External Documentation

Created by

**UPTOWN IT**

For

**<<CUSTOMER>>**

**Student Name:** << Name >>

**Student Number:** << Student Number >>

**PROJECT REFERENCE:**

**DATE:**

Contents

[1 Input process output (IPO) chart 2](#_Toc83252066)

[2 Task object event (TOE) chart 2](#_Toc83252067)

[3 Stylised data flow diagram (DFD) 3](#_Toc83252068)

[4 Selected pseudo code algorithms 3](#_Toc83252069)

[4.1 Algorithm name 3](#_Toc83252070)

[5 Structured charts 3](#_Toc83252071)

[5.1 Managing the GUI elements 3](#_Toc83252072)

[5.2 Managing the user actions 4](#_Toc83252073)

[6 Class diagram/s 4](#_Toc83252074)

[7. Debugging and testing 4](#_Toc83252075)

[7.1 Testing Plan 4](#_Toc83252076)

[7.2 Test cases 4](#_Toc83252077)

[7.3 Evidence of debugging 5](#_Toc83252078)

[8 System generated program documentation 5](#_Toc83252079)

[9 User Manual 5](#_Toc83252080)

[10 Evidence of Correspondence 5](#_Toc83252081)

Remove all blue text once your report is completed. The blue text is intended at giving you some pointers for the report content.

# 1 Input process output (IPO) chart

Provide an introduction.

Describe how the business processes information. Write the descriptions in plain English – not coding required at this stage.

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | PROCESS | OUTPUT | STORAGE |
| E.g. Save button | E.g. Save all the entries in RAM Memory into the specified output file. | E.g. Export of all records (entries). | E.g. All entries saved in a new copy of the data file. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Add rows s required |  |  |  |

# 2 Task object event (TOE) chart

Provide an introduction.

Identify the tasks. For each task identify the objects that the task requires and the event/s.

|  |  |  |
| --- | --- | --- |
| TASK | OBJECT | EVENT |
| E.g. Clear screen ready for new entry | E.g. New button | E.g. Click |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Add rows as required. |  |  |

# 3 Stylised data flow diagram (DFD)

Prepare a simple data flow diagram (DFD) to map the flow of information for any process or system.

There are a number of symbols and notations used to create DFD select the one that is appropriate for the project and do not mix the notation of different DFD approaches.

# 4 Selected pseudo code algorithms

In this section you are expected to provide algorithms not code. Create algorithms for all project’s functionality. The number of algorithms will depend on the project requirements.

Make sure that your algorithms have the following characteristics:

* Precision
* Uniqueness
* Finiteness

Do not mix different pseudo code styles.

Create sub-headings for each algorithm

## 4.1 Algorithm name

Create sub-headings for each algorithm – 4.2, 4.3 etc.

**Sample algorithm**

**WriteFile**

Open the data file for writing data

For all entries in the arrays (For count = 1 to

NumberOfRecords)

Write a line of data into the data file

End For

Close the data file

# 5 Structured charts

Structured chart display the hierarchical structure of modules and their connections to other modules. Structured charts represent the software architecture of the system to a manageable low level.

Make sure that you clearly identify the modules, sub-modules, libraries, data flow, control flow and loops or repetitive calls.

## 5.1 Managing the GUI elements

Create a structured chart for the GUI elements

## 5.2 Managing the user actions

Create a structured chart for the user actions

# 6 Class diagram/s

Class diagrams are the building block of object-oriented modelling.

Use a UML diagramming tool such as Visual Paradigm or similar to create the diagrams.

# 7. Debugging and testing

## 7.1 Testing Plan

The approach to testing that will be used is as follows:

1. Review the project requirements
2. Prepare a solution design
3. Develop the required application.
4. **Unit testing** – test each component (method / class) as it is being developed
5. **Integrated Testing** - Prepare a set of data (test cases) to check how the application will behave when appropriate data is read in and entered via the screen.
6. Prepare several sets of data (test cases) to check how the application will behave when data that will generate exceptions is read in from the data file.
7. Enter several sets of data (test cases) to check how the application will behave when data that will generate exceptions is entered via the screen.
8. **System Testing** – Deploy the application to a computer system/network/web host site similar to that of the client’s and retest selected test cases.
9. **Acceptance Testing** - Deploy the application to the client’s computer system/network/web host site and look to the client to retest using sample live data.
10. Archive the test plan, test cases and results to an appropriate backup location.

Complete the section below.

**Test Objectives:** What is your testing aiming to achieve?

**Test Strategy**: How do you aim to complete your testing?

**Resource Requirements:** What personnel, computers, software, etc. will you need to complete your testing?

**Roles and Responsibilities:** Who will be involved and what will be their respective roles?

**Bug Reporting:** How will you be providing the results of your testing to your manager and/or the development team?

## 7.2 Test cases

Use the template provided to record the results of the tests and the test cases. File name: ICTPRG443\_Test\_Cases\_Report\_Template.xlsx.

## 7.3 Evidence of debugging

Provide screenshots of different instances of debugging. Make sure that you include examples of:

1. a breakpoint
2. a set of associated watches
3. an instance of tracing through several lines of code

# 8 System generated program documentation

Prepare automated program documentation using a facility provided within your IDE, such as Javadoc, or using a third-party product such as ESS Model.

Ensure this automated documentation is provided within your submission.

# 9 User Manual

Prepare a mostly graphical user manual for the system.

# 10 Evidence of Correspondence

Maintain appropriate ongoing communication with your manager and client.

This can be done with a series of applicable emails and meeting minutes *(see following pages)*.

1. Email your manager before the commencement of this project to secure the project specification and the organisation’s programming and documentation standards.
2. Present the meeting minutes for a meeting you have held with your project manager at the start of the project in which you have review the project requirements
3. Prepare one of more emails to your manager requesting clarification on one or more aspects of the project, as the project proceeds.
4. Prepare an email to your manager and separately to your client regarding the inclusion and utilisation of a Third Party Library.
5. Prepare a closing email to your manager regarding the success of your project development endeavours. You should suggest any areas of concern or risk to your manager for inclusion in the development team’s risk matrix.
6. Prepare a closing email for your client regarding the completion of the program coding, with an outline of the deployment process, plus plans for initial client staff training.

To: Project Manager / Manager

From:

CC:

BCC:

Date: 19/07/2027

Subject: Perfect Policies Quiz - Project Requirements

-------------------------------------------------------------

Hi <Project Manager>,

Based on the specification provided, the requirements for the Perfect Policies Quiz Project include:

1.

2.

3.

Regards,

<Lead Developer>

----------------------------------------------

< Lead Developer >

Lead Developer@UptownIT.com.au

-----------------------------------------------

Meeting Minutes

Project: Perfect Policies Quiz

Meeting Name: Project requirements clarification

Date:

Time:

Attendees:

-------------------------------------------------------------

**Agenda Item 1**: Review of the client data file

<< Discussion and Decision/Resolution >>

**Agenda Item 2**: Review of the sample interface and operation

<< Discussion and Decision/Resolution >>

**Agenda Item 3**: << Description >>

<< Discussion and Decision/Resolution >>

**Agenda Item 4**: << Description >>

<< Discussion and Decision/Resolution >>

To: Project Manager / Manager

From:

CC:

BCC:

Date: 19/07/2027

Subject: Perfect Policies Quiz - Project Clarification

-------------------------------------------------------------

Hi <Project Manager>,

I wish to request clarification on the following aspect(s) of the Perfect Policies Quiz project:

1.

2.

3.

I am good to meet in person to discuss this/these aspect(s) – should you prefer.

Regards,

<Lead Developer>

----------------------------------------------

< Lead Developer >

Lead Developer@UptownIT.com.au

-----------------------------------------------

To: Project Manager / Manager

From:

CC:

BCC:

Date: 19/07/2027

Subject: Perfect Policies Quiz Project – Third Party Library utilised

-------------------------------------------------------------

Hi <Project Manager>,

In the Perfect Policies Quiz project, we utilised the following Third Party Library:

Name of Library:

Areas of the project where utilised:

Licence detail:

Web / Server Link to Licence:

Regards,

<Lead Developer>

----------------------------------------------

< Lead Developer >

Lead Developer@UptownIT.com.au

-----------------------------------------------

To: << