|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Qualification details** | | | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Release 8.0) | | | | |
| **Qualification National**  **Code & Title:** | ICT40120 Certificate IV in Information Technology (Release 3) | | | **State code** | BFF9 |
| **Qualification National**  **Code & Title:** | ICT50220 Diploma of information Technology (Release 2) | | | **State code** | BGJ4 |
| **Assessment Title** | Assessment Task Two (Individual Project) | | | | |
| **Unit National Code & Title** | ICTPRG443 Apply intermediate programming skills in different languages | | | | |
| ICTICT430 Apply software development methodologies | | | | |
| ICTICT449 Use version control systems in development environments | | | | |
| **Date Due** | Week Eighteen | | **Date Received** |  | |
| **Student Name** |  | | | | |
| **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 | **Reassessment 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 Assessment | 8 Class Sessions (Week 11 - 19) |
| 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:  ICTPRG443 Apply intermediate programming skills in different languages   * Demonstrate your skills and knowledge by creating, coding, debugging and testing code * Establish user requirements and then research and collect information about data structures to provide suitable solutions. * Manage time and tasks to complete a series of coding and documentations problems   ICTICT430 Apply software development methodologies   * Select traditional and non-traditional systems development methodologies * Apply selected software methodology to project plan which identifies resources and control structures * Document analysis for approval to external stakeholders.   ICTICT449 Use version control systems in development environments   * Prepare and evaluate version control systems * Install and configure a version control system * Create and upload code to version control system * Test and review logs on version control system   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 Topic Two will include the following: Weekly Readings, Class notes, and Weekly Activities. |
| Required Resources | Web links and example code can be downloaded from the Blackboard portal.  PC with MS Visual Studio, MSOffice. |
| 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 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 portfolio 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. |
| Project contents | A project of programming 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 – Wiki Application Proposal * Question 2 – GUI Design * Question 3 – Class Details * Question 4 – Project Details * Question 5 – Sign Off and Approval * Question 6 – Programming Criteria * Question 7 – Testing Report * Question 8 – Finalise Project * Question 9 – Demonstration and Submission |

# Scenario

You have accepted the role of a Senior Programmer for CITE Managed Services, your task is to develop a fully functional Wiki application for the junior programmers. The initial Wiki prototype has been approved for full development, however, CITE management have requested some alterations to the original specifications.

## Background Information

There are many different categories and definitions for Data Structures used in software development; CITE management would like to see a uniform definition and cataloguing of this information. They have supplied some specific details but would like you to complete the task.

## Introduction

Before the project can commence you need to review the Management Criteria and complete the Wiki Application Proposal, GUI Design, Class Details and Project Details for submission. Once your proposal has been authorised (signed off) by the CITE representative (Your Lecturer) you can begin the next stage of the Wiki development.

You should consult with the CITE representative (Your Lecturer) if you are unsure about any of the problems or questions. Your primary research should focus on the resources on the Blackboard website, additional information can be collected from the Internet, ensure all sources are referenced at the end of your report. You should write your answers in the standard templates provided in this assessment document.

## Management Criteria

The Wiki application must use a List<T> of a simple class which implements an IComparable<T> interface. This single class must have the following attributes: Name, Category, Structure and Definition, (refer Data Structure Matrix at the end of this document). Following the success of the prototype, management would like the user to have the following functionality: user can add, edit and delete Data Structure information. During this process the system must be able to prevent duplicates and filter out numeric or special character input. The user can select a Data Structure name from the list of Names and the associated information will be displayed in the related text boxes.

The application must have a search feature so a user can find a specific Data Structure by entering the Name into the search textbox and clicking the search button, if found the associated information will be displayed in the related text boxes. The search textbox must clear when the search is completed and refocus the I beam icon into textbox.

The Wiki application will save data when the form closes. There are two buttons for the manual open and save option; this must use a dialog box to select a file or rename a saved file. All wiki data is stored/retrieved using a binary file format.

All user interactions must have full error trapping and feedback messaging which is displayed in a status strip at the bottom of the form. Use a message box for all critical errors with caption and icon.

