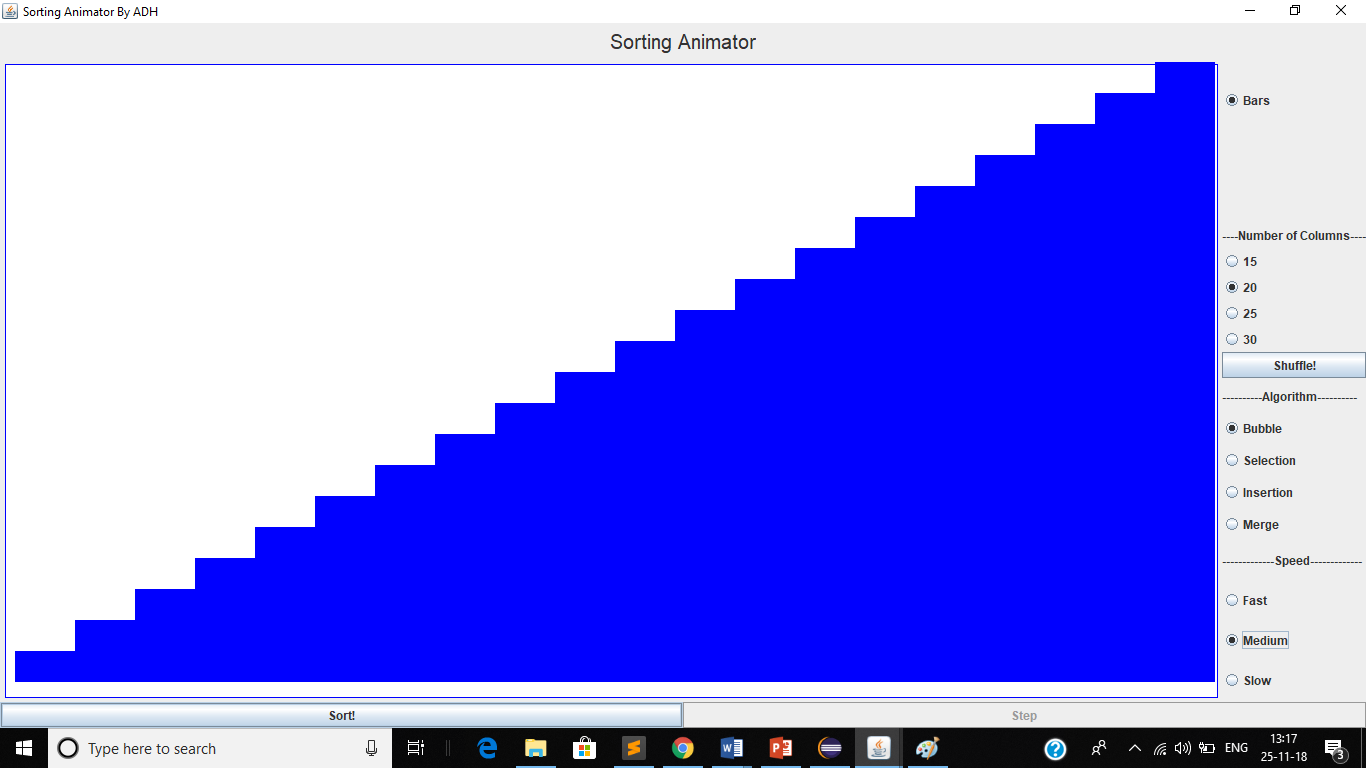
**6.2 Snapshots**

6.2.1 Welcome Page



6.2.2 Working Model



****

**Chapter 7**

**Conclusion**

This project provides an interactive interface for the users who wish to visualize the sorting algorithm.

**Reference**

* Herbert Schildt, “Java The Complete Reference”, 9th Edition, McGraw-Hill Education, 2014.
* Ian Sommerville, “Software engineering”, 9th edition, Pearson Education, 2014.
* W3Schools -https://www.w3schools.com/java/java\_intro.asp
* Stack Over Flow-https://stackoverflow.com/search?q=learn+java
* YouTube-https://www.youtube.com/watch?v=Av6zh817QEc