|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Release 7.2) | | |
| **Qualification National Code and Title** | ICT40120 Certificate IV in Information Technology (Release 3) | **State code** | BFF9 |
| **Qualification National Code and Title** | ICT50220 Diploma of information Technology (Release 1) | **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 Ten | | **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 | 8 class sessions (Weeks 2-10) |
| 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 Diagram  Question 4 – Development Methodologies  Question 5 – Methodology Analysis and Selection  Question 6 – Project Plan  Question 7 – Manage Version Control System  Question 8 – Prototype Development  Question 9 – Version Control Update  Question 10 – Data Structure Matrix  Question 11 – UML Activity Diagram  Question 12 – Debug Test Report  Question 13 – Post Development Analysis  Question 14 – 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.

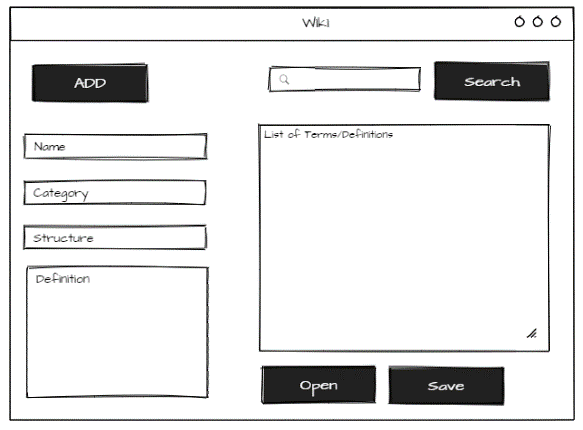
Before the project can move to the next stage CITE management would like a report on the full development process and related documentation. Review the proposed program criteria and answer the associated questions. Use the supplied template forms to present your answers. Finally develop a working prototype 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.

## Proposed Program Criteria

|  |  |
| --- | --- |
| * The user can select an item from the list box and the corresponding information will be displayed in the four text boxes (on the Left), * The user can search for an item which will be displayed in the four text boxes, * Search input box must clear if search unsuccessful, * The user can add a new item, * All user interactions must have full error trapping and feedback messaging, * The wiki prototype will load and save data when the buttons are clicked, | * All wiki data is stored/retrieved using a binary file format, * A double mouse click in the search text box will clear the search input box, * The prototype must use a two-dimensional array of type string, * The Data Structure information must be defined using the following attributes: Data Structure Name, Category, Structure and Definition, (refer Data Structure Matrix in Question 10). |

## Proposed Interface Design



## Question 1

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 Prototype Application | | |
| Date | | | 1/02/2022 | | |
| Developer Name | | | Kyle Watson | | |
| User Requirements | | | | | |
| Req. # | | Description | | Importance | Notes |
| 1 | | The user can select an item from the list box and the corresponding information will be displayed in the four text boxes on the Left | | 3 | Important for user feedback |
| 2 | | The user can search for an item which will be displayed in the four text boxes | | 3 | Necessary for ease of functionality |
| 3 | | The user can add, delete and edit items | | 1 | Basic requirement for the application to operate. |
| 4 | | The wiki prototype will load and save data when the buttons are clicked | | 2 | Essential so the user doesn’t lose their work |
| 5 | | A double mouse click in the search text box will clear the search input box | | 5 | A simple shortcut for |
| 6 | | All user interactions must have full error trapping and feedback messaging | | 4 | Can be completed during debug phase of development once essential functionality is complete |
| User Interaction and Specifications | | | | | |
| How will the application behave and what GUI specifications are required. | | | | | |
|  | The user can select an item from the list box and the corresponding information will be displayed in the four text boxes on the Left | | | | |
|  | The user can search for an item which will be displayed in the four text boxes | | | | |
|  | Search input box must clear if search unsuccessful | | | | |
|  | The user can add, edit and delete items | | | | |
|  | All user interactions must have full error trapping and feedback messaging | | | | |
|  | The wiki prototype will load and save data when the buttons are clicked | | | | |
|  | All wiki data is stored/retrieved using a binary file format | | | | |
|  | A double mouse click in the search text box will clear the search input box | | | | |
|  | The prototype must use a two-dimensional array of type string | | | | |
|  | The Data Structure information must be defined using the following attributes: Data Structure Name, Category, Structure and Definition | | | | |
| *Add more lines as required* | | | | | |

