|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Qualification details** | | | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Release 8.1) | | | | |
| **Qualification National**  **Code & Title:** | ICT40120 Certificate IV in Information Technology (Release 4) | | | **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 14 | | **Date Received** |  | |
| **Student Name** | James Chellew | | | | |
| **Student Declaration** | I declare that the evidence submitted is my own work:  Yes | | | | |
| **Assessor Name** | Stewart Godwin | | | | |
| **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 | 7 Class Sessions (Week 8 - 14) |
| 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 Topics 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 duplicate Names 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 the Name is 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 cursor into the Name textbox. After a successful search the Data Structure Name in the list must be highlighted; a double mouse click in the Name input box which will clear all textboxes, this must have an associated tool tip.

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 | | DataClassWiki | | | |
| Date | | 15/09/2023 | | | |
| Student Name | | James Chellew | | | |
| Client Requirements | | | | | |
| Req. # | Description | | | Importance | Notes |
| 1 | Must use a List<T> | | |  |  |
| 2 | Must implement a IComparabe<T> interface | | |  |  |
| 3 | Contains a single extra class Contains Name, Category, Category, Structure attributes | | |  |  |
| 4 | User can add, edit and delete data structure information objects | | |  |  |
| 5 | No duplicate names | | |  |  |
| 6 | Filter special Characters | | |  |  |
| 7 | Selecting data structure name will display associated information in textboxes | | |  |  |
| 8 | Can search for name, matches will display information, highlight it and focus on the name textbox | | |  |  |
| 9 | Double click on name field will clear all text boxes | | |  |  |
| 10 | Tooltips for all parts of form | | |  |  |
| 11 | Save data on close, also save and open data manually with buttons. File format is .bin | | |  |  |
| 12 | Full error trapping and feedback. Feedback displayed in text strip at the bottom of form. Critical errors displayed in message box with caption and icon. | | |  |  |
|  |  | | |  |  |
|  | *Add more lines as required* | | |  |  |
| User Interaction and Specifications | | | | | |
| GUI Component | | | Related Functions | | |
| List box | | | 1. Display Data of each item of information in the view box for user viewing 2. Upon clicking/selecting list box item, display respective information in the appropriate GUI elements (text box for name, radio button for linear or non-linear etc.) 3. Update display when modifying or adding data. Clear selection after modifying. | | |
| Data structure name text box | | | 1. Display "Name" column of respective item from the list box. 2. Allow modification of this element that will update the list item upon edit button push. | | |
| Category Combo box | | | 1. Selects the category in combo box according to selected item in list view. 2. Displays a drop down of all categories for user to select 3. If user edits selection, it should update in list upon edit button press | | |
| Structure Radio button | | | 1. Linear and non-linear options 2. Checking one or the other will cause the data to be saved as linear or non linear in the structure variable of information class. | | |
| Structure group box | | | 1. Used to check for what radio button has been selected. | | |
| Definition text box (multi-line). | | | 1. Display "Definition" column of respective item from the list box. Information should be spread over multiple lines. 2. Allow modification of this element that will update the list item element upon edit button push. | | |
| Add button | | | 1. Will add an element to the list if all (4) fields are filled with information. 2. Throw an error if not all fields filled. Displayed in error message text box. | | |
| Edit button | | | 1. Modify all fields of selected element. 2. Throw an error if modification leaves a field empty. 3. Check for each text box and give feedback to which ones are not filled. | | |
| Delete button | | | 1. Delete respective object item selected in the list view 2. Show message box to confirm element deletion. | | |
| Confirmation message box | | | 1. YES/NO message type message box prompting user if they are sure to delete the selected item. 2. Continue with deletion if YES, abort deletion if no. 3. Message boxes should include appropriate icons. | | |
| Search text box | | | 1. Use input from this field to search objects based on name comparator 2. Upon hit, select the respective list box element (selection will trigger display of data) 3. Focus back on the search text box (have typing curser active) 4. Display unsuccessful search in user feedback text box. | | |
| Search button | | | 1. Use input from search text box to search. 2. Should also be able to use the enter key to search. | | |
| Save button | | | 1. Display save file dialog on button push. | | |
| Savefiledialog | | | 1. By default, open to working directory. 2. Filter for \*.bin (binary file format) 3. When save is pressed in dialog box, save contents of list into a bin file. 4. Cancel saving of file if the dialog is exited or cancel is pressed. 5. Should prompt user to save before exiting program. | | |
| Open button | | | 1. Display open file dialog on button push. | | |
| Openfiledialog | | | 1. By default, open to working directory. 2. Filter for \*.bin files 3. When open is pressed, Clear list, read contents of file to list. | | |
| Labels | | | 1. Label each GUI element | | |
| User feedback and error message text box | | | 1. Display brief description of each action of user. 2. Display error message for blocked/invalid actions. | | |
| Tooltip | | | 1. Display description of each GUI element when hovering mouse over the respective element. | | |
| Other | | | 1. All elements tab index chronological 2. Enter key prompts a search. | | |

