# Project Assessment: Develop and test an application

## Criteria

### Unit code, name and release number

ICTPRG403 Develop data-driven applications (1)

ICTPRG410 Build a user interface (1)

ICTPRG404 Test applications (1)

### Qualification/Course code, name and release number

ICT40518 - Certificate IV in Programming (1)

## Student details

### Student number

### Student name

## Assessment Declaration

By submitting this assessment online you acknowledge:

* This assessment is my original work and no part of it has been copied from any other source except where due acknowledgement is made.
* No part of this assessment has been written for me by any other person except where such collaboration has been authorised by the assessor concerned.
* I understand that plagiarism is the presentation of the work, idea or creation of another person as though it is your own. Plagiarism occurs when the origin of the material used is not appropriately cited. No part of this assessment is plagiarised.

Version: 20200108

Date created: 8 January 2020

Date modified: 13 January 2020

For queries, please contact:

Technology and Business Services SkillsPoint

Location: Ultimo

© 2019 TAFE NSW, Sydney  
RTO Provider Number 90003 | CRICOS Provider Code: 00591E

This assessment can be found in the: [Learning Bank](https://share.tafensw.edu.au/share/access/searching.do?doc=%3Cxml%2F%3E&in=P7ac4831b-430a-4b8d-8b56-f7b32ed5b9cf&q=&type=standard&sort=rank&dr=AFTER)

The contents in this document is copyright © TAFE NSW 2019, and should not be reproduced without the permission of the TAFE NSW. Information contained in this document is correct at time of printing: 23 November 2021. For current information please refer to our website or your teacher as appropriate.

## Assessment instructions

Table 1 Assessment instructions

| Assessment details | Instructions |
| --- | --- |
| **Assessment overview** | The objective of this assessment is to assess your knowledge and performance as required to develop and test a multi-layer data-driven application. |
| **Assessment Event number** | 1 of 3 |
| **Instructions for this assessment** | This is a project-based assessment and will be assessing you on your knowledge and performance of the unit.  This assessment is in two parts:  Research  Develop an order management system.  The assessment also contains:   * Assessment Checklist 1-2 * Observation Checklist 1-2 * Assessment Feedback.   **Check the Assessment and Observation checklists to ensure that you’ve covered all the required tasks.** |
| **Submission instructions** | On completion of this assessment, you are required to upload it or hand it to your assessor for marking.  Ensure you have written your name at the bottom of each page of your assessment.  Submit the following documents for each part:   * Part 1: Research   + This completed document * Part 2: Develop an order management system   + OMS project code   + This completed document.   It is important that you keep a copy of all electronic and hardcopy assessments submitted to TAFE and complete the assessment declaration when submitting the assessment. |
| **What do I need to do to achieve a satisfactory result?** | To achieve a satisfactory result for this assessment all questions must be answered correctly and all items in the Assessment and Observation Checklists must be marked Satisfactory. |
| **Assessment conditions** | Assessment conditions will be safe and replicate the workplace. Noise levels, production flow, interruptions and time variances must be typical of those experienced in the programming and software field of work.  Assessment may be undertaken in normal classroom conditions, which is assumed to be noisy and similar to workplace conditions, or within the workplace. This may include phones ringing, people talking and other interruptions. |
| **What do I need to provide?** | * USB drive or other storage method with enough free space to save work to. |
| **What will the assessor provide?** | * An integrated development environment (IDE) * Word processing software * Internet access * Code to be tested (student’s code verified by assessor) * Code to be tested (student’s code verified by assessor) * Coding standard as listed in instructions * Specific licensed tools for the platform you are using (API):   + Visual Studio 2019 (or higher)   + SQL Server + Management Studio 2017 (or higher) * Database management system software and files:   + Database Script (DB\_Script.sql)   + Database test data (DB\_Test\_Data\_Script.sql) |
| **Due date and time allowed** | Indicative time to complete assessment:   * In class: Four hours * Out of class: 16 hours. |
| **Assessment location** | Parts 2.4 and 2.10 will be completed in the classroom. All other parts will be completed both in and outside of the classroom.  The student may access their referenced text, learning notes and other resources. |
| **Supervision** | All other parts are an unsupervised, take-home assessment. Your assessor may ask for additional evidence to verify the authenticity of your submission and confirm that the assessment task was completed by you. |
| **Reasonable adjustment** | If you have a permanent or temporary condition that may prevent you from successfully completing the assessment event(s) in the way described, you should talk to your assessor about ‘reasonable adjustment’. This is the adjustment of the way you are assessed to take into account your condition, which must be approved BEFORE you attempt the assessment. |
| **Assessment feedback, review or appeals** | In accordance with the TAFE NSW policy *Manage Assessment Appeals,* all students have the right to appeal an assessment decision in relation to how the assessment was conducted and the outcome of the assessment. Appeals must be lodged within **14 working days** of the formal notification of the result of the assessment.  If you would like to request a review of your results or if you have any concerns about your results, contact your Teacher or Head Teacher. If they are unavailable, contact the Student Administration Officer.  Contact your Head Teacher for the assessment appeals procedures at your college/campus. |

