Task 1 : Update and add anything we need

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint One Project Specifications | | | | |
| Project Details | | | | |
| Release | | 1.0 | | |
| Date | |  | | |
| Team Name | | Astro | | |
| Scrum Master | |  | | |
| Team Member | |  | | |
| Project Tasks | | | | |
| Task # | Description | | Priority | Notes |
| 1 | Design user interface for the Astronomical Processing application | | 1 | Form-based GUI with list box, text boxes, and buttons for processing |
| 2 | Implement data storage using an array of 24 integers | | 1 | Array simulates hourly data for a day |
| 3 | Develop the Bubble Sort algorithm for sorting the array | | 2 | Required for sorting data |
| 4 | Develop the Binary Search algorithm for searching the array | | 2 | Required for searching data |
| 5 | Implement input validation and error messages | | 3 | Ensures robust user interaction |
| 6 | Generate random integers to fill the array for simulation | | 4 | Simulates incoming data stream |
| 7 | Implement functionality to edit data values | | 4 | Allows data modification by the user |
| 8 | Integrate GitHub for source control and project management | | ? | Ensure all team members have access |
| 9 | Document meeting discussions and algorithm designs | | ? | Add documentation to source control |
| 10 | Test the application thoroughly to ensure all functionalities work | | 5 | Ensure application meets client requirements |
| Functional Requirements | | | | |
| A functional requirement is **describing the behaviour of the system** as it relates to the system's functionality. | | | | |
| **Data Storage:**  Store the number of neutrino interactions per hour as integers in an array of size 24.  **User Interface:**   * Display the data in a list box. * Allow the user to input data via text boxes. * Provide buttons to sort and search the data. * Include an input field for data editing.   **Sorting Method:**   * Implement Bubble Sort algorithm to sort the array.   **Searching Method:**   * Implement Binary Search algorithm to search the array.   **Input Validation:**   * Generate error messages if the input text box is empty or if the search is not successful. * Generate success messages if the search is successful.   **Data Simulation:**   * Fill the array with random integers (between 10 and 90) to simulate data stream. | | | | |
| Non-Functional Requirements | | | | |
| A non-functional requirement elaborates a performance characteristic of the system. | | | | |
| **Performance:**   * Ensure the application processes data efficiently and quickly.   **Usability:**   * Design the user interface to be intuitive and easy to navigate.   **Reliability:**   * Ensure the application handles invalid inputs gracefully and provides appropriate feedback.   **Maintainability:**   * Write clear and maintainable code with proper comments and documentation.   **Compatibility:**   * Ensure the application works across different versions of the operating system if applicable. | | | | |

**WINDOW FORM TOOLS AND PROPERTIES NAMES:**

• ListBox (Name: listBoxData)

• Labels (Names: lblSearch, lblEditIndex, lblEditValue)

• TextBoxes (Names: textBoxSearch, textBoxEditIndex, textBoxEditValue)

• Buttons (Names: buttonSort, buttonSearch, buttonEdit)

**Properties for Each Component:**

• **ListBox:**

• Name: listBoxData

• **Labels:**

• lblSearch: Text = “Search”

• lblEditIndex: Text = “Index”

• lblEditValue: Text = “Value”

• **TextBoxes:**

• textBoxSearch: Name = “textBoxSearch”

• textBoxEditIndex: Name = “textBoxEditIndex”

• textBoxEditValue: Name = “textBoxEditValue”

• **Buttons:**

• buttonSort: Text = “Sort”, Name = “buttonSort”

• buttonSearch: Text = “Search”, Name = “buttonSearch”

• buttonEdit: Text = “Edit”, Name = “buttonEdit”

What do you think about it?

Any ideas?

Update?

Add your screenshot here!

Design ideas!

A screenshot of a computer

Description automatically generated

For list data numbers from 10 -90 For 24 hours

public partial class Form1 : Form

{

private int[] dataArray = new int[24];

public Form1()

{

InitializeComponent();

InitializeDataArray();

DisplayDataArray();

}

private void InitializeDataArray()

{

Random rand = new Random();

for (int i = 0; i < dataArray.Length; i++)

{

dataArray[i] = rand.Next(10, 91); // Random integers between 10 and 90

}

}

private void DisplayDataArray()

{

listBoxData.Items.Clear();

for (int i = 0; i < dataArray.Length; i++)

{

listBoxData.Items.Add($"Hour {i + 1}: {dataArray[i]}");

}

Need code for each button and other values: