|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Release 8.1) | | |
| **Qualification National Code and Title** | ICT40120 Certificate IV in Information Technology (Release 4) | **State code** | BFF9 |
| **Qualification National Code and Title** | ICT50220 Diploma of information Technology (Release 2) | **State code** | BGJ4 |
| **Assessment Title** *(as per DAP)* | Assessment Task One (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 Seven | | **Date Received** | | 19/07/2023 | |
| **Student Name** | Mehraneh Hamedani | | | | | |
| **Student Declaration** | I declare that the evidence submitted is my own work: Yes.  my student ID: 30062786 | | | | | |
| **Assessor Name** |  | | | | | |
| **Assessment Decision** | Satisfactory | | | Not Yet Satisfactory | | |
| **Assessor Signature** |  | | | **Date** | |  |
| **Is student eligible for reassessment (Re-sit)?** | No | Yes | | **Re-assessment Date:** | | Week Twentyl |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Portfolio |
| Duration of the assessment | 7 class sessions (Weeks 1-7) |
| 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: SDLC, Weekly readings (PDF), Example programs and Independent Outside of Class Activities |
| Required Resources | Web links and example code can be downloaded from the Blackboard portal.  PC with MS Visual Studio, MSOffice.  Internet Access to MSDN, GitHub and www.citems.com.au/ |
| 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 programming 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 project 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. |
| Portfolio Description | 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 – Project Specifications  Question 2 – Version Control Specifications  Question 3 – UML Class Details  Question 4 – Development Methodologies  Question 5 – Methodology Analysis and Selection  Question 6 – Manage Version Control System  Question 7 – Project Plan  Question 8 – Sign off and Approval  Question 9 – Prototype Development  Question 10 – Version Control Update  Question 11 – Data Structure Matrix  Question 12 – UML Activity Diagram  Question 13 – Debug Test Report  Question 14 – Post Development Analysis  Question 15 – 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. In Computer Science there are many different categories and definitions for Data Structures, most of these terms are used in the CITE software development department, however, CITE management would like to see a uniform definition and cataloguing of this information. They have supplied some basic details but would like you to complete a feasibility study and create a working prototype application. A rudimentary interface design has been provided along with a list of proposed program criteria which the prototype application must include.

Before the project can move to the next stage CITE management would like a report on the full development process and related documentation. Review the proposed program criteria and answer the associated questions. Use the supplied template forms to present your answers. Finally develop a working prototype using Microsoft Visual Studio C# and GitHub version control. The purpose of the assessment is to demonstrate to CITE management how this project can be achieved. If you do not have a GitHub account you should sign up for the free version, this will be used again in other courses (https://github.com).

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 and CITEMS website, additional information can be collected from the Internet, ensure all sources are referenced at the end of your submission. You must write your answers in the standard templates provided in this assessment task document.

## Client Program Criteria

The client was interviewed, and the following information was recorded. The client would like the end user to select a record from a display list and then have the corresponding information displayed in four text boxes on the left side of the interface. The end user should be able to search for an item which will be displayed in the four text boxes; after the search the search input box must clear and retains focus (cursor is inside the search text box) allowing the user to search for a new term.

The client requires that the end user has the option to add/edit/delete any of the four fields associated with an individual data structure record, the four fields are: Data Structure Name, Category, Structure and Definition. The user must be prompted via a popup box during the deletion process.

