|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Version 8.1) | | |
| **Qualification National Code and Title** | ICT50220 Diploma of information Technology (Release 2) | **State code** | BGJ4 |
| **Assessment Title** *(as per DAP)* | Assessment Project One (Individual Project) | | |
| **Unit National Code & Title** | ICTPRG535 Build advanced user interfaces | | |
| ICTPRG547 Apply advanced programming skills in another language | | |
| ICTICT517 Match ICT needs with the strategic direction of the organisation | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date Due** | Week Five | | **Date Received** | | 28/02/2023 | |
| **Student Name** | Satbir Singh | | | | | |
| **Student Declaration** | I declare that the evidence submitted is my own work: | | | | | |
| **Assessor Name** |  | | | | | |
| **Assessment Decision** | Satisfactory | | | Not Yet Satisfactory | | |
| **Assessor Signature** |  | | | **Date** | |  |
| **Is student eligible for reassessment (Re-sit)?** | No | Yes | | **Re-assessment Date:** | | Week Twenty |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feedback to student** | | | |
| *Via Blackboard (LMS) – Please check [Grade] section.* | | | |
| **Feedback from student** | | | |
| *Via Blackboard (LMS) – Please use [Comment] section during submission.* | | | |
| **Student signature** |  | **Date** |  |

|  |  |
| --- | --- |
| **Assessment Instructions** | |
| **TO THE ASSESSOR** |  |
| Type of Assessment | Individual Project |
| Duration of the assessment | 5 class sessions (Weeks 1 - 5) |
| Location of assessment | Classroom |
| Conditions | Assessor to ensure that the noise levels, natural interactions and time variances are maintained as it would be in the Software Development industry.  Learners are required to complete the required tasks in class and submit the required documentation electronically via Blackboard |
| Elements and Criteria | As detailed in the assessment plan  You are required to make sure that all students meet the elements, performance criteria and oral communication items as outlined in the provided solution |
| **TO THE STUDENT** |  |
| Purpose of Assessment | You are required to show you can:  ICTPRG5335 Build advanced user interfaces   * Plan and design a UI solution according to organisational requirements, * Apply interactions designs and implement validation requirements against the design plan, * Create and display the UI with graphics according to UI requirements.   ICTPRG547 Apply advanced programming skills in another language   * Code advanced data structures using hashing, sorting and searching algorithms, * Apply third party libraries and communication technologies for data exchange, * Test and evaluate the code to resolve logical and syntactical designs flaws, * Create and document the application according to technical specifications.   ICTICT517 Match ICT needs with the strategic direction of the organisation   * Interpret, analysis and report the strategic organisational plan * Propose and document changes for the implementation of a ICT system * Provide action plan and schedule   The student must demonstrate the ability to complete the tasks outlined in this assessment and is expected to use systematic analytical processes and effect time management to meet the goals/deadlines outlined in the DAP. |

|  |  |
| --- | --- |
| Allowable Materials | Blackboard Topics, SDLC, Weekly readings (PDF), Example programs and Independent Outside of Class Activities |
| Required Resources | Web links and example code can be downloaded from the Blackboard portal.  PC with Notepad++, Visual Studio, GitHub, MSOffice.  Internet Access to GitHub and www.citems.com.au/ |
| Reasonable Adjustment | In some circumstances, adjustments to assessments may be made for you. If you require support for literacy and numeracy issues; support for hearing, sight or mobility issues; change to assessment times/venues; use of special or adaptive technology; considerations relating to age, gender and cultural beliefs; format of assessment materials; or presence of a scribe you need to inform your lecturer. |
| Assessment Submission | All questions and programming activities must be attempted. All written answers must be submitted in this assessment document in the appropriate space.  Use of research tools and peers in formulating answers are acceptable – but work submitted must be your own work.  Final project documentation is to be uploaded to the appropriate area in the Blackboard course created for this unit.  If you are marked as NYS (Not Yet Satisfactory) on your first attempt, you will be provided with another opportunity to re-attempt the assessment. |
| Portfolio Description | A project of web coding tasks and written questions which should be completed in class and finished in the students’ own time on a weekly basis as per the Delivery and Assessment schedule.  Question 1 – Design Specification  Question 2 – Version Control  Question 3 – Design Approval  Question 4 – Application Development  Question 5 – Debug and Testing  Question 6 – Evaluation  Question 7 – Assessment, Feedback and Signoff |

# Scenario

You are employed as the Senior Programmer with CITE Managed Services and have been assigned the Satellite Data Processing Project for an organisation called Malin Space Science Systems (MSSS). This project will require the planning, development, testing and reporting of an application that will sort and search complex data sets recorded during satellite operations. The senior scientists at Malin would like two sorting and two searching algorithms so they can compare the processing of raw data as it is collected. This application will utilise two advanced sorting algorithms and two binary searching techniques. The details and criteria are provided in the following paragraphs.

