**A white cover with colorful lines and text

AI-generated content may be incorrect.**

**SAT Rules and Regulation**

* This SAT will be completed both in class and outside of class.
* The solution of this task must be your own work and completed by yourself.
* There is a three stage authentication process. Any inconsistencies will risk in you facing the BHS internal assessment board.

1. Your declaration
2. The journal with teacher comment and signature;
3. The compass submissions;

* It is your responsibility to make sure the submission is of a valid format so that it can be opened and viewed by the teacher. If it is corrupt in any way, it will not be assessed for that session. It is recommended that upon submission, you download your submission and check for yourself.
* Your submission must be made before the end of the due date. Failure to do so may result in a 0 score.
* Do not share your solutions with anyone during the timeframe of the SAT.

I understand and agree to abide by the above rules and regulation.

Student name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SCHOOL-ASSESSED TASK**

**VCE Software Development UNIT 3 OUTCOME 2**

Unit 3 Outcome 2: Analysis and Design

Document a problem, need or opportunity, formulate a project plan, document an analysis, and generate design ideas and a preferred design for creating a software solution.

**Nature of task**

A brief outlining the proposed solution and a project plan (Gantt chart) indicating tasks, times, milestones, dependencies and the critical path

**And**

Analytical tools that depict the interactions between systems, users, data and networks

**And**

An analysis that defines the requirements, constraints and scope of a solution in the form of a software requirements specification

**And**

A folio of design ideas and evaluation criteria

**And**

Detailed design specifications of the preferred design.

**Scope of task**

Development of a brief and project plan

In preparation for the SAT students will need to be able to identify a real-world problem, need or opportunity that can be developed as a software solution for a client or potential users.

Criterion 1 assesses students’ skills in developing a brief and in project management. Students will need to prepare a brief that documents their problem, need or opportunity.

Students will prepare a Gantt chart using an appropriate software tool that documents all the stages and the activities of the problem-solving methodology for Unit 3 Outcome 2 and Unit 4 Outcome 1 (both parts of the SAT).

Students will need to document all the relevant tasks, sequencing, time allocations, milestones, dependencies and critical path.

The evidence from this task is observed through Progress Check 1 & 2 and assessed through Criterion 1.

Documentation of the analysis

Criterion 2 assesses students’ skills in documenting the analysis. Students are required to collect and prepare data for analysis using appropriate data collection methods. The data collected will enable students to document the relevant features of the selected analytical tools and depict the relationships between the data, users and systems. The process of data collection may involve students communicating back-and-forth with their clients or potential users.

The data collected and the analytical tools will assist in the development of a software requirements specification in Criterion 3.

The evidence from this task is observed through Progress Check 3 & 4 and assessed as part of Criterion 2.

Development of a software requirements specification

Criterion 3 assesses students’ skills in documenting a software requirements specification. Students will document the functional and non-functional requirements, constraints and scope as well as the user characteristics and the technical environment for the proposed software solution. The software requirements specification is to be presented as a formal document. Analytical tools developed in Criterion 2 are to be included in the appendix of the software requirements specification.

Students will document evidence of their critical and creative thinking using questions and follow-up questions to clarify the development of the solution requirements as part of the Analysis stage in Criterion 2 and 3.

The evidence from this task is observed through Progress Check 5 and assessed through Criterion 3.

Generating design ideas and developing evaluation criteria

Criterion 4 assesses students’ skills in generating design ideas and developing evaluation criteria. Students will generate design ideas for the software solution using a range of ideation tools. Design ideas are to be annotated to explain the appearance and functionality of the software solution. They will develop evaluation criteria for their design ideas and the software solution. Evaluation criteria will reference the functional and non-functional requirements for the design ideas and be used to measure the efficiency and the effectiveness of the software solution in Criterion 10. Students will need to justify which of the elements of the design ideas should be further developed into detailed designs.

The evidence from this task is observed through Progress Check 6 and assessed through Criterion 4.

Producing detailed designs

Criterion 5 assesses students’ skills in producing detailed designs. Students will produce their detailed designs for the software solution using a range of design tools. They will also document the use of design principles that influence the appearance and functionality of the detailed designs as well as the characteristics of user experience that affect the detailed designs.

Students will document evidence of their critical and creative thinking through the development of design ideas, solution requirements and the detailed designs as part of the Design stage in Criterion 5.

The evidence from this task is observed through Progress Check 7 and assessed through Criterion 5.

**SCHOOL-ASSESSED TASK**

**VCE Software Development UNIT 4 OUTCOME 1**

This outcome will be assessed separately. However, the project plan for Criterion 1 in Unit 3 Outcome 2 must include these tasks.

Unit 4 Outcome 1: Development and Evaluation

Develop and evaluate a software solution that meets requirements and assess the effectiveness of the project plan.

**Nature of task**

A software solution that meets the software requirements specification

**And**

Preparation and conduction of beta testing

**And**

* + an evaluation of the efficiency and effectiveness of the software solution
  + an assessment of the effectiveness of the project plan (Gantt chart) in monitoring project progress

in one of the following:

* + a written report
  + an annotated visual plan.

**Scope of task**

Development of the software solution

Criterion 6 assesses students’ skills in using the features of the programming language to develop the software solution. In order to develop the software solution students are required to use an appropriate programming language that meets the prescribed list of software tools and functions, and outcome specific requirements of the study. Students will also use a range of appropriate data types, data structures and data sources.

The evidence from this task is observed through Progress Check 8, 9 &10 and assessed through Criterion 6.

Criterion 7 assesses students’ skills in developing the software solution. Students will apply suitable naming conventions, write internal documentation and apply appropriate validation techniques.

The evidence from this task is observed through Progress Check 8, 9 &10 and assessed through Criterion 7.

Debugging and alpha testing of the software solution

Criterion 8 assesses students’ skills in debugging and alpha testing the software solution. Students will document the use of debugging and testing techniques to ensure their software solution functions as expected.

Students will document evidence of their critical and creative thinking through the modification of designs, evaluation criteria and the development of the software solution as part of the Development stage in Criterion 8.

The evidence from this task is observed through Progress Check 9 &10 and assessed through Criterion 8.

Beta testing

Criterion 9 assesses students’ skills in conducting beta testing. Students will document the preparation of a beta testing plan and test scenarios and then conduct the beta testing. After performing the beta testing with their clients or potential users, students will document the results. Students will use the results of the beta testing to make recommendations for modifications to the software solution. Students could choose to make modifications to the software solution or to document the actual modifications they would make to the software solution in a written report.

The evidence from this task is observed through Progress Check 11 and assessed through Criterion 9.

Evaluation of the software solution and assessment of the project plan

Criterion 10 assesses students’ skills in evaluating the software solution and assessing the project plan. Students will document the evaluation of the efficiency and effectiveness of the software solution using the evaluation criteria developed in Criterion 4. This includes the extent to which it meets the functional and non-functional requirements. Students will then propose an evaluation strategy to be conducted in the future.

Students will also need to document evidence of their critical and creative thinking through the evaluation of the process they followed through the analysis, design and development stages and discuss improvements that could be made to the software solution as part of the Evaluation stage in Criterion 10.

Students will document the modifications made to the initial project plan throughout the duration of the project and then assess the effectiveness of the project plan.

The evidence from this task is observed through Progress Check 12 and assessed through Criterion 10.

**SAT Timeline**

|  |  |
| --- | --- |
| **Task** | **Progress Check Date** |
| Unit 3 Outcome 2 (SAT Part A) | |
| Criterion 1 | March 7  March 14 |
| Criterion 1: Brief Submission | March 17 |
| Criterion 2 | April 4  May 5 |
| Criterion 3 | May 19 |
| Criterion 4 | May 26 |
| Criterion 5 | June 9 |
| Unit 3 Outcome 2 Submission | June 13 |
| Unit 4 Outcome 1 (SAT Part B) | |
| Criterion 6 & 7 | July 4  July 25  August 15 |
| Criterion 8 | June 25  August 15 |
| Criterion 9 | August 29 |
| Criterion 10 | September 12 |
| Unit 4 Outcome 1 Submission | September 17 |

**Note:** Weekly interviews will start from Term 1 Week 9 (Tuesday 1st April) to help with the authentication of student work and monitoring progress.

