|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Release 8.0) | | |
| **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** | Joshua Farrell | | | | | |
| **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 Topic One: 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

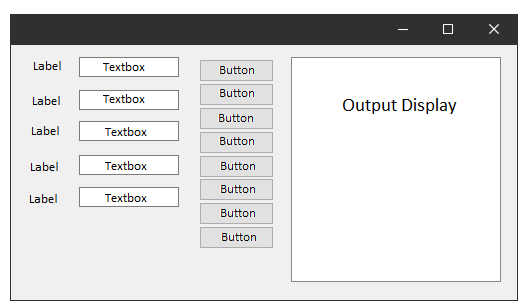
The client was interviewed, and the following information was recorded. The client would like the end user to select a record from a display list box and then have the corresponding information displayed in four text boxes on the left side of the interface. The end user should be able to search for an 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 question ten 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)



## 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 Application | | |
| Date | | | 20/07/2022 | | |
| Developer Name | | | Joshua Farrell | | |
| Client Requirements | | | | | |
| Req. # | | Description | | Importance | Notes |
| 1 | | A record can be selected from a list box by the user and have corresponding information displayed in four text boxes. | | 1 | Record data needs to use a mouse click event |
| 2 | | Records can be searched by the user which will be displayed in the four text boxes, which are named Data Structure Name, Category, Structure and Definition. The search box must be cleared and retain focus. | | 1 |  |
| 3 | | Records can be added, edited and deleted by the user. | | 1 |  |
| 4 | | If a record is to be deleted, the user must be prompted by a popup box. | | 1 |  |
| 5 | | Records must be stored using a two-dimensional array of type string. The array size is 12 rows and 4 columns. | | 1 |  |
| 6 | | If the name text box is double clicked, all four text boxes must be cleared. | | 1 |  |
| 7 | | Records must be loaded and saved using load/save buttons using a binary file format. | | 1 |  |
| 8 | | The end user must be able to select an alternative data file. | | 1 |  |
| 9 | | The program must have full error trapping and give feedback to the user using a status strip | | 1 | Error trapping may use try-catch blocks with IOException |
|  | | *Add more lines as required* | | Could be order of tasks |  |
| User Interaction and Specifications | | | | | |
| How will the application behave and what GUI specifications are required. | | | | | |
| 1 | Records must be stored using a two-dimensional array (Client Requirement [CR] 5) and be output to a list box. | | | | |
| 2 | Records must be output to four text boxes named Data Structure Name, Category, Structure and Definition (CR 2). The Name text box must clear all text boxes if it is double clicked (CR 6) | | | | |
| 3 | The list box must have a selection event which will output the index of the selected record (CRs 1, 3 to 4). The index will be used to output the correct record’s contents to the four text boxes. | | | | |
| 4 | Records must be searched, added, edited and deleted using buttons (CRs 2 to 3). Deleting a record must also have a message box to warn the user (CR 4) | | | | |
| 5 | Records must be loaded and saved using buttons (CRs 7 to 8) | | | | |
| 6 | The program must output general feedback to a status strip and only use Message Boxes for serious errors (e.g., IO exceptions) (CRs 4 and 9) | | | | |
|  |  | | | | |
|  |  | | | | |
| *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: | | Subversion |
| URL: | | https://github.com/ | URL: | | https://subversion.apache.org/ |
| Major Features | | | | | |
| List all the major features associated with each version control system (ref: https://www.softwaretestinghelp.com/version-control-software/) | | | | | |
| 1. | GitHub supports the use of branches to test new features in a non-linear development environment. | | 1. | Subversion copies a master directory to use as a branch. | |
| 2. | Each developer has their own copy of the repository which enables the developer to work without requiring a constant connection to GitHub. | | 2. | Each developer connects to a centralised repository server using a client-server model to download a working copy. This could be used in environments where code is to be kept on-premises, however it could cause problems when a developer needs to work from home. | |
| 3. | Git was originally developed to handle large projects such as the Linux kernel, allowing complex projects to be controlled. | | 3. | A developer can lock (or check out) a file in a repository so that certain files such as graphics files can be modified on a working copy without the risk of file corruption by another developer. | |
| 4. | GitHub (mostly) handles branch merges by incorporating changes from the feature branch to the master branch. If there are any merge conflicts, GitHub will tell the developer what code to keep. | | 4. | Subversion locally handles merges between a repository and a working copy. If there are any merge conflicts, Subversion will allow the developer to view all changes and edit the merged file to manually specify what code to keep. | |
| 5. | GitHub is platform-independent, provided that the system is capable of running and uploading a local Git repository. | | 5. | Subversion is distributed by source code, provided that an operating system is capable of compiling the program, and connecting to a remote repository to use as a working copy. | |
| 6. | GitHub can link repositories to projects which can be displayed using a Kanban board. | | 6. | Subversion commits the whole working copy as a single atomic transaction, which creates a new revision on the repository. | |
| 7. | GitHub allows any developer to track changes between commits on any branch | | 7. | Subversion allows any developer to track the current state of a file between a working copy and the repository. | |
|  |  | |  |  | |
| *Add more lines as required* | | |  | | |
| Recommendation: Which VCS would you choose and why? | | | | | |
| *I would select GitHub as it is directly supported by Microsoft Visual Studio, whereas Subversion would be run by an external program. GitHub is a well known system that uploads a local Git repository to GitHub.* | | | | | |

## 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 row = 12;  static int col = 4;  private string[,] ArrayWiki = new string[row, col]; |

## 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 (Scrum) |
| Description: Agile is a non-traditional methodology that emphasises short sprints with limited tasks that contribute to meeting a sprint goal. This methodology is usually driven by the developers with limited input from the client, except to gather requirements. |
| Diagram: |
| Advantages Agile methodologies such as Scrum allows small teams to work on a portion of a larger project for a short period of time (no longer than 2 months). |
| Disadvantages Agile is claimed by larger organisations to be inefficient as they believe that most agile methodologies are too extreme without receiving any benefits. |
| 2nd Methodology Name Waterfall |
| Description: The waterfall methodology is a traditional system development methodology that breaks down project activities into sequential phases. This methodology is usually driven by the client with limited input from the developer. Waterfall was initially used in manufacturing and construction where any changes would be almost impossible to implement. |
| Diagram: |
| Advantages This methodology is well known in the business environment, particularly the manufacturing and construction industries, as they generally have stable and structured physical environments. |
| Disadvantages Any changes to a design are generally not possible, or prohibitively expensive. If applied to code, if there was a bug within a method which outputs an incorrect value, the project deliverable would not be accepted by the client and a new waterfall would begin. |
| 3rd Methodology Name Rapid Application Development (or Small Scale Application Development) |
| Description: Rapid Application Development (RAD) is a non traditional development methodology that is used with individual or small team projects who are responsible for managing their own project. This is a less formal methodology that can be used for personal projects that have a small source code size (typically one form with very few methods). |
| Diagram: |
| Advantages RAD allows an individual or a small team to work on a project without formal oversight. RAD also allows teams to work on an application that performs one or a few related tasks. |
| Disadvantages RAD would be difficult to implement in large teams as it would be hard to coordinate work between the team members without a centralised management and governance structure. RAD would be impractical to use in large projects where there are many unrelated tasks, such as writing an operating system. |
| 4th Methodology Name Iterative |
| Description: The iterative model is a traditional development methodology that allows a team to work on a subset of requirements, which is used to identify any further requirements in the next phase of the project. |
| Diagram: |
| Advantages The iterative model can be used in projects where the requirements are not fully known, and unlike the waterfall model, it allows a project team to revisit code generated from earlier stages to improve it. |
| Disadvantages The iterative model can be seen as a series of smaller waterfall projects, so any bugs identified in the current cycle would have to be fixed in the next cycle. |
| References |
| *Add references as required* |