# Question 2

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: | GitLab |
| URL: | | http://github.com | URL: | https://about.gitlab.com/ |
| Major Features | | | | |
| List all the major features associated with each version control system | | | | |
| 1. | Any of these: | | https://www.softwaretestinghelp.com/version-control-software/ | |
| 2. | Supports non-linear development | | Issue management, bug tracking and boards | |
| 3. | Distributed repository model | | Code review functionality and review apps tool | |
| 4. | Compatible with existing systems and protocols | | Continuous integration and deployment tool | |
| 5. | Capable of handling small to large scale projects | | Code climate for full visibility | |
| 6. | Cryptographic authentication of history | | Service desk utilizing a ticketing system | |
| 7. | Pluggable merge strategies | | Repository mirroring and high availability | |
| 8. | Toolkit based design | | Integration with IDEs like Visual Studio | |
| 9. | Periodic explicit object packing | | On-premise or cloud-based installations | |
| 10. | Garbage accumulates until collected | | Hosting static websites | |
| *Add more lines as required* | | |  | |
| Recommendation: Which VCS would you choose and why? | | | | |
| *Add your detailed answer here…* | | | | |

## Question 3

Create a UML 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 Diagram template to answer question three.

|  |  |
| --- | --- |
| UML Diagram | |
| Attributes | |
| 1 | static string[ , ] |
| 2 | static int rowSize |
| 3 | static int colSize |
| 4 | string filename |
| Data Structure | |
| Static string [ , ] myArray = new String [ rowSize, colSize ]; | |
| Diagram | |
|  | |

## Question 4

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 Model (Traditional) |
| Description: a software development process that follow a linear sequential flow meaning that new phases in the development process only begin if the previous phase is complete. |
| Diagram:  Diagram  Description automatically generated |
| Advantages:   * Clear concise project requirements * Suitable for smaller projects where a clear understanding of the needs is required * Easy to understand and use * Simple to arrange the tasks efficiently * Release level changes are allowed * Procedure and results are well and easily documented |
| Disadvantages:   * No parallel deliverable meaning it is difficult for two teams to work together * Does not provide the requirement changes and requirement review * Changes falling between phases are not allowed because of the dependence of phases on one another * Backward tracking is not possible. * Time consuming process as new phases cannot begin until prior phases are complete |
| 2nd Methodology Name: Agile Model (Non – Traditional) |
| Description: a methodology that operates by breaking up a project into smaller incremental builds provided in iterations that may be worked on simultaneously by cross functional teams. |
| Diagram: |
| Advantages:   * Takes a realistic approach to software development in that it is flexible and adaptive to the changing product requirements * Tested frequently through release iterations which minimize the risk of major failures * Promotes teamwork and cross training * Functionality can be developed and demonstrated rapidly * Delivers early partial working solutions * Little to no planning is required * Allows for concurrent development * Easy to manage * Gives flexibility to developers |
| Disadvantages:   * Not suitable for handling complex dependencies * More risk of sustainability, maintainability and extensibility * Requires an overall plan and competent leader capable of agile based project management * Delivery management dictates the scope of functionality to be delivered and adjustments necessary to meet deadlines * Heavy customer interaction is necessary to establish development requirement which without the team may be driven in the wrong direction * High individual dependency within teams is necessary due to the minimal amount of documentation generated * New team members may be challenged |
| 3rd Methodology Name: V – Model Application (Traditional) |
| Description: like the waterfall model it follows sequential progression. Each phase in the development cycle has a directly associated testing phase with the next phase only beginning after completion of previous phases. |
| Diagram: |
| Advantages:   * Is a highly disciplined model with phases being completed one at a time * Works well for smaller projects where requirements are well understood * Simple and easy to understand and use * Easy to manage due to the rigidity of the model with each phase having specific deliverables and a review process. |
| Disadvantages:   * High risk and uncertainty * Not a good model for complex and OOP projects * Pool model for long and ongoing projects * Not suitable for the projects where requirements are at a moderate to high risk of changing * Once an application is in the testing stage, it is difficult to go back and change a functionality * No working software is produced late during the life cycle |
| 4th Methodology Name: Big Bang Model |
| Description: a model in which no specific process is followed. Development starts with the required money and efforts as the input and the software developed being the output which may or may not be as per customer requirements. |
| Diagram: |
| Advantages:   * Simple and easy to follow * Little/No planning required * Easy to manage * Very little resources required * Flexibility is in the hands of developers * A good learning aid for newcomers or students |
| Disadvantages:   * High risk and uncertainty * Not a good model for complex and OOP projects * Pool model for long and ongoing projects * Can be expensive if requirements are misunderstood |
| References |
| *Add references as required*  Big bang model:  <https://www.tutorialspoint.com/sdlc/sdlc_bigbang_model.htm#:~:text=The%20Big%20Bang%20model%20is,be%20as%20per%20customer%20requirement>.  Waterfall, Agile and Spiral models:  Software Development Life Cycle (SDLC) PDF found on Cluster – C Sharp Two – Topic Six, Software Development Life Cycle |