**Unit 3 Outcome 2 SAT Final Submission Checklist**

Below is a list of all of the tasks/work/documents that need to be submitted for Part A of the SAT

**Criterion 1**

* MS Word – Design Brief
* The Project Plan for the entire SAT (Unit 3 Outcome 2 and Unit 4 Outcome 1)
* Include the following:
  + All PSM Stages and Activities
  + Relevant tasks that include sequencing, time allocations and dependencies:
  + Milestones (teacher and own)
  + Critical Path

**Criterion 2**

* Evidence of data collected
  + Video, Audio transcripts
  + Interview/survey questions and responses
* Analytical Tools
  + Context Diagram
  + Data Flow Diagram
  + Use Case Diagram

**Criterion 3**

* MS Word – SRS
  + Functional and Non-Functional Requirements
  + Scope and Constraints
  + Audience and Technical Environment
  + Appendix (Criteria 3 Evidence)
* Critical and Creative Thinking
  + Blog/Journal

**Criterion 4**

* MS Word – Design Ideas
  + Brainstorming
  + Mind Maps
  + Mood Boards
  + Sketches (minor annotations)
* MS Word – Evaluation Criteria
  + Design Ideas
  + Completed Software Solution
* MS Word – Design Ideas Evaluation Evidence and Justification

**Criterion 5**

* MS Word – Software Solution Design
  + Interface Designs (annotated), Data Dictionary, IPO Chart, Pseudocode, Object Description
* Critical and Creative Thinking
  + Blog/Journal

**Unit 3 Outcome 2 SAT Tasks**

Criterion 1: Skills indeveloping a Brief and in Project Management

**Indicator 1 – Writing a Brief**

You will describe and justify how your proposed solution addresses an identified problem, need, or opportunity. Your response should include the following details:

* **Identify the Problem, Need, or Opportunity**
  + State the issue your project aims to solve.
  + Explain why it is a problem, need, or opportunity.
* Describe How the Solution Addresses the Problem
  + Outline the key features of your solution.
  + Identify the users or clients who will benefit from it.
* Describe Relevant Features of the Programming Language
  + Explain why a specific programming language was chosen.
  + Mention features that make it suitable for the project.
* Justify Feasibility, Originality, and Effectiveness
  + Explain why the solution is practical and achievable.
  + Highlight what makes it innovative.
  + Justify how it effectively solves the identified problem.

**Indicator 2 – Stages and Activities on the PSM**

Start with the Problem-Solving Methodology (PSM) stages and activities as your key headings and use this document to help you fill in the blanks. Even some elements within this document will need further detail and break down.

**Indicator 3 – Tasks Identification**

Make a list of all the tasks that must be completed. Then, next to each task, try to estimate how long you think that task will take – this is the key challenge of managing projects. In this instance, there will be tasks required that you have never done before and therefore would not know how long they will take – you will need to use your best judgement to allocate time to these tasks.

Your final Project Plan should include the following:

* Tasks
* Sequencing
* Time Allocation
* Milestones
* Dependencies
* Critical Path

|  |  |
| --- | --- |
| **SAT Progress Check** | |
| SAT PC: Design Brief | March 7 (Fri) |
| SAT PC: Gantt chart 1st Draft | March 14 (Fri) |
| SAT Submission: Design Brief | March 17 (Mon 6pm) |

**Unit 3 Outcome 2 SAT Tasks**

Criterion 2: Skills in documenting the Analysis

**Indicator 1 – Data Collection**

Identify the most appropriate methods of collecting data and justify why that method is better than alternatives. You might:

* interview a range of stakeholders to get different perspectives
* observe people doing their current work to find inefficiencies
* collect current system documentation (including manuals, inputs and outputs)
* review similar products in the marketplace to see what functionality is common and unique to them
* survey current/potential users, clients, etc.

**Indicator 2 – Using Analytical Tools**

You are required to create three diagrams to show the Functional Requirements and how data will flow and be manipulated by the software solution. The diagrams that need to be created are:

* Use Case Diagram (UCD)
* Context Diagram
* Data Flow Diagram (DFD)

You will be marked on the use of features in each diagram and following the rules that each diagram must follow.

You will be marked on the accuracy of the three diagrams that you have created and how they relate to the data that you have collected in indicator 1.

|  |  |
| --- | --- |
| **SAT Progress Check** | |
| SAT PC: Data Collection | April 4 (Fri) |
| SAT PC: Analytical Tools | May 2 (Fri) |

**Unit 3 Outcome 2 SAT Tasks**

Criterion 3: Skills in documenting a Software Requirements Specification

**Indicator 1 – Functional and Non-Functional Requirements, Scope and Constraints**

Functional Requirements:

* describe the way your solution will manage data input, transformation, storage and output in the form of a context diagram and data-flow diagrams
* describe the interactions between your solution and its users and/or other systems in the form of a use case diagram or diagrams.

