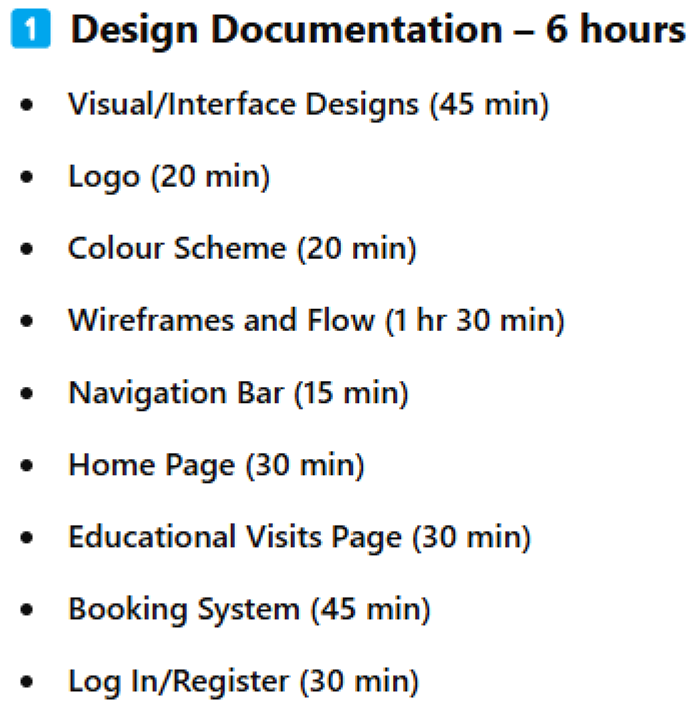
Task 1 A (B)

Visual Interface



1 Design Documentation – 6 hours

- Visual/Interface Designs (45 min)
- Logo (20 min)
- Colour Scheme (20 min)
- Wireframes and Flow (1 hr 30 min)
- Navigation Bar (15 min)
- Home Page (30 min)
- Educational Visits Page (30 min)
- Booking System (45 min)
- Log In/Register (30 min)

Wireframes (Low and High fidelity)

Main Tip: You will need to make two types of wireframes, low and high fidelity.

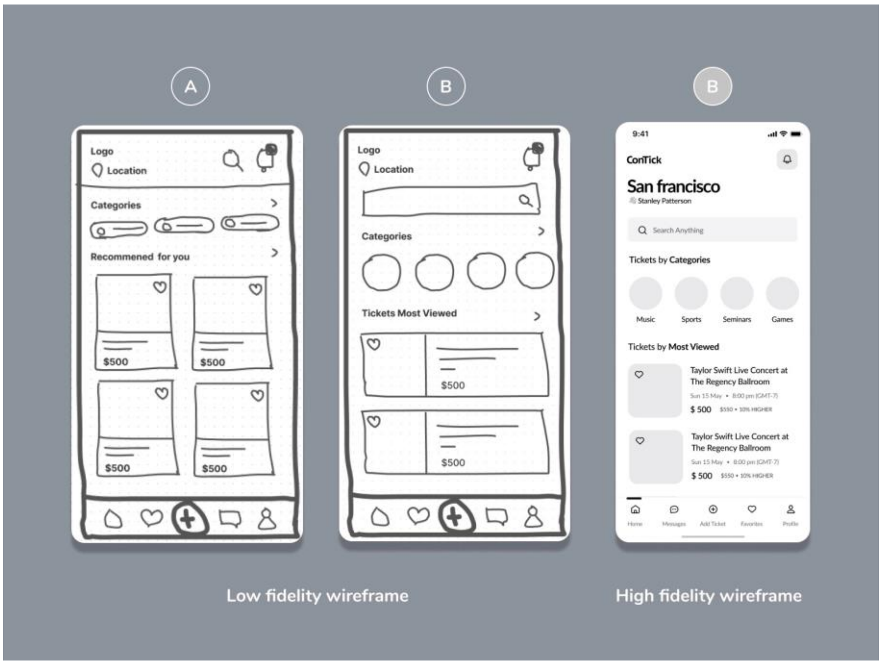
KEEP THE WIREFRAMES SIMPLE, the simpler you make these, the easier it will be to make your website.

Low fidelity has the least detail and a lot of placeholders. This can’t have colour. You shouldn’t spend too much time on these, do it as quick as you can.