## Question 1 Wiki Application Proposal

Complete the Wiki Application Proposal form below with the project details, then provide a comprehensive list of user requirements and features that satisfy the management criteria outlined by the CITE management team.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Wiki Application Proposal | | | | |
| Project Name | | Wiki Application for Junior Developers | | |
| Date | | 31/05/2022 | | |
| Student Name | | Kyle Watson | | |
| Client Requirements | | | | |
| Req. # | Description | | Importance | Notes |
| 1 | Wiki application must use a List<T> of a simple class which implements IComparable<T> interface. | | 1 | Base functionality |
| 2 | The class must have Name, Category, Structure and Definition as attributes. | | 1 | Base functionality |
| 3 | Can, add edit and delete Data Structure information | | 2 | Core methods |
| 4 | Must include a method for preventing duplicates and filter out numeric or special character input | | 5 | Low priority feature |
| 5 | User can select a Data Structure name from the list of Names and associated data will be displayed in the related boxes | | 2 | Core methods |
| 6 | User can search to find a specific Data Structure by entering a Name into a search textbox and clicking “search”. If found, the associated information will be displayed in the connected textboxes. | | 2 | Core methods |
| 7 | The search textbox must clear when the search is completed and refocus the I beam into the textbox. | | 5 | Low priority feature |
| 8 | The Wiki application will save data when the form closes. | | 3 |  |
| 9 | There are two buttons for the manual open and manual save options. These must use a dialog box to select a file or rename a saved file. | | 3 |  |
| 10 | All wiki data is store/retrieved using a binary file format | | 3 | Semantic |
| 11. | User interactions need full error trapping and feedback messages through a status strip at the bottom of the form. | | 4 | User experience |
| 12. | All critical errors require a message box with captions and icons. | | 4 | User experience |
|  | *Add more lines as required* | |  |  |

|  |
| --- |
| User Interaction and Specifications |
| How will the application behaves and what GUI specifications are required. |
| * Can add, edit, delete, search, save and open data structure information. |
| * User interactions require feedback via a status strip at the bottom of the app and critical errors require a message box with captions and icons. |
| * Searching for an existing Data structure item or selecting it from the listview box will display the associated information in the connected textboxes. |
| * The input for Category must be a ComboBox with six choices |
| * The input for the Structure must be two radio buttons inside a GroupBox. |
| * The input for the Definition must be a multi-lined textbox. |
| * The main list of Data Structures is a ListView which displays the Name and Category |
| *Add more lines as required* |

## Question 2 GUI Design

Create a detailed GUI Design along with details of all the components using a suitable GUI design software package (design cannot be hand drawn). Ensure your design covers all the features that are outlined by the CITE management team. Use the following form to complete this question.

Combo box

Add, edit, delete buttons

Open button

Listview box

Search box

|  |  |  |  |
| --- | --- | --- | --- |
| GUI Design | | | |
| Software Name | Wiki Application for Junior Developers | URL | [Wiki Application FIGMA link](file:///C:\Users\30048165\Downloads\Wiki%20Application%20FIGMA%20link) |
| Graphical User Interface Layout | | | |
| Input textbox    Display box for definition  Radio buttons  Save button | | | |

## Question 3 Class Details

Create a UML class diagram of the Wiki data object and then define the data structure for the List in the target language. Use the following form to complete this question.

|  |
| --- |
| Class Details |
| UML Class Diagram |
|  |
| Data Structure Definition |
| * List<T> is collection of strongly typed objects that can be accessed by index and has methods for sorting, search and modifying the list. |

## Question 4 Project Details

Using a CITE approved agile software development methodology, create a project plan. List and describe all the tasks required to complete the development of the Wiki Application. Create a new GitHub repository and then use the GitHub Project template to answer this question. Complete the following form as evidence of your work.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Details | | | |
| Project Name | Wiki Application for Junior Developers | | |
| Version Number | 1.0 | Date | 31/05/2022 |
| Repository Name: | Wiki\_Application\_for\_Junior\_Developers | | |
| URL | https://github.com/Kwatson-1/Wiki\_Application\_for\_Junior\_Developers | | |
| Initial Project Tasks | Screen Shots | | |

|  |  |
| --- | --- |
| Repository Details | Screen Shots |

## Question 5 Sign-off and Approval

You will need to arrange for the previous questions to be reviewed by the Lecturer/Assessor for sign off, approval and feedback before you start Question 6.