## Question 5 Methodology Analysis and Selection

Refer back to question four 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 |
| An individual can work on the project |
| The project is mostly self-managed with limited client interaction |
| The project is limited in scope by the client |
| *Add more lines as required* |
| Methodology Selection |
| Methodology Name Rapid Application Development |
| Justification (reasons for selection) |
| The project can be worked on by an individual with limited client involvement |
| The project is limited by the client requirements, which reduces the scope of the project |
|  |
| *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 | | v0.0.1-alpha-2-e5397acd | Date | 10/08/2022 | | |
| Supported Platforms | | Windows 10/11 x64 | | | | |
| 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: | WikiApplication | | | | | |
| URL | https://github.com/joshuafarrell95/WikiApplication | | | | | |
| Local Source Control Screen Shot |  | | | | | |
| Cloud Screen Shot |  | | | | | |

## Question 7 Project Plan

Using your recommended development methodology from question five, 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 question 6 create a Project within the repository and add your project plan.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Plan | | | |
| Project Name | WikiApplication | | |
| Date | 10 August 2022 | | |
| Developer Name | Joshua Farrell | | |
| Development Tasks | | | |
| Task Name | Task Type | Task Description | Input/Output Parameters |
| Create records array | Coding | Create two-dimensional array to store 12 records with 4 attributes | Output to ListView |
| Create list view | Coding | Create ListView to display Array contents | Input from ArrayWiki |
| Create record text boxes | Coding | Create Data Structure Name, Category, Structure and Definition textboxes and associated labels | Input from ArrayWiki  Output to ArrayWiki |
| Create search text box | Coding | Self explanatory | Output to search method |
| Create record buttons | Coding | Create Add, Edit, Delete and Search buttons | Add and Delete – output to ArrayWiki  Edit – input and output to ArrayWiki  Search – input from ArrayWiki |
| Create utilities to initialise and display array | Coding | Self explanatory | Initialisation – output to ArrayWiki  Display – Input from ArrayWiki to ListView |
| Create method to clear text boxes | Coding | Create a reusable method that will be called by a MouseDoubleClick event | Output to all text boxes |
| Create methods to load and save data | Coding | Data must be stored in a binary format and the end user must be able to load and save to an alternative file | Load – Input from Binary Reader  Save – Output to Binary Writer |
| Trap errors | Testing | Use try-catch blocks to trap errors |  |
| Give feedback using a status strip or dialog box | Testing | General feedback will use a status strip to reduce user load. Deleting a record will require a Dialog Box. | Output to status strip or dialog box (dependent on feedback severity) |

## 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? |
| Stewart Godwin | Lecturer |  | 17/08/2022 | Yes |
|  |  |  |  |  |
| Lecturer Feedback | | | | |
| Q1 – Change importance to order of tasks  Q7 – Add task type (coding, debug, testing, admin, etc.) | | | | |

## Question 9 Prototype Development

Create the wiki prototype to demonstrate how a collection of information can be stored using a Windows Forms Application. 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 Question 6/7 to manage your code during the development; ensure you record these commits/branches as a series of screen shots to be included in Question 9 (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 string array.

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, column),~~
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 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: | WikiApplication |
| URL | https://github.com/joshuafarrell95/WikiApplication |
| Desktop Screen Shots | Start |
| Working |
| Final |
| Cloud Screen Shots | Start |
| Working |
| Final |