Non-Functional Requirements:

* describe the key attributes the solution should possess such as usability, reliability, portability, robustness, maintainability; be sure to link these to characteristics of the users/potential users themselves and/or the operating environment where appropriate.

Constraints imposed on the new solution or its development, such as:

* economic, legal, social, technical and usability factors
* vulnerability to security threats
* authentication and data protection
* accessibility.

Scope of the new solution:

* consideration of features that may not be feasible or essential to the implementation of the proposed solution and which may provide opportunities for further development.

Stakeholders of the new solution:

* identify any relevant organisations and their structure
* user characteristics of the eventual users.

Environment Characteristics:

* document the hardware and network requirements (using network diagrams where appropriate).

**Indicator 2 – Critical and Creative Thinking**

Critical Thinking - justifying any changes, comparing alternatives and documenting the reasoning.

Creative Thinking - how you overcame challenges along the way and adapted designs and perhaps even the evaluation criteria.

|  |  |
| --- | --- |
| **SAT Progress Check** | |
| SAT PC: SRS | May 16 (Fri) |

**Unit 3 Outcome 2 SAT Tasks**

Criterion 4: Skills in generating Design Ideas and developing Evaluation Criteria

**Indicator 1 – Generating Design Ideas**

You must produce two or three alternative design ideas for your solution. The use of the following ideation tools must be used:

* mood boards
* brainstorming
* mind maps
* sketches

The alternative design ideas should be “distinctive, feasible and original” and should take into account both the appearance and the functionality of your solution.

**Indicator 2 – Evaluation Criteria for Design Ideas and Final Software Solution**

Develop a set of criteria which can be used to measure the alternate designs, the preferred design can be chosen, and the final solution can be evaluated in terms of efficiency and effectiveness. You should consider the data collected from clients and/or potential users as to preferences and possibilities.

Each evaluation must measure the efficiency and effectiveness of the Design/Final Solution

Efficiency – Time, Effort and Cost

Effectiveness - Accessibility, Accuracy, Attractiveness, Clarity, Communication of Message, Completeness, Readability, Relevance, Timeliness and Usability

|  |  |
| --- | --- |
| **SAT Progress Check** | |
| SAT PC: Design Ideas and Evaluation Criteria | May 23 (Fri) |

**Unit 3 Outcome 2 SAT Tasks**

Criterion 5: Skills in producing detailed Designs

**Indicator 1 – Producing detailed Design for the Software Solution**

There is a range of design tools available for you to use including those listed below. Any comprehensive solution design will involve the use of multiple tools to accurately describe the solution.

You will need to create the following:

* Input, Process, Output (IPO) charts
* Algorithm representation (pseudocode) –written representation of the logical sequence of processing steps required for the solution. This should be used to describe the total solution, broken into logical sections or modules.
* Data dictionaries – list all the variables and data objects, their data type, purpose and scope.
* Object descriptions – similar to data dictionaries, this is used for object-oriented solutions that use classes and includes details of events and methods relevant to each object.

**Indicator 2 – Demonstrating Design Principles and User Experience in Detailed Designs**

You will annotate and provide written explanations to demonstrate how design principles and user experience (UX) characteristics have been applied in your detailed designs.

Review the key design principles and UX characteristics relevant to your project. These may include:

* Design Principles: Alignment, contrast, repetition, proximity, balance, and white space.
* User Experience (UX) Characteristics: Usability, accessibility, responsiveness, efficiency, and aesthetics.

Annotation details:

* Use Callouts or Labels: Add annotations (text boxes, arrows, or comments) to highlight specific design elements.
* Link to Design Principles and UX: Clearly state which principle or UX characteristic is applied in each part of your design.

Example Annotation:

*"The navigation menu is positioned at the top of the page to ensure consistency across all screens (Principle: Repetition)."*

**Indicator 3 – Critical and Creative Thinking**

Critical Thinking - justifying any changes, comparing alternatives and documenting the reasoning.

