# Part 1 – Research and investigation

* Read the task sheet – specifically Project Conventions for this task. It’s different to the other ones. Pro tip: There are exactly 3 parts and **at least 2 separate mind maps**. There are other differences too – such as ongoing evaluations. **Task specifications and ISMGs don’t match well in IA3**. You have to do both.
* This document has been set up to meet the QCAA page length and page size requirements. **Make sure you remove all of these scaffold notes after use**.
* Start researching theory first, this is Unit 4 so it is new content and on your exam: <https://digisoln.com/digital.html>, specifically slideshows: **21 REST**, **22 ENCRYPTION** AND **27 OTHER THEORY**
* Read the whole task sheet and scaffold before starting – some evaluations the come last require criteria to be set early on about security strategies, etc.

|  |  |  |
| --- | --- | --- |
| AO1: **Recognise** and **describe**: | Evidence your research **retrieval** and **comprehension** of: |  |
| i. Key elements of an application | Required elements of your web application:   * Stimulus / technical proposal requirements / required functionality – use a **Mind Map** (yawn) and / or text box annotations to complement the mind map * The QCAA sample uses a single GUI design for this. Unbelieveable, but true. Confirmers will likely be looking for it, since everyone is probably going to follow suit. **Better draw up a few GUIs** (sigh). **Not as many as IA1 or IA2**, and don’t annotate with useability or design principles this time. Annotate with explanations of the actual parts or components of the GUI instead – what they are for, how they work, event triggers, state changes, etc. | Page 1 – **mind map**  Page 2 – **GUIs**  Tag both with:  “**Recognition and Description of Key Elements of an Application**” |
| ii. Components of data exchange systems | **Full stack** **diagram** recognising and describing all hardware and software components involved in this ‘distributed’ system, showing exposed endpoints or live feeds, with annotations (aka. tech stack) | Page 1 or 2 – full stack diagram – feel free to label this:  “**Recognition and Description of Components of Data Exchange Systems**” |
| AO3: **Analysis** of data exchange problem – *relate* all of the information in both **mindmap**, **GUIs** and **tech stack** to the **current** problem and context |
| iii. Data security processes | This is pure research, which may include a summary of:   * encryption (via decryption key)   + transport layer encryption (~~SSL~~ >> TLS) – *more on the actual HTTP protocol later* * authorization (via signature, code or token)   + authorization vs authentication – difference? * checksums (checking for data corruption) * hashing (checking for malicious interference) | Page 1 or 2 (wherever there is room) – in a floating text field titled:    “**Recognition and Description of Data Security Processes**” |
| AO3: **Analysis** – *relate* all **relevent** information of data security presented here to the **current** problem and context |

Recognition and Description of Key Elements of an Application – User Interface

|  |  |  |
| --- | --- | --- |
| AO2: **Symbolise** and **explain**: | Evidence your research **retrieval** and **comprehension** of: |  |
| i. Data interface | "In computing, an interface is a shared boundary across which two or more separate components of a computer system exchange information." There are multiple interfaces in the OSI model, within code modules, and within data exchange.  Given that this ISMG specifically refers to **data interface**, I’d suggest:   * explain what **Really Simple Syndication** is (and use the Heritage Register RSS to help explain). * explain what an **Application Programming Interface** is (and use the weeds API to help explain) – especially its **uniform URL interface** (*refer to one of the REST principles below of uniformity*) * Difference between RSS and API | Page 3: In a floating text field titled:    “**Explanation of Data Interface**” |
| ii. Structures and specifications | **XML vs JSON** – explain both format notation, specification or syntax. Use snippets of the **two** data sources to help explain – and arrows, to ‘point out’ the elements | Page 3: In a floating text field titled:    “**Explanation of Structures and Specifications**” |
| iii. Data flow relationships within and between systems | Draw a **DFD level 0** and then refine to **level 1** to illustrate data flow throughout entire system | Page 3: In a floating text field titled:    “**Symbolisation and Explanation of Data flow Relationships Within and Between Systems** ” |
| iv. Digital methods of exchanging data | Other topics that could be of use in demonstrating “retrieving and comprehending” to ***symbolise and******explain data interface, or digital methods of exchanging data*** include:   * A summary of HTTP data exchange – such as: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview> * Explaination of RESTful principles – and how the API is STATELESS, which how this affects the **method** of exchanging data * API vs SOAP * A very brief retrieval and comprehension of **other / alternative** **digital methods of exchanging data**, including but not limited to:   + File transfer (FTP), or another protocol mentioned underneath, and how this is different from HTTP   + SMTP, P2P, RPC, UDP and streamed packets – see the OSI model for more info * There is scope here to mention Python libraries and their **methods** available to implement the exchange:   + Requests library for HTTP exchange – <https://docs.python-requests.org/en/master/>   + Element Tree for XML     - Methods - <https://docs.python.org/3/library/xml.etree.elementtree.html>   + JSON. 🡪 <https://docs.python.org/3/library/json.html>     - dump() 🡪 Dictionary to JSON file     - dumps() 🡪 Dictionary to JSON string     - load() 🡪 JSON file to Dictionary (deserialise)     - loads() 🡪 JSON string to Dictionary | Page 3: In a floating text field titled:    “**Explanation of Digital Methods of Exchanging Data** ” |

|  |  |  |
| --- | --- | --- |
| AO3: **Analyse**: |  |  |
| Data exchange problem | Analysis of stimulus / technical proposal. |  |
| Relevent information related to data security | Anaylsis of information – see above |  |
| to **identify**: |  |  |
| i. Data structures | Analysis, according to QCAA, is “*dissect to ascertain and examine constituent parts*”. I’d suggest:  Draw up a **data spec** **table** showing all key / elements available within the **two** data sources, a sample value / text, attribute if any (XML), data type (especially for JSON), and 100% any **additional comments or annotations** in your own language (e.g. ‘*this element can be used to identify a green event’*). If there is hierachical (nested) data for any key / element, ensure you reflect this relationship in your table or diagram (e.g. indentation or colour key to show child elements). | Page 4  Tag: “**Analysis of problem and information to identify data structures**” |
| ii. Data exchange methods | Explicitly Identify the two data sources, providers, and any other caveats, legal, usage requirements, etc. From this, identify the **data exchange methods** to be used:   * Communcation protocol (e.g. HTTP / HTTPS over TCP / IP – OSI layer?). * Architectural pattern (i.e. API / RSS) * Data format (i.e. JSON / XML)   Now pick out the relevant **Digital methods of exchanging data from AO2** and **Data security processes from AO1** and make decisions as to what might be **used** or **useful** in this solving this particular data exchange problem (and state why). Obviously, mention anything else of use in AO1 or AO2 (Retreival and comprehension) here. | Page 4  Tag: “**Analysis of problem and information to identify data exchange methods**” |
| iii. Risks to data | Security risks:   * confidentiality (e.g. unauthorized or unauthenticated access)   + consequences? * integrity (e.g. malicious corruption) * malicious interference to cause an availability risk   Technical risks:   * latency, jitter, QoS guarantee and timeliness   There are many risks, especially to this **localised** solution. What others can you can think of? *Mitigation strategies*? | Page 4  Tag: “**Analysis of problem and information to identify risks to data**” |
| iv. Coded components | Use the **first A4 code page below** to identify core code implementation of the base level functionality **described so far**.  You must show code refinements later on, so don’t delete any **code** that is over written from here. | |