## 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 is a linear data structure containing a number of elements which are identified by at least one index. Arrays can be used by an iteration construct such as a FOR loop by passing an index through the array. Arrays are addressed through zero-based indexing, that is, the first element of the array is stored in index 0, and ends at index n – 1, n being the length (or number of potential elements) of the array. |
| Two Dimension Array | Linear | A two-dimension array is a linear data structure which contains an array within each element, that is an array of an array. Each array contains a number of elements, which are identified by two separate indices (e.g., this definition is stored within a two-dimensional array with indices [1,3]). |
| List | List | Linear | A list is a linear data structure which are a defined number of elements. Lists uses an Array (implemented by the IList<T> generic interface), which are dynamically increased by using pointers, or terminators. |
| Linked list | Linear | A linked list is a linear data structure that has an element and a pointer that points directly to the next list, or a terminator, requiring an iteration construct to iterate through the list. Linked lists are read sequentially, that is, to read an element n, the linked list must be read from index 0 to n. |
| Self-Balance Tree | Tree | Non-Linear | A self-balance tree is a non-linear data structure that is node based. A tree can be used to maintain an ordered list such as a priority queue (not to be confused with Queue) which has an additional priority attribute associated within the self-balance tree. |
| Heap | Non-Linear | A heap is a non-linear data structure that uses an array to represent a tree-based data structure. The root element is stored in index 0 and the nodes connected to that element are stored in indexes 1 and 2 represented by the equations 2i + 1 and 2i + 2 (where i is the index) to find a child element. A new element can be added to a heap using insertion, deleted using extraction to remove an element, or edited using replacement. |
| Binary Search Tree | Non-Linear | A binary search tree is a non-linear data structure that uses a tree to represent nodes, which has at most two children for each node. The number of nodes in a binary tree is at least 2h + 1 and at most 2^(h+1)-1. |
| Graph | Graphs | Non-Linear | A graph is a non-linear data structure that uses a finite amount of nodes (or vertices) which are able to connect to other nodes using edges. Edges can be directed which requires an ordered pair of nodes, or undirected which can connect to any amount of nodes. Nodes and edges can have attributes to describe a value such as cost, capacity, etc. |
| Set | Abstract | Non-Linear | A set is a non-linear data structure that is unordered (or unsorted). Each element in a set must be unique as a set has no indexes. Sets can interact with each other to find the union, intersection or exception of the sets. |
| Queue | Linear | A queue is a linear data structure that is implemented using first in first out principles. An element can be enqueued into the back of a queue and dequeued from the front of the queue. Once an element is in the queue, if there is a previously enqueued element in the queue, that element must wait until it is in the front of the queue. Inserting (or enqueueing) and deleting (or dequeuing) an element has a time complexity of O(1), meaning that any changes to a queue always takes the same amount of time to execute. |
| Stack | Linear | A stack is a linear data structure that is implemented using last in first out principles. An element can be pushed or popped on top of a queue. To access an element that is not on the top of the stack, every element above that element must be popped off the stack. In the worst case, accessing an element from a stack would have a time complexity of O(n), meaning that all other elements must be accessed first. |
| Hash Table | Hash | Non-Linear | A hash table is a non-linear data structure that uses a dictionary (or array) to store hashed elements. A key is hashed using a hashing algorithm, then it is stored into a bucket (or slot). If two unique keys have the same hash, this is a hash collision and a collision resolution algorithm would need to be used to resolve the collision. |
| References | | | |
| *Add references as required* | | | |

## 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 is provided for clarification. Complete the following UML Activity Diagram template below.

|  |  |
| --- | --- |
| UML Activity Diagram | |
| Diagram | C# Code |
| Note: PDF version of diagram is also attached at the project root | private void BinarySearch(string searchString)  {  bool isItemFound = false;  int min = 0;  int max = row - 1;  while (min <= max)  {  int mid = ((min + max) / 2);  if (searchString.CompareTo(ArrayWiki[mid, 0]) == 0)  {  isItemFound = true;  HighlightRecord(mid);  break;  }  else if (searchString.CompareTo(ArrayWiki[mid, 0]) < 0)  {  max = mid - 1;  }  else  {  min = mid + 1;  }  }  if(!isItemFound)  {  MessageBox.Show("Record " + searchString + " not found.", "Search unsuccessful", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  else  {  statusStrip.Items.Add("Record " + searchString + " found and highlighted.");  }  textBoxSearch.Clear();  } |

## 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 Question 8.5) 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 Question 10 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 | | WikiApplication | | | | | | | | |
| **Method** | | Functional test | | | | | | | | |
| **Description** | | This will test whether the wiki application is able to | | | | | | | | |
| **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 SDLC 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 Question 10. 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