Your documents will include:

* Question 1 – Wiki Application Proposal.
* Question 2 – GUI Design.
* Question 3 – Class Details.
* Question 4 – Project Details (screen shots from source control)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Approval (Lecturer/Assessor use only) | | | | |
| Approver Name | Title | Signature | Date | Approved? |
| Stewart Godwin |  |  | 31/05/2022 | Yes |
|  |  |  |  |  |
| Lecturer Feedback | | | | |
|  | | | | |

## Question 6 Programming Criteria

Create and code the Wiki Application to demonstrate how a collection of information can be stored using a Windows Forms Application. This application will utilise a List<T> of a class object**.** Use the hardware and software supplied in the classroom to accomplish the development, debugging and internal documentation of the final Wiki Application. Use the Version Control System outlined in Question Four to manage your code during the development; ensure you record these commits/fetch as a series of screen shots to be included in the final report (Question 8, ie start, working, final). Your code must adhere to the CITEMS software development standards. (refer http://www.citems.com.au/)

The following programming criteria and features are required, the exact requirements of the Programming Criteria are essential. Any variation from them will need to be corrected in order to achieve a satisfactory performance. The following GUI layout is provided as a guide for your development.

Graphical user interface

Description automatically generated

### Addendum

* The input for Category must be a ComboBox with six choices, (refer Data Structure Matrix at end of document).
* The input for the Structure must be two radio buttons inside a GroupBox.
* The input for the Definition must be a multi-lined textbox.
* The main list of Data Structures must be a ListView which displays the Name and Category (You are not permitted to use a ListBox).

### Programming Criteria

6.1 Create a separate class file to hold the four data items of the Data Structure (use the Data Structure Matrix as a guide). Use auto-implemented properties for the fields which must be of type “string”. Save the class as “Information.cs”.

6.2 Create a global List<T> of type Information called Wiki.

6.3 Create a button method to ADD a new item to the list. Use a TextBox for the Name input, ComboBox for the Category, Radio group for the Structure and Multiline TextBox for the Definition.

6.4 Create and initialise a global string array with the six categories as indicated in the Data Structure Matrix. Create a custom method to populate the ComboBox when the Form Load method is called.

6.5 Create a custom ValidName method which will take a parameter string value from the Textbox Name and returns a Boolean after checking for duplicates. Use the built in List<T> method “Exists” to answer this requirement.

6.6 Create two methods to highlight and return the values from the Radio button GroupBox. The first method must return a string value from the selected radio button (Linear or Non-Linear). The second method must send an integer index which will highlight an appropriate radio button.

6.7 Create a button method that will delete the currently selected record in the ListView. Ensure the user has the option to backout of this action by using a dialog box. Display an updated version of the sorted list at the end of this process.

6.8 Create a button method that will save the edited record of the currently selected item in the ListView. All the changes in the input controls will be written back to the list. Display an updated version of the sorted list at the end of this process.

6.9 Create a single custom method that will sort and then display the Name and Category from the wiki information in the list.

6.10 Create a button method that will use the built in binary search to find a Data Structure name. If the record is found the associated details will populate the appropriate input controls and highlight the name in the ListView. At the end of the search process the search input TextBox must be cleared.

6.11 Create a ListView event so a user can select a Data Structure Name from the list of Names and the associated information will be displayed in the related text boxes combo box and radio button.

6.12 Create a custom method that will clear and reset the TextBboxes, ComboBox and Radio button

6.13 Create a double click event on the Name TextBox to clear the TextBboxes, ComboBox and Radio button.

6.14 Create two buttons for the manual open and save option; this must use a dialog box to select a file or rename a saved file. All Wiki data is stored/retrieved using a binary file format.

6.15 The Wiki application will save data when the form closes.

6.16 All code is required to be adequately commented. Map the programming criteria and features to your code/methods by adding comments above the method signatures. Ensure your code is compliant with the CITEMS coding standards (refer http://www.citems.com.au/).

## Question 7 Testing Report

Once you have completed coding the application, ensure your code is error free and functions correctly. You will need to complete the following testing report using the trace features built-in to Visual Studio. Add suitable code to the methods from Question 6.5, 6.7 and 6.8 then run the application and record the results. Your Test Report must include appropriate evidence that your code functions as expected (references to screen captures). Complete the following Test Report template below.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Testing Report | | | | | | | | | |
| Project Name | | Wiki Application FJP | | | | | | | |
| Methods | | ValidName(String textBoxName), method, Delete method, Edit method | | | | | | | |
| Description | | Testing of the software’s functional and non-functional requirements as per criteria 6.5, 6.7, 6.8 | | | | | | | |
| Level of Testing | | System testing | | | | | | | |
| Developer | | Kyle Watson | | Tester | Kyle Watson | | Date | 7/06/2022 | |
| Test Case No | Test Case Name | | Test steps | | | Test Data | Expected result | | Pass / Fail |
| 1 | Edit button testing. | | Load data into the List<T>. Select an item to edit, changes the values and press the edit button. | | | Name: ‘Array’ changed to ‘ArrayTest’ | Selected item updated. See Figure 1 of the trace file | | Pass |
| 2 | ValidName testing. | | Load data into the List<T>. Try add new records which evaluate to true and false. | | | Name: Array added twice, first to test true and then to test false | ‘Array’ violates ValidName while ‘ArrayTest’ does not. See Figure 2. | | Pass |
| 3 | Delete button testing. | | Load data into the List<T>. Check the count prior and post to deleting. | | | Name: ‘Array’ | Selected item deleted. See Figure 3. | | Pass |
| 4 | Add button – duplicate | | Load data into the List and attempt to add a duplicate entry | | | Name: ‘Array’ | Item not added into the list. | | Pass |
| 5 | Add button –  Blank | | Attempt to add an object with fields missing data | | | Alternate missing name, category, structure, and definition | Item cannot be added with blank fields. | | Pass |
| 6 | Add button –  special chars | | Attempt to add a data name which contains characters and digits. | | | Name: ‘123!@#’ | Digits and symbols cannot be entered due to input handling. | | Pass |
| 7 | Add button – valid entry | | Add data with all fields filled in and is a non-duplicate | | | Name: ‘Test’ | Item added successfully | | Pass |
| 8 | Delete button –  no selection | | Click delete without selecting an item from the list view | | | N/A | Message box error appears. | | Pass |
| 9 | Delete method - double click | | Double click an item in the list view | | | Delete: ‘Array’ | Double clicked item removed from the list. | | Pass |
| 10 | Edit button –  blank | | Click the edit button without selecting an item to edit | | | N/A | Message box error appears. | | Pass |
| 11 | Edit button – duplicate entry | | Select an item from the list and attempt to edit it so that it is a duplicate | | | Select ‘Heap’ and attempt to edit it to ‘Array’ | Error appears stating that the item already exists. | | Fail |
| 12 | Edit button – no selection | | Click the edit button with no selection | | | N/A | Message box error appears. | | Pass |
| 13 | Save button - yes | | Press the save button, enter a name, and press yes | | | Save file as ‘TestSave.bin’ | File saved successfully | | Pass |
| 14 | Save button – cancel | | Press the save button, enter a name and press cancel | | | Enter file name as ‘TestSave.bin’ | File not saved and program does not close | | Pass |
| 15 | Save method – on close yes | | Run the program, click escape and then press yes and save the file as “TestSaveClose” | | | Save file name as ‘Test Save Close’ | File saved successfully | | Pass |
| 16 | Save method – on close no | | Run the program, click escape and then press no and check file is not saved | | | N/A | File not saved | | Pass |
| 17 | Save method – on close cancel | | Run the program, click escape and then press cancel and ensure program returns to normal functionality | | | N/A | File not saved and program not closed | | Pass |
| 18 | Open file - cancel | | Click open file and press cancel | | | N/A | File not opened and data not loaded into the list. | | Pass |
| 19 | Open from file | | Click open file and select “Data Structure Matrix.bin” | | | File to open: ‘Data Structure Matrix’ | Data loaded successfully into the list. | | Pass |
| 20 | Search - existing | | Load data into the list view, enter the name of an existing item the list and click search | | | Search item name: ‘Queue’ | ‘Queue’ found and corresponding fields filled out correctly | | Pass |
| 21 | Search – non-existing | | Load data into the list view, enter the name of a non-existing item and click search | | | Search item name: ‘Balance’ | Message showing item not found. | | Pass |
| 22 | Search blank | | Click search with nothing entered into the search box | | | N/A | Error message explaining nothing was entered. | | Fail |

# Trace statement documentation:

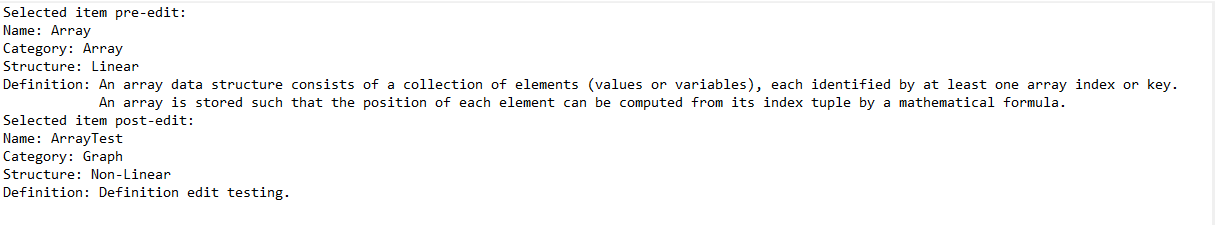


Figure : Edit button & method

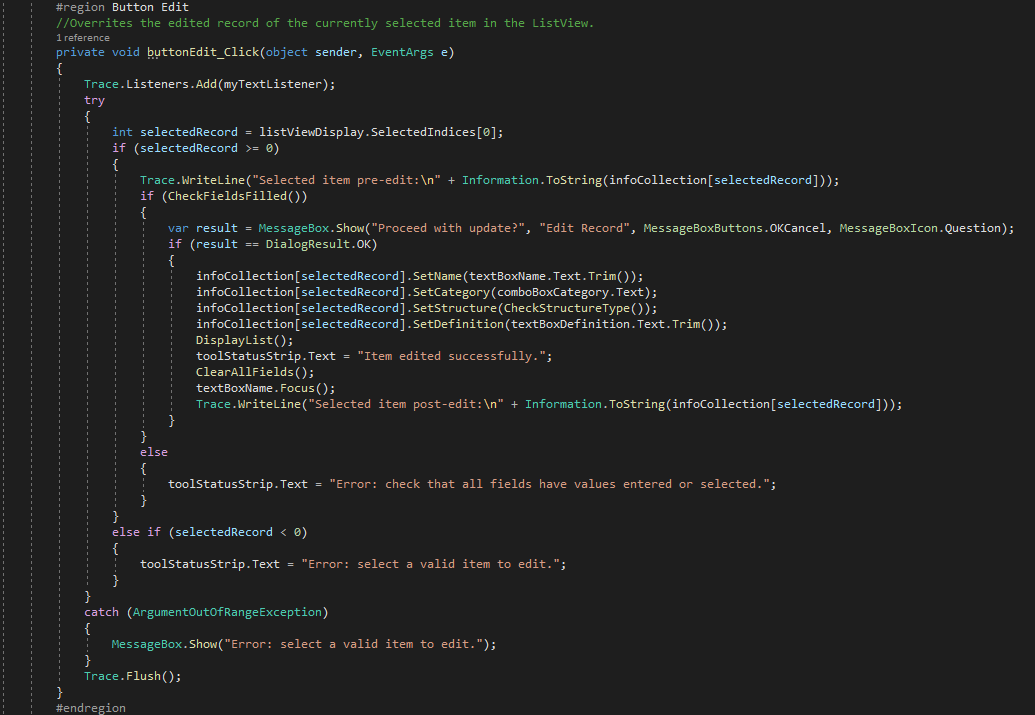


Figure : Text file output

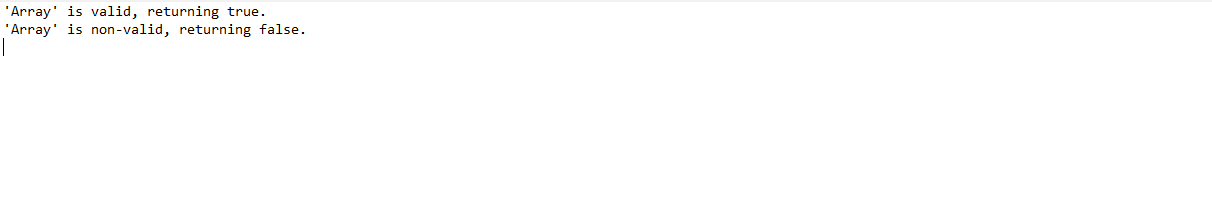


Figure : ValidName method

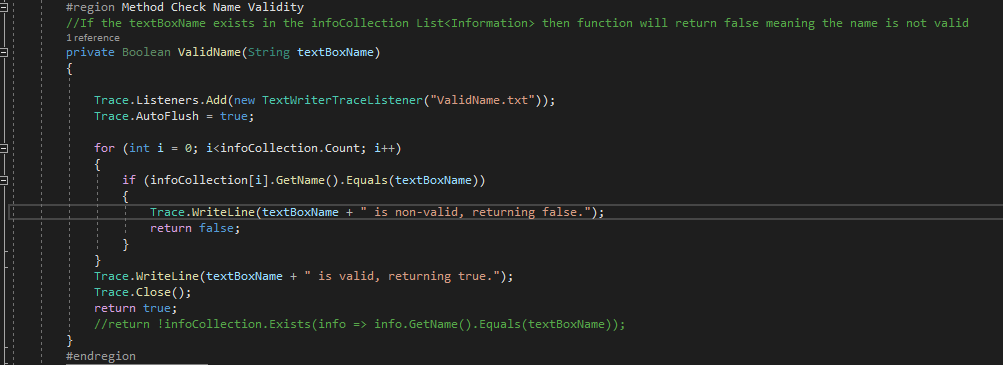


Figure Text file output



Figure : Delete method

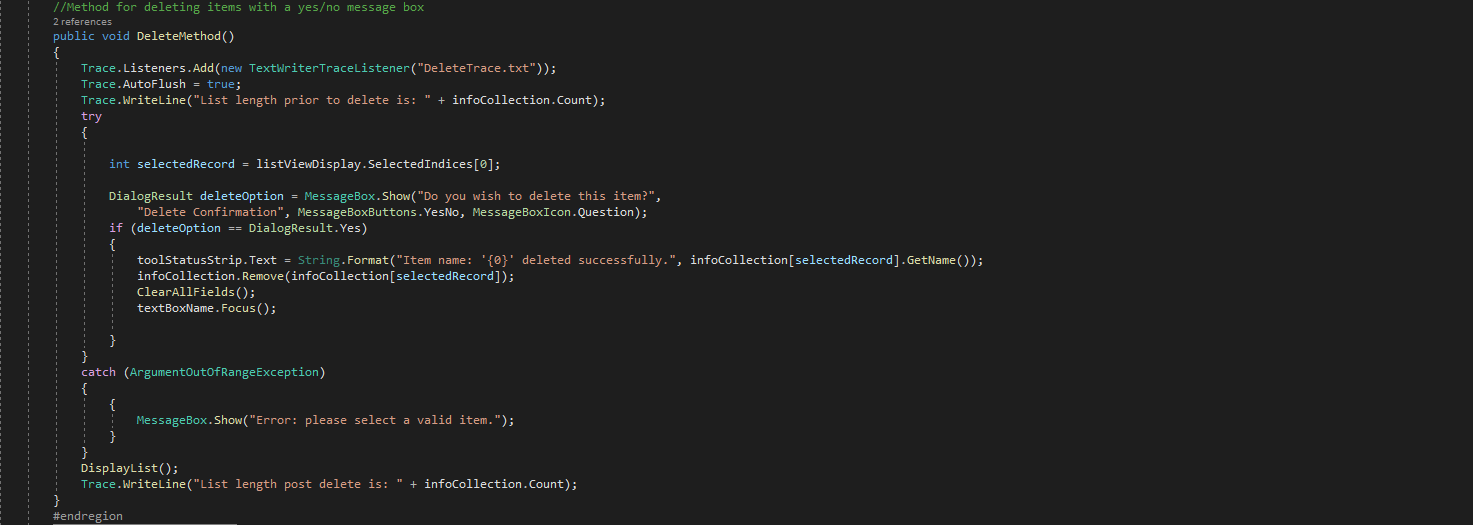


Figure : Text file output

## Question 8 Finalise Project

Once you have completed coding and testing of the Wiki Application you should review and reflect on the process before answer the following questions and completing the template below. Add your screen shots from Question 6 along with a suitable description of the activity/task each screen shot is demonstrating.

1. How effective was your project plan in developing the final Wiki Application?
2. Describe your experience when using the VCS.
3. Explain how you were able to utilise all the feature s of the VCS
4. Describe the disruptions/interruptions experienced during the project.
5. What problems did you encounter and how could this project be improved?

|  |
| --- |
| Finalise Project |
| Questions |
