# Digital Solutions 2019 v1.2

## Unit 1 assessment instrument

### Project - digital solution

#### Unit objectives

This assessment instrument is used to determine student achievement of the following Unit objectives:

1. recognise and describe programming elements and useability principles

2. symbolise and explain information, ideas and interrelationships related to programming problems

3. analyse problems and information related to a selected technology context

4. determine user-experience and programming requirements, and self-determined and prescribed criteria of a programming problem

5. synthesise information and ideas to determine possible digital solutions

6. generate user interface and programmed components of the prototype digital solution

7. evaluate impacts, components and solutions against criteria to make refinements and justified recommendations

8. make decisions about and use mode-appropriate features, language and conventions for particular purposes and contexts.

*The assessment objectives used in the ISMG below have been contextualised to reflect these unit objectives.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject** | Digital Solutions | | |
| **Technique** | Project – digital solution | | |
| **Unit** | Unit 1: Creating with code | | |
| **Topics** | Topic 1: Understanding digital problems  Topic 2: User experiences and interfaces  Topic 3: Algorithms and programming techniques  Topic 4: Programmed solution | | |
| **Conditions** | | | |
| **Duration** | Up to 8 weeks | | |
| **Mode** | Multimodal | **Length** | * 8-10 A3 pages * 2-4 minute demonstration of the functionality of the user interface and coded components of the digital solution by video recording * 4-6 A4 pages of code with annotations |
| **Individual / group** | Individual | **Other** | * The reference list is not included in the page count. * Schools implement authentication strategies that reflect QCAA guidelines. |
| **Resources available** | * Computers * Internet * Stimulus (technical proposal) | | |
| **Context** | | | |
| Lost property in schools leads to item wastage, and increased costs for parents and students in purchasing replacements. Recently schools have encouraged students to “tag” their items (either with permanent marker or engraved) with their name or student ID, however this approach has not been widely successful, as it is not the owner intention for an item to be lost, and as such not every item brought to school is tagged. As schools grow, lost property becomes an increasing issue, both in terms of cost and storage space. | | | |
| **Task** | | | |
| You must document the problem-solving process used to develop and generate the user interface and programmed components of a prototype for a new lost property management application. The new web application must address the requirements of the scenario in the stimulus provided. Demonstrate the functionality of the components of the prototype web application in a video recording. | | | |
| **To complete this task, you must:** | | | |
| * recognise and describe   + programmed and user-interface components   + useability principles including accessibility, effectiveness, safety, utility and learnability * symbolise   + the user and developer problem using mind maps and one or more constructed sketches, annotated diagrams, images or screenshots   + algorithms communicated in pseudocode that demonstrate knowledge and understanding of programming features   + interrelationships between user experiences and programming in the prototype web application * explain   + essential features of the problem   + code samples and screenshots from the prototype web application with annotations   + the prototype web application from a user-experience perspective, communicated through a collection of annotated images of the user-interface components   + how programming elements and user-interface components connect, communicated in an annotated diagram   + functionality, useability and efficiency of the coded components, communicated through code comments and annotations on the 4–6 A4 pages * analyse the prototype web application and information to identify   + user interface and programmed components and their relationships to the structure of the prototype web application   + the prototype web application’s personal and social impacts * determine   + constraints and limitations   + existing code and user-interface solutions   + solution requirements that include essential elements and features of the user interface based on useability principles   + prescribed and self-determined criteria * synthesise ideas and information about solutions for   + user interfaces and programmed components of the prototype web application, for example annotated diagrams identifying and describing proposed components of the prototype web application * generate   + sample code for the digital prototype on the 4–6 A4 pages, demonstrating     - selection, iteration, user input     - data structures to manage and manipulate data, both server-side and client-side     - efficiency, robustness, eloquence and technical sophistication of coding ability     - consistent standards and explicit commenting throughout all source files   + a user interface and programmed components of a prototype for a web application * evaluateagainst criteria relating to   + personal, social and economic impacts supported by a collection of data samples or representations   + accuracy and efficiency of the coded components supported by a collection of annotated code segments in tables, diagrams and written paragraphs identifying errors and actions   + the prototype web application from a user-experience perspective supported by a collection of annotated images of the provided user interface components * make refinements and justified recommendations for current and future improvements. | | | |
| **Stimulus** | | | |
| See Stimulus material at end of document | | | |
| **Checkpoints** | | | |
| □ Term 1 Week 10: Submit exploration of solutions, identification of algorithms and user interface sketches | | | |
| □ Term 2 Week 3: Complete draft submission | | | |
| □ Term 2 Week 5: Final submission | | | |