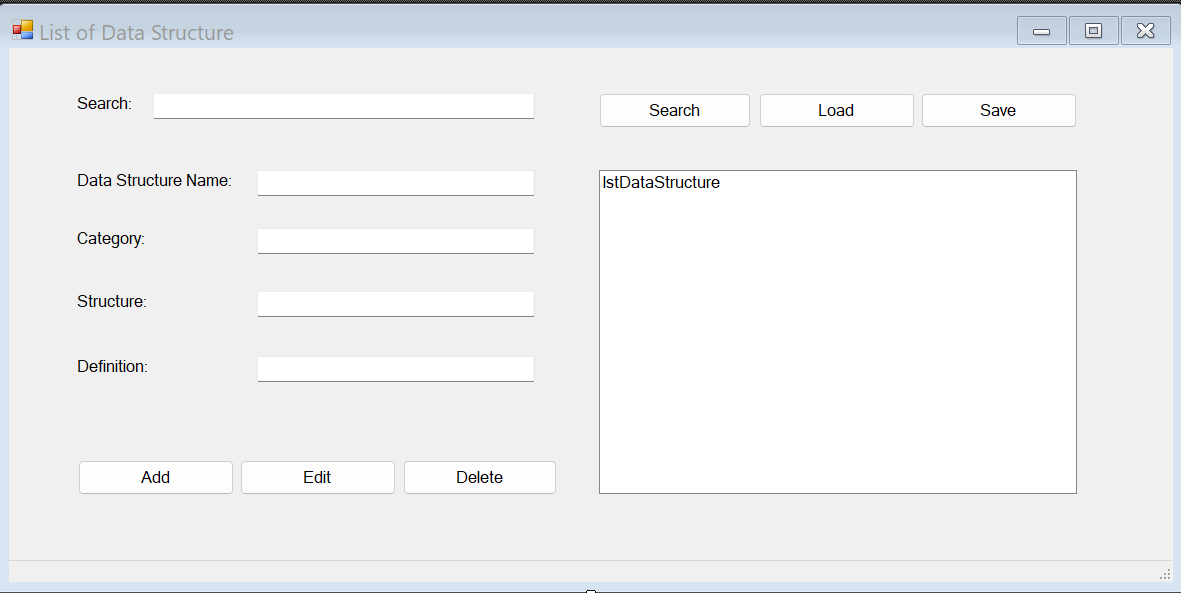
The prototype must use a two-dimensional array of type string to store each record. Refer to the program criteria to determine the exact size of the array, there is no provision for additional data! A double mouse click in the name text box will clear all four text boxes and focus the cursor into the name text box.

The wiki prototype will load and save data when the appropriate button is clicked, and all the wiki data is stored/retrieved using a binary file format. During the load and save process the end user must have the option to select an alternative data file. All end user interactions must have full error trapping and feedback messaging via a status strip at the bottom of the interface.

