|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sensor Data-View Application Development Plan | | | | |
| Project Name | | SensorShare | | |
| Date | | 30/10/2025 | | |
| Student Name | | Samuel Bailey | | |
| Scope of Work | | A Windows Forms Application   * No Real Time Data-View * Simple File Reading | | |
| Constraints | | Due Date (Certain Time Allotted), C# Required | | |
| Stakeholders | | **Samuel Bailey:** Programmer, **Alisa Blakeney:** Client/Lecturer | | |
| Project Requirements | | | | |
| Req. # | Description | | Importance | Notes |
| 1 | Data Grid | | High | Visual representation of sensor readings. Displays values in a tabular format with colour coding. |
| 2 | Save and Load Menus | | High | Allows users to save or load sensor datasets. |
| 3 | Average Display | | Medium | Shows calculated average values of the current dataset. |
| 4 | Bound Inputs | | High | Text fields for setting upper and lower acceptable bounds. |
| 5 | Dataset Nav icons | | Medium | Buttons or icons to move between multiple loaded datasets. |
| 6 | Visual Indicators | | High | Uses colours to show value ranges: Green = acceptable, Red = above range, Blue = below range. |
| 7 | Sorted Dataset | | High | Dataset must be sorted automatically for efficient operations. |
| 8 | Average Calculation | | High | Automatically compute average for each dataset. |
| 9 | Binary Search | | High | |  | | --- | |  |   Custom binary search for locating values |
| 10 | Dataset Switching | | Medium | Ability to move between multiple loaded datasets. |
| User Interaction and Specifications | | | | |
| * Data Loading: Upon loading the application and also upon pressing the load button, a menu will come up to allow users to choose while file to load. That menu will feed the file path to the controller, which gives it to the processor to create the SensorData object that the grid will display. | | | | |
| * Navigation: Tabs will be clickable for loaded datasets to allows users to go back and forth between datasets. | | | | |
| * Data Input: Users will be able to set bounds and searches in textbox inputs to allow them to find and view certain data. | | | | |
| * Visual Feedback: Bound inputs will be displayed via DataGrid colours. | | | | |
| * Error Handling: Error messages will appear upon invalid inputs to help the user navigate the program. | | | | |

|  |
| --- |
| Data Processing Control Structure |
| A screenshot of a computer  AI-generated content may be incorrect.Provide your data processing class along with all methods their inputs, outputs, and detail their function.  **ATTRIBUTES:**  **Instance: DataProcessor** = Since it will be a singleton class, we must ensure that there is only one instance of this class. Upon calling it, the *getInstance()* method will either fetch the internally held instance or create it upon first use. It will be private, as only the methods will be able to fetch or create this instance.  **loadedData: List<SensorData>** = the DataProcessor class will hold all the SensorData objects that have been loaded in this session.  **currentDatasetIndex: int** = this is a simple pointer value that will tell the class which of the held SensorData objects to perform its functions on.  **METHODS:**  **getInstance()** = getInstance will either fetch the private value or use the constructor method DataProcessor() to create the initial instance and store it.  **loadData(filePath): SensorData** = the class will use the given filePath string to find a binary file. It will then take that file and turn it into a SensorData object and append it to the list. The currentDatasetIndex will be changed and the object will be fed to the controller via return.  **saveData(filePath)** = the class will take the current dataset and turn it into a binary file, then save that binary file to the allotted file path.  **calculateAverage(SensorData): float** = the processor will take the current SensorData, using *getValues(),* it will calculate the average and return it as a float value.  **binarySearch(float): (int, int)?** = a custom binary search that will search the label column via a binary search and return a row and column value of the found value. This will be sent to the DataView class to use to highlight the found row.  **changeDatasets()** = This is a simple function that is called to update the current SensorData object to one that has been previously loaded.  **sortData(SensorData)** = a simple sort function to allow for the binary search to function as needed. |