|  |  |  |
| --- | --- | --- |
| AO4: **Determine**: |  |  |
| i. Data exchange requirements | Determine data exchange requirements for the weeds API or Heritage Register RSS web feed:   * URLs, URI segments, parameters or any **filters** * Keys, tokens or signatures required (probably none, but specify this) * Handling the expected response – suggest you map expected response format to any data structure(s) (“deserialisation”) identified in the target format in AO5 synthesis * Required code libraries or modules * Refer to any previous analysis, retrieval and comprehension work, that affects these requirements, particularly:   + Stateless communication   + Data structures / source format notation or syntax, and how this affects parsing requirements | Page 5  Tag: “**Determination of Data Exchange Requirements**” |
| ii. Security strategy for data | There is a base level of security already implemented in your system, when endpoints use TLS over HTTP, or require OAuth tokens or API keys. Explicitly identify these as “determined as part of the overall security strategy”. Additional measures in your security plan could include:   * Delivery of web application over HTTPS * Error handling for timeouts * Turn off debug messages on live systems * For event creation of weed removal events (read the stimulus), you’ll be acquiring user input. This is an entry point for malicious code, so consider:   + Santisation of input   + Authentication or authorization mechanism * Data integrity checks * **Acutally explain how you are going to implement these** | Page 5  Tag: “**Determination of Security Strategy for Data**” |
| iii. Code for the data conversion program | Modular code snippet(s). Not all of it – just the data transfer component(s). | 1 x A4 portrait page of code below |
| iv. Prescribed and self-determined criteria | **Prescribed** and **self-determined** criteria – list (you’ve done this before in IA1 and IA2), avoid using the QCAA task specifciations as criteria.  *The confusing part of IA3 is that on the task sheet you are asked to “****evaluate the most suitable process for exporting and importing data between the digital systems******against prescribed and self-determined criteria”****. Yep. Though this doesn’t “match” to ISMG well, best to include a small part of evaluation here to meet project conventions. Basically make this one of your criteria, and evaluate it “on-the-spot”.*  *Also, keep checking the task sheet for anything else I’ve missed..* | **Page 5**  Tag: “**Determination of Prescribed and Self-determined criteria**” |

A4 page for sample code snippets for **core** server-side code mechanisms (not the completed Web Application – e.g.: no Jinja) to illustrate ISMG "astute determination of code for the data conversion program". This might include:

* Use of a request module and endpoint URL, for URL request for a resource.
* Code for programmatic transformation of interchange formats, e.g.:
  + JSON to Dictionary
  + XML to Tree
  + CSV to Table
* Code to parse resulting formats to extract data.
* Sanitization or error handling
* From the start – **annotate refinements made to code** and keep any 'first attempts' so you can point out your **refinements to code**. This is a new, different evaluation criterion to IA2 or IA1 you will discover later.
* There are more pages later if this all does not fit. You can also use any left over room on the A3 sheets above.

# Part 2 – Data Exchange Solution

* *On this first page, include* ***another mind map****. Once again the task sheet doesn’t match well to ISMG, but a mind map is required here.*
* *Again, the task sheet mentions ‘data flow relationships within and between systems’ for symbolisation, which could mean taking a process from* ***Part 1 – Research and investigation*** *and magnifying it (e.g., into a part there-of Level 2 DFD with data flow arrows labelled).*
* *Part 2 also mentions “Determine Prescribed and Self-determined criteria”. Perhaps consider adding to or elaborating, considering what Part 2 asks you to evaluate (see below)*