.

## Part 1: Research

To complete this part of the assessment, you are required to research technologies, techniques and best practices available for the development of a multi-layer data application.

In the Moodle Section “Knowledge Assessment for Cluster 2” complete the questions given in “Project Part 1 Question” Quiz.

## Part 2: Develop an order management system

### Scenario

XYZ & Co. has requested your services for developing a multi-layer data-driven desktop application. They need an Order Management System (OMS) so their employees can record and manage custom orders of their available stock items.

XYZ & Co. has already been working with a consultant and provided you with the client requirements for the product as listed below.

Download and unzip the resource folder (Cl\_Programming\_2\_AE\_Pro\_1of3\_SR1.zip) to access files referred to within the assessment.

#### Product description

The OMS will be used for recording and managing customer orders and tracking stock item levels.

##### Adding a new order

A typical scenario of use starts when a customer contacts XYZ & Co and places an order. At this point an employee of XYZ & Co will start to enter the details of the order into the system. When a new order is created, a unique identifier is generated and assigned to the order. The current date and time is also captured and associated with the order. If at any time the employeecancels entering the order, any data relating to the Order that has already been saved to the database must be deleted.

##### Adding order line items

The employee will then add the order line items by selecting a stock item, entering the quantity and clicking the ‘Add Item’ button. The employee will follow this process for every order line item.

If the quantity entered for a stock item is higher than what is currently available in stock a warning must be displayed. Validation is also required to ensure a value greater than zero is entered for the quantity.

When an order is first created it is flagged as ‘New’ indicating order line items are still being added. After all of the items have been added the employee will ‘Submit’ the order, this will result in the order being flagged as ‘Pending’. An order can be pending for any period of time but most orders are processed within 2 – 3 business days.

It is also required that order items can be deleted from an order whilst it is flagged as ‘New’.

##### Viewing and processing orders

When the application is launched, the employee will be presented with a view that lists all of the orders in a data grid.

Each row will display the unique identifier (Id), date/time, number of line items, total and the current state of the order (New, Pending, Complete or Rejected).

The employee will then be able to select one of the orders and navigate to the ‘Order Details’ view, which will display the order details including the line items (SKU, Name, Price, Quantity and Total).

If the order is flagged as ‘Pending’ a ‘Process’ button will be visible so the order can be processed. When an order is processed the quantity of each item ordered is checked against the quantity currently in stock. If there is enough items in stock for the quantity ordered of all line items, the quantity ordered is decremented from the stock available and the order it is flagged as ‘Complete’. If there is not enough stock available at the order is flagged as ‘Rejected’, see **Figure 1 - Order State Flow Diagram**.

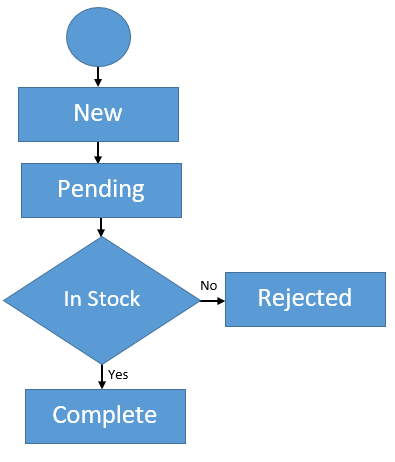


Figure 1 - Order State Flow Diagram

##### Stock items

Each stock item has a unique identifier (SKU), name, price and current stock level (number of items currently in stock). When an order is processed the number of items in stock is decremented accordingly.

##### Navigation

Design your UIs and provide a diagram to show how your application navigates between multiple views.

### Technical specifications

The architecture listed below must be used for the application. This includes one Windows Presentation Foundation (WPF) project, and the files needed to create the Domain, Data access and user interface.

* OrderManagementSystem (Solution)

### Development principles and practices

* All coding must comply with the coding standards as described at [C# Coding Standards and Naming Conventions](https://www.dofactory.com/reference/csharp-coding-standards).
* All methods must have internal XML documentation as illustrated below in **Figure 3: C# Internal XML comments (example)**; further information is available at [Recommended Tags for Documentation Comments (C# Programming Guide)](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/xmldoc/recommended-tags-for-documentation-comments).

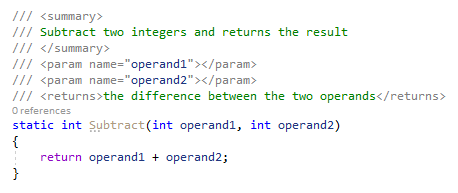


Figure 3: C# Internal XML comments (example)

### Task 1: User interface requirements

Document the requirements for the application presentation layer (user interface) according to the Product description from the client. List the requirements for each user interface (view) of the application in **Table 5 - User Interface Requirements.**

Each record must include an itemised list containing the following views:

* Responsibilities: What function(s) is the view responsible for providing?
* Components: What components are required? E.g. button
* Actions: What actions can the end-user perform? E.g. process order
* Events: What events can occur? E.g. button clicked
* Inputs/outputs: What input data is required and what output data is visible?

A minimum of four rows is required; add additional rows to the table if required.

Table 5 - User Interface Requirements

|  |  |
| --- | --- |
| View | Requirements |
| View and process orders | Responsibilities: Landing page, shows user list of orders in a listview with ID, Date, no of line items, total and status  Components: Button to refresh listview, view order details, add items button, new order button  Actions: ViewOrderDetails, RefreshOrders, CreateOrder adds automatic ID and time/date stamp, EditOrder  Events: OnClick refresh, OnSelectionChanged select order to edit/view, OnClick View details, OnClick CreateOrder, OnClick Edit order details, OnClick sort listview columns  Inputs/outputs: Output all orders as datarow, inputs are mouse click, output time.now to new order. |
| Order details | Responsibilities: Show details of order: SKU, name, price, quantity, total  Components: Cancel order button, Process order button  Actions: CancelOrder, ProcessOrder  Events: click cancel button, click process button, OnSelectionChanged determine visibility of ProcessOrder  Inputs/outputs: Remove order from db (& associated orderItems), decrement stock from available, change status to rejected or complete |
| Create order | Responsibilities: User can view all items that form part of an order. They can see stock items and select which to add to order.  Components: Button to submit, button to cancel, button to save draft, Item number selector (arrows), add to order button, remove from order button, warning message box, currentorder listview, stock items cbobox, txtbox to show currently available stock, txtbox to show cost to add selected items, txtbox to show total cost of order.  Actions: SubmitOrder: changes order status from ‘new’ to ‘pending’, CancelOrder: removes order from db if status is ‘new’, AddItemToOrder, RemoveItemFromOrder, SaveDraftOrder: closes current window and refreshes view of all orders.  Events: click submit order button, click cancel order button, click save draft order button, click add to order button, click remove from order button, OnSelectionChanged cboStockItems, click less and more quantity buttons  Inputs/outputs: output all order items from current order as listview, input item quantity, output order, output order status to pending, output available stock |
| View stock | Responsibilities: User can view all stock items  Components: stock items listview, txtboxes to display stock id, name, PPU and quantity in stock. Buttons to add, update, delete and cancel  Actions: AddStock, UpdateStock, DeleteStock  Events: OnSelectionChanged lvStockItems, OnClick btnAdd, OnClick btnUpdate, OnClick btnDelete, OnClick btnCancel  Inputs/outputs: output all stock items as listview, input name, price & stock on update |
| Message Box | Responsibilities: Displays warning or success message  Components: OK button, yes/no buttons  Actions: Close()  Events: OnClick button  Inputs/outputs: mouse click |

