|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Version 8.1) | | |
| **Qualification National Code and Title** | ICT50220 Diploma of information Technology (Release 2) | **State code** | BGJ4 |
| **Assessment Title** | Assessment Task Three – Part B (Individual Project) | | |
| **Unit National Code & Title** | ICTPRG535 Build advanced user interfaces | | |
| ICTPRG547 Apply advanced programming skills in another language | | |
| ICTICT517 Match ICT needs with the strategic direction of the organisation | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date Due** | Week Eighteen | | **Date Received** | | 1/10/24 | |
| **Student Name** | JACK DU BOULAY | | | | | |
| **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 Nineteen |

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Project |
| Duration of the assessment | 4 class sessions (Weeks 15-18) |
| 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:  ICTPRG535 Build advanced user interfaces   * Plan and design a UI solution according to organisational requirements, * Apply interactions designs and implement validation requirements against the design plan, * Create and display the UI with graphics according to UI requirements.   ICTPRG547 Apply advanced programming skills in another language   * Code advanced data structures using hashing, sorting and searching algorithms, * Apply third party libraries and communication technologies for data exchange, * Test and evaluate the code to resolve logical and syntactical designs flaws, * Create and document the application according to technical specifications.   ICTICT517 Match ICT needs with the strategic direction of the organisation   * Interpret, analysis and report the strategic organisational plan * Propose and document changes for the implementation of a ICT system * Provide action plan and schedule   The student must demonstrate the ability to complete the tasks outlined in this assessment and is expected to use systematic analytical processes and effect time management to meet the goals/deadlines outlined in the DAP. |

|  |  |
| --- | --- |
| Allowable Materials | Blackboard Topics, SDLC, Weekly readings (PDF), Example programs and Independent Outside of Class Activities |
| Required Resources | Web links and example code can be downloaded from the Blackboard portal.  PC with Notepad++, Visual Studio, GitHub, MSOffice.  Internet Access to 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 web coding 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.  Part A  Question 1 – Project Specification  Question 2 – Strategic Objectives Evaluation  Question 3 – Design Approval  **Part B**  **Question 4 – General Interface Programming Criteria (Dictionary)**  **Question 5 – Admin Interface Programming Criteria**  **Question 6 – General Interface Programming Criteria (SortedDictionary)**  **Question 7 – Admin Interface Programming Criteria**  **Question 8 – Testing and Code Optimisation**  **Question 9 – Implementation Plan**  **Question 10 – Demonstration, Feedback and Signoff** |

# Scenario

You are employed as the Senior Programmer with CITE Managed Services, and you have been assigned the Master File Project for an organisation called Malin Space Science Systems (MSSS). This project will require the planning and development of a multi-window application that will manage the staff details for the MSSS organisation. Ensure your development follows an Agile methodology that is recorded and maintained using your GitHub account. The details and criteria are provided in the following paragraphs.

You should consult with the CITE representative (your Lecturer) if you are unsure about any of the problems or questions in this assessment. Your primary research should focus on the resources on the Blackboard LMS and CITE web site, additional information can be collected from the Internet, ensure all sources are referenced in your submission. You must demonstrate your working applications before uploading to Blackboard, your Lecturer (Assessor) will sign off to ensure all the criteria are satisfied.

## Organisational Background

The staff at Malin are required to use their unique ID to gain access to all the facilities within the Malin organisation. All staff are issued a mobile phone and laptop as part of their employment conditions and induction training; this ensures they can work from any location utilising hot-desks and 5GHz Wi-Fi. The management at Malin’s ICT Security use the mobile phone numbers as the unique staff ID for login to security doors, science buildings, communication and server systems. Therefore, when a staff member arrives at work, they will use their staff mobile number to gain entry ito any building. Once they connect their laptop at a hot-desk they will logon to the computer network using the same mobile phone number.

The list of staff names and mobile numbers is stored on a secure cloud server in a single table data structure, which is separate to the encrypted password server that manages the passwords and monitoring systems. Since the original company was founded in London all mobile phones are issued using UK digits, so mobile numbers are nine digits starting with 07xxx, however the leading zero has been removed. Your application is to be a temporary solution whilst the development team are working on a global system that incorporates a single sign-on and biometric authentication. However, your solution is expected to be included in the global system.