## Background Information

Marlin Space Science Systems operate a satellite with a Sun-synchronous orbit, which means they can record data from the same spot at the same local time each day. The advantage for scientists means they can compare data from a fixed location over a long period of time to detect changes. The data is collected as raw numbers from two sensors on the satellite as it passes over the receiving station. This stream of numbers forms a Normal Distribution and is then transmitted and saved onto a server for further examination. The processing occurs when each data stream is read into the application using a mathematical formula embedded in a Dynamic Link Library called Galileo. Currently the scientists use a 32bit command line application which is executed from a console, this old software is not able to run successfully on new multi-platform hardware. There has been a series of recordings that are outside the expected range of normal readings and the scientists at Malin require a replacement application which could provide accurate information. Therefore, the organisation has requested a .NET Multi-platform App UI that is compatible with contemporary hardware. The analysis team have consulted with Marlin and provided an interface layout for the display of satellite information with the required features.

Graphical user interface

Description automatically generated Chart, histogram

Description automatically generated

Ref: <https://en.wikipedia.org/wiki/Sun-synchronous_orbit>

Ref: <https://www.esa.int/ESA_Multimedia/Images/2020/03/Polar_and_Sun-synchronous_orbit>

## Application Requirements

The Marlin scientists require a .NET WPF App (refer Appendix) using a C# code base to process the data using the Galileo DLL and two LinkedList’s; one for each satellite sensor. The data from each sensor must be read into a simple LinkedList of type “double” and then sorted using a Selection and Insertion sort algorithm. Next, the scientist will enter an integer value into a search textbox and select a Recursive or Iterative binary search algorithm. Finally, for each of the four algorithms the processing time must be measured and displayed.

The Marlin scientists would like buttons for each action and several display options to examine the raw data and processed information. The application must have a single form that shows all the information from the two sensors. The following features are mandatory:

* A List Box to display the data from each Sensor.
* A List View with two columns to display the data from both sensors (Sensor A and Sensor B).
* A numeric input to control the standard deviation (Sigma) and mean (Mu) of the data.
* A button to load data into the two LinkedLists and then display these LinkedLists in a List View.
* A series of buttons to process the sort and search functions.
* Textbox display for the processing times each sort and search.

You should consult with the CITE representative (your Lecturer) if you are unsure about any of the problems or questions in this assessment. Your primary research should focus on the resources on the Blackboard LMS and CITE web site, additional information can be collected from the Internet, ensure all sources are referenced in your submission. Ensure your development follows an Agile methodology that is recorded and maintained using your GitHub account. You must demonstrate your working applications before uploading to Blackboard, your Lecturer (Assessor) will sign off to ensure all the criteria are satisfied.

## Question 1 Design Specification

Provide a suitable description/explanation for each client requirement, and then complete the UI design specification which outlines all the user interactions that occur when buttons and related components are clicked or selected. Complete the following Design Specification template to answer this question.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Design Specification | | | | |
| Developer Name | Satbir Singh | | Date | 06/02/2023 |
| Client Technical Requirements | | | | |
| Requirements | | Description | | |
| 1. Galileo DLL | | The Dynamic Link Library must be used for processing data and formula should be used to form a Normal Distribution that is in DLL. | | |
| 1. Two Linked Lists | | Two Linked Lists should be used for processing data for each sensor, because client wants to compare the collected data. | | |
| 1. Sorting Data | | Data should be sorted using Selection Sort and Insertion sort | | |
| 1. Searching Data | | Binary search iterative and Recursive should be used for searching for data. | | |
| 1. Time Measurements | | For selection and insertion sort time counter should be used and display the time taken for performing sorting. | | |
| 1. Multi-Platform | | This App should be compatible with multi-platform hardware as client’s current app is not compatible with new hardware platform | | |
| 1. Display and Input | | Appropriate List View, List Box, buttons and text boxes should be used for Input and Output of Data. So, it is easy to use. | | |
| UI Design Specification  What are the User Interactions when a form component is clicked or select? | | | | |
| UI Action | | Description | | |
| Numeric Up Down | | Two Numeric Ups and Down will be provides one for standard deviation and other for mean that user need to select | | |
| Load Data Button | | Based on numbers provided in numeric up and down if user will click on this button, then it will load the data using DLL. | | |
| List View | | A list view with two columns one for each sensor will display the data collected from DLL based on the standard deviation and mean. | | |
| Search Buttons | | Two search clickable buttons (Iterative and Recursive) for each Sensor will be provided with a search target text box for inputs and two non-editable text boxes for displaying the ticks or time to perform the search. | | |
| Display Sensor’s Data | | Two List boxes one for each sensor will display the data individually for performing the sort and search operations on it. | | |
| Sort Buttons | | Two search buttons for each Sensor will be given to perform sorting. | | |

