|  |  |  |  |
| --- | --- | --- | --- |
| **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** | Rishon Jacobs | | | | | |
| **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.

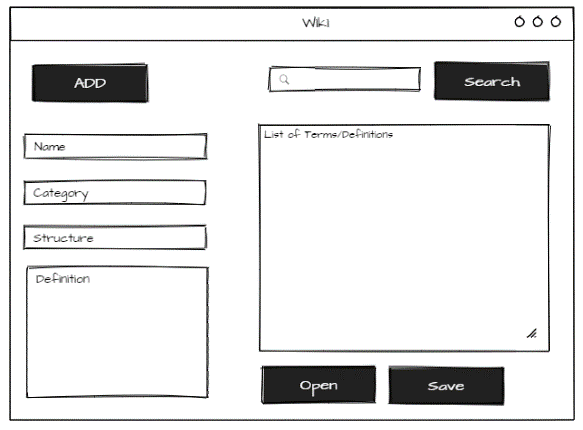
Before the project can move to the next stage CITE management would like a report on the full development process and related documentation. Review the proposed program criteria and answer the associated questions. Use the supplied template forms to present your answers. Finally develop a working prototype 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 | | |  | | |
| Date | | | 2/02/2022 | | |
| Developer Name | | | Rishon Jacobs | | |
| User Requirements | | | | | |
| Req. # | | Description | | Importance | Notes |
| 1 | | Load Wiki Data Button | | 1 |  |
| 2 | | Save Wiki Data Button | | 1 |  |
| 3 | | Add Wiki Data Button | | 1 |  |
| 4 | | Ability to search for specific data in the List box, and display it in the four textboxes | | 1 |  |
| 5 | | Wiki Data stored/retrieved using binary file format | | 2 |  |
| 6 | | Data Structure Information must be defined using the attributes: Data Structure Name, Category, Structure and Definition. | | 1 |  |
| 7 | | Double mouse click clears search input box | | 2 |  |
| 8 | | Must use a two-dimensional array of type string | | 1 |  |
| 9 | | Selecting an item from the list will display its information in the four text boxes. | | 1 |  |
| 10 | | Include error messages and error trapping | | 1 |  |
| User Interaction and Specifications | | | | | |
| How will the application behave and what GUI specifications are required. | | | | | |
|  | User able to select item from list box and 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: | JEDI VCS |
| URL: | | http://github.com | URL: | http://jedivcs.sourceforge.net/ |
| Major Features | | | | |
| List all the major features associated with each version control system | | | | |
| 1. | **Security** – Secures your code as it is written and identifies vulnerabilities before production. Layers of security to protect repositories. | | **Versatile Use -** Connects to other software to build an integrated development environment. Can be used as standalone software.  Delphi IDE Integration | |
| 2. | **Easy access** – Github can be accessed from any connected device. | | **Project Management Tools** – Built-in project management tools. Tracks and prioritises application issues for prompt troubleshooting. Streamlines workflow and makes sure all tasks are related to your projects.  Time Tracker  To-Do Manager  Project Tree View of Repository | |
| 3. | **Project Management** – Visual tracking, creation of milestones / goals and labels for teams and track issues. | | **Change Tracker –** Finds the lines which may cause problems, assisting in locating bugs after moving versions. Has a difftool that can be customised for easy pinpointing of modifications.  Bug Tracker  Project & Line History  Customisable Diff Checker  Command Line/ Console Application | |
| 4. | **Team Administration** – Option to assign roles, with specific permissions, to team members. Team sync, invitations, Audit log API, verification etc. | | **Flexible Rights Settings** – Supports project-based rights management, can configure access controls every time. Option to rotate the leadership and key roles, depending on the project and skills required. | |
| 5. | **Collaborative Coding** – Pull requests, code owner’s permissions, team discussions, multiline commenting, public repositories and easy access for team members. | |  | |
|  |  | |  | |
|  |  | |  | |
|  |  | |  | |
| *Add more lines as required* | | |  | |
| Recommendation: Which VCS would you choose and why? | | | | |
| *Add your detailed answer here…*  I would choose GitHub, because it provides almost the same features as JEDI VCS but with more options when it comes to access. GitHub can be accessed on nearly any device and provides options from Small Businesses to Large Enterprises, whereas JEDI VCS can only be accessed from Windows and provides a maximum company size of a Medium Business. GitHub also has more available support for users. | | | | |

## 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 int rowSize |
| 2 | Static int colSize |
| 3 | Static string[,] |
| 4 | String fileName |
| Data Structure | |
| Static string[ , ] myWikiArray = 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 \_\_Iterative Model\_\_\_\_\_\_\_\_\_ |
| Description: Iterative model begins by specifying and implementing just part of the software, which is reviewed in order to identify further requirements. A new version of the software is produced at the end of the model. |
| Diagram: |
| Advantages:   * Risks identified and resolved during iteration; each iteration Is an easily managed milestone * Easier to manage risk – High-risk part is done first * Initial Operating time is less * Supports changing requirements * Results obtained early and periodically * Testing and debugging during smaller iteration is easy * Risk analysis is better * Issues & risks identified from each increment can be utilized to the next increment |
| Disadvantages:   * May require more resources * More management attention is required * Management complexity is more * Not suitable for smaller projects * Not all requirements may be gathered at the beginning of the entire life cycle which can cause design issues * Cost of change is cheaper but it is not very suitable for changing requirements * Highly skilled resources are required for risk analyis |
| 2nd Methodology Name \_\_\_V-model\_\_\_\_\_\_\_\_ |
| Description: is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase. |
| Diagram: |
| Advantages:   * Highly disciplined model and phases are completed one at a time * Works well for smaller projects where requirements are understood * Easy to manage due to the rigidity of the model * Simple and easy to understand and use. |
| Disadvantages:   * High risk and uncertainty * Not a good model for complex and object-oriented projects * Poor for long and ongoing projects * Not suitable for projects where requirements change * Difficult to change functionality after entering the testing stage * No working software is produced until late during life cycle |
| 3rd Methodology Name \_Agile Model\_\_\_\_\_\_\_\_\_\_ |
| Description: Agile model combines the iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of a working software product. |
| Diagram: |
| Advantages:   * Realistic approach to software development * Resource requirements are minimum * Minimal rules, documentation easily employed * Little or no planning required * Easy to manage * Provides flexibility to developers * Good for environments that change steadily * Suitable for fixed or changing requirements * Delivers early partial working solutions * Enables concurrent development and delivery within an overall planned context |
| Disadvantages:   * High individual dependency * Transfer of software to new teams may be challenging due to a lack of documentation * Strict delivery management d |
| 4th Methodology Name \_\_\_Bang Bang Model\_\_\_\_\_\_\_\_ |
| Description: Bang Model is SDLC model where there is no formal development followed and very little planning is required. Even the customer is not sure about what exactly he wants and the requirements are implemented on the fly without much analysis. |
| Diagram: Big Bang Model: What Is It And How Do You Use It? | TIGO Software Solutions |
| Advantages:   * Very simple model * Little to no planning required * Easy to manage * Few resources required * Flexibility to developers * Good learning aid for newcomers or students |
| Disadvantages:   * Very high risk and uncertainty * Not a good model for complex and object-oriented projects * Poor model for long and ongoing projects * Can turn out to be expensive if requirements misunderstood |
| References |
| *Add references as required* |

## 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 |
| * Good for a smaller project where requirements are understood |
| * Simple and easy to use |
| * Few resources required |
| * Sequential phases for development |
| * Easy to manage |
| *Add more lines as required* |
| Methodology Selection |
| Methodology Name \_\_\_**Agile**\_**Kanban**\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Justification (reasons for selection) |
| * Every simple and easy to understand approach |
| * Adaptability to ongoing changes |
| * Reduces costs and wastage |
| * Management and supervision is very easy when using a Kanban Board |
| * Promotes continuous and sustainable improvements |
| *Add more lines as required* |

## 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 Data Structure | | |
| Date | 09/03/2022 | | |
| Developer Name | Rishon Jacobs | | |
| Development Tasks | | | |
| Task Name | Task Type | Task Description | Input/Output Parameters |
| 1. Determine user requirements & program criteria | Planning | Determine the layout of the program and what is being accessed. Understand the purpose of the requirements. | Input: CITE program criteria  Output: match with client criteria to create development goals |
| 1. Program design | Planning | - design UML diagrams  - determine how the application will behave  - plan components for user interface |  |
| 1. User Interface Development | Development | - Draw out button locations and usage  - Link github & visual studio |  |
| 1. Coding and establishing user requirements & program criteria | Development | Write code according to program requirements and meet criteria. |  |
| 1. Implementation of tooltips and statusStrip information | Development | - Button usage information  - Status Strip  - Labels |  |
| 1. Clean up and error trapping | Development | - Add obvious error messages and trapping | Input: Information entered into text boxes  Output: Displayed in list box |
| 1. Debugging & Testing | Testing / documentation | - Test the prototype and record the information in the related table  - document findings  - document changes | Input: null or invalid data  Output: Error message |
| 1. Make changes and prepare for demonstration. | Presentation | Presentation to client for final submission or receive feedback for final changes. |  |
| 1. Finalise and submit. | Final Changes | If necessary changes are needed, make them and submit work. |  |
| *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.