## Question 5

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 |
| * Optimal for smaller projects with well defined requirements |
| * Must facilitate development that follows sequential phases |
| * Easy to manage and enable production of specific deliverables at each phase |
| * Simple to follow |
| * Little resources required |
| *Add more lines as required* |
| Methodology Selection |
| Methodology Name \_\_\_**Waterfall**\_\_\_\_ |
| Justification (reasons for selection) |
| * Clear and concise project requirements |
| * Unlikeliness of having to return to previous phases |
| * Procedure and results are easily documented |
| * The methodology is suited to smaller projects |
| * Client is unlikely to be involved in the project meaning the end goals are unlikely to change |

## Question 6

Using your recommended 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.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Plan | | | |
| Project Name | Wiki Prototype Application | | |
| Date | 01/03/2022 | | |
| Developer Name | Kyle Watson | | |
| Development Tasks | | | |
| Task Name | Task Type | Task Description | Input/Output Parameters |
| 1. Establish program criteria | Planning | * Determine standards by which the program will be assessed | **Input:** CITE representative program criteria  **Output:** adaptation of clients proposed criteria to realistic developmental goals |
| 2. Determine user requirements and specifications | Planning | * Establish how the application will behave * Plan necessary GUI components |  |
| 3. Program design phase | Design | * Determine program’s primary data structure * Determine necessary class attributes via UML diagram |  |
| 4. User Interface Development | Development | * Plan program layout and port to Visual Studio forms |  |
| 5. Program development following established user requirements and program criteria | Development | * Implement program code for basic functionality to meet program requirements and criteria | **Input 1:** wiki data entered via text boxes  **Output 1:** data displayed in the list box  **Input2:** existing wiki data in the application  **Output2:** binary file with saved data  **Input3:** pre-saved binary file  **Output3:** information retrieved from pre-existing binary and loaded into the application |
| 6. Initial error trapping | Development | * Develop necessary error trapping to support basic program development |  |
| 7. Implementation of quality-of-life features and user feedback interfacing | Development | * Develop necessary message boxes, status strips, labels, tooltips and quality of life features |  |
| 8. Debugging and testing with associated documentation. | Testing/Documentation | * Thorough testing and documentation |  |
| 9. Program demonstration and submission. | Presentation | * Presentation to client for final submission or to inform further program refinement |  |
| *Add more lines as required* |  |  |  |

## Question 7

CITE has authorised the usage of GitHub as the recommended Version Control System (VCS) because it is fully integrated into Visual Studio, create your own GitHub account (you can use your existing account) then download and install the desktop interface onto your local PC. Create a suitable folder structure for both the local and cloud systems to accommodate the development of the wiki prototype application. Complete the following Manage Version Control System template to answer this question.