## Application Requirements

Malin Space Science Systems requires a Windows Application which loads data from a .csv file into a read-only list box display. The user can then filter the data into a second list box for further examination. When a record is selected from the second list box the details are displayed in several related text boxes. The filter must be able to find and display partial results for both the ID number and the staff name. The Create, Update and Delete features must be performed in a second GUI. The Create option must prevent duplicate ID numbers, while the Update and Delete option must have a confirmation message (Status Strip). If a record is selected in the first general GUI and requires amendment the user can open the second admin GUI which will receive the data from the first general GUI. Refer to the following basic flowchart for basic UI actions and information flow. The full programming criteria for both applications are listed in Question 4, 5 and 6. The development can be completed using Windows Presentation Foundation or Windows Forms but must be consistent for both applications and GUIs. Any alternatives must be discussed and approved with the client (your Lecturer) prior to coding and testing.

This assessment is a demonstration and comparison of two keyboard driven applications; therefore, all actions **must** utilise the Alt, Ctrl and CharKey combinations. The testing and optimization of the code is to demonstrate the most efficient file IO and search, then compare the two different data structures.

## UI Flowchart

**Start**

**End**

**Load data from Excel**

**Open General GUI & Display data in List box 1**

**Open Admin GUI**

**Filter Data**

**Display in List box 2**

**Display data from Textbox**

**Select and Display one data record**

**Enter ID for Display**

**Perform Edit Operations**

**Save Data and Close Admin GUI**

**No Data Found**

Application Requirements

In the following questions you will develop/code/debug two separate solutions; these cannot be located in the same Project but must have a separate and distinct NameSpace. The first Windows Application will use a **Dictionary data structure** while the second Windows Application will use a **Sorted Dictionary**. Both applications must have the same code and method structures for all the criteria. You will be testing each application to determine the “best” option for the client.

## Question 4 General Interface Programming Criteria (Dictionary)

Your first programming task is to create the General GUI which will load the data and display records. The second task is to create the Admin GUI to perform Create, Update and Delete operations. The csv file can be obtained from your lecturer once your Design has been approved.

### Create a Dictionary data structure with a TKey of type integer and a TValue of type string, name the new data structure “MasterFile”.

### Create a method that will read the data from the .csv file into the Dictionary data structure when the GUI loads.

### Create a method to display the Dictionary data into a non-selectable display only list box (ie read only).

### Create a method to filter the Staff Name data from the Dictionary into a second filtered and selectable list box. This method must use a text box input and update as each character is entered. The list box must reflect the filtered data in real time.

### Create a method to filter the Staff ID data from the Dictionary into the second filtered and selectable list box. This method must use a text box input and update as each number is entered. The list box must reflect the filtered data in real time.

### Create a method for the Staff Name text box which will clear the contents and place the focus into the Staff Name text box. Utilise a keyboard shortcut.

### Create a method for the Staff ID text box which will clear the contents and place the focus into the text box. Utilise a keyboard shortcut.

### Create a method for the filtered and selectable list box which will populate the two text boxes when a staff record is selected. Utilise the Tab and keyboard keys.

### Create a method that will open the Admin GUI when the Alt + A keys are pressed. Ensure the General GUI sends the currently selected Staff ID and Staff Name to the Admin GUI for Update and Delete purposes and is opened as modal. Create modified logic to open the Admin GUI to Create a new user when the Staff ID 77 and the Staff Name is empty. Read the appropriate criteria in the Admin GUI for further information.

1. Add suitable error trapping and user feedback via a status strip or similar to ensure a fully functional User Experience. Make all methods private and ensure the Dictionary is static and public.
2. Ensure all code is 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/).

## Question 5 Admin Interface Programming Criteria (Dictionary)

The Admin GUI must be in the same name space and solution folder as the General GUI from the previous question.

1. Create the Admin GUI with the following settings: GUI is model, all Control Box features are removed/hidden, then add two text boxes. The text box for the Staff ID should be read-only for Add, Update and Delete purposes.
2. Create a method that will receive the Staff ID from the General GUI and then populate text boxes with the related data.
3. Create a method that will create a new Staff ID and input the staff name from the related text box. The Staff ID must be unique starting with 77xxxxxxx while the staff name may be duplicated. The new staff member must be added to the Dictionary data structure.
4. Create a method that will Update the name of the current Staff ID.
5. Create a method that will Remove the current Staff ID and clear the text boxes.
6. Create a method that will save changes to the csv file, this method should be called as the Admin GUI closes.
7. Create a method that will close the Admin GUI when the Alt + L keys are pressed.
8. Add suitable error trapping and user feedback via a status strip or similar to ensure a fully functional User Experience. Make all methods private and ensure the Dictionary is updated.
9. Ensure all code is 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 and MS coding standards (refer http://www.citems.com.au/).
10. Addendum

This development utilises keyboard input for all General and Admin GUI events, you are permitted to reassign the key combinations from those described in the list of criteria; however, ensure these are recorded/displayed for the user’s convenience.

## Question 6 General Interface Programming Criteria (Sorted Dictionary)

Your second programming task is to create the General GUI which will load the data and display records. The final task is to create the Admin GUI to perform Create, Update and Delete operations. The csv file can be obtained from your lecturer once your Design has been approved.

### Create a SortedDictionary data structure with a TKey of type integer and a TValue of type string, name the new data structure “MasterFile”.

### Create a method that will read the data from the .csv file into the SortedDictionary data structure when the GUI loads.

### Create a method to display the SortedDictionary data into a non-selectable display only list box (ie read only).

### Create a method to filter the Staff Name data from the SortedDictionary into a second filtered and selectable list box. This method must use a text box input and update as each character is entered. The list box must reflect the filtered data in real time.

### Create a method to filter the Staff ID data from the SortedDictionary into the second filtered and selectable list box. This method must use a text box input and update as each number is entered. The list box must reflect the filtered data in real time.

### Create a method for the Staff Name text box which will clear the contents and place the focus into the Staff Name text box. Utilise a keyboard shortcut.

### Create a method for the Staff ID text box which will clear the contents and place the focus into the text box. Utilise a keyboard shortcut.

### Create a method for the filtered and selectable list box which will populate the two text boxes when a staff record is selected. Utilise the Tab and keyboard keys.

### Create a method that will open the Admin GUI when the Alt + A keys are pressed. Ensure the General GUI sends the currently selected Staff ID and Staff Name to the Admin GUI for Update and Delete purposes and is opened as modal. Create modified logic to open the Admin GUI to Create a new user when the Staff ID 77 and the Staff Name is empty. Read the appropriate criteria in the Admin GUI for further information.

1. Add suitable error trapping and user feedback via a status strip or similar to ensure a fully functional User Experience. Make all methods private and ensure the Dictionary is static and public.
2. Ensure all code is 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/).

## Question 7 Admin Form Programming Criteria (Sorted Dictionary)

The Admin GUI must be in the same name space and solution folder as the General GUI from the previous question.

1. Create the Admin GUI with the following settings: Form is model, all Control Box features are removed/hidden, then add two text boxes. The text box for the Staff ID should be read-only for Update and Delete purposes.
2. Create a method that will receive the Staff ID from the General GUI and then populate text boxes with the related data.
3. Create a method that will create a new Staff ID and input the staff name from the related text box. The Staff ID must be unique starting with 77xxxxxxx while the staff name may be duplicated. The new staff member must be added to the SortedDictionary data structure.
4. Create a method that will Update the name of the current Staff ID.
5. Create a method that will Remove the current Staff ID and clear the text boxes.
6. Create a method that will save changes to the csv file, this method should be called as the Admin GUI closes.
7. Create a method that will close the Admin GUI when the Alt + L keys are pressed.
8. Add suitable error trapping and user feedback via a status strip or similar to ensure a fully functional User Experience. Make all methods private and ensure the SortedDictionary is updated.
9. Ensure all code is 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 and MS coding standards (refer http://www.citems.com.au/).

## Addendum

This development utilises keyboard input for all General and Admin GUI events, you are permitted to reassign the key combinations from those described in the list of criteria, ensure these are recorded/displayed for the user’s convenience.

## Question 8 Testing and Code Optimisation

### File IO Optimisation

Ensure your code is error free and functions correctly, then test the method code for the major iterative File IO structures in the general and/or admin interfaces, ensure you record the performance details. Research alternatives to determine if your code is the best solution for reading and writing CSV files. Modify and optimise your methods as required, comment out older/slower code. Utilise the Trace and Stopwatch features in Visual Studio to collect data.

