**Project type: Software design and development with database design and development (object-oriented programming)**

**Project title:**

*<insert your project title here>*

Contents

[Analysis of the problem (10 marks) 2](#_Toc206580510)

[Description of the problem 2](#_Toc206580511)

[UML use case diagram 3](#_Toc206580512)

[Requirements specification 4](#_Toc206580513)

[Project plan 6](#_Toc206580514)

[Design of the solution (20 marks) 7](#_Toc206580515)

[Design of Advanced Higher concepts 7](#_Toc206580516)

[Design of integration 8](#_Toc206580517)

[Design of additional functional requirements 9](#_Toc206580518)

[User interface design 10](#_Toc206580519)

[Implementation (30 marks) 11](#_Toc206580520)

[Log of ongoing testing 12](#_Toc206580521)

[Advanced Higher concepts 13](#_Toc206580522)

[Implementation of integration 14](#_Toc206580523)

[Implementation of additional requirements 15](#_Toc206580524)

[User interface implementation 16](#_Toc206580525)

[Research and development of new skills and/or knowledge 17](#_Toc206580526)

[Testing the solution (15 marks) 19](#_Toc206580527)

[Persona and test cases 19](#_Toc206580528)

[Comprehensive test plan 20](#_Toc206580529)

[Evidence of requirements testing 22](#_Toc206580530)

[Evaluation (5 marks) 23](#_Toc206580531)

[Fitness for purpose 23](#_Toc206580532)

[Future maintainability 23](#_Toc206580533)

[Robustness 23](#_Toc206580534)

[Appendix 24](#_Toc206580535)

# 

# Analysis of the problem (10 marks)

Before you begin designing and developing a solution, you must analyse the problem that you are going to solve, to ensure that you fully understand every aspect of it. This stage of your project should take around 5 hours.

## **Description of the problem**

**(2 marks)**

**Outline of the problem**

Your outline should identify the Software design and development Advanced Higher concepts and the integration with a database that will be used in your project.

Refer to page 16 of the Coursework Assessment Task to help with this.

(150-300 words)

|  |
| --- |
|  |

**Constraints**

Describe any constraints for your project.

(100 words)

|  |
| --- |
|  |

## UML use case diagram

(2 marks)

Draw a UML use case diagram for your problem. Your diagram should define the following:

* actors
* use cases
* relationships

Your diagram should include the integration with the database that you intend to implement.

<insert evidence here>

## Requirements specification

**(4 marks)**

**End user requirements**

Identify a maximum of six end-user requirements for your project.

|  |  |
| --- | --- |
|  | The end-users of the solution should be able to: |
| EU1 |  |
| EU2 |  |
| EU3 |  |
| EU4 |  |
| EU5 |  |
| EU6 |  |

**Functional requirements**

**Advanced Higher concepts**

Complete the pre-populated functional requirements for Advanced Higher concepts to describe how they will be applied in your project.

At least one class (with properties and methods) must be designed and implemented by you. This class must not be generated by frameworks or software plug-ins.

|  |  |
| --- | --- |
|  | The solution is required to: |
| FR1 | Have an object-oriented solution with a developer defined class to <insert text> |
| FR2 | Use an array of objects to <insert text> |
| FR3 | Use a bubble sort/insertion sort/binary search algorithm (delete as appropriate) |
| FR4 | Apply the algorithm in FR3 to the data structure in FR2 to <insert text> |

**Integration**

Complete the pre-populated functional requirements for integration to describe how they will be applied in your project.

|  |  |
| --- | --- |
|  | The solution is required to: |
| FR5 | Have a database table to store <insert text> |
| FR6 | Connect to the database to execute a query to <insert text> |
| FR7 | Generate an interface to receive query input values and display formatted query output |

**Additional functional requirements**

Identify between four and eight additional functional requirements, a minimum of two of these must relate to validation of keyboard input.

This could include SQL queries in addition to the one required for integration.

A single functional requirement must relate to a single function the program will carry out. This should not include refinements of a functional requirement with multiple bullet points.

|  |  |
| --- | --- |
|  | The solution is required to… |
| FR8 |  |
| FR9 |  |
| FR10 |  |
| FR11 |  |
| … |  |

## Project plan

**(2 marks)**

Create a project plan for the four remaining stages of your project.

Your project plan should include:

* the tasks you complete in each stage
* any resources you need to implement your solution
* an estimate of how long each stage and tasks will take

Guidance for producing a project plan

Your list of tasks for each stage could include:

* user-interface design
* implementation of input validation
* ongoing testing

Resources can include access to specialist software and development tools. Some of these could be available at any time, while others may only be available at certain times. You need to plan to ensure that your project is not held up waiting for resources.

Your timings should allow for holidays, or other events that affect how much time you can spend on your project.

You should review and update your project plan as you work through each stage.

Insert the **final version** of your project plan including timings and list any resources needed to implement your solution.

<insert evidence here>

# Design of the solution (20 marks)

The design stage of your project should take around 10 hours, including time to revisit and adapt as the project progresses.

**Design of Advanced Higher concepts**

**(6 marks)**

Use structure diagrams or pseudocode to design of the Advanced Higher concepts identified during your analysis.

**Design for FR1 – UML class diagram indicating the class you have defined**

<insert evidence here>

**Design for FR2 – Design of array of objects**

<insert evidence here>

**Design for FR3 and FR4 – Design of refinement of standard algorithm applied to array of objects in FR2**

<insert evidence here>

## Design of integration

**(5 marks)**

Design the integration identified during your analysis.

**Design of FR5 – data dictionary and entity relationship diagram (if more than one table)**

<insert evidence here>

**Design of FR6 –**

**Design of connection to the database and design of execution of query (using pseudocode)**

<insert evidence here>

**Design of query**

<insert evidence here>

**Design of FR7 – this evidence should be shown in user-interface design**

## Design of additional functional requirements

**(4 marks)**

Design the additional functional requirements (maximum of 8), including validation of keyboard input, identified during your analysis.

**Design of FR8**

<insert evidence here>

**Design of FR9**

<insert evidence here>

**Design of FR10**

<insert evidence here>

**Design of FR11**

<insert evidence here>

**…..**

## User interface design

**(5 marks)**

Design of the user interface for your solution.

Your user interface design should be annotated wireframes showing:

* Input forms showing validation of keyboard input
* Buttons and menu options showing underlying processes
* Output screens including error messages resulting from input validation

Your design should refer to relevant end-user and/or functional requirements.

<insert evidence here>

# Implementation (30 marks)

The implementation stage of your project should take around 15 hours, including time to revisit and adapt as the project progresses.

Your implemented code should meet the functional requirements identified at the analysis stage and match you completed design.

The full code for your project must be added to your submission as an appendix at the end.

## Log of ongoing testing

**(5 marks)**

As you implement your solution, you will encounter errors or problems that you need to solve before you can continue. Take notes of errors, solutions and any reference materials you use, for example websites, forums, textbooks or learning resources. You will need to refer to these notes to produce evidence of ongoing testing.

Complete the log of ongoing testing below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Describe how functional requirement was tested** | **Describe any issues encountered during testing** | **Describe how you resolved the issue noting any references** |
| FR1 |  |  |  |
| FR2 |  |  |  |
| FR3 and FR4 |  |  |  |
| FR5 |  |  |  |
| FR6 |  |  |  |
| FR7 |  |  |  |
| FR8 |  |  |  |
| FR9 |  |  |  |
| … |  |  |  |

## Advanced Higher concepts

**(12 marks)**

Provide evidence of your implemented Advanced Higher concepts.

This evidence will include code and, where appropriate, ‘before’ and ‘after’ screenshots to show that the code works correctly.

**Implementation of FR1- implementation of the class, including properties and methods.**

<insert evidence here>

**Implementation of FR2 - implementation of array of objects.**

<insert evidence here>

**Implementation of FR3 and FR4 – implementation of standard algorithm applied to array of objects in FR2.**

<insert evidence here>

## Implementation of integration

**(6 marks)**

Provide evidence of your implemented integration.

This evidence will include code and, where appropriate, ‘before’ and ‘after’ screenshots to show that the code works correctly.

**Implementation of FR5 – implementation of database structure and initial values in table(s).**

<insert evidence here>

**Implementation of FR6 – implementation of connection to the database and execution of query.**

<insert evidence here>

**Implementation of FR7 – this evidence should be shown in implementation of the user-interface**

## Implementation of additional requirements

Provide evidence of your implemented additional functional requirements (maximum 8).

This evidence will include code and, where appropriate, ‘before’ and ‘after’ screenshots to show that the code works correctly.