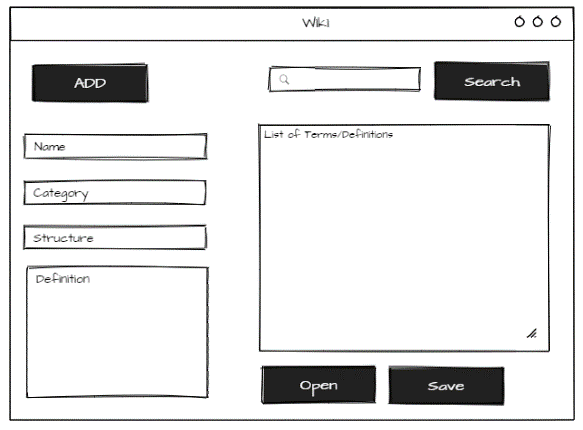
Consult with your lecturer if you wish to use an alternative source control service. Any alternative source control must support a local desktop installation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Manage Version Control System | | | | | | |
| VCS Name | | Github | | | | |
| Version Number | | 1.0 | Date | 1/3/2022 | | |
| Supported Platforms | | Windows | | | | |
| Installation Information | | | | | Yes | No |
| Has the cloud VCS account been created and named correctly? | | | | |  |  |
| Has the cloud VCS folder structure been created? | | | | |  |  |
| Has the desktop VCS software installed correctly? | | | | |  |  |
| Has the desktop VCS folder structure been created? | | | | |  |  |
| Are the two VCS resources linked? | | | | |  |  |
| Comments/Issues/Problems | | | | | | |
|  | | | | | | |
| Account Details Checklist | | | | | | |
| Repository Name: | Wiki-Prototype-Application | | | | | |
| URL | https://github.com/Kwatson-1/Wiki-Prototype-Application | | | | | |
| Desktop Screen Shot |  | | | | | |
| Cloud Screen Shot |  | | | | | |

## Question 8

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 7 to manage your code during the development; ensure you record these commits 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/)

The following user interface is provided as a starting point for your prototype development.



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

The following programming criteria and features are required;

### 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 a CLEAR method to clear the four text boxes so a new definition can be added,
4. 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 (by reference) the array element to be swapped (do not use any built-in array methods),
5. 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),
6. Create a display method that will show the following information in a List box: Name and Category,
7. Create a method so the user can select a definition (Name) from the Listbox and all the information is displayed in the appropriate Textboxes,
8. 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,**
9. Create a LOAD button that will read the information from a binary file called ***definitions.dat*** into the 2D array,
10. All code is required to be adequately commented. Map the programming criteria and features to your code/methods by adding comments above the method signatures. Ensure your code is compliant with the CITEMS coding standards (refer http://www.citems.com.au/).

**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 9

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: | Wiki Prototype Application |
| URL | https://github.com/Kwatson-1/Wiki-Prototype-Application/projects/1 |
| Desktop Screen Shots |  |
| Cloud Screen Shots |  |

## Question 10

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 an object of fixed length that stores variables of the same type indexed from 0 to n-1 where n is the declared length. It allows multiple values to be stored within a single variable with a single declaration. |
| Two Dimension Array | Linear | A two-dimensional array shares the same properties as a one-dimensional array however differs in its declaration and is used to store array within an array thus making it an array of arrays. |
| List | List | Linear | A list is used to store a collection of similarly typed objects that may be accessed by index. The length of a list is flexible and can be resized as necessary for the number of elements present. |
| Linked list | Linear | A linked list is a data structure which stores elements in the non-contiguous location meaning that acquired memory space is taken from different locations thus reducing wastage of memory. |
| Self-Balance Tree | Tree | Non-Linear | A binary tree is a non-linear data structure in which a node can have 0, 1 or 2 nodes. Each node is represented either as a parent node or a child node of which a parent node can have 2. |
| Heap | Non-Linear | In programming the heap is an area of pre-reserved computer main storage (memory) that a program process can use to store data in some variable amount that won’t be known until the program is running. |
| Binary Search Tree | Non-Linear | A binary search tree follows some order to arrange the elements in which the value of the left node must be smaller than the parent node and the value of the right node must be greater than the parent node. |
| Graph | Graphs | Non-Linear | A graph consists of a finite set of vertices (nodes) and set of edges which connect a pair of nodes. Graphs are used to represent networks which may include city paths, telephone networks, circuit networks or social networks. |
| Set | Abstract | Non-Linear | A set is an unordered collection of distinct elements generally of the same type however may be made up of elements of different types. |
| Queue | Linear | A queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. |
| Stack | Linear | A stack is an array or list structure of function calls and parameters used in computer programming and CPU architecture. Elements in a stack are added or removed from the top of the stack in a “last in”, “first out” procedure. |
| Hash Table | Hash | Non-Linear | A hash table implements an associative array abstract data type (structure that can map keys to values). A hash table uses a hash function to compute an index into an array of slots/buckets from which the desired value can be found. |
| References | | | |
| *Add references as required* | | | |