|  |  |  |
| --- | --- | --- |
| File IO Optimisation | | |
| **File IO Testing** | | |
| Read CSV | Description | Speed |
| Before | try  {    OpenFileDialog dialog = new OpenFileDialog(); // Open new Dialog  dialog.Filter = "CSV | \*.csv"; // Filter file types for CSV  Stopwatch sw = new Stopwatch(); // <-- create new stopwatch  if (dialog.ShowDialog() == DialogResult.OK) // if user clicked OK then read csv  {  sw.Start(); // <-- start stopwatch  MasterFile.Clear();  previousPath = dialog.FileName; // Collect the previous location  using (StreamReader reader = new StreamReader(dialog.FileName))  {  while (!reader.EndOfStream)  {  var line = reader.ReadLine(); // read current line of csv  if (line != ",") // The CSV has many ',' at the end of the file - this will ignore that error  {  var values = line.Split(','); // Split the data into:  int num = int.Parse(values[0]); // Key  string name = values[1]; // Value  MasterFile.Add(num, name); // Add Key and Value  }  }  }  }  sw.Stop(); // <-- stop stopwatch  TimeSpan ts = sw.Elapsed;  MessageBox.Show($"{ts}");  1. Opens a file dialog to select a CSV file.    2. Starts a stopwatch.    3. Clears the MasterFile dictionary and stores the file path.    4. Reads the CSV file line by line, splitting and adding valid lines to MasterFile.    5. Stops the stopwatch and shows the elapsed time. | Test 1:Test 2:  Test 3: |
| After | MalinStaffNamesV3.csv  try  {    OpenFileDialog dialog = new OpenFileDialog(); // Open new Dialog  dialog.Filter = "CSV | \*.csv"; // Filter file types for CSV  Stopwatch sw = new Stopwatch(); // <-- create new stopwatch  if (dialog.ShowDialog() == DialogResult.OK) // if user clicked OK then read csv  {  sw.Start(); // <-- start stopwatch  MasterFile.Clear();  previousPath = dialog.FileName; // Collect the previous location    using (var reader = new StreamReader(dialog.FileName))  using (var csv = new CsvReader(reader, new CsvConfiguration(CultureInfo.InvariantCulture)))  {  while (csv.Read())  {  var num = csv.GetField<int>(0);  var name = csv.GetField<string>(1);  MasterFile.Add(num, name);  }  }    }  sw.Stop(); // <-- stop stopwatch  TimeSpan ts = sw.Elapsed;  MessageBox.Show($"{ts}");  DisplayData(); // Display entries  }  catch (Exception ex)  {  MessageBox.Show(ex.Message); // An error occured  }  1. Opens a file dialog to select a CSV file.    2. Starts a stopwatch.    3. Clears the MasterFile dictionary and stores the file path.    4. Reads the CSV file and adds its contents to MasterFile.    5. Stops the stopwatch and shows the elapsed time.    6. Displays the data.  7. if it doesn’t complete any of those tasks then it will show an error message if something goes wrong | Test 1:Test 2:  Test 3: |
| Write CSV | Description | Speed |
| Before | Stopwatch sw = new Stopwatch();  sw.Start();  using (StreamWriter txtwriter = new StreamWriter(fileName))  {  foreach (var entry in masterFile)  {  string key = entry.Key.ToString();  string value = entry.Value;  string line = string.Format("{0},{1}", key, value);    txtwriter.WriteLine(line);  }  txtwriter.Close();  sw.Stop();  TimeSpan ts = sw.Elapsed;  MessageBox.Show($"{ts}");  }   1. Creates and starts a new stopwatch 2. Initializes are stream writer 3. Read each line from dictionary 4. Writes to the .csv file 5. Close stream writer 6. Stops stop watch then displays the time elapsed | Test 1:Test 2:Test 3: |
| After | var sw = Stopwatch.StartNew();    using (var txtWriter = new StreamWriter(fileName))  {  foreach (var entry in masterFile)  {  string line = $"{entry.Key},{entry.Value}";  txtWriter.WriteLine(line);  }  }    sw.Stop();  MessageBox.Show($"Time elapsed: {sw.Elapsed}");   1. Creates and starts a new stopwatch 2. Initializes are stream writer 3. Read each line from dictionary 4. Writes to the .csv file 5. Close stream writer 6. Stops stop watch then displays the time elapsed | Test 1:  Test 2:  Test 3: |
| **Conclusion and Recommendations** | | |
| For the reading tests, my original method proved to be more efficient for reading the CSV data compared to the other method provided to me by AI. CSVReader proved to be cumbersome to use as it required the program to include a new library, and the proof is in the time it takes to read the files. | | |
| For writing the data back to the csv or creating a new one, the optimized string formatting proved to be slightly better in terms of performance compared to the manual formatting and temporary storing of strings. | | |
| According to people on Stack overflow and AI such as copilot and ChatGPT, the CSVReader library is the most optimal to use for more complex data sets. So, that is the recommendation should the assignment require it. | | |

### Data Structure Optimisation

The Dictionary<T,V> data structure represents a basic hash table , while the alternative is the SortedDictionary<T,V> which is a basic binary search tree implementation.