| 1. How effective was your project plan in developing the final Wiki Application?  My project plan was effective for developing the final Wiki Application as the majority of the core functionality was developed in a short period of 3 days. The effectiveness could have been improved by planning a framework for developing the error messaging and user feedback in a more structured manner. |
| 2. Describe your experience when using the VCS.  The VCS was simple to use and allowed for me to work across two different locations and machines thanks to interfacing with its desktop application version controlling. |
| 3. Explain how you were able to utilise all the feature s of the VCS  I was able to access most of the features of Git through its desktop application which allowed me to push changes to a development branch and then merge those into the main branch when certain changes were up to standard and unlikely to cause problems. The remaining features I accessed through the VCS website itself included Kanban boards (for project progression monitoring) and ability to examine of others’ profiles and projects to find code which helped inform development decisions in my project. |
| 4. Describe the disruptions/interruptions experienced during the project.  During development I encountered an error that prevented running and saving of my project. The error itself was a bug with Visual Studio that had reportedly been patched. Because my work was backed up however I lost minimal work and time by restoring a previous version. |
| 5. What problems did you encounter and how could this project be improved?  The main problem I encountered was keeping track of how I was handling user feedback, specifically the tool status strips and pop-up message box errors. In certain places I was unsure if I had already handled certain feedback messages to the user via the button method or core method itself leading to multiple instances of non-DRY code. This project could be improved by creating a framework to inform decisions regarding where error messaging and user-feedback should be located in the code so that it is more uniform across the application as a whole. |

|  |  |  |  |
| --- | --- | --- | --- |
| Version Control Update | | | |
| Repository Name: | | Wiki\_Application\_for\_Junior\_Developers | |
| URL | | [GitHub - Kwatson-1/Wiki\_Application\_for\_Junior\_Developers: A wiki application that stores data structure information.](https://github.com/Kwatson-1/Wiki_Application_for_Junior_Developers) | |
| VCS Screen Shots | | | Description |
| Update 1 |  | |  |
| Update 2 |  | |  |
| Update 3 |  | |  |
| Update 4 |  | |  |
| Update 5 |  | |  |

