|  |  |  |  |
| --- | --- | --- | --- |
| **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** | |  | |
| **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 | | **Re-assessment 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 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 quest ion one.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Specifications | | | | | |
| Project Name | | | 2D Array Wiki | | |
| Date | | | 30/01/2023 | | |
| Developer Name | | | Peter Halligan | | |
| Client Requirements | | | | | |
| Req. # | | Description | | Importance | Notes |
| 1 | | Load/Save array as binary File | | 1 | The records must be stored in a binary file format |
| 2 | | Save/load prompt on form load | | 2 | When opening the form, the user will be prompted if they wish to load a form |
| 3 | | Add/edit/search/delete/Save/Load buttons | | 2 | Buttons for basic navigation and manipulation of the form and its stored data |
| 4 | | Delete prompt box | | 2 | User will be prompted to confirm deletion |
| 5 | | 2D array of type String | | 1 | Data type must be a 2D array all with the type of string |
| 6 | | Search box will refocus and clear after use | | 3 | After a search the search textbox will regain focus and clear |
| 7 | | Full error catching | | 1 | Try catch like a boss |
| 8 | | Message feedback on status strip | | 3 | Status strip |
| User Interaction and Specifications | | | | | |
| How will the application behave and what GUI specifications are required. | | | | | |
| 1 | User can select a record from a display list, related info will display in 4 text boxes on the left. | | | | |
| 2 | Search bar should be able to do the same, after search, clear search box and focus on search box. | | | | |
| 3 | The user can add/edit/delete any of the four fields independently. | | | | |
| 4 | The four fields are: Data Structure Name, Category, Structure and Definition. | | | | |
| 5 | Popup confirmation during deletion. | | | | |
| 6 | 2D array of type sting for data storing (with set size in program criteria). | | | | |
| 7 | Double mouse click will clear 4boxes and focus cursor on name box. | | | | |
| 8 | Save and load function/buttons. | | | | |
| 9 | Data saves as a binary file format. | | | | |
| 10 | Save/Load prompts user for file name and location. | | | | |
| 11 | Full error trapping. | | | | |
| 12 | Feedback status strip. | | | | |
| *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: | | GitHub | VCS Name: | | Mercurial |
| URL: | | https://github.com/ | URL: | | https://www.mercurial-scm.org/ |
| Major Features | | | | | |
| List all the major features associated with each version control system (ref: https://www.softwaretestinghelp.com/version-control-software/) | | | | | |
| 1. | Protected Branches | | 1. | High scalability | |
| 2. | Cross platform | | 2. | Client central server | |
| 3. | Merge Protection | | 3. | Fast due to being local | |
| 4. | High scalability | | 4. | Allows all platforms | |
| 5. | Wiki Pages, document sharing | | 5. | Open source allowing greater control | |
| 6. | Continuous Development | | 6. | Wide array of extensions | |
| 7. | Github Pages, easy website hosting | | 7. | Speed (in part due to point2) | |
| *Add more lines as required* | | |  | | |
| Recommendation: Which VCS would you choose and why? | | | | | |
| *Add your detailed answer here…*  I would recommend Github as it is a well-documented well explored VCS and would be easier to learn therefore allowing faster launch speed of the development stage. This is further backed by its fantastic organisation and safety feature that mean I almost never have to worry about data loss, also it has massive room for scale. | | | | | |

## 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) |
| static int rows = 12;  static int columns = 4;  int pointer = 0;  string[,] multiArray = new string[rows, columns];  Example:  multiArray[0, 0] = "Name A";  multiArray[0, 1] = "Category B";  multiArray[0, 2] = "Structure A";  multiArray[0, 3] = "Definition A"; |