## Question 11

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.

|  |
| --- |
| C# Code |
| private void buttonSearch\_Click(object sender, EventArgs e)  {  int startIndex = -1;  int finalIndex = counter; // set size of data in array  bool flag = false;  int foundIndex = -1;  if (listViewOne.SelectedItems.Count != 0)  {  listViewOne.SelectedItems[0].Selected = false;  }  while (!flag && !((finalIndex - startIndex) <= 1)) //breakpoint  {  int newIndex = (finalIndex + startIndex) / 2;  // The string.Compare(a,b) method compares 2 strings a and b and returns an integer value  // -1 if a is less than b, 0 if they are equal, 1 if b is less than a  if (string.Compare(wikiArray[newIndex, 0], textBoxOne.Text) == 0) //breakpoint  {  foundIndex = newIndex;  flag = true;  break;  }  else  { //breakpoint  if (string.Compare(wikiArray[newIndex, 0], textBoxOne.Text) == 1)  finalIndex = newIndex;  else  startIndex = newIndex;  }  }  if (flag) //breakpoint  {  textBoxName.Text = wikiArray[foundIndex, 0];  textBoxCategory.Text = wikiArray[foundIndex, 1];  textBoxStructure.Text = wikiArray[foundIndex, 2];  textBoxDefinition.Text = wikiArray[foundIndex, 3];  listViewOne.Items[foundIndex].Selected = true;  listViewOne.HideSelection = false;  toolStripStatusLabel.Text = "Found.";  }  else  toolStripStatusLabel.Text = "Not found."; //breakpoint  } |
| UML Activity diagram |
|  |

## Question 12

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 | | Wiki Prototype Application | | | | | | | | |
| **Method** | | Source code debugger | | | | | | | | |
| **Description** | | Utilisation of breakpoints to check the values of variables and ensure proper method function. | | | | | | | | |
| **Level of Testing** | | Unit testing | | | | | | | | |
| **Developer** | | Kyle Watson | | **Tester** | Kyle Watson | | **Date** | | 5/04/2022 | |
| **Test Case No** | **Test Case Name** | | **Test steps** | | | **Test Data** | | **Expected result** | | **Pass / Fail** |
| 1 |  | |  | | |  | |  | |  |
| 2 |  | |  | | |  | |  | |  |
| 3 |  | |  | | |  | |  | |  |
| 4 |  | |  | | |  | |  | |  |
| 5 |  | |  | | |  | |  | |  |
| 6 |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |

## Question 13

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?  Waterfall |
| How effective was your project plan in developing the final prototype?  The waterfall methodology while preceded quickly initially proved costly for time when bugs were encountered during the development phase as it meant that the project could not continue until they were solved. |
| What alternative data structures could be used?  A good alternative data structure would have been a List type as it would allow for simpler deletion and insertion. While the array structure requires less memory, because the application does not require much memory to begin with this is not a concern. Finally the list structure type has inbuilt methods for sorting and searching which would have cut down on the requirements for development time. |
| What constructive advice could you provide for the development of a similar project?  Agile would have been a good alternative as it would allow |

## Question 14

### 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 User Requirements and User Interaction Specifications. |  |  |  |
| Q2 | UML Class Diagram: List four attributes and a simple UML class diagram showing Array |  |  |  |
| Q3 | Development Methodologies: List four different SDLC methodologies, No variations of a single methodology. |  |  |  |
| Q4 | Methodology Analysis: List several (three or more) selection criteria, Lists several (three or more) recommendations for SDLC |  |  |  |
| Q5 | Project Plan: Reflect selection from Question 4 and has all the SDLC tasks outlined |  |  |  |
| Q6 | Manage Version Control System: All fields are completed and there are screen shots of GitHub desktop and cloud |  |  |  |
| Q7 | Prototype Development: all program criteria and feature have been coded, software standard have been implemented in layout and comments |  |  |  |
| Q8 | Data Structure Matrix: All 12 data structure have a suitable definition. |  |  |  |
| Q9 | UML Activity Diagram: The code has breakpoints in the Decision and Iterative constructs, the activity diagram reflects the C# code. |  |  |  |
| Q10 | Testing and Debugging: All four bullet points have been tested and the results recorded. |  |  |  |
| Q11 | 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