For each Application run optimisation tests to determine the most efficient operational performance. Use the Performance Profiler and a release version to measure the CPU Usage, File IO and Memory Usage of both you Applications.

Record your results and determine the best performance; add comments to the code with the best and worst performance.

Your Application Optimisation report must include appropriate evidence that your multi-window application functions as expected (references to screen captures) and how the code could be optimised. Complete the following report template to answer this question.

|  |  |  |
| --- | --- | --- |
| Application Optimisation | | |
| **Dictionary<> Testing** | | |
| Method/Event | Description/Definition | CPU, Memory Usage, File IO |
| Display Data () | Displays and enters the keys and values into both the nonselectable listbox and the selectable listbox. |  |
| Textbox\_Search\_Value\_TextChanged() | Changes the label for what is being searched.  Clears the listbox items.  Gets the string from the textbox.  Inserts records from the dictionary that contain the arrangement of chars in the string that was collected into the selectable listbox. |  |
| textBox\_Search\_Key\_TextChanged() | Changes the label for what is being searched.    Clears the listbox items.    Gets the string from the textbox.  Inserts records from the dictionary that contain keys starting with the collected string into the selectable listbox. |  |
| Clear\_TextBox\_Search\_Value() | Clears the text from the search by value textbox |  |
|  |  |  |
| **SortedDictionary<> Testing** | | |
| Method/Event | Description/Definition | CPU, Memory Usage, File IO |
| Display Data () | Displays and enters the keys and values into both the nonselectable listbox and the selectable listbox. |  |
| Textbox\_Search\_Value\_TextChanged() | Changes the label for what is being searched.  Clears the listbox items.  Gets the string from the textbox.  Inserts records from the sorted dictionary that contain the arrangement of chars in the string that was collected into the selectable listbox. |  |
| textBox\_Search\_Key\_TextChanged() | Changes the label for what is being searched.    Clears the listbox items.    Gets the string from the textbox.  Inserts records from the sorted dictionary that contain keys starting with the collected string into the selectable listbox. |  |
| Clear\_TextBox\_Search\_Value() | Clears the text from the search by value textbox |  |
|  |  |  |
| **Conclusion and Recommendations** | | |
| The sorted dictionary has proved to be less cumbersome when using the display method.  The unsorted dictionary has proved to be slightly more effective when searching through the dictionary by value.  The sorted dictionary has proved to be slightly less memory intensive than the unsorted dictionary when searching by key.  The unsorted dictionary has proved to be less memory intensive when clearing the textbox with a keypress (f5) as this will also refresh the selectable listbox. | | |
|  | | |
|  | | |

# Question 9 Implementation Plan

Once all the development and testing has been completed you will need to consider an implementation plan. This plan will need to include a schedule (timeline), the activity/training priority and comment/feedback to ensure a trouble-free adoption of your application.

Complete the following Implementation Plan to answer this question.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Implementation Plan | | | | |
| Developer Name |  | | **Date** |  |
| Week/Month | Activity/Explanation/Training | Comment/Feedback | | |
| Week One | User training sessions | Get feedback on user experience | | |
| Week Two | Deployment to production environment | Monitor performance issues | | |
| Week Three | Addressing initial bugs | Collect reports and logs | | |
| Week Four | Follow-up training on advanced features | Get feedback for suggestions and improvements | | |
|  | Add additional rows as required |  | | |

## Question 10 Demonstration, Feedback and Signoff

Ensure your code is fully commented with your Name, ID, and Date placed above the main code body of each file. Check all the above documentation has been completed and is ready for inspection. Email your Lecturer (Assessor) and arrange a date and time to demonstrate your working applications, use the following Checklist to ensure you have completed all the assessment criteria.

|  |  |  |  |
| --- | --- | --- | --- |
| Checklist | | Completed | |
| **Questions** | | YES NO | |
| Q4 | General Interface Programming Criteria (Dictionary) |  |  |
| Q5 | Admin Interface Programming Criteria |  |  |
| Q6 | General Interface Programming Criteria (SortedDictionary) |  |  |
| Q7 | Admin Interface Programming Criteria |  |  |
| Q8 | Testing and Code Optimisation: All the fields in the Report have been filled in. |  |  |
| Q9 | Implementation Plan: The plan is complete with timeline, activities and feedback |  |  |
| Q10 | Demonstration: The multi-window application is complete and all components work correctly. |  |  |

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

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

End of Assessment Task Three – Part B