## Question One

### Instructions

Fill in all sections of the Sprint One Project Specifications form, start by completing the Project Details. Then list all the Project Tasks and assign a Priority from Very Important = 1 to Least Important = 5. Next, list all the functional and non-functional requirements based on the client requirements and program functionality.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint One Project Specifications | | | | |
| Project Details | | | | |
| Release | | 1.0 | | |
| Date | | 7/September/2021 | | |
| Team Name | | Elljam | | |
| Scrum Master | | James Boyd | | |
| Team Member | | Ellena Begg | | |
| Project Tasks | | | | |
| Task # | Description | | Priority | Notes |
| 1 | Decide on a suitable GUI prototype tool | | 1 |  |
| 2 | Design GUI and components including action/events | | 3 |  |
| 3 | Create Scrum Board on Github | | 1 |  |
| 4 | Complete Functional Requirements of application | | 2 |  |
| 5 | Complete Non-Functional Requirements of application | | 2 |  |
| 6 | Complete GUI components including action/events | | 5 |  |
| 7 | Design, write and review Bubble Sort/Binary Search algorithms | | 4 |  |
| 8 | Design, write and review Bubble Sort/Binary Search pseudocode | | 4 |  |
| Functional Requirements | | | | |
| A functional requirement is **describing the behaviour of the system** as it relates to the system's functionality. | | | | |
| * Store 24 integers as an array * Generate 24 random integers between 10 and 99 to populate array with a button press | | | | |
| * Display integers in a Listbox | | | | |
| * Allow user to add, edit or remove entries * Only allow integer input | | | | |
| * Sort data (ascending order) using a Bubble Sort algorithm | | | | |
| * Enable user to search for an entry using a Binary Search algorithm * Show clearly understandable error messages for incorrect input, no item selected, unsuccessful search * Show clearly understandable success messages for search results, display where in list search item was found | | | | |
| Non-Functional Requirements | | | | |
| A non-functional requirement elaborates a performance characteristic of the system. | | | | |
| * Useability (text large enough to read, buttons easy to click, understandable error messages) | | | | |
| * Clear text box after search/edit/add | | | | |
| * Redisplay list after every update | | | | |
| * Tooltip messages when hovering cursor over buttons | | | | |
| * Yes/No confirmation before deleting items | | | | |

## Question Two

### Instructions

Fill in all sections of the GUI Design Specifications form. List all the UI components (buttons, textbox, listbox, etc) and describe what action or event is associated with each. Insert a detailed image/picture/screen capture of the GUI design, ensure all the features are identified and labelled.

|  |  |
| --- | --- |
| GUI Design Specifications | |
| GUI Development Tool | https://figma.com |
| Application Development IDE | Microsoft Visual Studio 2019 |
| UI Components | |
| List all UI components and their associated action/event (describe what each component does).  Textbox: Allows user to type in data  Buttons (x6):   * Add Item: Checks for valid input in Textbox, adds integer item from Textbox to the array * Remove Item: Checks for valid input in Textbox or item selected in Listbox, searches array for item. If found, asks user for confirmation, removes item from the array if user clicks Yes * Edit Item: Checks for valid input in Textbox or item selected in Listbox, searches array for item, updates item if found * Bubble Sort: Sorts items in ascending order using Bubble Sort algorithm and redisplays sorted array in Listbox * Binary Search: Check for valid input in Textbox, sorts array using Bubble Sort method, searches array using Binary Search algorithm, returns success/failure with Status Strip message, highlights item in Listbox * Fill Array: Generates 24 random integers between 10 and 99 to fill array, displays array in Listbox unsorted   Listbox: Displays data, allows user to select data with mouseclick  Status bar: Display error messages and confirmations | |
| GUI Layout | |
| Insert Your Diagram/Image hereC:\Users\30041547\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Sprint One GUI Layout.png | |

## Question Three

### Instructions

Fill in the two sections for the algorithm and pseudo code for the Binary Search and Bubble Sort. Refer to the Blackboard resources to ensure your answer reflects the differences between an Algorithm and Pseudo Code.

|  |  |  |
| --- | --- | --- |
| Algorithm/Pseudo Code Design for Binary Search | | |
| Put the Algorithm for the Binary Search here;  1. Sort data in ascending order. Find the middle number of the array. 2. If the search target is less than the middle number, go to Step 3. If the search target is greater than the middle number, go to Step 4. If the middle number is the search target, end search. 3. Perform new search using the middle number from the previous Step as the new highest bound of the list. Find the middle number again. Repeat Step 2 4. Perform new search using the middle number from the previous Step as the new lowest bound of the list. Find the new middle number. Repeat Step 2. | Put the Pseudo Code for the Binary Search here; lowBound = lowest entry in list  highBound = highest entry in list  target = user’s search target  midPoint = middle entry in list  while(lowBound is less than highBound)  midPoint = (lowBound + highBound) / 2  if midPoint = target  print(“target found”)  break  else if target is less than midPoint  highBound = midPoint - 1  else if target is greater than midPoint  lowBound = midPoint + 1 | |
| Algorithm/Pseudo Code Design for Bubble Sort | | |
| Put the Algorithm for the Bubble Sort here;  1. Run through list repeatedly swapping adjacent elements if they are in the wrong order. 2. Return to beginning of the list, repeat step 1. 3. Continue until list is in sorted order. | | Put the Pseudo Code for the Bubble Sort here; For I = 0 to arrayLength - 1    For J = 0 to arrayLength - 1        If (Array[J] > Array[J + 1]             Temp = Array[J]             Array[J] = Array[J + 1]             Array[J + 1]= Temp          End-If       End-For End-For |

## Question Four

### Instructions

Fill in the two sections for the Scrum Board snapshot and meeting agenda/minutes. Ensure these documents are reflected in the Source Control.

|  |  |
| --- | --- |
| Scrum Board and Meeting Notes | |
| Put the Scrum Board and meeting notes here;    C:\Users\30041547\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Scrum Board 07.09.2021.png | C:\Users\30041547\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Meeting Notes 07.09.2021.png |