## Question 2 Version Control

Malin Space Science Systems would like you to use GitHub as the primary source control, setup an appropriate structure in your GitHub account to manage the Satellite Data Processing Project development. Add a project to your repository which reflects the basic Agile development process you intend to pursue. Complete the following GitHub Version Control template to answer this question.

|  |  |  |  |
| --- | --- | --- | --- |
| GitHub Version Control | | | |
| Repository Name: | Space-Science-System-WPF-App | | |
| URL | https://github.com/satbircoder/Space-Science-System-WPF-App.git | Date | 06/02/2023 |
| Screen Shot(s) |  | | |

## Question 3 Design Approval

Once you have complete questions 1 & 2 arrange for your document to be reviewed by the Lecturer/Assessor for approval, sign off and feedback before completing the development and testing.

* Question 1 Design Specification
* Question 2 Version Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Design Approval (Lecturer/Assessor use only) | | | | |
| Approver Name | Title | Signature | Date | Approved? |
| Stuart | Lecturer |  |  |  |
|  |  |  |  |  |
| Lecturer Feedback | | | | |
|  | | | | |

## Question 4 Application Development

When you have received approval from the client (or Lecturer) your next task is to create the .NET Multi-platform App UI which satisfies the following criteria and uses the supplied Galileo dll. The exact requirements are critical as the Malin scientists are using this application to determine anomalies in the data collected over the Mount Etna volcano in Italy. As you develop the application ensure you debug and test each algorithm to ensure compliance with the client’s criteria.

**Important:** You can only declare two global variables which represent the LinkedLists for Sensor A and Sensor B. You are not permitted to use any other global variables; all methods should utilise input parameters and returns types in order to process information between methods.

**Global Methods**

1. Create two data structures using the LinkedList<T> class from the C# Systems.Collections.Generic namespace. The data must be of type “double”; you are not permitted to use any additional classes, nodes, pointers or data structures (array, list, etc) in the implementation of this application. The two LinkedLists of type double are to be declared as global within the “public partial class”.
2. Copy the Galileo.DLL file into the root directory of your solution folder and add the appropriate reference in the solution explorer. Create a method called “LoadData” which will populate both LinkedLists. Declare an instance of the Galileo library in the method and create the appropriate loop construct to populate the two LinkedList; the data from Sensor A will populate the first LinkedList, while the data from Sensor B will populate the second LinkedList. The LinkedList size will be hardcoded inside the method and must be equal to 400. The input parameters are empty, and the return type is void.
3. Create a custom method called “ShowAllSensorData” which will display both LinkedLists in a ListView. Add column titles “Sensor A” and “Sensor B” to the ListView. The input parameters are empty, and the return type is void.
4. Create a button and associated click method that will call the LoadData and ShowAllSensorData methods. The input parameters are empty, and the return type is void.

**Utility Methods**

1. Create a method called “NumberOfNodes” that will return an integer which is the number of nodes(elements) in a LinkedList. The method signature will have an input parameter of type LinkedList, and the calling code argument is the linkedlist name.
2. Create a method called “DisplayListboxData” that will display the content of a LinkedList inside the appropriate ListBox. The method signature will have two input parameters; a LinkedList, and the ListBox name. The calling code argument is the linkedlist name and the listbox name.

**Sort and Search Methods**

1. Create a method called “SelectionSort” which has a single input parameter of type LinkedList, while the calling code argument is the linkedlist name. The method code must follow the pseudo code supplied below in the Appendix. The return type is Boolean.
2. Create a method called “InsertionSort” which has a single parameter of type LinkedList, while the calling code argument is the linkedlist name. The method code must follow the pseudo code supplied below in the Appendix. The return type is Boolean.
3. Create a method called “BinarySearchIterative” which has the following four parameters: LinkedList, SearchValue, Minimum and Maximum. This method will return an integer of the linkedlist element from a successful search or the nearest neighbour value. The calling code argument is the linkedlist name, search value, minimum list size and the number of nodes in the list. The method code must follow the pseudo code supplied below in the Appendix.
4. Create a method called “BinarySearchRecursive” which has the following four parameters: LinkedList, SearchValue, Minimum and Maximum. This method will return an integer of the linkedlist element from a successful search or the nearest neighbour value. The calling code argument is the linkedlist name, search value, minimum list size and the number of nodes in the list. The method code must follow the pseudo code supplied below in the Appendix.

**UI Button Methods**

1. Create four button click methods that will search the LinkedList for an integer value entered into a textbox on the form. The four methods are:
   1. Method for Sensor A and Binary Search Iterative
   2. Method for Sensor A and Binary Search Recursive
   3. Method for Sensor B and Binary Search Iterative
   4. Method for Sensor B and Binary Search Recursive

The search code must check to ensure the data is sorted, then start a stopwatch before calling the search method. Once the search is complete the stopwatch will stop, and the number of ticks will be displayed in a read only textbox. Finally, the code/method will call the “DisplayListboxData” method and highlight the search target number and two values on each side.

1. Create four button click methods that will sort the LinkedList using the Selection and Insertion methods. The four methods are:
   1. Method for Sensor A and Selection Sort
   2. Method for Sensor A and Insertion Sort
   3. Method for Sensor B and Selection Sort
   4. Method for Sensor B and Insertion Sort

The button method must start a stopwatch before calling the sort method. Once the sort is complete the stopwatch will stop, and the number of milliseconds will be displayed in a read only textbox. Finally, the code/method will call the “ShowAllSensorData” method and “DisplayListboxData” for the appropriate sensor.

1. Add two numeric input controls for Sigma and Mu. The value for Sigma must be limited with a minimum of 10 and a maximum of 20. Set the default value to 10. The value for Mu must be limited with a minimum of 35 and a maximum of 75. Set the default value to 50.
2. Add two textboxes for the search value; one for each sensor, ensure only numeric integer values can be entered.
3. All code is required to be adequately commented. Map the programming criteria and features to your code/methods by adding comments/regions above the method signatures. Ensure your code is compliant with the CITEMS coding standards (refer http://www.citems.com.au/).

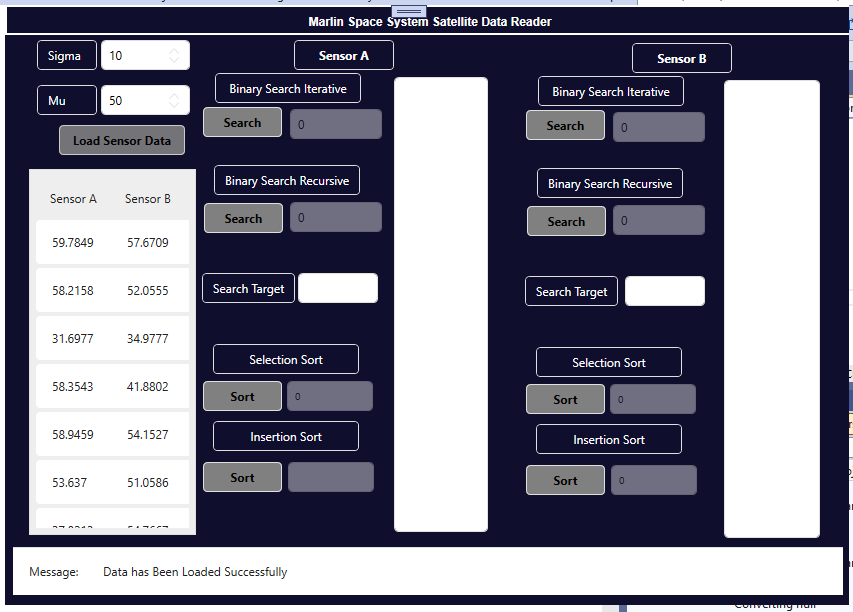
## Question 5 Debug and Testing

Ensure your code is error free and functions correctly, then design and test the applications using several different sets of data. During these tests check the returned times are record correctly and formatted into the appropriate scientific units. Your Test Report must include appropriate evidence that your application functions as expected (references to screen captures).

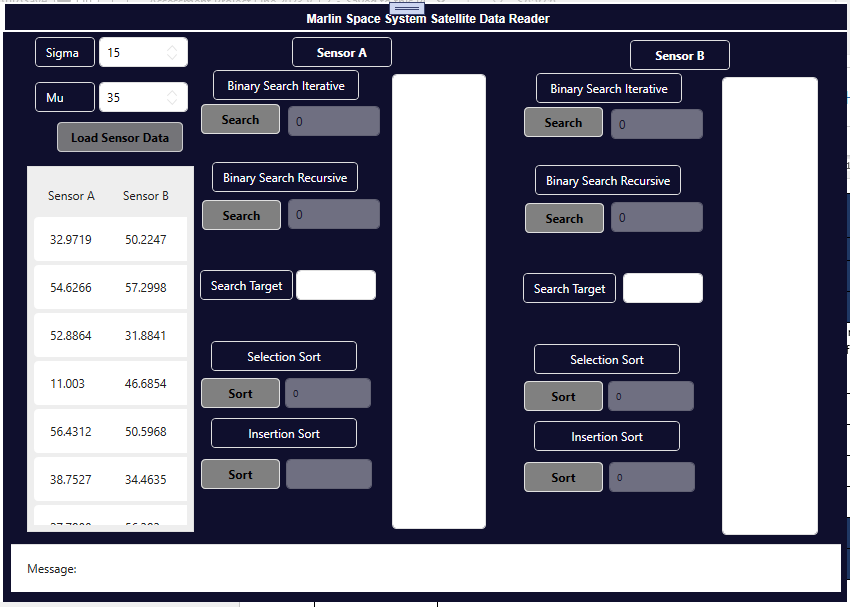
Finally, test the application to ensure all the user interactions are working as expected. Complete the following Test Report template to answer this question.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Report | | | | | |
| Developer Name | Satbir Singh | | **Date** | 27/02/2023 | |
| Sensor Data Test | | | | | |
| Input Data | Description | Actual Result | | | Pass / Fail |
| Sigma 10, Mu 50 | Load Data button has been pressed after putting these values and data should be displayed in List View for appropriate Sensors | Displaying the data in List View Collecting from Galileo6 and Sensor Linked Lists | | | Pass Refer to  Figure 1 |
| Sigma 15, Mu 35 | Load Data button has been pressed after putting these values and data should be displayed in List View for appropriate Sensors | Displaying the data in List View Collecting from Galileo6 and Sensor Linked Lists | | | Pass Refer to  Figure 2 |
| Sigma 20, Mu 40 | Load Data button has been pressed after putting these values and data should be displayed in List View for appropriate Sensors | Displaying the data in List View Collecting from Galileo6 and Sensor Linked Lists | | | Pass Refer to  Figure 3 |
| Sigma 17, Mu 45 | Load Data button has been pressed after putting these values and data should be displayed in List View for appropriate Sensors | Displaying the data in List View Collecting from Galileo6 and Sensor Linked Lists | | | Pass Refer to  Figure 4 |
| Sigma 10, Mu 55 | Load Data button has been pressed after putting these values and data should be displayed in List View for appropriate Sensors | Displaying the data in List View Collecting from Galileo6 and Sensor Linked Lists | | | Pass Refer to  Figure 5 |
| Sigma 10, Mu 55  (Tried on same Sigma and Mu Values) | Load Data button has been pressed after putting these values and data should be displayed in List View for appropriate Sensors.  Values should be different from Previous Values based on Sigma and Mu Values. | Displaying the data in List View Collecting from Galileo6 and Sensor Linked Lists | | | Pass Refer to  Figure 6 |
| Sigma 15, Mu 60 | Load Data button has been pressed after putting these values and data should be displayed in List View for appropriate Sensors | Displaying the data in List View Collecting from Galileo6 and Sensor Linked Lists | | | Pass Refer to  Figure 7 |
| Sigma 20, Mu 75  (Max Values for both) | Load Data button has been pressed after putting these values and data should be displayed in List View for appropriate Sensors | Displaying the data in List View Collecting from Galileo6 and Sensor Linked Lists | | | Pass Refer to  Figure 8 |
|  |  |  | | |  |
| User Experience Tests | | | | | |
| UI Component | Description | | | | Pass / Fail |
| Selection Sort Button (Sensor A) Test 1 | After Clicking on Selection Sort Button, Data from linked list of Sensor A should be displayed after sorting in List Box Assigned to Sensor A and display the sorting time in appropriate Text Box. | | | | Pass Refer to Figure 9 |
| Selection Sort Button (Sensor A) Test 2 | After Clicking on Selection Sort Button, Data from linked list of Sensor A should be displayed after sorting in List Box Assigned to Sensor A and display the sorting time in appropriate Text Box. (This Is Performed on different Data) | | | | Pass Refer to Figure 10 |
| Selection Sort Button (Sensor A) Test 3 | After Clicking on Selection Sort Button, Data from linked list of Sensor A should be displayed after sorting in List Box Assigned to Sensor A and display the sorting time in appropriate Text Box. (This Is Performed on different Data) | | | | Pass Refer to Figure 11 |
| Selection Sort Button (Sensor B) Test 1 | After Clicking on Selection Sort Button, Data from linked list of Sensor B should be displayed after sorting in List Box Assigned to Sensor B and display the sorting time in appropriate Text Box. | | | | Pass Refer to Figure 9 |
| Selection Sort Button (Sensor B) Test 2 | After Clicking on Selection Sort Button, Data from linked list of Sensor B should be displayed after sorting in List Box Assigned to Sensor B and display the sorting time in appropriate Text Box. (On Different Data) | | | | Pass Refer to Figure 10 |
| Selection Sort Button (Sensor B) Test 3 | After Clicking on Selection Sort Button, Data from linked list of Sensor B should be displayed after sorting in List Box Assigned to Sensor B and display the sorting time in appropriate Text Box. (On Different Data) | | | | Pass Refer to Figure 11 |
| Insertion Sort Button (Sensor A) Test 1 | After Clicking on Insertion Sort Button, Data from linked list of Sensor A should be displayed after sorting in List Box Assigned to Sensor A and display the sorting time in appropriate Text Box. | | | | Pass Refer to Figure 12 |
| Insertion Sort Button (Sensor A) Test 2 | After Clicking on Insertion Sort Button, Data from linked list of Sensor A should be displayed after sorting in List Box Assigned to Sensor A and display the sorting time in appropriate Text Box. | | | | Pass Refer to Figure 13 |
| Insertion Sort Button (Sensor A) Test 3 | After Clicking on Insertion Sort Button, Data from linked list of Sensor A should be displayed after sorting in List Box Assigned to Sensor A and display the sorting time in appropriate Text Box. | | | | Pass Refer to Figure 14 |
| Insertion Sort Button (Sensor B) Test 1 | After Clicking on Insertion Sort Button, Data from linked list of Sensor B should be displayed after sorting in List Box Assigned to Sensor B and display the sorting time in appropriate Text Box. | | | | Pass Refer to Figure 12 |
| Insertion Sort Button (Sensor B) Test 2 | After Clicking on Insertion Sort Button, Data from linked list of Sensor B should be displayed after sorting in List Box Assigned to Sensor B and display the sorting time in appropriate Text Box. | | | | Pass Refer to Figure 13 |
| Insertion Sort Button (Sensor B) Test 3 | After Clicking on Insertion Sort Button, Data from linked list of Sensor B should be displayed after sorting in List Box Assigned to Sensor B and display the sorting time in appropriate Text Box. | | | | Pass Refer to Figure 14 |
| Iterative Search Button (Sensor A) Test 1 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 15 |
| Iterative Search Button (Sensor A) Test 2 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 16 |
| Iterative Search Button (Sensor A) Test 3 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 17 |
| Iterative Search Button (Sensor B) Test 1 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 18 |
| Iterative Search Button (Sensor B) Test 2 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 19 |
| Iterative Search Button (Sensor B) Test 3 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 20 |
| Recursive Search Button (Sensor A) Test 1 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 21 |
| Recursive Search Button (Sensor A) Test 2 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 22 |
| Recursive Search Button (Sensor A) Test 3 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 23 |
| Recursive Search Button (Sensor B) Test 1 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 24 |
| Recursive Search Button (Sensor B) Test 2 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 25 |
| Recursive Search Button (Sensor B) Test 3 | After Clicking on Search Button, Data should be searched after entering the search value in the appropriate text box and highlight the range of data in List Box and display time in Ticks in Text Box. | | | | Pass Refer to Figure 26 |
| Search Box Empty (Sensor A) | Should ask for data in Search Box | | | | Pass Refer to Figure 27 |
| Search Box more than Three digits (Sensor B) | Should not perform Searching, should ask for Number Less Than four digits | | | | Pass Refer to Figure 28 |
| Recommendations and Limitations | | | | | |
| Limitations With Sorting: - These algorithms fails when there is an occurrence of duplicates number and stop sorting the data.  This issue can be resolved by using stable sorting algorithm that take care of duplicate occurrence.  Moreover, Galileo Library should return data with more decimal points so, chances of duplicate occurrence will be minimal or null.  Moreover, No clear about how user wants their feedback messages, for example in form of message boxes or status bar. | | | | | |

## Sensors Data Test Screen Shots



Figure



Figure



Figure



Figure



Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure



Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

## Selection Sort Buttons Testing Screen Shots



Figure



Figure



Figure

## Insertion Sort Buttons Testing Screen Shots



Figure



Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

## Iterative Search Buttons Screen Shots

A screenshot of a computer

Description automatically generated with medium confidence

Figure



Figure

Graphical user interface, application

Description automatically generated

Figure



Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

A screenshot of a computer

Description automatically generated with medium confidence

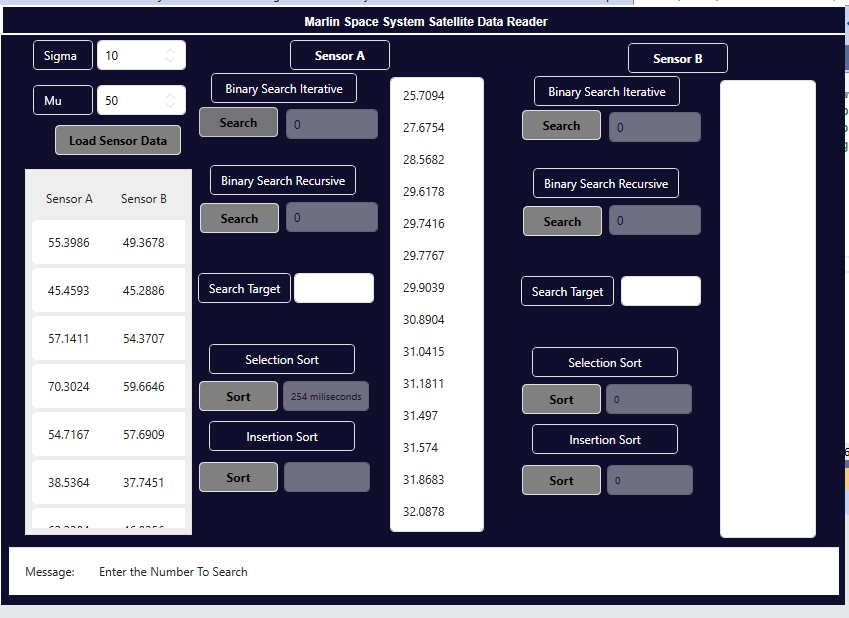
Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

## Other Utilities Testing



Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

## Question 6 Evaluation

Once testing has been completed you can answer the following questions regarding the implementation and usage of your application.

|  |  |  |  |
| --- | --- | --- | --- |
| Evaluation Report | | | |
| Developer Name | Satbir Singh | **Date** | 20/02/2023 |
| Reflective Questions | | | |
| Your WPF application will replace the current 32Bit command line program, What are the impacts of this change?  The command line programs can only be operated by using Command-line interface. These receives only command from the user, but new App uses WPF that has a user interface to interact not commands and It can be used on any platform and user interface provides that facility to use clicks for performing functions not commands. So, it will be easier and faster to use as compared to current system in use. | | | |
| What are Malin’s objectives for requesting the new software application?  The senior scientists at Malin’s wants the collected data from their satellite sensors to be sorted using two different algorithms and also two searching algorithms should be used for searching of a particular range of data in the collected data from sensors. Moreover, their current system is not compatible with new hardware platform so they want a new application that can run on any hardware platform. | | | |
| How will your WPF based application meet the objectives of the Malin organisation?  This App has included all the functionality that was requested by Malin’s. It will fulfill all the requirements of the scientist such as two sorts and two search algorithms are implemented and tested as well. However, new system can not be perfect, but improvements will be made in future. | | | |
| What are the perceived difficulties of introducing the new WPF based application?  First, as new system, it is hard to accept for the scientists because they are using their old system from long time. Moreover, some of the features might be changed as compared to previous system because of completely new UI. | | | |
| What training could be required to use the new WPF based application?  GUI training should be required because no command or code will be required to use this App and all the usage of buttons and display items (List View and list boxes) should be explained in the training and how convenient to use these tools as compared to previous system. | | | |
| What time schedule would you recommend for the implementation of training.  All components of the App has added tool tips on them so once user will go over a particular components it will display what is the functionality of this component and if user will try to perform any inappropriate function that is not lies in that components functionality than appropriate message will be displayed. So, with all these features training should not be more than one day because once all these features will be explained properly then there won’t be need of formal training. | | | |
| How would you seek feedback on the success or failure of your WPF based application?  As the system implemented the feedback will be collected by different employees at Malin’s using feedback form based on this App on different days because it is hard to compare large number of feedbacks and then one common issue, if occurred will be resolved as first priority. | | | |

## Question 7 Assessment, Feedback and Signoff

Check all the above documentation has been completed and is ready for inspection. Contact your Lecturer (Assessor) and arrange to demonstrate your working applications, use the following Marking Guide and Observation Checklist to ensure you have completed all the assessment criteria.

### Assessor Marking Guide

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Marking Guide and Observation Checklist | | Satisfactory | | Feedback |
| **Questions** | | YES NO | |  |
| Q1 | Design Specification: All fields of the Design Specification are filled in. |  |  |  |
|  | Client Technical Requirements contains information that is correct. |  |  |  |
|  | UI Specifications show a detailed information when a GUI component is clicked or select? |  |  |  |
| Q2 | Version Control: All fields of the template are filled in. There are screen shots of GitHub showing the Project and Repository. |  |  |  |
|  | Lecturer has observed the GitHub account and it reflects an Agile project methodology. |  |  |  |
|  | Lecture has observed the GitHub account which reflects a repository with appropriate files. |  |  |  |
| Q3 | Design Approval has been signed off by Lecturer |  |  |  |
|  | Suitable feedback has been provided |  |  |  |
|  | Anomalies have been corrected |  |  |  |
| Q4 | There are only two global variables (two LinkedLists) |  |  |  |
|  | The DLL has been referenced in the solution. The data from each sensor has been added to the relative LinkedList. All criteria in 4.2 are correct. |  |  |  |
|  | The sensor data can be displayed in a ListView and called from a Button click method |  |  |  |
|  | The two utility methods have the correct input parameters. |  |  |  |
|  | The sort and search methods have the correct algorithm. Each input parameter is correct and all criteria in 4.7 - 4.10 are correct. |  |  |  |
|  | The four UI buttons for the searches have the correct “calls” and return the correct time units. |  |  |  |
|  | The four UI buttons for the sorts have the correct “calls” and return the correct time units. |  |  |  |
|  | The GUI controls meet the 4.13 – 4.15 criteria, All code is correctly commented |  |  |  |
| Q5 | Testing: All the fields in the Testing Report have been filled in. |  |  |  |
|  | The sensor test data has tested each search and sort for both sensors, (8 tests) |  |  |  |
|  | Each test has been run more than once. |  |  |  |
| Q6 | Evaluation: The evaluation questions have been answered. |  |  |  |
| Q7 | Feedback: The client (Lecturer) has provided suitable feedback for the assessment task. |  |  |  |
|  | Feedback: The client (Lecturer) has observed the Github project and completed development plan. |  |  |  |
| **General Feedback:** | | | | |
|  | **Assessment Decision**  Satisfactory  Not Yet Satisfactory | | | |

**Note:** All documentation must use the supplied templates/forms.

**Submit the zipped solution folder with relevant documents to Blackboard**

End of Assessment One

## Appendix

### Selection Sort

integer min => 0

integer max => numberOfNodes(list)

for ( i = 0 to max )

min => i

for ( j = i + 1 to max )

if (list element(j) < list element(min))

min => j

END for

**// Supplied C# code**

LinkedListNode<double> currentMin = list.Find(list.ElementAt(min))

LinkedListNode<double> currentI = list.Find(list.ElementAt(i))

**// End of supplied C# code**

var temp = currentMin.Value

currentMin.Value = currentI.Value

currentI.Value = temp

END for

### Insertion Sort

integer max = numberOfNodes(list)

for ( i = 0 to max – 1 )

for ( j = i + 1 to j > 0, j-- )

if (list element(j - 1) > list element(j))

**// Supplied C# code**

LinkedListNode<double> current = list.Find(list.ElementAt(j))

**// End of supplied C# code**

**// Add Swap code here by swapping**

**// current previous value with current value.**

END if

END for

END for

### Binary Search Iterative

while (minimum <= maximum - 1)

integer middle = minimum + maximum / 2

if (search value = list element(middle))

return ++middle

else if (search value < list element(middle))

maximum => middle - 1

else

minimum => middle + 1

END while

return minimum

### Binary Search Recursive

if (minimum <= maximum - 1)

integer middle = minimum + maximum / 2

if (search value = list element(middle))

return middle

else if (search value < list element(middle))

return binarySearchRecursive(list, search value, minimum, middle - 1)

else

return binarySearchRecursive(list, search value, middle + 1, maximum)

END if

return minimum

### Galileo DLL

**<?xml version="1.0"?>**

**<doc>**

**<assembly>**

**<name>**Galileo**</name>**

**</assembly>**

**<members>**

**<member** **name="T:Galileo.ReadData">**

**<summary>**

A data recording library that returns the readings from two sensors.

**</summary>**

**</member>**

**<member** **name**=**"M:Galileo.ReadData.SensorA(System.Double,System.Double)">**

**<summary>**

Sensor A records data that represents a Gaussian distribution

(also known as normal distribution) it is a bell-shaped curve.

It is assumed that during any measurement, values will follow a normal

distribution with an equal number of measurements above and below the mean value (mu).

**</summary>**

**<param** **name**=**"mu">**The mean of the distribution**</param>**

**<param** **name**=**"sigma">**The standard deviation of the distribution**</param>**

**<returns>**A data reading within a Gaussian Distribution rounded to 4 decimal points**</returns>**

**</member>**

**<member** name=**"M:Galileo.ReadData.SensorB(System.Double,System.Double)">**

**<summary>**

Sensor B records data that represents a Box-Muller distribution

(also known as normal distribution) it is a bell-shaped curve.

It is assumed that during any measurement, values will follow a normal

distribution with an equal number of measurements above and below the mean value (mu).

**</summary>**

**<param** **name**=**"mu">**The mean of the distribution**</param>**

**<param** **name**=**"sigma">**The standard deviation of the distribution**</param>**

**<returns>**A data reading within a Box-Muller Distribution rounded to 4 decimal points**</returns>**

**</member>**

**</members>**

**</doc>**

### WPF Framework

### 