High fidelity is more specific, a lot of the place holders will be filled in, usually the images are still a placeholder. High fidelity needs to have colour as well. You can spend more time on these, but don’t spend too long.

- Place holders for images is a box with a cross in it.
- Place holders for text is lines

Examples:



Sources to use:
Examiner reports:

Page Specification

Main Tip:
Examples:
Sources to use:
Examiner reports:

Colours

Main Tip: Use the contrast checker above, primary colours + secondary colours, colour combination, explain why you picked those colours etc.
Examples:
Sources to use:
Examiner reports:

Sources

Main Tip: Links to sources, to prove the information
Examples:

https://www.londonzoo.org/

https://www.chesterzoo.org/

https://www.marwell.org.uk/

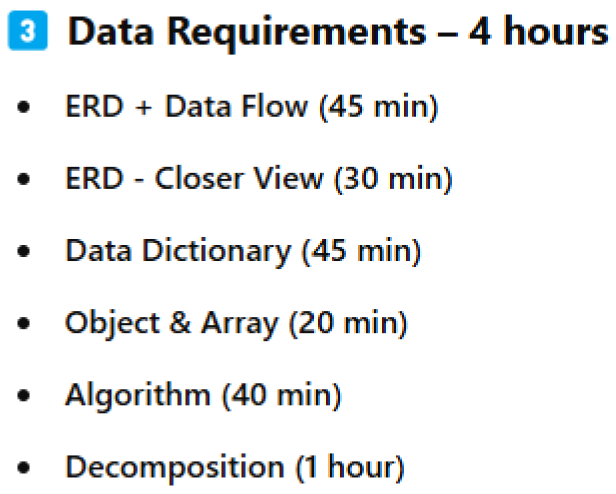
Sources to use:
Examiner reports:

Site Map

Main Tip: Add a map
Examples:
Sources to use:
Examiner reports:

Navigation Flow Diagram

Main Tip:
Examples:
Sources to use:
Examiner reports:



3 Data Requirements – 4 hours

- ERD + Data Flow (45 min)
- ERD - Closer View (30 min)
- Data Dictionary (45 min)
- Object & Array (20 min)
- Algorithm (40 min)
- Decomposition (1 hour)

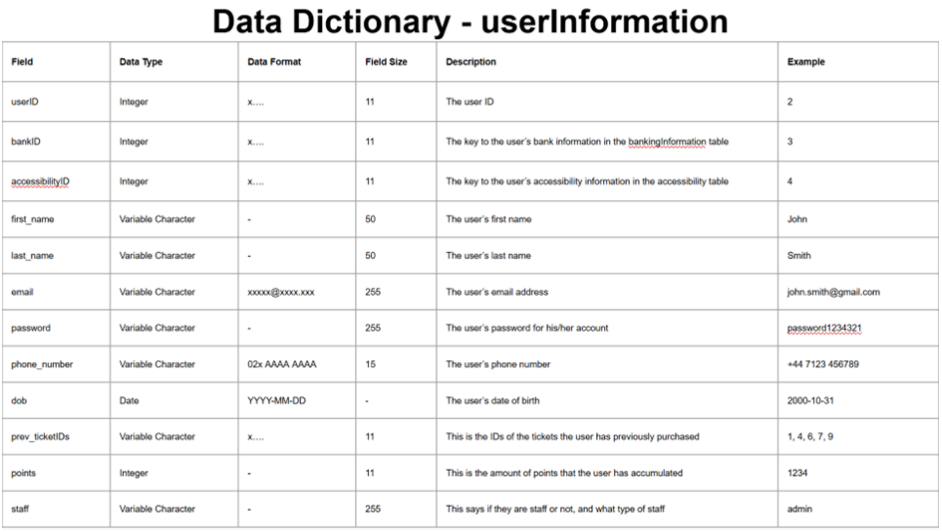
Data

Data Tables/Data Types

Main Tip:
Examples:
Sources to use:
Examiner reports:

Data Dictionary

Main Tip: Data dictionary to show the informaton



Data Dictionary - userInformation

Field	Data Type	Data Format	Field Size	Description	Example
userID	Integer	x....	11	The user ID	2
bankID	Integer	x....	11	The key to the user's bank information in the bankingInformation table	3
accessibilityID	Integer	x....	11	The key to the user's accessibility information in the accessibility table	4
first_name	Variable Character	-	50	The user's first name	John
last_name	Variable Character	-	50	The user's last name	Smith
email	Variable Character	xxxxx@xxxx.xxx	255	The user's email address	john.smith@gmail.com
password	Variable Character	-	255	The user's password for his/her account	password1234321
phone_number	Variable Character	02x AAAA AAAA	15	The user's phone number	+44 7123 456789
dob	Date	YYYY-MM-DD	-	The user's date of birth	2000-10-31
prev_ticketIDs	Variable Character	x....	11	This is the IDs of the tickets the user has previously purchased	1, 4, 6, 7, 9
points	Integer	-	11	This is the amount of points that the user has accumulated	1234
staff	Variable Character	-	255	This says if they are staff or not, and what type of staff	admin

Examples:

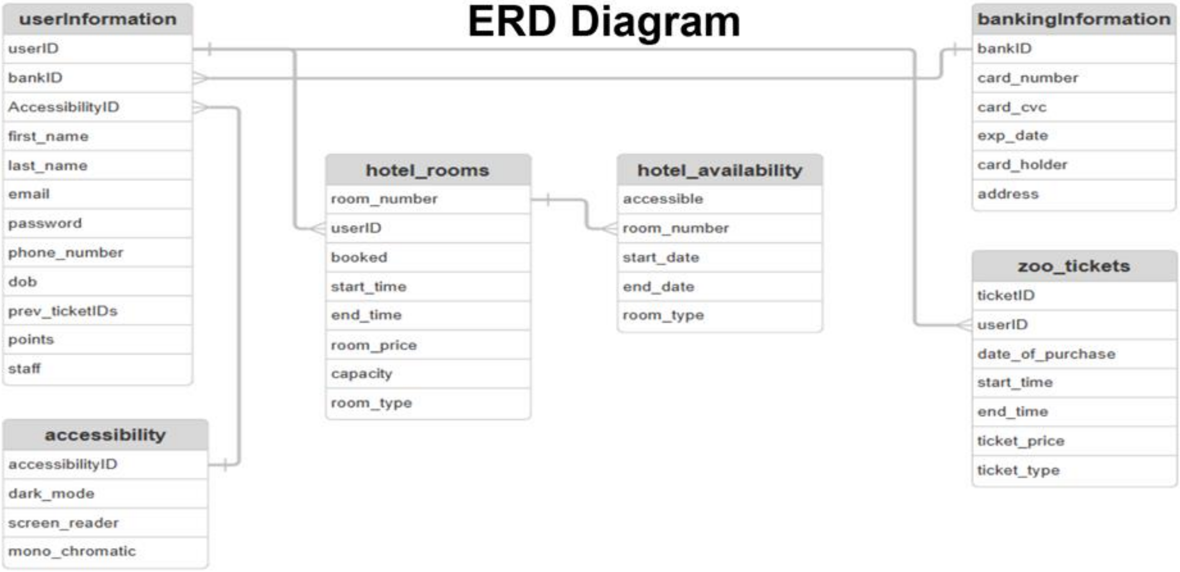
Sources to use:

Examiner reports:

ERD Diagram

Main Tip: To show how things work

Examples:



Sources to use:

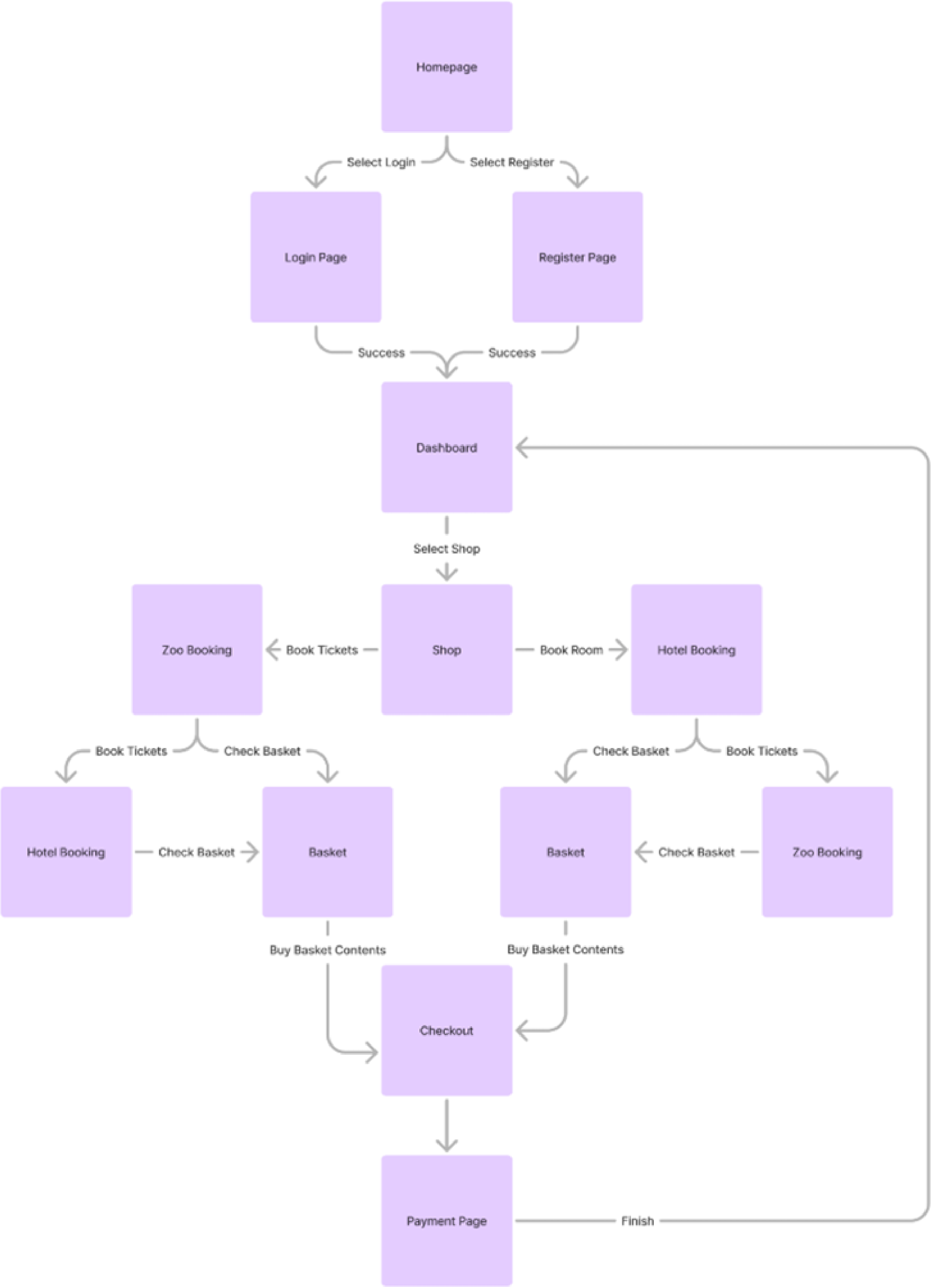
Examiner reports:

Data Flow Diagram

Main Tip:

Examples:

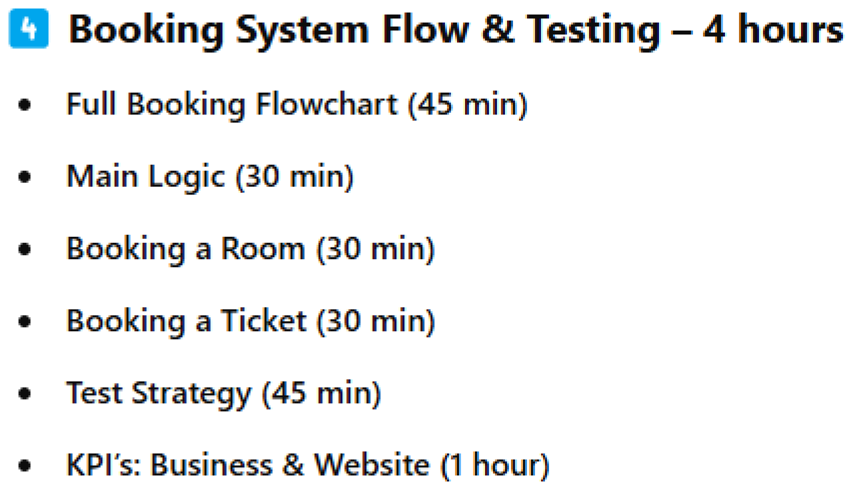
User Flow Diagram



Sources to use:

Examiner reports:

Algorithms



4 Booking System Flow & Testing – 4 hours

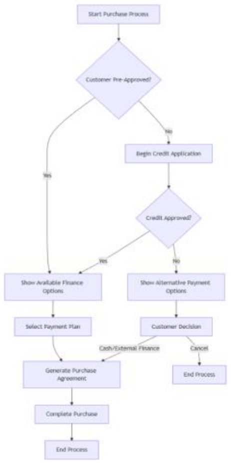
- Full Booking Flowchart (45 min)
- Main Logic (30 min)
- Booking a Room (30 min)
- Booking a Ticket (30 min)
- Test Strategy (45 min)
- KPI's: Business & Website (1 hour)

Flowcharts/Pseudocode/Natural Language

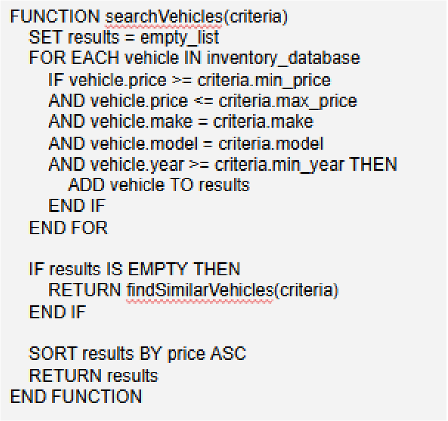
Main Tip: Pick whatever is more appropriate for your code.

This is to decompose you code, you just have to pick a way to decompose it and then put it on your document.

Examples:



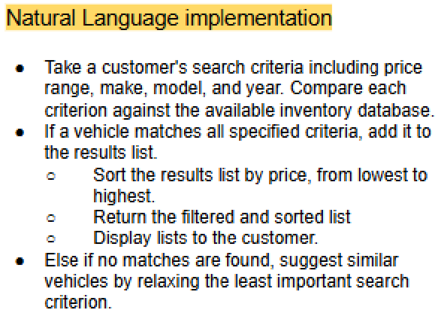
Pseudocode



```
FUNCTION searchVehicles(criteria)
    SET results = empty_list
    FOR EACH vehicle IN inventory_database
        IF vehicle.price >= criteria.min_price
        AND vehicle.price <= criteria.max_price
        AND vehicle.make = criteria.make
        AND vehicle.model = criteria.model
        AND vehicle.year >= criteria.min_year THEN
            ADD vehicle TO results
        END IF
    END FOR

    IF results IS EMPTY THEN
        RETURN findSimilarVehicles(criteria)
    END IF

    SORT results BY price ASC
    RETURN results
END FUNCTION
```



Natural Language implementation

- Take a customer's search criteria including price range, make, model, and year. Compare each criterion against the available inventory database.
- If a vehicle matches all specified criteria, add it to the results list.
  - Sort the results list by price, from lowest to highest.
  - Return the filtered and sorted list
  - Display lists to the customer.
- Else if no matches are found, suggest similar vehicles by relaxing the least important search criterion.

Sources to use:

Examiner reports:

Testing

Testing Strategy

Main Tip:

Examples:

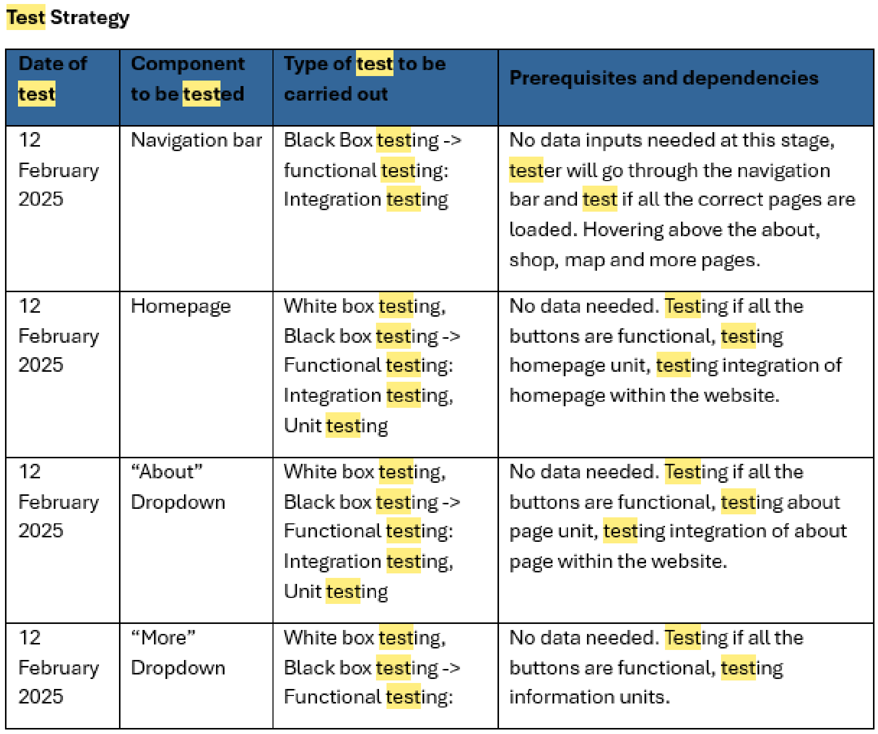
Sources to use:

Examiner reports:

Test Plan

Main Tip: Test strategy to show when testing was executed

Examples:



Test Strategy

Date of test	Component to be tested	Type of test to be carried out	Prerequisites and dependencies
12 February 2025	Navigation bar	Black Box testing -> functional testing: Integration testing	No data inputs needed at this stage, tester will go through the navigation bar and test if all the correct pages are loaded. Hovering above the about, shop, map and more pages.
12 February 2025	Homepage	White box testing, Black box testing -> Functional testing: Integration testing, Unit testing	No data needed. Testing if all the buttons are functional, testing homepage unit, testing integration of homepage within the website.
12 February 2025	"About" Dropdown	White box testing, Black box testing -> Functional testing: Integration testing, Unit testing	No data needed. Testing if all the buttons are functional, testing about page unit, testing integration of about page within the website.
12 February 2025	"More" Dropdown	White box testing, Black box testing -> Functional testing:	No data needed. Testing if all the buttons are functional, testing information units.

Sources to use:

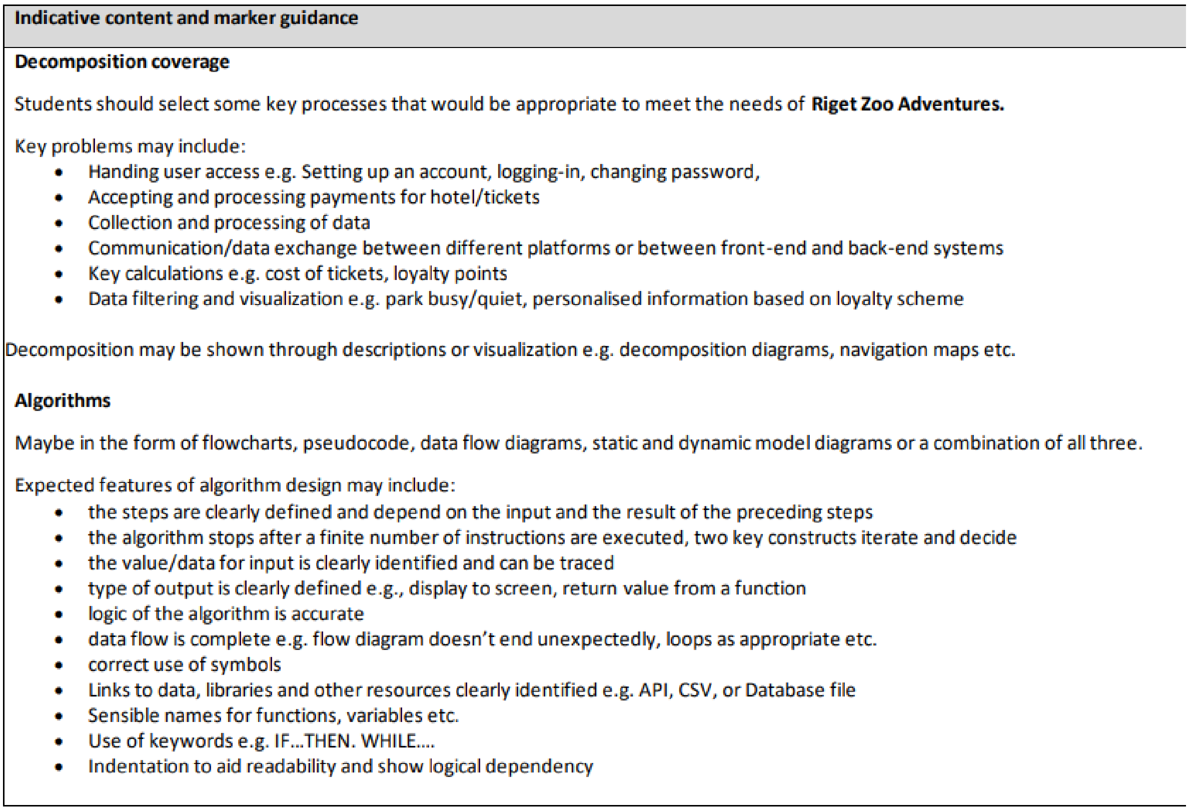
Examiner reports:

7 Final Review – 1 hour

- Check completeness & refine

Examiners Report:

Task 1 Activity B – The Design –Algorithm Design



Indicative content and marker guidance

Decomposition coverage

Students should select some key processes that would be appropriate to meet the needs of **Riget Zoo Adventures.**

Key problems may include:

- Handing user access e.g. Setting up an account, logging-in, changing password,
- Accepting and processing payments for hotel/tickets
- Collection and processing of data
- Communication/data exchange between different platforms or between front-end and back-end systems
- Key calculations e.g. cost of tickets, loyalty points
- Data filtering and visualization e.g. park busy/quiet, personalised information based on loyalty scheme

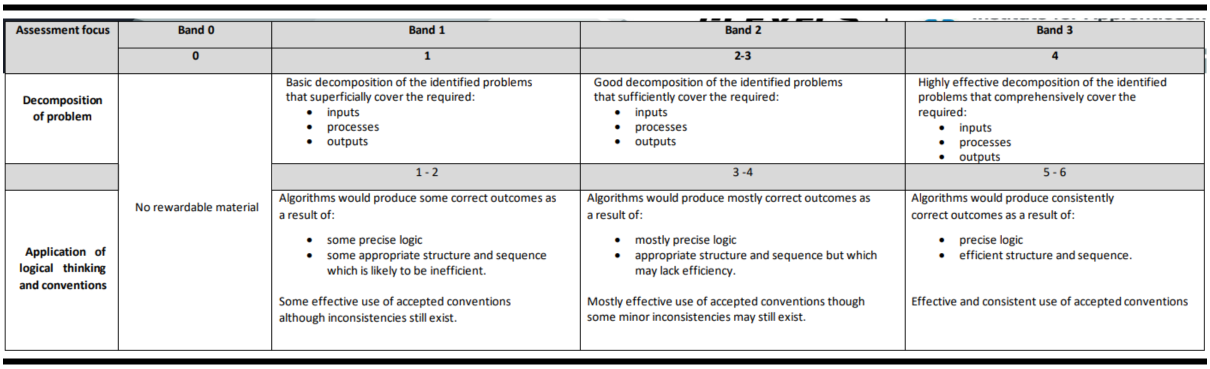
Decomposition may be shown through descriptions or visualization e.g. decomposition diagrams, navigation maps etc.

Algorithms

Maybe in the form of flowcharts, pseudocode, data flow diagrams, static and dynamic model diagrams or a combination of all three.

Expected features of algorithm design may include:

- the steps are clearly defined and depend on the input and the result of the preceding steps
- the algorithm stops after a finite number of instructions are executed, two key constructs iterate and decide
- the value/data for input is clearly identified and can be traced
- type of output is clearly defined e.g., display to screen, return value from a function
- logic of the algorithm is accurate
- data flow is complete e.g. flow diagram doesn't end unexpectedly, loops as appropriate etc.
- correct use of symbols
- Links to data, libraries and other resources clearly identified e.g. API, CSV, or Database file
- Sensible names for functions, variables etc.
- Use of keywords e.g. IF...THEN. WHILE....
- Indentation to aid readability and show logical dependency