## Question 9 Demonstration and Submission

Complete the documentation for Questions 6, 7, and 8 then request a suitable time to demonstrate your Wiki Application for approval. The demonstration will ensure your code is compliant and your documentation conforms to CITE standards. Use the following marking guide to check you have completed all the tasks.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assessor Marking Guide | | Satisfactory | | Comment |
| Questions | | YES NO | |  |
| Q1 | Wiki Application Proposal is complete. |  |  |  |
| Q2 | GUI Design has software details and layout diagram. |  |  |  |
| Q3 | Class Details has UML class diagram and data structure specification |  |  |  |
| Q4 | Project Details is complete with several screen captures of the VCS. |  |  |  |
| Q5 | Sign-off and Approval has been authorised by the Lecturer. |  |  |  |
| Q6 Programming Criteria, all the following criteria have been completed | Q6.1 Separate class file |  |  |  |
| Q6.2 List of Class |  |  |  |
| Q6.3 Add button |  |  |  |
| Q6.4 ComboBox array |  |  |  |
| Q6.5 ValidName method |  |  |  |
| Q6.6 Two methods for radio buttons |  |  |  |
| Q6.7 Delete button |  |  |  |
| Q6.8 Edit button |  |  |  |
| Q6.9 Display method for ListView |  |  |  |
| Q6.10 Binary search using built in sort and Icomparable |  |  |  |
| Q6.11 ListView select method |  |  |  |
| Q6.12 Method to clear input controls |  |  |  |
| Q6.13 Double click method to clear text box |  |  |  |
| Q6.14 Two methods to open and save to binary file |  |  |  |
| Q6.15 Save data when form closes |  |  |  |
| Q6.16 Comments mapping criteria to code and all code meets organisational standards |  |  |  |
| Q7 | Testing Report, three methods have trace code to ensure suitable testing. |  |  |  |
| Q8 | Finalise Project complete the five questions and update the VCS screen shots. |  |  |  |
| Q9 | Demonstration and Submission |  |  |  |
|  | **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 Task Two

## Addendum

|  |  |  |  |
| --- | --- | --- | --- |
| Data Structure Matrix | | | |
| NAME | CATEGORY | STRUCTURE | DEFINITION |
| Array | Array | Linear | An array data structure consists of a collection of elements (values or variables), each identified by at least one array index or key. An array is stored such that the position of each element can be computed from its index tuple by a mathematical formula. |
| Two Dimension Array | Linear | A two-dimensional array can be visualised as a grid (or table) with rows and columns. Positions in a two dimensional array are referenced like a map using horizontal and vertical reference numbers. They are sometimes called matrices. |
| List | List | Linear | A list or sequence is an abstract data type that represents a finite number of ordered values, where the same value may occur more than once. |
| Linked list | Linear | A linked list is a linear collection of data elements whose order is not given by their physical placement in memory. Instead, each element points to the next. It is a data structure consisting of a collection of nodes which together represent a sequence. |
| Self-Balance Tree | Tree | Non-Linear | A self-balancing tree is any node-based binary search tree that automatically keeps its height (maximal number of levels below the root) small in the face of arbitrary item insertions and deletions. |
| Heap | Non-Linear | A heap is a specialized tree-based data structure which is essentially an almost complete tree that satisfies the heap property. The heap is one maximally efficient implementation of an abstract data type called a priority queue, priority queues are often referred to as "heaps". |
| Binary Search Tree | Non-Linear | A binary search tree (BST), also called an ordered or sorted binary tree, is a rooted binary tree data structure whose internal nodes each store a key greater than all the keys in the node’s left subtree and less than those in its right subtree. |
| Graph | Graphs | Non-Linear | A graph data structure consists of a finite set of vertices, together with a set of unordered pairs of these vertices for an undirected graph or a set of ordered pairs for a directed graph to implement the undirected graph and directed graph concepts from the field of graph theory within mathematics. |
| Set | Abstract | Non-Linear | A set is an abstract data type that can store unique values, without any particular order. It is a computer implementation of the mathematical concept of a finite set. Unlike most other collection types, rather than retrieving a specific element from a set, one typically tests a value for membership in a set. |
| Queue | Linear | A queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. |
| Stack | Linear | A stack is an abstract data type that serves as a collection of elements, with two main principal operations: Push, which adds an element to the collection, and Pop, which removes the most recently added element that was not yet removed. |
| Hash Table | Hash | Non-Linear | A hash table is a data structure that implements an associative array abstract data type, a structure that can map keys to values. A hash table uses a hash function to compute an index, also called a hash code, into an array of buckets or slots, from which the desired value can be found. |
|  | | | |