Creative Thinking - how you overcame challenges along the way and adapted designs and perhaps even the evaluation criteria.

|  |  |
| --- | --- |
| **SAT Progress Check** | |
| SAT PC: Software Solution Design | June 6 (Fri) |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **1. Skills in developing a brief and in project management.** | * Documents a problem, need or opportunity as a brief. | Insufficient evidence | Identifies a problem, need or opportunity. | Describes how the proposed solution aims to address the identified problem, need or opportunity.  Identifies the users or clients of the proposed solution. | Explains how the proposed solution aims to address the identified problem, need or opportunity.  Describes the users or clients of the proposed solution. | Describes the relevant features of the programming language to be used within the proposed solution. | Justifies how the development of the proposed solution will be feasible, original and address the identified problem, need or opportunity. |
| * Prepares a Gantt chart using software that documents all stages and activities from the problem-solving methodology for Unit 3 Outcome 2 and Unit 4 Outcome 1. | Prepares a plan using software that documents the stages from the problem-solving methodology. | Prepares a Gantt chart using software that documents the stages from the problem-solving methodology for Unit 3 Outcome 2. | Prepares a Gantt chart using software that documents the stages from the problem-solving methodology for Unit 4 Outcome 1. | Prepares a Gantt chart that documents the stages and activities from the problem-solving methodology for Unit 3 Outcome 2 and Unit 4 Outcome1. | Prepares a Gantt chart that clearly and accurately documents the all the stages and activities from the problem-solving methodology for Unit 3 Outcome 2 and Unit 4 Outcome1. |
| * Documents all the relevant tasks, sequencing, time allocations, milestones, dependencies and critical path. |  | Lists relevant tasks. | Outlines a plan that includes tasks and time allocations. | Documents the appropriate sequencing of tasks, time allocations and teacher-provided milestones. | Documents the appropriate sequencing of student-provided milestones.  Explains why the project needs to be monitored. | Documents the dependencies and the critical path.  Discusses how the progress of the project will be monitored and documented. |
|  | 0 q | 1 q 2 q | 3 q 4 q | 5 q 6 q | 7 q 8 q | 9 q 10 q |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **2. Skills in documenting the analysis.** | * Documents and prepares the data for analysis using appropriate data collection methods. | Insufficient evidence | Identifies data that is required to inform the analysis. | Collects data using one of the following data collection methods:   * interviews * observations * surveys * reports.   Outlines how the data will be collected to inform the analysis. | Collects data using two of the following data collection methods:   * interviews * observations * surveys * reports.   Describes how the data collected will be used to determine requirements, constraints and scope. | Collects data using three or more of the following data collection methods:   * interviews * observations * surveys * reports.   Describes how the data collected will be used to determine user characteristics and the technical environment. | Prepares the data for analysis by labelling and categorising the data.  Explains the use of the selected data collection methods. |
| * Uses relevant features of the selected analytical tools and illustrates the relationships between users, data and systems. |  | Identifies features of the selected analytical tools for illustrating. | Illustrates the features of the context diagram/s.  Illustrates the relationships between the existing system, entities and data flows.  Some errors or omissions exist. | Illustrates the features of the data flow diagram/s.  Illustrates the relationships between the processes, entities, data stores and data flows.  Some errors, inconsistencies or omissions exist. | Illustrates the features of the use case diagram/s.  Illustrates the relationships between the systems boundary, actors, associations, relationships (includes and extends) and use cases.  Minor errors, inconsistencies or omissions exist. | Illustrates correctly all the relevant features of the three selected analytical tools.  Illustrates correctly all the relationships between users, data and systems.  No errors, inconsistencies or omissions exist. |
|  | 0 q | 1 q 2 q | 3 q 4 q | 5 q 6 q | 7 q 8 q | 9 q 10 q |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **3. Skills in documenting  a software requirements specification.** | * Documents the proposed software solution as part of  an SRS. | Insufficient evidence | Lists solution requirements. | Outlines the purpose and requirements of the proposed software solution in a formal document. | Documents the functional and non-functional requirements of the proposed software solution.  Documents the constraints that may impact the development of the proposed solution.  Documents the analytical tools. | Describes the characteristics of the users for the proposed software solution.  Describes the scope of the proposed software solution. | Describes the technical environment in which the proposed software solution will operate.  Organises the formal document using clear headings, sections and appendices. |