|  |  |  |
| --- | --- | --- |
| **Criterion** | **Marks allocated** | **Result** |
| **Retrieving and comprehending**  Assessment objectives 1, 2 | 8 |  |
| **Analysing**  Assessment objectives 3, 4 | 8 |  |
| **Synthesising and evaluating**  Assessment objectives 5, 6, 7 | 10 |  |
| **Communicating**  Assessment objective 8 | 4 |  |
| **Total** | 30 |  |
| **Authentication strategies** | | |
| * Students will provide documentation of their progress at indicated checkpoints. | | |
| * Students must acknowledge all sources. | | |
| * Students must submit a declaration of authenticity. | | |
| * The teacher will collect copies of the student response and monitor at key junctures. | | |
| * The teacher will conduct interviews or consultations with each student as they develop the response. | | |
| **Scaffolding** | | |
| Your response must include:   * headings that organise and communicate the iterative phases of the Digital Solutions problem-solving process * A3 pages that   + demonstrate all phases of the problem-solving process   + communicate knowledge and understanding by way of annotated sketches, diagrams, images or screenshots * a video   + in mp4 file format   + no larger than 200 MB   + demonstrating the functionality of the prototype web application’s user interface and coded components * A4 pages of code with annotations of analysis, synthesis and evaluation related to the code element or problem * referencing of sources, following the school’s referencing style * written and visual features, as well as grammatically accurate language conventions, to communicate decision-making | | |

# Stimulus

|  |  |
| --- | --- |
| **Identification** | |
| **Your role** | Local, independent, web application developer, contracted by the school to prototype the lost property management web application. |
| **Product proposed** | Lost Property – a “*lost property management solution via a web interface*” |
| **Server IP** | * Production: 127.0.0.1:5000 (local) * Live: 192.168.X.X (LAN address, yet to be determined) |
| **Potential *interactions* with the proposed product** | |  |  | | --- | --- | |  | **Name**: Gladys  **Job**: Student  **Requirements**: Gladys is in year 12 and has lost her car keys. She drove her friends to school, then put her keys down for volleyball training. Gladys has a laptop but doesn’t have time to check the lost property office with her studies.  **Primary concern**: Privacy and getting back her missing car keys. | |  | **Name**: Allister  **Job**: Lost Property Manager  **Requirements**: Needs to log items when they are handed in to lost property. Enters the required data and uploads a photo (optional). Can delete items when handed back to owners.  **Primary concerns**: Ensuring accurate descriptions and details of items are posted. | |  | **Name**: Leoni  **Job**: Bursar  **Requirements**: Keen to stress test the system to determine limits and acceptance. Also, keen to ensure liability, accessibility and legal agreements for the college are met in any new digital product introduced to the student body.  **Primary concerns**: Minimising risks of the web application, including looking at ways to mitigate irresponsible, unethical or illicit practises. | |
| **Specifications** | |
| **Data** | **No database is required** for this application. Generated data (such as items) can be managed via appropriate **hard coded data structures** (such as lists, dictionaries or tuples) |
| **Algorithms and Code** | All planned functionality involving computational logic for your lost property web application must be written out in plain *pseudocode* first.  At minimum, your web application should enable the following functionality across two separate sections or pages:   * Allow students to browse all items currently available in lost property * Allow the lost property manager to add a new lost property item to the items currently being displayed |
| **Extended functionality** | In addition to the core functionality, it may be ideal if some of the following functionality could be enabled in your web application. This is not an exhaustive list, but potential ideas that could extend the value of your web application:   * For lost property manager:   + Manipulate items listed:     - Rename or edit details of lost item     - Remove item from display   + Upload picture of lost item   + Gather meta data on items     - Count of all items currently lost * For students:   + Filter items by category   + Search items by keyword   + Sort items, e.g. by date added or category * For security:   + Allow a secure login for offsite access (via a *session variable*)   Other aesthetic ideas are welcome, such as client-side form validation using JavaScript, or CSS techniques to declutter the screen (such as grouping like functions together). The UX should be encouraging, with care taken to ensure the site is kept simple, clear and minimal. |
| **Interface** | * A responsive web template is provided for this task. Ensuring a quality UX using this template is left as a job for the developer. * Modifications to the template are welcome provided:   + Intuitiveness (e.g. breadcrumb navigation) and accessibility (e.g. tooltips, alt text, contrasting colours) are guiding principles.   + Modifications conform to the W3 standards. A good benchmarking tool for this is the Queensland Government Consistent User Experience Standard, available here: <https://www.forgov.qld.gov.au/cue> |

# Instrument-specific marking guide

Criterion: Retrieving and comprehending

### Assessment objectives

1. recognise and describe programming elements, user interface components and useability principles

2. symbolise and explain programming information and ideas, code structures and interrelationships between user experiences and coding of the digital prototype

|  |  |
| --- | --- |
| **The student work has the following characteristics:** | **Marks** |
| * accurate and discriminating recognition and discerning description of relevant programming elements, user-interface components and useability principles * adept symbolisation and discerning explanation of algorithms and relevant programming information and ideas, code structures and interrelationships between user experiences and coding of the digital prototype. | 7-8 |
| * accurate recognition and effective description of relevant programming elements, user-interface components and useability principles * methodical symbolisation and effective explanation of algorithms and relevant programming information and ideas, code structures and interrelationships between user experiences and coding of the digital prototype. | 5-6 |
| * appropriate recognition and description of some programming elements, user-interface components and useability principles * competent symbolisation and appropriate explanation of algorithms and some information and ideas, and interrelationships between user experiences and coding of the digital prototype | 3-4 |
| * variable recognition and superficial description of programming elements, user-interface components or useability principles * variable symbolisation and superficial explanation of information, ideas or interrelationships. | 1-2 |
| * does not satisfy any of the descriptors above. | 0 |