### Task 2: User interface technology and programming language

Identify an appropriate application development language to use for building the user interface. Explain why you have chosen this language and how it will meet current and/or future needs (approximately 50-100 words).

|  |
| --- |
| User interface technology and programming language |
| *(provide a summary and details of the selected technology and programming language here)*  The user interface will be written with the markup language Xaml, using Visual Studio to create a WPF application. Xaml is chosen as it is flexible and easy to use, and provides all the required functionality such as data pickers, dropdown menus, buttons etc. Xaml is also easy to transfer data between systems. It allows for scalability if the project needs to grow and include more functions. The functionality will be written using the programming language C#. C# is a class-based language. The classes will be useful for modelling the data from the provided database and assigning functionality to objects. |

### 

### Task 3: User interface prototype

Develop a prototype of the user interface for all the views identified and documented in Task 1, using the selected language from Task 2. Remember, a prototype is not a fully functioning application and is a tool used to demonstrate the form and function proposed for the finished application. Paste screenshots of the prototype screens in the space provided below.

|  |
| --- |
| User interface prototype |
|  |

### Task 4: Review prototype

You now need to review the prototype with the client in a role play, and consult with them on the design.

* For face-to-face students, your assessor will observe the role play and complete Observation Checklist 1.
* For online students, make an appointment with your assessor for an online meeting.

Role play participants:

* **Client** – arrange for another person to act as the client
* **Developer** (this is you).

After presenting the prototype, document any feedback, editing or actions required in **Table 6 - Prototype review**. Add additional rows as required.

Table 6 - Prototype review

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Comments | Signature (client) | Actions (student) |
|  |  |  |  |
|  |  |  |  |

### Task 5: Develop domain

Create your domain classes and write code for all the relevant classes included in the project. The domain **cannot** contain any data persistence logic; all responsibilities for persistence must be included in the data access layer.

### Task 6: Develop data access layer

Create the required Data access layer class(es) responsible for inserting, updating, deleting and retrieving data from the database.

To achieve this, you will need to select and reference an Application Programming Interface (API) for connecting with a database. Document the API and database connectivity details using internal XML comments in the relevant classes.

Use the provided SQL script **DB\_Script.sql** to create the database. Executing this script will create a database named **OrderManagementDb,** which contains all the database tables and initial data for **Stock Items**.

Create and manage the connection string to connect to the database.

### Task 7: Develop the application logic

Create the application logic to suit the requirements.

### Task 8: Build user interface

Build the user interface (presentation layer) with the required functionality, as determined by the prototype from**: Task 3: User interface** prototype**.**

**NOTE:** Be sure to handle any exceptions from the lower layers in the application architecture e.g., data access layer. Use try/catch blocks and display an appropriate message for any exceptions that may occur.

### Task 9: Test application

For this task, you are required to design, develop and execute tests for the application.

#### Part A – Test plan

Develop a test plan for the application and document the plan in **Table 9 - Test plan**.

The test plan must include:

* Scope
* Test objectives
* Roles and responsibilities
* Test types
* Unit Test
* Scripted UI Test
* Deliverables
* Testing tools.

Table 9 - Test plan

|  |  |
| --- | --- |
| Item | Description |
| **Scope** | 1. Adding a new order 2. Deleting an order 3. Adding items to an order 4. Removing items from an order 5. Submitting order for processing 6. Processing an order |
| **Test objectives** | 1. Adding a new order    * Insert new row in OrderHeaders table    * Unique OrderId is generated    * Default value of OrderStateId is 1 (New)    * Create new instance of OrderHeader    * Instance has correct values for variables 2. Deleting an order    * User confirms if they want to delete    * The row is removed from the OrderHeaders table    * Confirmation message 3. Adding items to an order    * Instance of OrderItem is created with user values and key value to the selected OrderHeader    * Row is inserted to OrderItems table 4. Removing items from an order    * Instance of OrderItem is created using values from selected row    * Row is removed from OrderItem table using unique identifiers 5. Submitting order for processing    * User is warned that no more changed will be permitted    * OrderHeader value for OrderStateId is changed to 2 (Pending)    * Row is updated in OrderHeader table using instance of selected OrderHeader 6. Processing an order    * Check if selected OrderHeader OrderStateId is 2 (Pending)    * For each OrderItem associated with the OrderHeader, compare its requested quantity with available stock for the associated StockItem. A StockItem instance is created for this purpose using the OrderItem values.    * If any of the OrderItems exceeds available stock, show a notification and stop the process. Update OrderHeader OrderStateId to 3 (rejected).    * If all of the OrderItems have available stock, display a success message, update the OrderHeader OrderStateId to 4 (complete), then update each StockItem with the subtracted quantities. |
| **Roles and responsibilities** | Test manager – the person writing this document defines the tests  Test developer – the person who writes their logic  Tester – the person who conducts the tests and logs their results |
| **Test types** | Unit testing for each of the methods involved in the objectives listed above  Integration testing for the objectives themselves – the units working together  User interface testing – do the components that a user sees perform as required |
| **Deliverables** | Test plan document  Test cases documents  Test scripts  Test data  Execution logs |
| **Testing tools** | *Identify and list a minimum of two testing tools that will be used in the testing plan. Provide a brief description (approximately 25-50 words) for each tool.*  Unit testing tools in Visual Studio  Visual studio supports the MSTest framework. It has a dedicated namespace TestTools.UnitTesting. A [TestClass] attribute is required for any class that contains unit test methods. Each test must be prefixed with [TestMethod]. Test methods must return void and cannot have parameters.  User interface testing tools in Visual Studio  User selects a new Coded UI Test Project and records actions to perform during the test. Then for example the developer can disable certain functionality and let the recorded actions run. Test results will show that the test failed if the test attempts invalid action. |

#### Part B – Test cases

Develop a minimum of two test cases for the application. Each test case must include the following:

* Test Case Identifier (Test #), name and description
* Priority (Low/Med/High)
* Test type/technique
* Test data and expected results
* Status (Pass/Fail)
* Steps taken.

**NOTE:** Use **Table 10: Test case 1** and **Table 11: Test case 2** provided below for each test case.

Table 10: Test case 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test # | Name and description | | | Priority |
| 1 | Add Order Header => This will test if a new OrderHeader is instantiated and then populated with the correct default values. | | | High |
| **Test type/technique** | | | | |
| **Unit test** | | | | |
| **Test data** | | **Expected** | **Actual** | **Pass/Fail** |
| “New”, DateTime.Now | | **The object was created with default values DateTime.Now, and Status is set to ‘new’.** | **As expected** | **Pass** |
| **Steps** | | | | |
| **1** | **User clicks btnCreateNewOrder** | | | |

Table 11: Test case 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test # | Name and description | | | Priority |
| 2 | Add Order Item => This will test if a new OrderItem is instantiated using the values input by the user. | | | High |
| **Test type/technique** | | | | |
| **Unit Test** | | | | |
| **Test data** | | **Expected** | **Actual** | **Pass/Fail** |
| 1, 1, “Chair”, 25, 1 [OrderHeaderId, StockItemId, Description, Price, Quantity] | | **The OrderItem is instantiated and populated with the values input for each property.** | **As expected** | **Pass** |
| **Steps** | | | | |
| **1** | User navigates to Window: Edit order details | | | |
| **2** | User selects stock item from combobox | | | |
| **3** | User selects desired quantity | | | |
| **4** | User clicks Add to Order | | | |

#### 

#### Part C – Test framework

Identify the test framework that you will use for testing the application, explaining why you have chosen this compared to other options (approximately 100-150 words).

Table 12 – Test Framework

|  |
| --- |
| Test Framework |
| Microsoft unit test framework for managed code. This framework is installed with Visual Studio and provides the framework for testing .NET code. The application is developed within Visual Studio and Microsoft provides an easy way to develop and implement testing within the same IDE. MTest framework has tools required for the unit testing described above, and user interface testing as is also required. |

#### Part D –Automated testing

##### Prepare test algorithm

Document the steps of a test algorithm for one of the unit tests used for testing the application. Provide the steps of the algorithm in **Table 13 - Test algorithm (pseudo-code)**.

Table 13 - Test algorithm (pseudo-code)

|  |  |
| --- | --- |
| Step | Logic and condition |
| **1** | The default constructor for an OrderHeader is called with properties for OrderStateId (“New”) and OrderDate(Time.Now). |
| **2** | The newly instantiated object contains properties with the correct default values |

##### Implement the test algorithm

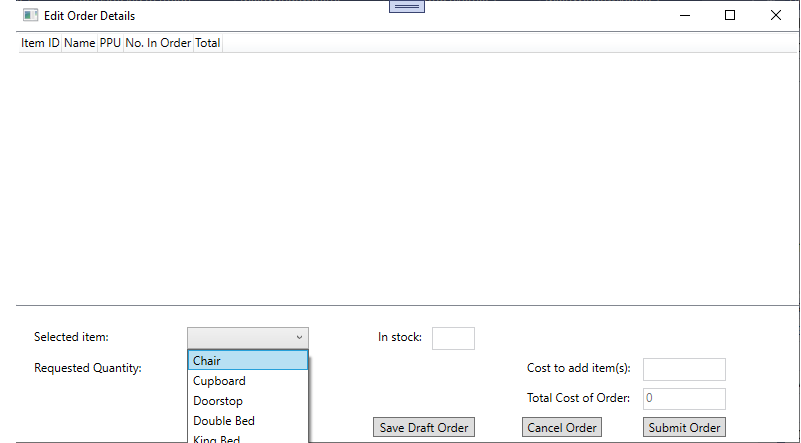
Write a script that implements your algorithm to test the application.

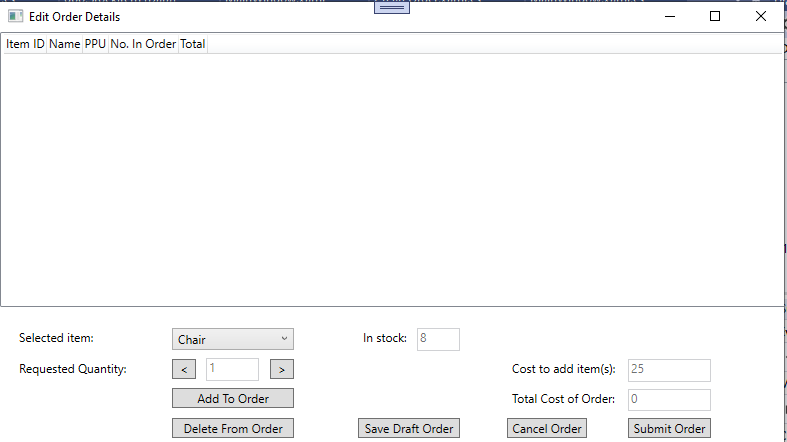
##### Execute the test script

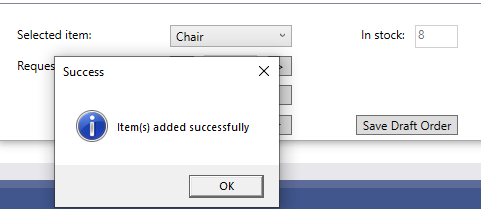
Document and analyse the results of the script in the test case document from **Part B – Test cases**. Include one or more screenshots of the test being run.

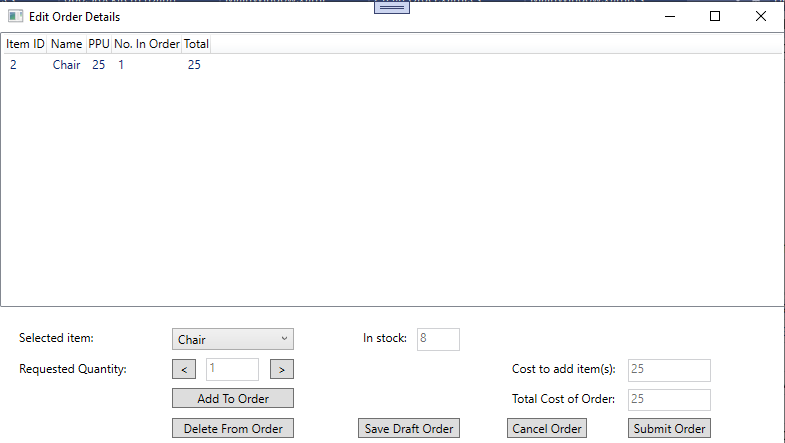
#### Part E – User interface testing

Using the second test case, perform a physical test of the UI. Document and analyse the results of the UI testing in the test case document from **Part B – Test cases**. Include one or more screenshots of the test being run.









#### Part F – Errata (bugs and fixes)

Document any errors (bugs) found during the development and/or testing of the application along with actions taken to resolve them and the status (has the error/bug been fixed). You must ensure that any errors are fixed so that your UI meets requirements.

Table 14 - Errata (bugs and fixes)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Description | Actions | Date/Time | Status |
| **1** | Loading OrderItems caused null exception error | Ensure all data types are correct and bindings correct | 07/12/2021 | Resolved |
|  |  |  |  |  |

#### Part G – Test progress

Document the progress of the application testing including all activities undertaken or to be undertaken, expected completion time, relevant notes and status.

Table 15 - Test progress report

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test # | Name and description | Expected Completion Date / Time | Notes | Status |
| **1** | Add Order Header | 07/12/2021 |  | Complete |
| **2** | Add Order Item | 07/12/2021 |  | Complete |
| **3** | UI Test | 07/12/2021 |  | Complete |

### Task 10: Obtain sign-off

Once your UI is complete and functional, you need to obtain client sign-off in a role play.

* For face-to-face students, your assessor will observe the role play and complete Observation Checklist 2.
* For online students, make an appointment with your assessor for an online meeting.

Role play participants:

* **Client** – arrange for another person to act as the client
* **Developer** (this is you).

Discuss the UI with the client and obtain their sign-off, completing **Table 16 – Completed UI sign-off**.

Table 16 – Completed UI sign-off

|  |  |  |
| --- | --- | --- |
| Date | Comments | Signature (client) |
|  |  |  |

## Assessment Checklist 1

The following checklist will be used by your assessor to mark your performance against the assessment criteria of Part 1. Use this checklist to understand what skills and/or knowledge you need to demonstrate in your submission. All the criteria described in the Assessment Checklist must be met. The assessor may ask questions while the submission is taking place or if appropriate directly after the task/activity has been submitted.

Table 17: Assessment Checklist 2

| TASK/STEP # | Instructions | S | U/S | Assessor Comments |
| --- | --- | --- | --- | --- |
| **1** | Researches and investigates techniques |  |  | *Assessors are to record their observations in sufficient detail to demonstrate their judgement of the student’s performance against the criteria.* |
| **2** | Interprets complex technical documents |  |  |  |

## Assessment Checklist 2

The following checklist will be used by your assessor to mark your performance against the assessment criteria of Part 2. Use this checklist to understand what skills and/or knowledge you need to demonstrate in your submission. All the criteria described in the Assessment Checklist must be met. The assessor may ask questions while the submission is taking place or if appropriate directly after the task/activity has been submitted.

Table 18: Assessment Checklist 2

| TASK/STEP # | Instructions | S | U/S | Assessor Comments |
| --- | --- | --- | --- | --- |
| **1.1** | Gathers and documents the user interface (UI) requirements from the scenario, following relevant protocols |  |  | *Assessors are to record their observations in sufficient detail to demonstrate their judgement of the student’s performance against the criteria.* |
| **1.2** | Formulates the content flow (inputs/ outputs) |  |  |  |
| **1.3** | Defines the UI actions |  |  |  |
| **1.4** | Itemises the UI events |  |  |  |
| **2.1** | Determines and documents the appropriate development language |  |  |  |
| **3.1** | Designs the UI components |  |  |  |
| **3.2** | Builds a prototype UI using the determined language |  |  |  |
| **5.1** | Documents code using appropriate format |  |  |  |
| **5.2** | Ensures application deals with disconnected data |  |  |  |
| **6.1** | Determines the data access API |  |  |  |
| **6.2** | Creates and manages a connection string |  |  |  |
| **6.3** | Connects to the data source |  |  |  |
| **6.4** | Queries data from the data source |  |  |  |
| **6.5** | Retrieves data from the data source |  |  |  |
| **6.6** | Inserts/updates data |  |  |  |
| **6.7** | Documents the database connectivity |  |  |  |
| **6.8** | Manages data integrity |  |  |  |
| **6.9** | Manages exceptions |  |  |  |
| **8.1** | Builds the UI with required functionality |  |  |  |
| **9.1** | Develops a test plan according to requirements |  |  |  |
| **9.2** | Identifies testing types and testing tools |  |  |  |
| **9.3** | Defines and designs two test cases to cover test requirements |  |  |  |
| **9.4** | Analyses and identifies test data using different test case design techniques |  |  |  |
| **9.5** | Selects and evaluates a unit test framework |  |  |  |
| **9.6** | Design and documents an accurate algorithm for testing in an appropriate format |  |  |  |
| **9.7** | Algorithm uses basic mathematical equations |  |  |  |
| **9.8** | Implements the algorithm in the test procedures |  |  |  |
| **9.9** | Tests the UI for completeness, consistency and functionality, according to requirements |  |  |  |
| **9.10** | Accurately records the test results in an appropriate format |  |  |  |
| **9.11** | Analyses test results |  |  |  |
| **9.12** | Iterates the UI build until it meets requirements |  |  |  |
| **9.13** | Creates test progress reports based on recorded results in an appropriate format |  |  |  |

## Observation Checklist 1

The Observation Checklist will be used by your assessor to mark your performance in any of the previous three event types. Use this Checklist to understand what skills you need to demonstrate in Part 2 Task 4. The Checklist lists the assessment criteria used to determine whether you have successfully completed this assessment event. All the criteria must be met. Your demonstration will be used as part of the overall evidence requirements of the unit. The assessor may ask questions while the demonstration is taking place or if appropriate directly after the task/activity has been completed.

Table 19 Observation Checklist

| Task # | Task/Activity Performed | S | U/S | Assessor Comments (Describe the student’s ability in demonstrating the required skills and knowledge) |
| --- | --- | --- | --- | --- |
| **1** | Reviews UI prototype with client |  |  | *Date of Observation:*  *Assessors are to record their observations in sufficient detail to demonstrate their judgement of the student’s performance against the criteria required.* |
| **2** | Participates in conversation relevant to own role |  |  |  |
| **3** | Initiates and takes the lead where appropriate |  |  |  |

## Observation Checklist 2

The Observation Checklist will be used by your assessor to mark your performance in any of the previous three event types. Use this Checklist to understand what skills you need to demonstrate in Part 2 Task 10. The Checklist lists the assessment criteria used to determine whether you have successfully completed this assessment event. All the criteria must be met. Your demonstration will be used as part of the overall evidence requirements of the unit. The assessor may ask questions while the demonstration is taking place or if appropriate directly after the task/activity has been completed.

Table 20 Observation Checklist

| Task # | Task/Activity Performed | S | U/S | Assessor Comments (Describe the student’s ability in demonstrating the required skills and knowledge) |
| --- | --- | --- | --- | --- |
| **1** | Obtains and documents client sign-off for completed UI |  |  | *Date of Observation:*  *Assessors are to record their observations in sufficient detail to demonstrate their judgement of the student’s performance against the criteria required.* |
| **2** | Participates in conversation relevant to own role |  |  |  |
| **3** | Initiates and takes the lead where appropriate |  |  |  |
| **4** | Follows relevant protocols |  |  |  |
| **5** | Meets expectations with own role |  |  |  |

## Assessment Feedback

*NOTE: This section* ***must*** *have the assessor signature and student signature to complete the feedback.*

### Assessment outcome

Satisfactory

Unsatisfactory

### Assessor feedback

☐ Has the Assessment Declaration been signed and dated by the student?

☐ Are you assured that the evidence presented for assessment is the student’s own work?

Was the assessment event successfully completed?

If no, was the resubmission/re-assessment successfully completed?

Was reasonable adjustment in place for this assessment event?  
*If yes, ensure it is detailed on the assessment document.*

Comments:

### Assessor name, signature and date:

### Student acknowledgement of assessment outcome

Would you like to make any comments about this assessment?

### Student name, signature and date

***NOTE: Make sure you have written your name at the bottom of each page of your submission before attaching the cover sheet and submitting to your assessor for marking.***