| * Documents the process of critical and creative thinking through critical analysis, the use of questions and follow-up questions to clarify the development of the software requirements specification. | Identifies the data that needs to be collected to inform the development of the software requirements specification. | Outlines the use of questions to critically analyse the data collected to inform the development of the software requirements specification. | Write questions to critically analyse the development of the software requirements specification. | Evaluates questions to critically analyse the development of the software requirements specification. | Writes follow-up questions to clarify the data collected to inform the development of the software requirements specification. |
|  | 0 q | 1 q 2 q | 3 q 4 q | 5 q 6 q | 7 q 8 q | 9 q 10 q |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **4. Skills in generating design ideas and developing evaluation criteria.** | * Generates design ideas for the software solution. | Insufficient evidence | Lists design ideas for the software solution. | Illustrates design ideas for the software solution using one of the following ideation tools and techniques:   * mood boards * brainstorming * mind maps * sketches. | Generates design ideas for the software solution using two of the following ideation tools and techniques:   * mood boards * brainstorming * mind maps * sketches.   Uses annotations to identify the appearance of the software solution. | Generates design ideas for the software solution using three or more of the following ideation tools and techniques:   * mood boards * brainstorming * mind maps * sketches.   Uses annotations to explain the appearance and functionality of the software solution. | Generates feasible, original and distinctive design ideas using ideation tools and techniques that fully represent the software solution.  Uses annotations to describe and justify the appearance and functionality of the software solution. |
| * Develops evaluation criteria with reference to design ideas and the efficiency and effectiveness of the software solution. | Identifies measures to evaluate design ideas or the software solution. | Outlines criteria to evaluate the design ideas and the software solution, with reference to the functional and non-functional requirements. | Develops and applies evaluation criteria to determine which elements of the design ideas should be further developed into detailed designs. | Uses the evaluation criteria to explain which elements of the design ideas should be further developed into detailed designs. | Uses the evaluation criteria to justify which elements of the design ideas should be further developed into detailed designs. |
|  | 0 q | 1 q 2 q | 3 q 4 q | 5 q 6 q | 7 q 8 q | 9 q 10 q |
| **VCE Software Development: School-assessed Task 2025** | | | | | | | |
| **Assessment Criteria** | **Levels of Performance** | | | | | | |
| **Indicators** | **Not shown** | **1–2 (very low)** | **3–4 (low)** | **5–6 (medium)** | **7–8 (high)** | **9–10 (very high)** |
| **Unit 3 Outcome 2**  **5. Skills in producing detailed designs.** | * Produces detailed designs for the software solution. | Insufficient evidence | Uses a mock-up to represent the user interface for the software solution. | Uses annotated mock-ups to represent the user interfaces within the software solution.  Uses a data dictionary with reference to data types. | Uses a detailed data dictionary, with reference to data types and data structures.  Uses an IPO chart and pseudocode to represent the functionality of the software solution.  Applies the relevant features of the design tools. | Uses a detailed data dictionary, with reference to data types, data structures and data sources.  Uses IPO charts and multiple modules of pseudocode to represent the functionality of the software solution.  Applies the relevant features of the design tools correctly. | Uses object descriptions and pseudocode to represent the functions and methods used within objects and classes.  Designs are feasible and complete. |
| * Documents the use of design principles and the characteristics of user experience in the detailed designs. | List design principles and characteristics of user experience that have been considered as part of the detailed designs. | Uses brief annotations to identify how the design principles have been applied within the detailed designs. | Uses annotations or written descriptions to describe how design principles have been applied within the detailed designs, with reference to appearance and functionality. | Uses annotations and written descriptions to document how relevant characteristics of user experience have been applied within the detailed designs. | Uses annotations and written explanations to demonstrate how the selected design principles and characteristics of user experience have been applied within the detailed designs. |
| * Documents the process of critical and creative thinking through the development of design ideas and the detailed designs. | Identifies existing and possible solutions to inform design ideas. | Outlines the connections between the design ideas using annotations. | Documents the connections between the design ideas and solution requirements. | Documents the connections between the design ideas, solution requirements and the detailed designs. | Documents possible contingencies when developing solution designs.  Documents possible solutions to mitigate issues. |
|  | 0 q | 1 q 2 q | 3 q 4 q | 5 q 6 q | 7 q 8 q | 9 q 10 q |