## 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 \_Waterfall\_\_\_\_\_\_\_\_\_ |
| Description:  Waterfall is the oldest of the SDLC’s. It uses a linear flow to work through the entire development process. Meaning that a phase can only be started after the last one is complete. |
| Diagram: |
| Advantages:  Waterfall is simple so it may be understood and used in all cases.  Clearly Defined Stages  Phases are processed one at a time |
| Disadvantages:  High amount of uncertainty as no software is produced until late in the cycle.  Not suitable for long runs or large scale.  Cannot change requirements until the end of a run. |
| 2nd Methodology Name \_Iterative\_\_\_\_\_\_\_\_\_\_ |
| Description:  An iterative SDLC works like a water fall but there is a lessor focus on getting all the requirements at the very beginning as at the end they can be updated and a second build will be started where the process is repeated but with the first build as a base for the second and so on. |
| Diagram: |
| Advantages:  Fast at getting a functional software early.  Can plan parallel development  Changing Scope/Requirements is less costly.  Superior risk analysis. |
| Disadvantages:  More resource heavy.  End of project may not be known.  Highly skilled resources are required for risk analysis. |
| 3rd Methodology Name \_Spiral\_\_\_\_\_\_\_\_\_\_ |
| Description:  A combination of iterative model and sequential linear development model, or, Waterfall with a very high increase in risk analysis. |
| Diagram: |
| Advantages:  Early System.  Requirements can be changed.  More extensive use of prototypes. |
| Disadvantages:  More complex management.  End of project may not be known.  Process is complex.  Spiral may be endless. |
| 4th Methodology Name \_RAD \_\_\_\_\_\_\_\_\_\_ |
| Description: The RAD model focuses on gathering customer requirements, early testing of prototypes, reuse of existing components and continuous integration and rapid delivery. |
| Diagram: |
| Advantages:  Progress can be measured.  Higher productivity with fewer people in shorter time.  Increased reusability of components. |
| Disadvantages:  Dependency on technically strong team members for identifying business requirements.  Can only be used for systems that can be modularized.  Requires highly skilled developers/engineers. |
| References |
| *Software Development Life Cycle (SDLC) on blackboard C# Topic Six:*  *https://learn-ap-southeast-2-prod-fleet01-xythos.content.blackboardcdn.com/5fcebc1997e08/3438057?X-Blackboard-S3-Bucket=learn-ap-southeast-2-prod-fleet01-xythos&X-Blackboard-Expiration=1675674000000&X-Blackboard-Signature=d8v%2BUFZ%2B8Wao5wOSiYO4RZmyAARaaGpmeACiAJk%2BTl0%3D&X-Blackboard-Client-Id=309467&X-Blackboard-S3-Region=ap-southeast-2&response-cache-control=private%2C%20max-age%3D21600&response-content-disposition=inline%3B%20filename%2A%3DUTF-8%27%27SDLC.pdf&response-content-type=application%2Fpdf&X-Amz-Security-Token=IQoJb3JpZ2luX2VjELP%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaDmFwLXNvdXRoZWFzdC0yIkcwRQIhAOjWtm9oKnzAZ22ggH0JMiMVBLm%2FGoqzUJYvWztzJInJAiA%2BdtyFkw%2BtHEivAUsSf8yp%2Fjmg8FziEKxaiqUmWPJDyCrbBAg8EAAaDDU1NjkwMzg2MTM2MSIMCcjkCg1Xt5MGpMIYKrgEuOj0xik%2BF7WyAqAUe4leaDwkXFo9d8D%2FtjEqYxb1MFxJblwnt2QtOkrcCluI0H%2BGVx3z0%2FEqQODm9DxmULcKuYAItd69B2hxnugtORW2lZoTJNtM3I35rD42vSBfA%2BTFc7ZN6vshA5QLiQDJ72XIPfiXv%2F%2BKGO5fsJAKD%2BB9cmJf16rArsFAP7GLc6utZ34rSj3ywS0VgIP9dXjNCHuwQVH6PO2JP1ywL1oa3kEhWqCg0jYaACyJQag1MsoR8cRysQEa4AtVngTJ%2Bgs5lgE0bU1Kt3IDvfAgMxpMOklYipQU9jBT%2FJV2qsWYEhpSPZtbKIepv9f5lohsy%2FDjHoL7FJbtvx3ylnZwpRdqLezing%2BMR4AHDSTxNgq5oCJ7epJPJlnurCv4BqmFgAz%2B9ca%2F2oKxkLqgn45SakbVauRqn0UN%2FopPzWJwE2Zpg0zSeJbR4ArMk1KE4tdoRSj8ZoW0HV%2FXRjY69xe%2Brw%2BZFnvakW%2Frg0oz%2BiwVK88P%2BV9gw38wqWt1N3FAr9AoKIaAYm9dH%2FFaRuHIBp2PhAqbM5aNvnNesAJI7MhsQO8p80ncJHmad%2FxaYguaydRAu2rN0cTzSy81%2FI8q34D560OCsLj5NZUGOFyvvU1e6rU9bZg4soZvWCii%2B3zHKAzsTYP4HO6Tn5Kkgbh3e8go6YQBK8OES9SVdVa%2BrcYmsybtdnkVDCkRNOok98Q7Lj3HrTQ0uM0apm35MrJzAMEpRDk%2FPmMwi5b72dr%2Fx18d3jD42YGfBjqpAdtXNoZtPKF6wDwnYA55kFs1rneztq2yA5AFVnVwfgaJmN98qZQfS43TRAzXkFVITkVCcs6c2jq2Wpv%2BT3ydzkC0BeD44CrNDbimygcdIYWdTb0wD9FnckIVPa4XbCLFZvbYhCVCIYBR6%2FBJnlny%2BhAeonVHUPrwkC3%2FeD4YrPad5Nr32n8bBJ1tPfR6D%2BX9FU98d%2FWnJFwCiCgpxvEDYq%2F3kEgPJlqpdOo%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20230206T030000Z&X-Amz-SignedHeaders=host&X-Amz-Expires=21600&X-Amz-Credential=ASIAYDKQORRYQ7I4X45V%2F20230206%2Fap-southeast-2%2Fs3%2Faws4\_request&X-Amz-Signature=117516a4cb80511cc5fb436affaef4949df45c4765861acb53b9718edb8bd45f* |