Criterion: Analysing

### Assessment objectives

1. analyse the problem and information related to the stimulus for a low-fidelity prototype digital solution

2. determine user interface, programmed and solution requirements of the digital solution and prescribed and self-determined criteria

|  |  |
| --- | --- |
| **The student work has the following characteristics:** | **Marks** |
| * insightful analysis of the problem and relevant contextual information to identify the essential elements and features of user interface and programmed components and their relationships to the structure of the low-fidelity prototype digital solution * astute determination of the user interface, programmed and solution requirements of the digital solution and essential prescribed and self-determined criteria. | 7-8 |
| * considered analysis of the problem and relevant contextual information to identify the relevant elements and features of user interface and programmed components and their relationships to the structure of the low-fidelity prototype digital solution * logical determination of the user interface, programmed and solution requirements of the digital solution and effective prescribed and self-determined criteria. | 5-6 |
| * appropriate analysis of the problem and contextual information to identify some elements and features of user interface and programmed components and their relationships to the structure of the low-fidelity prototype digital solution * reasonable determination of the user interface, programmed and solution requirements of the digital solution and some prescribed and self-determined criteria. | 3-4 |
| * superficial analysis of the problem or partial information to identify aspects of elements or features of the low-fidelity prototype digital solution * vague determination of some solution requirements of the digital solution and some criteria. | 1-2 |
| * does not satisfy any of the descriptors above. | 0 |

Criterion: Synthesising and evaluating

### Assessment objectives

1. synthesise information and ideas to determine user interface and programmed components for a digital solution

2. generate user interfaces and programmed components of the digital solution

3. evaluate impacts, components and the digital solution against prescribed and self-determined criteria to make refinements and justified recommendations

|  |  |
| --- | --- |
| **The student work has the following characteristics:** | **Marks** |
| * coherent and logical synthesis of relevant information and ideas to determine user interface and programmed components for a digital solution * purposeful generation of efficient user interface and programmed components of the digital solution * critical evaluation of impacts, user experience and coded components and the digital solution against essential prescribed and self-determined criteria to make discerning refinements and astute recommendations justified by data. | 9-10 |
| * logical synthesis of relevant information and ideas to determine user interface and programmed components for a digital solution * effective generation of user interface and programmed components of the digital solution * reasoned evaluation of impacts, user experience and coded components and the digital solution against effective prescribed and self-determined criteria to make effective refinements and considered recommendations justified by data. | 7-8 |
| * simple synthesis of information and ideas to determine user interface and programmed components for a digital solution * adequate generation of user interface and programmed components of the digital solution * feasible evaluation of impacts, user experience and coded components and the digital solution against some prescribed and self-determined criteria to make adequate refinements and fundamental recommendations justified by data. | 5-6 |
| * rudimentary synthesis of partial information or ideas to determine user interface or programmed components * partial generation of user interface and programmed components of the digital solution * superficial evaluation of impacts, user experience components or the solution against some criteria. | 3-4 |
| * unclear combination of information, ideas or solution components * identification of a change to an idea or a solution. | 1-2 |
| * does not satisfy any of the descriptors above. | 0 |

Criterion: Communicating

### Assessment objectives

1. make decisions about and use mode-appropriate features, written language and conventions for a technical audience

|  |  |
| --- | --- |
| **The student work has the following characteristics:** | **Marks** |
| * discerning decision-making about, and fluent use of   + written and visual features to communicate about a solution   + language for a technical audience   + grammatically accurate language structures   + referencing and project conventions. | 3-4 |
| * variable decision-making about, and inconsistent use of   + written and visual features   + suitable language   + grammar and language structures   + referencing or project conventions. | 1-2 |
| * does not satisfy any of the descriptors above. | 0 |

|  |  |
| --- | --- |
| **Marks** | **LOA** |
| **30** | **A+** |
| **29** |
| **28** | **A** |
| **27** |
| **26** | **A-** |
| **25** |
| **24** | **B+** |
| **23** |
| **22** | **B** |
| **21** |
| **20** | **B-** |
| **19** |
| **18** | **C+** |
| **17** |
| **16** | **C** |
| **15** |
| **14** | **C-** |
| **13** |

ISMG to LOA  
Note: Your grade will be awarded holistically. The “marks-to-grade” ratios shown below are a guide, and not to be taken as a determinant of final award:

|  |  |
| --- | --- |
| **Marks** | **LOA** |
| **12** | **D+** |
| **11** |
| **10** | **D** |
| **9** |
| **8** | **D-** |
| **7** |
| **6** | **E+** |
| **5** |
| **4** | **E** |
| **3** |
| **2** | **E-** |
| **1** |