# Project Assessment

## Criteria

### Unit code, name and release number

ICTPRG502 - Manage a project using software management tools

ICTICT403 - Apply software development methodologies

ICTDBS502 - Design a database

ICTPRG425 - Use structured query language

ICTPRG527 - Apply intermediate object-oriented language skills

ICTPRG532 - Apply advanced object-oriented language skills

ICTPRG504 - Deploy an application to a production environment

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

ICT50718 - Diploma of Software Development (Release 1)

## Student details

### Student number

### Student name

## Assessment Declaration

* 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.

### Student signature and Date

Version: *1.0*

Date created: *2 August 2018*

Date modified: *30/01/2020*

For queries, please contact:

*SkillsPoint*

*Location*

© 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: 30 January 2020. 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 would be required to develop client user interface, develop mobile application and use XML effectively. |
| **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 XX parts and includes an Assessment Feedback form:   1. Research based 2. Product development 3. Assessment Checklist |
| **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 this assessment.  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. |
| **What do I need to provide?** |  |
| **What the assessor will provide?** | *Computers, data sheets, reference text, organisational policy etc that is referenced in the assessment. These may be hard copy or made available online.*  *(Adjust as required)* |
| **Due date and time allowed** | *Enter due date and time allowed* |
| **Assessment feedback, review or appeals** | Appeals are addressed in accordance with Every Students Guide to Assessment. |

## Specific task instructions

The instructions and the criteria in the tasks and activities below will be used by the assessor to determine if you have satisfactorily completed this assessment event. Use these instructions as a guide to ensure you demonstrate the required knowledge.

## Part 1: Research based

For this part of the assessment you are required to research and document your findings for each of the topics listed in **Table 2 - Research**. The topics relate to technologies, methods and tools relating to application development which will provide you with the under pinning knowledge required for Part 2: Product Development.

**\*NOTE: Each individual student must perform their own research and document the findings for each topic in their own words. Any references to websites or books must be acknowledged by providing either the URL or name of the book and accredited author.**

Table 2 - Research

|  |
| --- |
| **Traditional versus non-traditional system development methodologies**  \*Provide a brief summary and evaluation of the Software Development Lifecycle (SDLC) including the main differences between traditional (Waterfall) and non-traditional (Agile) methodologies. |
| *(document research findings here)* |
| **Agile methodology processes, artefacts, tools, resources and roles**  \*Provide a brief summary of the chosen development methodology, include a brief description on the different processes, artefacts, tools, resources and roles used in the methodology. |
| *(document research findings here)* |
| **Source control**  \*Provide a brief summary on source control, include an evaluation of specific technologies available and a description of the standard workflow followed when using source control. |
| * *(document research findings here)* |
| **Database Management Systems DBMS**  \*Provide a brief summary on what is a database management system.  Include information on:   * Data analysis in regards to determining appropriate data types and data structures * The characteristics and benefits of creating a conceptual model * The process and benefits of normalisation and eliminating data redundancy * Security features including authentication, authorisation and encryption * Service availability including, concurrency, redundancy and scalability |
| *(document research findings here)* |
| **Structured Query Language SQL**  \*Provide a brief summary of SQL, more specifically include information on:   * Data Definition Language DDL, Data Control Language DCL and Data Manipulation Language DML * The characteristics of an SQL statement * The four main types of operations (CRUD) * Common keywords used for sorting and filtering * Grouping and aggregate functions * Sub-queries * Inner, outer, left and right Joins * Views, stored procedures, functions and triggers |
| *(document research findings here)* |
| **Coding Standards**  \*Provide a brief summary of coding standards, why they are important and some specific examples of how coding standards can be applied in application development. |
| *(document research findings here)* |
| **Object Oriented Programming**  \*Provide a brief summary of object oriented programming , include information on the fundamental concepts, the benefits of developing with an object oriented language and the main differences between object oriented and procedural programming languages. |
| *(document research findings here)* |
| **Advanced data structures**  \*Provide a description of some common advanced data structures including:   * Singly-Linked List * Doubly-Linked List * Binary tree * Hash table |
| *(document research findings here)* |
| **Space Time Complexity**  \*Provide a description of space time complexity, what is it and why is it important?  Include a description and comparison of:   * three sorting algorithms * three search algorithms |
| *(document research findings here)* |

## Part 2: Product Development

### Case Study

#### Background

**Bit by Byte Development Pty Ltd** has contracted your team for developing a reservation system for a restaurant named **Bean Scene**. Currently the restaurant has been managing reservations manually using pen and paper but would like to start managing all reservations electronically.