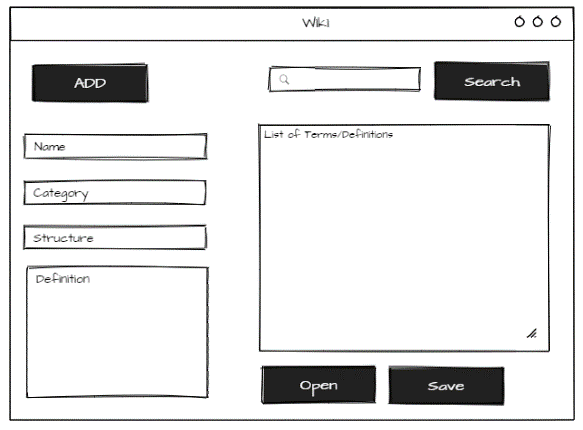
Consult with your lecturer if you wish to use an alternative source control service. Any alternative source control must support a local desktop installation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Manage Version Control System | | | | | | |
| VCS Name | | Github | | | | |
| Version Number | | 2.9.12 | Date | 10 / 3 / 22 | | |
| 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-Data-Structure | | | | | |
| URL | <https://github.com/rishonj/Wiki-data-structure> | | | | | |
| 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 Data Structure |
| URL | <https://github.com/rishonj/Wiki-data-structure> |
| 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 a data structure consisting of a collection of elements (values or variables), each identified by at least one array index or key. |
| Two Dimension Array | Linear | The two-dimensional array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns. |
| List | List | Linear | A list is an abstract data type that represents a finite number of ordered values, where the same value may occur more than once. |
| Linked list | Linear | A linked list is a data structure that stores elements in a non-contiguous location meaning that acquired memory space is taken from different locations thus reducing wastage of memory. |
| Self-Balance Tree | Tree | Non-Linear | Self-Balancing Binary Search Trees are height-balanced binary search trees that automatically keep the height as small as possible when insertion and deletion operations are performed on the tree. |
| Heap | Non-Linear | A 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 a set of edges that 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.

|  |  |
| --- | --- |
| UML Activity Diagram | |
| Diagram | C# Code |
|  | int startIndex = -1;  int finalIndex = counter;  bool flag = false;  int foundIndex = -1;  if (listViewCategory.SelectedItems.Count != 0)  {  listViewCategory.SelectedItems[0].Selected = false;  }  if (!string.IsNullOrEmpty(textBoxSearch.Text))  {  while (!flag && !((finalIndex - startIndex) <= 1)) //breakpoint  {  int newIndex = (finalIndex + startIndex) / 2;    if (string.Compare(wikiArray[newIndex, 0], textBoxSearch.Text) == 0) //breakpoint  {  foundIndex = newIndex;  flag = true;  break;  }  else  {  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];  listViewCategory.Items[foundIndex].Selected = true;  listViewCategory.HideSelection = false;  MessageBox.Show("Item found.");  toolStripStatusLabel.Text = "Item found at index " + foundIndex + ".";  }  else  { toolStripStatusLabel.Text = "Not found."; //breakpoint |

## 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 Data Structure | | | | | | | | |
| **Method** | |  | | | | | | | | |
| **Description** | |  | | | | | | | | |
| **Level of Testing** | |  | | | | | | | | |
| **Developer** | |  | | **Tester** |  | | **Date** | |  | |
| **Test Case No** | **Test Case Name** | | **Test steps** | | | **Test Data** | | **Expected result** | | **Pass / Fail** |
| 1 |  | |  | | |  | |  | |  |
| 2 |  | |  | | |  | |  | |  |
| 3 |  | |  | | |  | |  | |  |
| 4 |  | |  | | |  | |  | |  |
| 5 |  | |  | | |  | |  | |  |
| 6 |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |

## 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?  Agile Kanban |
| How effective was your project plan in developing the final prototype?  The agile Kanban methodology proved to be a great option as it allowed me to overview my work and tasks easily, and comes back to previous steps after changing code. The adaptability of this model proved to be a great assistance in completing the final prototype with ease. |
| What alternative data structures could be used?  List type would be a decent data structure alternative as it has inbuilt methods for searching and sorting. It would also allow for simpler deletion and insertion of data. |
| What constructive advice could you provide for the development of a similar project?  Agile Kanban or scrum are viable options for this project, as it is small and the requirements are well-defined. The Kanban board will assist in creating standards for elements prior to development as you will be able to see each task and what stage it is at. This model provides flexibility and reduces time spent trying to meet program criteria. |

## 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 screenshots 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