|  |  |  |  |
| --- | --- | --- | --- |
| AO5: **Synthesis** of relevant information and ideas to determine: | | Suggest page use: | |
| i. Data components | There is overlap with earlier ISMG here, but they key here for synthesis is to be **exact** in what you need:   * Data storage **variables and structure** server side (e.g. object, list, tuple, set, dict, or combination there-of – such as “Element Tree”). Be exact. Use code snippets or samples to illustrate hierarchies, indices, etc. * Session data? User data? Logs? User inputs? Data to render (output)? There will be data that sits “outside” the interchange data that must be synthesised.     **OPTIONAL – YOU DO NOT HAVE TO DO THE FOLLOWING, THEY ARE JUST IDEAS:**   * If you are feeling really tricky, try a *UML Class Diagram* or *Data Structure Diagram* (fancy method of illustrating / symbolising the above server-side data structures), which might include:   + Variables, fields, data types, constraints, useage, etc.   + Methods, e.g. remove, create new, search / sort, etc. – only specify the *function calls* and / or *parameters* in any diagram as illustration of modularisation / encapsulation. Then refer to the algorithms below that do the “heavy lifting” (i.e. implementation of these method declarations)     - Method declarations may include computational transformation (i.e. deserialisation) of data from data interface:       * Data Interface >>       * Server >>       * ^^ Client (serialisation back to server – unlikely?, or server >> ?) * ERD for any third party storage / data recording (if so, include some SQL as per IA1 / IA2). **AN EXTERNAL DATA STORAGE (e.g. SQLITE DATABASE) IS NOT A REQUIREMENT** * Much of this mark will be mirrored in the **AO4 determination result** – there is a significant “overlap” here | page 6 and / or page 7  Use subheading “**Synthesis of Data Components**” | 3 A3 pages landscape total |
| ii. Algorithm components (these aren’t as big of weighting as IA2 and IA1, but still important) | Use pseudocode and text fields to write modular algorithms for:   * **Deserialisation and parsing of data sources – suggest two different algorithms, one for XML, one for JSON** * Other / extended functionality algorithms that operate on shared data structures (e.g. sort, search, manipulate, append or filter) * Rendering records client-side algorithms, event triggers from UI, anything else? * Implementation of any methods mentioned in data components * Keep to syllabus standards as always: <https://digisoln.com/computational_thought/QCAAmaterials.html> | page 7 and / or page 8  Use subheading “**Synthesis of Algorithm Components**” |
| iii. Coded components | *Once again, the task sheet includes an eval that you have to include but doesn’t match well for the ISMG:*   * *evaluate the*   + *accuracy of code after testing and identify errors and actions to make improvements*   + *digital data exchange solution against prescribed and self-determined criteria*   + *functionality, useability and efficiency of the components of the digital solution* * *make refinements and justified recommendations for current and future improvements* | Set of 3 x A4 portrait pages below. | |

Code paste and annotate page 1.

* Avoid pasting any unnecessary or repeated code.
* The word "**efficient"** appears in the generation criteria on the ISMG. Point out where or how your code is **efficient** with comments / annotations
* Any imported libraries not native to the default language install should be acknowledged and referenced. *Code should be commented* and / or annotated.
* Include both Python and Jinja2 code, and **modularise** your programming components (e.g., into application path routes for Python, or sections of content rendering for Jinja2).
* Annotate **refinements made to code** throughout to evidence ISMG: "critical evaluation of impacts, coded components and a data exchange solution against essential prescribed and self-determined criteria to make discerning **refinements** of code and astute recommendations justified by data".

Code paste and annotate page 2 (if needed)

Code paste and annotate page 3 (if needed)

Redact any code that doesn’t fit here and refer to it in writing or video demonstration.

# Part 3 – Impacts

Remaining pages, address ISMG criteria: “*Evaluation of impacts, coded components and a data exchange solution against prescribed and self-determined criteria to make refinements* ***of code*** *and recommendations justified by data.*” To do this:

* Recollect using screen shots, explanations of **code** development issues and resolutions, and test data from anecdotal or formative end user testing as data for justification of **code** refinements (past tense).
* Conduct quantitive (e.g. systems criteria – speed, viewport responsiveness, browser compatibility) and anecdotal (e.g. end user feedback) testing as data for justification of recommendations (future directions).
* Match refinements and recommendations (justified by data) wherever possible against prescribed and self-determined criteria.

|  |  |  |  |
| --- | --- | --- | --- |
| **Prescribed Criteria** | **Evaluation of:**   * **impacts** (**of data transmission on personal, social, and economic needs)** * **coded components** * **data exchange solution** | **Data for justification**   * **testing** * **qualitive / quantitive** * **screen shots??** | **Refinements made to code /**  **Recommendations for future** |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Self-Determined Criteria** | **Evaluation of:**   * **impacts** (including Personal, Social, Economic, *Ethical, Legal and Sustainability impacts)* * **coded components** * **data exchange solution** | **Business woman pointing upData for justification** | **Refinements made to code /**  **Recommendations for future** |
|  |  |  |  |
|  |  |  |  |

Generally criteria that succeeds will have refinements, while criteria that fails will have recommendations.

Refinements **to code**!

Once again there is parts of task sheet that don’t align well to ISMG for part 3. Make sure you’ve got it all covered! **Especially everything about Security Strategy**:

Graphical user interface, text, application

Description automatically generated

Continue with Evaluation

Finish with proper **academic referencing standards**, both a reference list and in-text referencing**.** You **must** have at least 1 reference.