#### Workflow

The typical workflow for managing reservations is:

1. The restaurateur (or manager) **schedules sittings** on a quarterly basis, the characteristics of a sitting is as follows:
   1. a sitting is typically either **breakfast**, **lunch** or **dinner** (sometimes there are special events)
   2. a sitting has a **start time** and **end time**, for example breakfast 7am to 11am
   3. the restaurant **capacity** is defined for each sitting, for example there can be a maximum of 40 guests for the dinner sitting Friday 6th March 6pm – 11pm
   4. a sitting can be **closed** for reservations, this typically occurs due to the restaurant being **booked out** or the sitting is a **private event**
2. The restaurant receives a **reservation request** for a particular sitting either **in-person**, by **email** or **phone**, for example someone requests a reservation for 3 people for the dinner sitting Friday 6th March 6pm – 11pm.
3. A reservation request may also have **notes or additional requirements** that need to be captured such as “a table by the window” or “high chair required”
4. When a reservation request is first captured it’s status is flagged as **pending**
5. Typically one week (or less) before the sitting a staff member contacts the guest to confirm the reservation, if confirmation is successful the status of the reservation is updated to **confirmed**
6. If the guest contacts the restaurant to cancel the reservation it’s status is updated to **cancelled**
7. The reservation is assigned one or more tables of an area in the restaurant, this may be done any time before the start time of the reservation.
8. When the reservation starts and the guest(s) are taken to their table by a staff member and the status of the reservation is updated to **seated**
9. The reservation status is updated to **completed** when the reservation is over

#### System Requirements

**Bit by Byte Development Pty Ltd** has done some preliminary work for gathering requirements from the client which are listed below.

##### Making a reservation

The restaurant wants their guests to have many options when making a reservation, this includes:

* Online via the restaurant’s website
* Mobile app developed specifically for the restaurant guests to make reservations
* In person when the guest is in the restaurant prior to the date of the reservation request
* By email when the guest sends an email to the restaurant requesting a reservation
* By phone when the guest calls the restaurant requesting a reservation

In all of these cases the restaurant must be able to enter the reservation into the system in order to capture the details of the reservation. When entering a reservation in the system the following details must be captured:

* The guest’s name (first and last), email and phone number
* Which sitting the reservation is for e.g. Breakfast 30-01-2020 6:30 AM – 11:00 AM
* The start time of the reservation e.g. 8:00 AM
* The duration of the reservation e.g. 90 minutes (1.5 hours)
* The number of guests e.g. 3 people
* The source of the reservation e.g. Online, Mobile, In-person, Email or Phone
* Any additional notes or special requests

##### Areas and Tables

The restaurant has also requested for the system to be capable of managing the restaurant’s tables and areas. The restaurant is divided into different areas as follows: the main dining area, the outside area and the balcony area. Each area has its own tables with a unique identifier as shown in **Table 3 - Restaurant Areas & Tables**. After a reservation has been captured it is assigned to one or more tables any time prior to the commencement of the reservation.

Table 3 - Restaurant Areas & Tables

|  |  |
| --- | --- |
| Area | Tables |
| Main | M1, M2, M3, M4, M5, M6, M7, M8, M9, M10 |
| Outside | O1, O2, O3, O4, O5, O6, O7, O8, O9, O10 |
| Balcony | B1, B2, B3, B4, B5, B6, B7, B8, B9, B10 |

##### Authentication and Authorisation

The system will have three different types of users as listed below:

* **Manager**: the manager has the highest level of access and is responsible for scheduling sittings, capturing and updating reservation requests and viewing reports
* **Staff**: a standard staff member can only capture and update reservation requests, access to scheduling sittings and viewing reports must be restricted
* **Member**: if so desired a guest can register and create an account in the system. When this is done the guest is known as a member and has the ability to log in and view past, current and future reservations. If a member is logged in when making a reservation either online or by using the mobile app, the system automatically knows and fills in their details, this convenience is motivation for guests to register as a member.

### Major project – Development Tools, Methods and collaboration

Before commencing development of the applications you are required to evaluate and select the development tools and methods that will be employed to support managing the development process. To achieve this you are required to complete the information required in **Table 4 – Project management tools and methods**

Table 4 – Project management tools and methods

|  |
| --- |
| Project Management – Tools and methods |
| **Determine the software development methodology to be used**  Evaluate and select a software development methodology to be used for the development of the major project, document which methodology was selected and your reasons for selecting that methodology |
| (provide your answer here) |
| **Determine the project management software to be used**  Evaluate and select project management software to be used for the development of the major project, document the selected software and reason for making the selection. |
| (provide your answer here) |
| **Determine the source-control system to manage the source code and to handle conflicts**  Evaluate and select the source control technology to be used for the development of the major project, document the selected technology and reason for making the selection. |
| (provide your answer here) |
| **Determine the collaboration software to be used**  Evaluate and select the collaboration software to be used during development of the major project, document the selected software and reason for making the selection. |
| (provide your answer here) |

### Apply intermediate-advanced object-oriented language skills

During the development process of the project you are required to implement and provide evidence for criteria that demonstrates the required skill and knowledge for intermediate to advance object oriented language skills as listed in **Table 5 - Application and Evidence**. You are required document how each item has been implemented in the project along with supporting evidence such as external documentation, screen shots and an explanation of application.

As an example for the first item (1) you can describe the basic structure of your solution including namespaces and references assemblies along with a screen shot of Solution Explorer using the snipping tool. For the sixth item (6) you can provide a description of coding standards that have been applied in the project along with an example screen shot. For the item nineteen (19) you can provide a screen shot of you debugging the application with a brief description of the process being debugged. For item twenty one (21) you can submit an external document of the designed test plans etc…

Further to this your assessor will provide an opportunity for you to demonstrate your application of each item for observation and sign off on completion of all items along with providing any relevant feedback. Evidence will also be collected by submission of the actual project code.

Table 5 - Application and Evidence

|  |  |
| --- | --- |
| # | Application and Evidence |
| 1 | Organise multiple source code files into logical units and packages using namespaces, folders and assemblies |
| (provide your evidence here) | |
| 2 | Implement a minimum of two collections that allow for internal storage of data e.g. List, Array etc… |
| (provide your evidence here) | |
| 3 | Implement a minimum of one data sorting algorithm using a comparator |
| (provide your evidence here) | |
| 4 | Implement a minimum of one search algorithm |
| (provide your evidence here) | |
| 5 | Employ integrated-development environment facilities, to make files to automate program building |
| (provide your evidence here) | |
| 6 | Follow the guidelines for developing maintainable code, adhering to coding standards |
| (provide your evidence here) | |
| 7 | Use the facilities in the language for persisting objects to binary files |
| (provide your evidence here) | |
| 8 | Use method overloading facilities available in the language, at an introductory level |
| (provide your evidence here) | |
| 9 | Use exception-handling techniques to ensure program stability |
| (provide your evidence here) | |
| 10 | Use of a class that is based on multiple inheritances |
| (provide your evidence here) | |
| 11 | Design and implement programs that connect to a database |
| (provide your evidence here) | |
| 12 | Design and implement programs that use the language facilities to extract, update, and delete data stored in a database |
| (provide your evidence here) | |
| 13 | Design and implement programs that use the language facilities to manipulate database structure (query, create and delete) |
| (provide your evidence here) | |
| 14 | Write programs that deliver transactional integrity |
| (provide your evidence here) | |
| 15 | Employ the graphical user interface (GUI) framework, or text windowing interface, appropriate to the chosen language |
| (provide your evidence here) | |
| 16 | Use standard GUI components |
| (provide your evidence here) | |
| 17 | Use the facilities within the language for GUI objects to respond to user and program-generated event |
| (provide your evidence here) | |
| 18 | Use stand-alone debugging tools, or tools provided by the integrated development environment, to examine variables, and trace the running code |
| (provide your evidence here) | |
| 19 | Use the debugger to detect logical and coding errors |
| (provide your evidence here) | |
| 20 | Use the tracing of code and examination of variable contents during execution, to detect and correct errors |
| (provide your evidence here) | |
| 21 | Design and document limited tests of code |
| (provide your evidence here) | |
| 22 | Undertake limited testing of the produced code to ensure that it complies with the program specification |
| (provide your evidence here) | |
| 23 | Capture and document the test results |
| (provide your evidence here) | |
| 24 | Create and maintain program documentation |
|  | |
| 25 | implement nested classes |
| (provide your evidence here) | |
| 26 | design and implement dynamic data structures including a doubly linked list and a binary tree |
| (provide your evidence here) | |
| 26 | design and implement a data structure utilising a hash function |
| (provide your evidence here) | |
| 28 | implement two inter-process communication mechanisms |
| (provide your evidence here) | |
| 29 | implement a graphical user interface with drag and drop, help files and 2D graphics |
| (provide your evidence here) | |
| 30 | utilise an architectural framework and third party library |
| (provide your evidence here) | |
| 31 | implement a client-server application including data transfers |
| (provide your evidence here) | |
| 32 | utilise a version control system for code and documentation management |
| (provide your evidence here) | |
| **Observed by Assessor** | |
| Name:  Date:  Feedback | |

### Database development

As with writing code when developing the application layer of your project, database design, development and validation are common activities undertaken in field of software development. Throughout the development of the application you are required to apply and provide evidence for each item listed in **Table 6 - Database application and evidence**. You are required to provide evidence in the form of screen shots, description of how the item has been applied or supporting documentation.

Further to this your assessor will provide an opportunity for you to demonstrate your application of each item for observation and sign off on completion of all items along with providing any relevant feedback. Evidence will also be collected by submission of the actual project database and code.

Table 6 - Database application and evidence

|  |  |
| --- | --- |
| # | Application and Evidence |
| 1 | Meet with the client, and conduct a user-needs analysis, to determine database functionality |
| (provide your evidence here) | |
| 2 | Analyse the results of a user-needs analysis to identify technical requirements |
| (provide your evidence here) | |
| 3 | Develop a conceptual model of the database |
| (provide your evidence here) | |
| 4 | Submit the conceptual model to the client for review |
| (provide your evidence here) | |
| 5 | Evaluate client feedback and make changes as required |
| (provide your evidence here) | |
| 6 | Identify the attributes and determine the data types |
| (provide your evidence here) | |
| 7 | Undertake the normalisation of attributes |
| (provide your evidence here) | |
| 8 | Develop an entity-relationship (ER) diagram in order to clarify the cardinality of relationships |
| (provide your evidence here) | |
| 9 | Document attributes, normalised data, and the ER diagram |
| (provide your evidence here) | |
| 10 | Forward documentation to the client for confirmation |
| (provide your evidence here) | |
| 11 | Confirm primary and foreign keys for tables |
| (provide your evidence here) | |
| 12 | Review client business rules |
| (provide your evidence here) | |
| 13 | Identify the referential integrity constraints |
| (provide your evidence here) | |
| 14 | Establish database management system constraints and incorporate into database design |
| (provide your evidence here) | |
| 15 | Develop the validation rules for data |
| (provide your evidence here) | |
| 16 | Design indexes and develop the data dictionary |
| (provide your evidence here) | |
| 17 | Document the database design |
| (provide your evidence here) | |
| 18 | Design the user interface for database, including menus, input screens and outputs |
| (provide your evidence here) | |
| 19 | Design queries, based on requirements |
| (provide your evidence here) | |
| 20 | Design output reports, based on requirements |
| (provide your evidence here) | |
| 21 | Compare the physical design with the conceptual model, or user-needs analysis |
| (provide your evidence here) | |
| 22 | Incorporate changes as required |
| (provide your evidence here) | |
| 23 | Review the business security plan as a basis for commencing the access and security design |
| (provide your evidence here) | |
| 24 | Design the password and access system for the database |
|  | |
| 25 | Identify multiple-user requirements |
| (provide your evidence here) | |
| 26 | Develop client access profiles using the client business model |
| (provide your evidence here) | |
| 26 | Identify the database backup and recovery requirements |
| (provide your evidence here) | |
| 28 | Develop and document the database backup and restore procedures |
| (provide your evidence here) | |
| 29 | Submit the database, and documentation, to the client for final approval |
| (provide your evidence here) | |
| 30 | Retrieve all the data from a single table |
| (provide your evidence here) | |
| 31 | Retrieve data from specific columns in a single table |
| (provide your evidence here) | |
| 32 | Use 'order by' to sort query output |
| (provide your evidence here) | |
| 33 | Restrict the number of rows retrieved, by placing criteria in the 'where' clause |
| (provide your evidence here) | |
| 34 | Restrict the number of rows retrieved, by placing specific criteria in the select statement |
| (provide your evidence here) | |
| 35 | Use comparison operators in the 'where' clause to compare numeric, character, string, date and time data |
| (provide your evidence here) | |
| 36 | Use Boolean operators with the correct precedence |
| (provide your evidence here) | |
| 37 | Use criteria in the 'where' clause, to check for a range of values, to select values from a list, and to check for values that match a pattern |
| (provide your evidence here) | |
| 38 | Use SQL syntax to suppress duplicate values from query results |
| (provide your evidence here) | |
| 39 | Take action to exclude null values from a query result |
| (provide your evidence here) | |
| 40 | Use arithmetical operators with the correct precedence |
| (provide your evidence here) | |
| 41 | Use string functions, and operators, to obtain the required query output |
| (provide your evidence here) | |
| 42 | Use mathematical functions to obtain the required output |
| (provide your evidence here) | |
| 43 | Use date functions to obtain the required output |
| (provide your evidence here) | |
| 44 | Use SQL aggregate functions to obtain the required output |
| (provide your evidence here) | |
| 45 | Use 'group by' to aggregate data by multiple columns |
| (provide your evidence here) | |
| 46 | Sort aggregated data in the query output |
| (provide your evidence here) | |
| 47 | Filter aggregated data using the 'having' clause |
| (provide your evidence here) | |
| 48 | Employ the inner join syntax, to retrieve data from two or more tables |
| (provide your evidence here) | |
| 49 | Use 'left outer', 'right outer' and 'full outer' syntax, to join tables in the select statement |
| (provide your evidence here) | |
| 50 | Use correct syntax in the 'where' clause, to retrieve data from multiple tables |
| (provide your evidence here) | |
| 51 | Write a union query that retrieves data from more than one table |
| (provide your evidence here) | |
| 52 | Construct single and nested sub-queries |
| (provide your evidence here) | |
| 53 | Construct sub-queries that return a single row and multiple rows |
| (provide your evidence here) | |
| 54 | Use correlated sub-queries to retrieve required data |
| (provide your evidence here) | |
| 55 | Write sub-queries that use aggregates |
| (provide your evidence here) | |
| 56 | Identify the required columns, data types, keys, relationships, indexes and constraints |
| (provide your evidence here) | |
| 57 | Use the relevant naming conventions for database elements |
| (provide your evidence here) | |
| 58 | Create tables that implement the required elements |
| (provide your evidence here) | |
| 59 | Manipulate tables to meet specific requirements |
| (provide your evidence here) | |
| 60 | Create views that satisfy information requirements |
| (provide your evidence here) | |
| 61 | Use check constraints in a view |
| (provide your evidence here) | |
| 62 | Create and execute stored procedures, that use one or more parameters |
| (provide your evidence here) | |
| 63 | Drop a view from a database |
| (provide your evidence here) | |
| 64 | Drop a stored procedure from the database |
| (provide your evidence here) | |
| 65 | Create and test database triggers that automate data management, or perform specific required data-related functions |
| (provide your evidence here) | |
| **Observed by Assessor** | |
| Name:  Date:  Feedback | |