## 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 |
| I will be using the Agile SDLC method as it will allow me greater flexibility and faster to market time which will be very helpful as a student, Agile also has a high level of transparency allowing greater. Also Blake said so, so yeah that to. |
|  |
|  |
| *Add more lines as required* |
| Methodology Selection |
| Methodology Name \_Agile\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Justification (reasons for selection) |
| The Agile Software Development Life Cycle Methodology is the one I have selected for this assessment as it will allow me greater flexibility such as making changes as needed instead of waiting until the next iteration, this will be very helpful as a student that is still learning his craft. Another reason is that I will have a working prototype faster than if i used another SDLC like waterfall for example. Also Blake said so. |
|  |
|  |
| *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 | | V1 | Date | 14/02/23 | | |
| Supported Platforms | | Win | | | | |
| 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: | 2DarrayAT1 | | | | | |
| URL | [**https://github.com/theFlay/2DarrayAT1**](https://github.com/theFlay/2DarrayAT1) | | | | | |
| 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 | 2DarrayAT1 | | |
| Date | 20/02/23 | | |
| Developer Name | Peter Halligan | | |
| Development Tasks | | | |
| Task Name | Task Type | Task Description | Input/Output Parameters |
| List requirements | Design | Gather Client requirements |  |
| Initial UI Design | Design | Add required elements |  |
| Code template | Design | Create template to add code to |  |
| Build array | Code | Build array and basic features |  |
| Add/Edit/Remove Buttons | Code | Implement buttons |  |
| Save/Load Buttons | Code | Implement feature as binary file |  |
| Confirmation Box | Code | Feature implementation |  |
| Status strip | Code | Feature implementation |  |
| Add Error catching | Code | Feature implementation |  |
| Test | Testing | Testing features and error catching |  |
| Refine | Code | Tweak as needed |  |