Assessment focus	Band 0	Band 1	Band 2	Band 3
	0	1	2-3	4
Decomposition of problem	No rewardable material	Basic decomposition of the identified problems that superficially cover the required: • inputs • processes • outputs	Good decomposition of the identified problems that sufficiently cover the required: • inputs • processes • outputs	Highly effective decomposition of the identified problems that comprehensively cover the required: • inputs • processes • outputs
		1 - 2	3 -4	5 - 6
Application of logical thinking and conventions		Algorithms would produce some correct outcomes as a result of: • some precise logic • some appropriate structure and sequence which is likely to be inefficient. Some effective use of accepted conventions although inconsistencies still exist.	Algorithms would produce mostly correct outcomes as a result of: • mostly precise logic • appropriate structure and sequence but which may lack efficiency. Mostly effective use of accepted conventions though some minor inconsistencies may still exist.	Algorithms would produce consistently correct outcomes as a result of: • precise logic • efficient structure and sequence. Effective and consistent use of accepted conventions

Task 1 Activity B – The Design – The Data Requirements

Indicative content and marker guidance

Data design may be in the form of:

- data dictionaries,
- entity-relationship diagrams,
- data flow diagrams,
- static and dynamic model diagrams or a combination as appropriate to describe the planned solution

Note – data normalisation may not be required depending on the identified/proposed solution.

Data considered should be appropriate for the needs of **Riget Zoo Adventures** which may include:

- User profiles/accounts
- Hotel availability
- Ticket sales
- Educational materials – e.g. quizes and learning exercises (keeping scores), personalisation based on age

The design should show an understanding of error handling procedures which may include:

- Data validation rules
- Input masks
- Type casting
- Feedback to user/error messages
- Limiting field/variable length

Assessment Focus	**Band 0**	**Band 1**	**Band 2**	**Band 3**
	0	**1-2**	**3-4**	**5-6**
The design of the data requirements	No rewardable material	Data requirements for the proposed solution are somewhat appropriate, including (as required): • variables • data structures • data types Naming conventions used are mostly appropriate but are inconsistent. Effective error handling procedures are identified for some inputs/processes that require them.	Data requirements for the proposed solution are mostly appropriate, including (as required): • variables • data structures • data types Naming conventions used are appropriate and mostly consistent. Effective error handling procedures are identified for most inputs/processes that require them.	Data requirements for the proposed solution are fully appropriate, including (as required): • variables • data structures • data types Thoroughly appropriate and consistent naming conventions are used throughout. Thoroughly effective error handling procedures are identified for the inputs/processes that require them.

Task 1 Activity B – The Design – The Test Strategy

Indicative content and marker guidance

Students produce a strategy of how they will test the for a digital solution that they intend to develop.

The test schedule should cover:

- understanding of how components interrelate. This may include:
  - description of modules/functions that feed data into the component being tested
  - understanding of how correct/incorrect data form one part may impact on the component being tested
  - transition between unit testing and integration testing
  - Functional, non-functional and front-end testing
- the order in which components should be tested: This may cover:
  - identification of pre-requisites and dependencies
- the types of tests/testing techniques that are required to test the student's solution such as:
  - Acceptance testing.
  - Alpha testing.
  - Beta testing.
  - Black box testing.
  - White box testing/structural testing

Assessment focus	**Band 0**	**Band 1**	**Band 2**	**Band 3**
	0	**1-2**	**3-4**	**5-6**
Test strategy	No rewardable material	The test strategy demonstrates a basic understanding of: • how components interrelate • the order in which components should be tested • the types of tests that are required.	The test strategy demonstrates a good understanding of: • how components interrelate • the order in which components should be tested • the types of tests that are required	The test strategy demonstrates a thorough and detailed understanding of: • how components interrelate • the order in which components should be tested • the types of tests that are required.

Task 1 Activity B – The Design – The Design Documentation

Indicative content and marker guidance

Appropriateness of communication of the **design**:

- suitability for the intended audience
- clarity
- use of technical language
- choice of tools / how information is presented.

		1-2	**3-4**	**5-6**
Quality of communication	No rewardable material	Some effective communication of the design as a result of: • some use of appropriate techniques, methods and formats • some use of technical language that is appropriate for the intended audience	Mostly effective communication of the design as a result of: • the use of mostly appropriate techniques, methods and formats • the use of technical language that is mostly appropriate for the intended audience	Communication of the design is consistently effective as a result of: • the use of consistently appropriate techniques, methods and formats • the use of technical language that is consistently appropriate for the intended audience