### Deployment

During the development phases and on completion of the project you are required to deploy the project to a production environment. **Table 7 - Deployment evidence** lists all of the items and evidence required for demonstrating the deployment of your project. Evidence can be provided as screen shots, description of application in relation to your project and external documentation. Further to this your assessor will provide an opportunity for you to demonstrate your application of each item for observation and sign off on completion of all items along with providing any relevant feedback.

Table 7 - Deployment evidence

|  |  |
| --- | --- |
| # | Application and Evidence |
| 1 | Determine if a client system and a server system meet the requirements for installation |
| (provide your evidence here) | |
| 2 | Determine an installation method |
| (provide your evidence here) | |
| 3 | Review the security requirements |
| (provide your evidence here) | |
| 4 | Prepare a software-installation plan |
| (provide your evidence here) | |
| 5 | Create an install package for an application |
| (provide your evidence here) | |
| 6 | Test the install package in a test environment |
| (provide your evidence here) | |
| 7 | Deploy the install package to a production environment |
| (provide your evidence here) | |
| 8 | Create an uninstall package for removal of installed components |
| (provide your evidence here) | |
| 9 | Test the uninstall package in a test environment |
| (provide your evidence here) | |
| 10 | Deploy a database from the development environment to the production environment |
| (provide your evidence here) | |
| 11 | Specify the connection string to the database |
| (provide your evidence here) | |
| 12 | Configure the application to use parameters that are valid for the production environment |
| (provide your evidence here) | |
| 13 | Use the configuration files to modify the deployment variables |
| (provide your evidence here) | |
| 14 | Configure the security features in an application |
| (provide your evidence here) | |
| **Observed by Assessor** | |
| Name:  Date:  Feedback | |

### Development Activities

The following process of development activities must be completed and evidence provided. Supporting evidence can be reference to external documents such power point presentations, video recordings or presentation, report exported from project management tools. Further to this your assessor will provide an opportunity for you to demonstrate your application of each item for observation and sign off on completion of all items along with providing any relevant feedback.

|  |  |
| --- | --- |
| # |  |
| 1 | Interview client and document system requirements |
| 2 | Develop prototype, system proposal and present to client for approval |
| 3 | Implement any changes as per requirement of client feedback |
| 4 | Breakdown system requirements into tasks for prioritising and assignment to team members. |
| 5 | Develop project in iterations as per development methodology including activities such as testing and code reviews |
| 6 | Demonstrate completed project to client in presentation for acceptance, approval and feedback |
| 7 | Implement any changes as per requirement of client feedback |
| Supporting Evidence: | |
| **Observed by Assessor** | |
| Name:  Date:  Feedback | |

## Part 3: Assessment Checklist

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

| # | Instructions | S | U/S | Assessor Comments |
| --- | --- | --- | --- | --- |
| **1** | *Research completed to a satisfactory level* |  |  |  |
| **2** | *Development tools, methods and collaboration evaluation completed to a satisfactory level* |  |  |  |
| **3** | *Implementation and sufficient evidence provided for applying intermediate-advanced object oriented language skills* |  |  |  |
| **4** | *Implementation and sufficient evidence provided for database development* |  |  |  |
| **5** | *Implementation and sufficient evidence provided for database deployment* |  |  |  |
| **6** | *Development activities completed to a satisfactory level* |  |  |  |

## Assessment Feedback

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

### Assessment outcome

Satisfactory

Unsatisfactory

### Assessor Feedback

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.***