## Suggested Interface Design (example)

Table

Description automatically generated



## Question 1 Project Specifications

You are required to create a list of all the User Requirements for the wiki prototype application. Then list all the User Interactions and Specifications of the wiki application. Fill in the relevant sections of the following Project Specifications template to answer question one.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Specifications | | | | | |
| Project Name | | | WIKI Data | | |
| Date | | | 19/07/2023 | | |
| Developer Name | | | Mehraneh | | |
| Client Requirements | | | | | |
| Req. # | | Description | | Importance | Notes |
| 1 | | Display List: It provides a display list of records that the end user can select from. | | 2 |  |
| 2 | | Selection of Record: The end user should be able to choose a record from the display list. | | 2 |  |
| 3 | | Display information: Upon selecting a record, the corresponding information should be shown in four text boxes on the left side of the interface. | | 3 |  |
| 4 | | Search Functionality: The application should allow the end user to search for an item, and the search result should be displayed in the four text boxes. | | 2 |  |
| 5 | | Clear Search Input: After performing a search, the search input box must clear and retain focus, allowing the user to easily perform a new search. | | 3 |  |
| 6 | | Add/Edit/Delete Records: The end user must have the option to add, edit, or delete any of the four fields associated with a data structure record: Data Structure Name, Category, Structure, and Definition. | | 1 |  |
| 7 | | Confirmation for Deletion: During the deletion process, a popup box should prompt the user for confirmation before permanently deleting a record. | | 2 |  |
| 8 | | Data Storage: The prototype should utilize a two-dimensional array of type string to store each record's information. | | 1 |  |
| 9 | | Array Size Limitation: The program should have provisions for a fixed size of the array, with no provision for additional data beyond the defined limit. | | 1 |  |
| 10 | | Clearing Text Boxes: Double-clicking in the name text box should clear all four text boxes and focus the cursor back into the name text box. | | 3 |  |
| 11 | | Load and Save Data: The application must allow the end user to load and save data when the appropriate buttons are clicked. | | 2 |  |
| 12 | | Binary File Format: All wiki data should be stored and retrieved using a binary file format. | | 1 |  |
| 13 | | Alternative Data File Selection: During the load and save process, the end user can choose an alternative data file. | | 3 |  |
| 14 | | Error Trapping and Feedback: All end-user interactions should be accompanied by comprehensive error trapping and feedback messaging via a status strip at the bottom of the interface. | | 4 |  |
| 15 | | Tooltip: To show the description of each element. | | 4 |  |
|  | |  | |  |  |
|  | | *Add more lines as required* | |  |  |
| User Interaction and Specifications | | | | | |
| How will the application behave and what GUI specifications are required. | | | | | |
| Interface: | Display List: The interface should include a list where records are displayed for selection.  Text Boxes: There should be four text boxes on the left side of the interface to display information related to the selected record, and one for the user to enter search terms.  Buttons: The interface should have six buttons for the user to search, add, delete, edit, load, and save.  Status Strip: A status strip at the bottom of the interface should show feedback messages and error notifications.  Tooltips: To show the descriptions of all elements. | | | | |
| Record Selection: | Selection Mechanism: The end user should be able to select a record from the display list, either by clicking on it or using appropriate keyboard navigation.  Record Display: When a record is selected, its information (Data Structure Name, Category, Structure, and Definition) should be displayed in the respective text boxes. | | | | |
| Search Functionality: | Searching: The application should allow the end user to enter a search term in the search input box.  Display Search Result: Upon searching, if a matching record is found, its information should be displayed in the four text boxes.  Clear and Retain Focus: After the search, the search input box should clear, and the cursor should remain focused inside the box for easy input of a new search term. | | | | |
| Add/Edit/Delete Records: | Adding: The user should have the option to add a new record by providing the necessary information in the four text boxes.  Editing: The user should be able to edit the information of an existing record in the four text boxes.  Deletion: The user must be prompted with a popup box to confirm before deleting a record. Upon confirmation, the selected record should be removed from the display list and data storage. | | | | |
| Clearing Text Boxes: | Double-click: When the user double-clicks on the name text box, all four text boxes should be cleared, and the cursor should focus on the name text box for convenience. | | | | |
| Load and Save Data: | Loading: The application should allow the user to load data from a binary file using a designated button.  Saving: The application should allow the user to save data to a binary file using a designated button. | | | | |
| Alternative Data File Selection: | File Selection: During the load and save process, the user should be able to choose an alternative data file from their system. | | | | |
| Error Trapping and Feedback: | Comprehensive Error Handling: The application should handle errors gracefully and provide clear error messages to the user.  Status Strip Messages: All user interactions, including loading, saving, adding, editing, and deleting records, should trigger appropriate feedback messages displayed in the status strip at the bottom of the interface. | | | | |
|  |  | | | | |
| *Add more lines as required* | | | | | |

# Question 2 Version Control Specifications

CITE currently use GitHub as their primary source control; however, they would like you to investigate/research an alternative to GitHub. The purpose is to ensure CITE have chosen the best version control system for software development. Fill in the relevant sections of the following Version Control Specifications template to answer question two.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version Control Specifications | | | | | |
| GitHub VCS | | | Alternative VCS | | |
| VCS Name: | | Git | VCS Name: | | Mercurial |
| URL: | | <https://git-scm.com/> | URL: | |  |
| Major Features | | | | | |
| List all the major features associated with each version control system (ref: https://www.softwaretestinghelp.com/version-control-software/) | | | | | |
| 1. | distributed repository model. | | 1. | Fully distributed collaborative development. | |
| 2. | Support for non-linear development | | 2. | Handles both plain text and binary files robustly. | |
| 3. | Compatible with existing systems and protocols such as HTTP, FTP, and ssh. | | 3. | Possesses an integrated web interface. | |
| 4. | Handle small to large-sized projects. | | 4. | High performance and scalability. | |
| 5. | Cryptographic authentication of history. | | 5. | Decentralized. | |
| 6. | Pluggable merge strategies. | | 6. | Advance branching and merging capabilities. | |
| 7. | Toolkit-based design. | | 7. |  | |
| 8. | Periodic explicit object packing. | |  |  | |
| 9. | Garbage accumulates until collected. | |  |  | |
| *Add more lines as required* | | |  | | |
| Recommendation: Which VCS would you choose and why? | | | | | |
| *Add your detailed answer here…*  Both Git and Mercurial have their own strengths and weaknesses. Git is more popular and has a larger community, which has more resources available online. | | | | | |

## Question 3 UML Class Details

Create a simple UML Class Diagram for the 2D wiki data structure information. Ensure you have added the data structure and attributes. Fill in the relevant sections of the following UML Class Details template to answer question three.

|  |
| --- |
| UML Class Details |
| UML Diagram |
|  |
| Data Structure (use C# code) |
| using System;  using System.IO;  public class WikiData  {  Static int max = 10;  Static int fields = 4;  Int ptr = 0;  private string[][] dataStructuresArray = new string[max, fields];  private void SelectRecord(int index)  {  // Implementation to display corresponding information in the four text boxes.  }  private void ButtonSearch\_Click(object sender, EventArgs e)  {  // Implementation to search for an item and display the matching record's information.  }  private void ButtonAdd\_Click (object sender, EventArgs e)  {  // Implementation to add a new record with the provided data. By calling AddData, Clear\_TextBoxes, BubbleSort and DasplayListDataStr  }  private void AddData()  {  // Implementation to add a new record with the provided data.  }  private static void ‌BubbleSort()  {  // Implementation to sort records alphabetically and call Swap method.  }  private void Swap(int index)  {  // Implement to swap the records of the database regarding show data alphabetically in order.  }  private void DisplayListDataStr()  {  // Implement to show data of the records in the list box  }  private void ButtonEdit\_Click(object sender, EventArgs e)  {  // Implementation to edit an existing record with the updated data.  }  private void ButtonDelete\_Click(object sender, EventArgs e)  {  // Implementation to prompt and delete the selected record.  }  private void ClearTextBoxes()  {  // Implementation to clear all four text boxes and focus on the name text box.  }  private void ButtonLoad\_Click(object sender, EventArgs e)  {  // Implementation to load data from a binary file.  }  private void ButtonSave\_Click(object sender, EventArgs e)  {  // Implementation to save data to a binary file.  }  } |

## Question 4 Development Methodologies

You are required to create a comparison of four (4) software development methodologies that would be suitable to create the wiki prototype application. Your comparisons must include both traditional and non-traditional system development methodologies. Complete the following Development Methodologies template to answer question four.

|  |
| --- |
| Development Methodologies |
| 1st Methodology Name \_Agile\_\_\_\_\_\_ |
| Description: This methodology minimizes risk such as bugs, cost overruns and changing requirements when adding new functionality. Teams develop the software in iterations that contain mini-increments of the new functionality. There are different forms of the agile development method such as scrum, crystal, extreme programming (XP), and feature-driven development (FDD). |
| Diagram: |
| Advantages: It allows software to be released in iterations. Teams improve their efficiency by iterative releases to find and fix defects and align expectation early on. Users can understand software benefits earlier with frequent incremental improvements. |
| Disadvantages: Real-time communication is needed for Agile development methods. Therefore, new users often without the documentation they need to get up to speed. Because of user approval for developers in each iteration to complete each feature, they need a huge time commitment from users. It can be inefficient in large organizations. |
| 2nd Methodology Name \_DevOps\_\_\_ |
| Description: It is both development methodology and a set of practices that supports an organizational culture. It enhances collaboration between the departments responsible for different segments of the development life cycle, such as development, quality assurance, and operations. |
| Diagram: |
| Advantages: It is focused on improving time to market, decreasing the failure rate of new releases, shortening the lead time between fixes, and minimizing disruption while maximizing reliability. It automates continuous deployment to ensure everything happens smoothly and reliably. Reducing time, improving customer satisfaction, product quality, and employee productivity and efficiency are the benefits of using this methodology. |
| Disadvantages: Some customers don’t want continuous updates to their systems. Some industries have regulations that require extensive testing before a project can move to the operations phase. If different departments use different environments, undetected issues can appear in production. Some quality attributes require human interaction, which shows down the delivery pipeline. |
| 3rd Methodology Name \_\_Waterfall\_\_\_\_\_\_\_\_\_ |
| Description: It is a traditional method and rigid linear model that consists of sequential phases (requirements, design, implementation, verification, maintenance) focusing on distinct goals. Each phase must be 100% complete before the next phase can start. There is no process for going back to modify the project or direction. |
| Diagram: |
| Advantages: It is easy to understand and manage. It is suitable methodology for projects with clear objectives and stable requirements. It needs less experienced project managers and project teams. |
| Disadvantages: It is slow and costly because of its rigid structure and tight controls. |
| 4th Methodology Name \_\_Rapid\_\_\_\_\_\_\_\_\_ |
| Description: RAD is a condensed development process that produces a high-quality system with low investment costs. It can quickly adjust, so it has low investment cost. It has four phases, including planning, user design, construction, and cutover. The user design and construction phases repeat until the user confirms that the product meets all requirements. |
| Diagram: |
| Advantages: It is most effective for well-defined business objective projects and clearly defined user group, but not computationally complex. It is suitable for small to medium projects that are time sensitive. |
| Disadvantages: It needs a stable team composition with highly skilled developers and users who are deeply knowledgeable about the application area. Deep knowledge is necessary in a condensed development timeline that requires approval after each construction phase. |
| References |
| *Add references as required*  [Top 4 software development methodologies | Synopsys](https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/#:~:text=Top%204%20software%20development%20methodologies%201%201.%20Agile,method%20...%204%204.%20Rapid%20application%20development%20) |

## Question 5 Methodology Analysis and Selection

Refer back to the previous question and answer these two questions:

* What selection criteria determined your choice of the four development methodologies? Create a list of your section criteria.
* What methodology from question four would you recommend for this project? List your reasons why this is your preferred option.

Complete the following Methodology Analysis and Selection template to answer this question.

|  |
| --- |
| Methodology Analysis |
| Selection Criteria |
| Size of the team |
| Average experience of the developers |
| Stability and visibility of requirements |
| Time to complete the product |
| Teamworking |
| *Add more lines as required* |
| Methodology Selection |
| Methodology Name \_\_\_Agile\_\_\_\_\_\_ |
| Justification (reasons for selection) |
| It is suitable for small group. |
| Real time communication (face to face in class) |
| Developers have the same knowledge. |
| We need to document everything, so we don’t have the problem (lack of documentation) in Agile. |
| *Add more lines as required* |

## Question 6 Manage Version Control System

CITE uses GitHub as the primary Version Control System (VCS), create your own GitHub account (you can use your existing account). Ensure your development computer has Visual Studio installed with all the appropriate features to create and develop a C# .Net application. Open Visual Studio and create a new project; ensure the GitHub settings have been updated to reflect your GitHub account. Create a new repository for the project and check you have both a local and remote (Cloud) instance of the repository.

Complete the following Manage Version Control System template to answer this question.

Consult with your lecturer if you wish to use an alternative source control service. Any alternative source control must support a local desktop installation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Manage Version Control System | | | | | | |
| VCS Name | | Github | | | | |
| Version Details | | 2023 Github, Inc. | Date | 2023 | | |
| Supported Platforms | | Ubuntu Linux, Microsoft Windows, and macOS runners | | | | |
| Repository/Project Information | | | | | Yes | No |
| Has the cloud VCS account been created and named correctly? | | | | |  |  |
| Has the basic project solution been created? | | | | |  |  |
| Has the cloud VCS repository been created | | | | |  |  |
| Does the cloud VCS repository have a readme file? | | | | |  |  |
| Has the local source repository been created? | | | | |  |  |
| Are the two VCS resources linked? | | | | |  |  |
| Account Details Checklist | | | | | | |
| Repository Name: | wikiData | | | | | |
| URL | https://github.com/mehraneh1234/wikiData.git | | | | | |
| Local Source Control Screen Shot |  | | | | | |
| Cloud Screen Shot |  | | | | | |

## Question 7 Project Plan

Using your recommended development methodology from the previous question, create an initial project plan. List and describe all the tasks required to complete the development of the wiki application. Use the following Project Plan template to answer this question. Using the GitHub repository from the Manage VCS question create a Project within the repository and add your project plan.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Plan | | | |
| Project Name | Wiki Data | | |
| Date | 02/08/2023 | | |
| Developer Name | Mehraneh Hamedani | | |
| Development Tasks | | | |
| Task Name | Task Type | Task Description | Input/Output Parameters |
| analysis | documentation | Read the document of clients’ needs then analysed them. After that documentation of all requirements, user interactions, and specifications. By comparing the different VCSs chose GitHub then designed a UML Class diagram and with C# write the data structure. Agile is the chosen development Methodology to use in this project then manage GitHub by making a repository in local and cloud to start coding. | Read the client documentation/ document all the requirements |
| Sign off | Documentation and conversation | Display all work to the lecturer and had a conversation to send my document to him by email. | Show the documentation of all requirements to the lecturer/ start coding and completing the project |
| Code | programming | By C# write all codes | Class diagram and data structure of coding/ application of Wiki |
| Test | programming | After finishing the codes, test them. | Different testing/ resolve all bugs and errors |
| demonstration | presentation | Show how the application works. | Application/ pass the course |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Question 8 Sign-off and Approval

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

Question 1 – Project Specifications

Question 2 – Version Control Specifications

Question 3 – UML Class Details

Question 4 – Development Methodologies

Question 5 – Methodology Analysis and Selection

Question 6 – Manage Version Control System

Question 7 – Project Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Approval (Lecturer/Assessor use only) | | | | |
| Approver Name | Title | Signature | Date | Approved? |
|  |  |  |  |  |
|  |  |  |  |  |
| Lecturer Feedback | | | | |
|  | | | | |

## Question 9 Prototype Development

Create the wiki prototype to demonstrate how a collection of information can be stored using a Windows Application (WinForms). This prototype application will utilise a two-dimensional array with 12 rows and 4 columns (**use a** **simple 2D string array – not collections).** Use the hardware and software supplied in the classroom to accomplish the development, debugging and internal documentation of the prototype. Use the Version Control System from the previous Question to manage your code during the development; ensure you record these commits/branches as a series of screen shots to be included in Version Update Question (ie start, working, final). Your code must adhere to the CITEMS software development standards. (refer http://www.citems.com.au/)

**Note:** You are not permitted to use a class structure; this assessment is a demonstration of a simple 2D array of type string.

The following programming criteria and features are required, check the original project specifications for additional information.

### Programming Criteria

1. Create a global 2D string array, use static variables for the dimensions (row = 12, column = 4),
2. Create an ADD button that will store the information from the 4 text boxes into the 2D array,
3. Create an EDIT button that will allow the user to modify any information from the 4 text boxes into the 2D array,
4. Create a DELETE button that removes all the information from a single entry of the array; the user must be prompted before the final deletion occurs,
5. Create a CLEAR method to clear the four text boxes so a new definition can be added,
6. Write the code for a Bubble Sort method to sort the 2D array by **Name** ascending, ensure you use a separate **swap** method that passes the array element to be swapped (do not use any built-in array methods),
7. Write the code for a Binary Search for the **Name** in the 2D array and display the information in the other textboxes when found, add suitable feedback if the search in not successful and clear the search textbox (do not use any built-in array methods),
8. Create a display method that will show the following information in a ListView: Name and Category,
9. Create a method so the user can select a definition (Name) from the ListView and all the information is displayed in the appropriate Textboxes,
10. Create a SAVE button so the information from the 2D array can be written into a **binary file** called ***definitions.dat*** which is sorted by **Name,** ensure theuser has the option to select an alternative file. Use a file stream and BinaryWriter to create the file.
11. Create a LOAD button that will read the information from a binary file called ***definitions.dat*** into the 2D array, ensure theuser has the option to select an alternative file. Use a file stream and BinaryReader to complete this task.
12. All code is required to be adequately commented, and each interaction must have suitable error trapping and/or feedback. All methods must utilise the appropriate Dialog Boxes, Message Boxes, etc to ensure fully user functionality. Map the programming criteria (9.1 - 9.11) 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/).

**Note:** 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.

## Question 10 Version Control Update

At the conclusion of the code development record the Version Control changes, commits and pull requests with a series of suitable screen shots. Complete the following Version Control Update template to answer this question. A minimum of three screen captures is required (ie start, working, final)

|  |  |
| --- | --- |
| Version Control Update | |
| Repository Name: | wikiData |
| URL | [mehraneh1234/wikiData (github.com)](https://github.com/mehraneh1234/wikiData) |
| Desktop Screen Shots |  |
| Cloud Screen Shots |  |

## Question 11 Data Structure Matrix

Create test input data by researching and providing a definition for the 12 data structures listed in the Data Structure Matrix template below. The definitions must be between 20-40 words and contain real information which will be entered and saved by the wiki prototype. Add your definitions to the following Data Structure Matrix template to answer this question.

|  |  |  |  |
| --- | --- | --- | --- |
| Data Structure Matrix | | | |
| NAME | CATEGORY | STRUCTURE | DEFINITION |
| Array | Array | Linear | It stores a collection of elements in contiguous memory, which accessible by index. |
| Two Dimension Array | Linear | An array with two indices for representing tables or matrices. |
| List | List | Linear | It is a dynamic data structure to store a sequence of elements and allows insertions and deletions. |
| Linked list | Linear | Elements or nodes are connected by pointers and in this linear data structure can do insertions and removals. |
| Self-Balance Tree | Tree | Non-Linear | A binary tree that automatically adjusts its structure to ensure balanced heights, optimizing search operations. |
| Heap | Non-Linear | A binary tree-based data structure where the parent nodes have a specific order compared to their children. |
| Binary Search Tree | Non-Linear | A binary tree which left children containing smaller values and right children containing larger values. |
| Graph | Graphs | Non-Linear | A collection of nodes connected by edges (relationships between various entities. |
| Set | Abstract | Non-Linear | A data structure that stores unique elements without order and good for membership checking. |
| Queue | Linear | It has First-In-First-Out (FIFO) principle for element access. |
| Stack | Linear | It has Last-In-First-Out (FIFO) principle for element access. |
| Hash Table | Hash | Non-Linear | It stores key-value pairs, using hash functions to index values. |
| References | | | |
| *Add references as required*  [Learn Data Structures and Algorithms (programiz.com)](https://www.programiz.com/dsa) | | | |

## Question 12 UML Activity Diagram

Create a detailed UML Activity Diagram for the Binary Search method. Start by copying your C# code into the right side of the UML Activity Diagram section, then add your UML Activity Diagram in the left side. Now, using the code and the UML diagram identify breakpoints so all major pathways are tested. Update the C# Code on the right section to identify the breakpoints. The example in the appendix is provided for clarification. Complete the following UML Activity Diagram template below.

|  |  |
| --- | --- |
| UML Activity Diagram | |
| Diagram | C# Code |
| **Create your UML activity Diagram here** | **Copy of your Binary Search Code here**  private void BinarySearch(string searchName)  {  // Deselect previous selected item  if (listViewData.SelectedItems.Count > 0)  {  listViewData.SelectedItems[0].Selected = false;  }  // Clear the description on the bottom of the form  StatusStripDataStr.Items.Clear();  int left = 0;  int right = max - 1;  int result = -1;  // Loop to find the searchName in the 2d array by dividing the 2d array rows in two  // and comparing the searchName with the two divided parts and eliminating the part  // which does not have the searchName and repeat this loop upto find it.  while (left <= right)  {  int mid = left + (right - left) / 2;  if (dataArray[mid, 0] == searchName)  {  result = mid;  break;  }  else if (string.Compare(dataArray[mid, 0], searchName) < 0)  {  left = mid + 1;  }  else  {  right = mid - 1;  }  }  // If searched name is found then the data of that row is displayed in the text boxes.  if (result != -1)  {  // the searched name is selected in the listview  listViewData.Items[result].Selected = true;  listViewData.Focus();  txtDataStrName.Text = dataArray[result, 0];  txtCategory.Text = dataArray[result, 1];  txtStructure.Text = dataArray[result, 2];  txtDefinition.Text = dataArray[result, 3];    StatusStripDataStr.Items.Add("Target found.");  }  // If searched name is not found then displayed "Entry not found!" and clear textboxes  // and deselect the listview.  else  {  StatusStripDataStr.Items.Add("Target not found!");  txtSearch.Clear();  // Deselect previous selected item  listViewData.SelectedItems[0].Selected = false;    }  } |

## Question 13 Debug Test Report

Using the breakpoints shown in the previous Question as a starting point, utilise the debug features to debug, trace and test your Binary Search code. Ensure your code is error free and functions correctly (refer Programming Criteria) record and correct any errors. Your Debug Test Report must include appropriate evidence that your code functions as expected (references to screen captures). Complete the following Debug Test Report template below.