## 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 = 4, column = 12),
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: | 2DarrayAT1 |
| URL | https://github.com/theFlay/2DarrayAT1 |
| 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 | An array stores elements of the same type in a continuous and adjacent memory locations and assigns them a value to call on them starting at 0. It has a fixed size that’s defined when it is created |
| Two Dimension Array | Linear | How a 2D array differs from a normal array is it’s the same in every way except it stores a collection of collections. |
| List | List | Linear | A list works like an array except it can have its size changed after creation, however cannot have null entries meaning if you remove index 3 all other entries past three must move up one to fill the gap. |
| Linked list | Linear | A linked list is a list that does not store its items in a continuous wa but instead will have a pointer to the next location for each item, meaning if you want to change an item you have to iterate from the first item all the way though to the current one. |
| Self-Balance Tree | Tree | Non-Linear | A self balanced tree is a data structure that stores things in a hierarchical tree like structure and balances to ensure efficiency and speed. |
| Heap | Non-Linear | A heap stores elements in a complete binary tree and maintains the heap property to allow efficient retrieval of the max or min element. In a heap, the value of each parent node is greater (or smaller) than the values of its children nodes. |
| Binary Search Tree | Non-Linear | A Binary search tree stores data in a hierarchical tree structure where each node has at most two children. The binary search tree maintains the binary search property, where the value of the left child is less than the value of the parent node, and the value of the right child is greater than the value of the parent node. This allows for efficient search and insertion operations on the tree. |
| Graph | Graphs | Non-Linear | A graph is a set of vertices connected by edges it is used to model complex relationships between elements. |
| Set | Abstract | Non-Linear | A set is a collection of data where each element is unique and the order does not matter. |
| Queue | Linear | A Queue is a array or linked list that arranges is collection of elements in a First in First Out , it is useful for modelling real world scenarios such as waiting lines |
| Stack | Linear | A Stack is much like a reverse Queue, it’s a list or array of non unique elements ordered in a Last In First Out order. Its used for modelling things like browser history for example. |
| Hash Table | Hash | Non-Linear | A hash table is an array of elements where each item is mapped to an index, a unique hash value is then generated for each element that acts as the map to each element. |
| References | | | |
| [*https://en.wikipedia.org/wiki/Abstract\_data\_type*](https://en.wikipedia.org/wiki/Abstract_data_type)  [*https://en.wikipedia.org/wiki/Hash\_table*](https://en.wikipedia.org/wiki/Hash_table)  *asked chatpgt to summarise some of them from wiki because they use many big words*  *.* | | | |

## 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** |

## 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 | |  | | | | | | | | |
| **Method** | |  | | | | | | | | |
| **Description** | |  | | | | | | | | |
| **Level of Testing** | |  | | | | | | | | |
| **Developer** | |  | | **Tester** |  | | **Date** | |  | |
| **Test Case No** | **Test Case Name** | | **Test steps** | | | **Test Data** | | **Expected result** | | **Pass / Fail** |
|  |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |

## 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? |
| How effective was your project plan in developing the final prototype? |
| What alternative data structures could be used? |
| What constructive advice could you provide 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, use the following Assessor Marking Guide to ensure all code and documentation is compliant.

### Assessor Marking Guide

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assessor Marking Guide | | Satisfactory | | Comment |
| **Questions** | | YES NO | |  |
| Q1 | Project Specifications: All proposed program criteria is listed within the Client Requirements and User Interaction and Specifications. |  |  |  |
| Q2 | Version Control Specification: All fields are completed and there are there are suitable comparison between the source control systems |  |  |  |
| Q3 | UML Class Diagram: List four attributes and a simple UML class diagram showing Array |  |  |  |
| Q4 | Development Methodologies: List four different SDLC methodologies. |  |  |  |
| Q5 | Methodology Analysis: List several (three or more) selection criteria, Lists several (three or more) recommendations for SDLC |  |  |  |
| Q6 | Manage Version Control System: All fields are completed and there are there are suitable screen shots of GitHub. |  |  |  |
| Q7 | Project Plan: The plan reflects the selections from previous questions and has all the SDLC tasks outlined |  |  |  |
| Q8 | The Lecturer has assessed Q1-7 and provided feedback. |  |  |  |
| Q9 | Prototype Development: all program criteria and feature have been coded, software standard has been implemented in layout and comments.  Organisational Standards: all program criteria and feature have been coded, the software has satisfied the CITE organisational standards. |  |  |  |
| Q10 | Version Control Update: The GitHub account has been updated and there are suitable screen shots to demonstrate changes during the application development. |  |  |  |
| Q11 | Data Structure Matrix: All 12 data structure have a suitable definition. |  |  |  |
| Q12 | UML Activity Diagram: The code has breakpoints in the Decision and Iterative constructs, the activity diagram reflects the C# code. |  |  |  |
| Q13 | Testing and Debugging: All four bullet points have been tested and the results recorded. |  |  |  |
| Q14 | Post Development Analysis: All four questions have been answered. |  |  |  |
|  | **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 One

## Appendix

Diagram

Description automatically generated