## 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.

|  |  |  |  |
| --- | --- | --- | --- |
| GUI Design | | | |
| Software Name | DataClassWiki | URL |  |
| Graphical User Interface Layout | | | |
|  | | | |

## 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 |
| |  | | --- | | Information | | Variables | | -String name | | -string category | | -string structure | | -string definition | | Methods | | +GetName() | | +SetName(string) | | +GetCategory() | | +SetCategory(string) | | +GetStructure() | | +SetStructure(string) | | +GetDefinition() | | +SetDefinition(string) | | +CompareTo(Information) | |
| Data Structure Definition (in C# code) |
| namespace InformationClassWiki  {  internal class Information : IComparable<Information>  {  // Private initial variables  private string name;  private string category;  private string structure;  private string definition;  // get; set; Name  public string GetName()  {  return name;  }  public void SetName(string name)  {  this.name = name;  }  // get; set; Category  public string GetCategory()  {  return category;  }  public void SetCategory(string category)  {  this.category = category;  }  // get; set; Strucure  public string GetStructure()  {  return structure;  }  public void SetStructure(string structure)  {  this.structure = structure;  }  // get; set; Definition  public string GetDefition()  {  return definition;  }  public void SetDefinition(string definition)  {  this.definition = definition;  }  public int CompareTo(Information other)  {  return this.GetName().CompareTo(other.GetName());  }  }    } |

## 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 | Information Class Wiki | | |
| Version Number | V1.0 | Date | 22/09/2023 |
| Repository Name: | Information Class Wiki | | |
| URL | https://github.com/JamesChellew/InformationClassWiki | | |
| Initial Project Tasks |  | | |
| Repository Details |  | | |

## 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 | Lecturer |  | 33/09/2023 | 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 Application (WinForms). 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 the previous Question 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 (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 criteria for the GUI layout are provided as a guide for your development.

* 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 private properties for the fields which must be of type “string”. The class file must have separate setters and getters, add an appropriate IComparable for the Name attribute. 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 a custom method to populate the ComboBox when the Form Load method is called. The six categories must be read from a simple text file.

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 builtin 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 TextBoxes, 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 reader/writer 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 debug 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 | |  | | | | | | | | |
| Methods | |  | | | | | | | | |
| Description | |  | | | | | | | | |
| Level of Testing | |  | | | | | | | | |
| Developer | |  | | Tester |  | | Date | |  | |
| Test Case No | Test Case Name | | Test steps | | | Test Data | | Expected result | | Pass / Fail |
| 1 |  | |  | | |  | |  | |  |
| 2 |  | |  | | |  | |  | |  |
| 3 |  | |  | | |  | |  | |  |
| 4 |  | |  | | |  | |  | |  |
| 5 |  | |  | | |  | |  | |  |
| 6 | ETC | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |

## 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 the previous Question 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? | | | |
| 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? | | | |
| Version Control Update | | | |
| Repository Name: | |  | |
| URL | |  | |
| VCS Screen Shots | | | Description |
| Week 1 |  | |  |
| Week 2 |  | |  |
| … |  | |  |
| … |  | |  |
| Week 8 |  | |  |

## Question 9 Demonstration and Submission

Complete the documentation for the previous Questions and 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.

**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. |
|  | | | |