* Ensure you have entered 12 records from previous Question before you begin testing,
* Place a break point at each Decision and Loop construct and record the data as it changes,
* Use a test data item that will be found (ie Stack),
* Use a test data item that will not be found (ie ArrayList).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Debug Test Report | | | | | | | | | | |
| Project Name | | Wiki Data | | | | | | | | |
| **Method** | | Binary Search method | | | | | | | | |
| **Description** | | This binary search has done in an array (just for the rows and first column of the 2d array). It compares the searched item with the mid item in the array if they are the same that item is found, if not and the index of searched item is lower than mid then get rid of the other part and do the same with the remain part again. This procedure continues up to find the searched item otherwise it show a message that the searched item not found. | | | | | | | | |
| **Level of Testing** | | 7 | | | | | | | | |
| **Developer** | | Merhraneh | | **Tester** | Mehraneh | | **Date** | | 16/08/2023 | |
| **Test Case No** | **Test Case Name** | | **Test steps** | | | **Test Data** | | **Expected result** | | **Pass / Fail** |
| 1 | Boundary Value Analysis (the data which exists) | |  | | | Array | | Found True mid 0 Data Name Array | | Pass |
| 2 | Boundary Value Analysis (the data which exists) | |  | | | Two dimention array | | Found True mid 11 Data Name Two dimension array | | Pass |
| 3 | Boundary Value Analysis (the data which exists) | |  | | | Linked list | | Found True mid 5 Data Name linked list | | Pass |
| 4 | The data which does not exist | |  | | | mappp | | Target not found | | Pass |
| 5 | The data which does not exist | |  | | | 1254 | | Entry not found | | Pass |
| 6 | The data which does not exist | |  | | | b+ | | Entry not found | | Pass |
| 7 | Error Guessing Technique | |  | | | “” | | Please enter a search term. | | Pass |

## Question 14 Post Development Analysis

Once you have completed coding and testing of this prototype application you can answer the following questions and complete the Post Development Analysis section in the template below.

1. What Software Development Life Cycle did you use during the development of the Prototype Application?
2. How effective was your project plan in developing the final prototype?
3. What alternative data structures could be used?
4. What constructive advice could you provide for the development of a similar project?

|  |
| --- |
| Post Development Analysis |
| Questions |
| What SDLC did you use during the development of the Prototype Application?  I used Agile methodology because it allows for iterative development and quick feedback. |
| How effective was your project plan in developing the final prototype?  It was effective because it clarified the requisites to develop and complete the project on time. Because of the iterative process, regular testing, code review, and user feedback can help ensure the effectiveness of the project plan and the quality of the final prototype. |
| What alternative data structures could be used?  Instead of 2D string array, List of custom objects where each object holds the information for a definition. |
| What constructive advice could you provide for the development of a similar project?  It is complete because it has a plan (before coding plan out the project structure, and UI layout), modular code (it makes the code more readable, maintainable, and testable), version control (Git helps keep track of changes, collaborate with others, and maintain the history of the code), error handling, user feedback (through different actions and operations, it provides informative messages), testing, documentation, code review, incremental development, and read guidelines (to ensure to meet all the requirements). Therefore, I cannot provide any advice for the development of a similar project. |

## Question 15 Demonstration and Submission

Demonstrate your working program to your lecturer using the realistic data from the previous Question. Ensure your code is fully commented with your Name, ID, Date, Assessment Task placed above the workspace header. Ensure all the documentation has been completed and is ready for inspection.

**Note:** All documentation must use the supplied templates/forms.

**Submit the zipped solution folder with relevant documents to Blackboard**

End of Assessment Task One

## Appendix

Diagram

Description automatically generated