**Implementation of FR8**

<insert evidence here>

**Implementation of FR9**

<insert evidence here>

**Implementation of FR10**

<insert evidence here>

**Implementation of FR11**

<insert evidence here>

**…**

## User interface implementation

**(3 marks)**

Insert screenshot evidence of the implemented user interface including:

* input screens
* output showing results of processing
* gameplay (if appropriate)

This evidence should be clearly labelled and should refer to relevant end-user and/or functional requirements.

<insert evidence here>

## Research and development of new skills and/or knowledge

**(4 marks)**

When implementing your functional requirements, you may need to make use of some coding that extends beyond the content of the Advanced Higher course. If this is the case, you will need to carry out some research.

Complete the table below to describe:

* any new skills and/or knowledge that you researched
* why those new skills and/or knowledge were necessary
* how you applied these new skills and/or knowledge to your project

You should reference the resources you used to research and develop these new skills and/or knowledge

|  |  |  |
| --- | --- | --- |
| **New skill/knowledge** |  |  |
| **Functional Requirement(s) requiring new skill/knowledge** |  |  |
| **Why this new skill/knowledge was necessary** |  |  |
| **How you applied this new skill/knowledge to your project** |  |  |
| **Resources used to research and develop this skill/knowledge (if required)** |  |  |

# Testing the solution (15 marks)

Once you have fully implemented your design, you must carry out final testing on your solution. This testing should be systematic and comprehensive, and based on a test plan.

This stage of your project should take around 8 hours.

## Persona and test cases

**(3 marks)**

Describe the characteristics of one persona that you will adopt to test the end-user requirements.

|  |
| --- |
|  |

Describe the test cases that you will use to test each end-user requirement. Indicate the end-user requirement(s) that will be the focus of each test case.

| **Test case** | **Description each test case** | **End-user requirement(s) to be tested** |
| --- | --- | --- |
| Case 1 |  |  |
| Case 2 |  |  |
| Case 3 |  |  |
| Case 4 |  |  |
| … |  |  |
| … |  |  |

Test the end-user requirements by carrying out the test cases above.

Describe the results of testing using the persona and test cases described above.

(150-300 words)

|  |
| --- |
|  |

## Comprehensive test plan

**(6 marks)**

Plan how you will carry out final testing of all functional requirements by completing the tables below.

Your plan should be comprehensive, to fully test each functional requirement identified at the analysis stage.

**Test plan for final testing – functional requirements (AH concepts)**

| **Test Number** | **Functional requirement to be tested** | **How it will be tested (including values if required)** | **Evidence required** |
| --- | --- | --- | --- |
| Test 1 | FR1 |  |  |
| Test 2 | FR2 |  |  |
| Test 3 | FR3 and FR4 |  |  |

**Test plan for final testing – functional requirements (integration)**

| **Test Number** | **Functional requirement to be tested** | **How it will be tested (including values if required)** | **Evidence required** |
| --- | --- | --- | --- |
| Test 4 | FR5 |  |  |
| Test 5 | FR6 |  |  |
| Test 6 | FR7 |  |  |

**Test plan for final testing – additional functional requirements**

| **Test Number** | **Functional requirement to be tested** | **How it will be tested (including values if required)** | **Evidence required** |
| --- | --- | --- | --- |
| Test 7 | FR8 |  |  |
| Test 8 | FR9 |  |  |
| … |  |  |  |

## Evidence of requirements testing

**(6 marks)**

Test the functional requirements by carrying out the test plan above.

For each test, provide screenshot evidence of your testing.

| **Test Number** | **Screenshot evidence** |
| --- | --- |
| Test 1 |  |
| Test 2 |  |
| Test 3 |  |
| Test 4 |  |
| Test 5 |  |
| **…** |  |

# 

# Evaluation (5 marks)

The evaluation stage of your project should take around 2 hours.

## Fitness for purpose

**(3 marks)**

Describe how your solution matches all requirements in your requirements specification. (400 to 600 words)

|  |
| --- |
|  |

Describe the results of your testing. You should reflect on what happened during final testing of functional requirements, discuss what went well and identify any tests that didn’t work as expected. (200-300 words)

|  |
| --- |
|  |

## Future maintainability

**(1 mark)**

Describe how your code allows for future maintenance. (150-250 words)

|  |
| --- |
|  |

## Robustness

**(1 mark)**

Describe the robustness of your solution. (150-250 words)

|  |
| --- |
|  |

# Appendix

<Insert your full program code here>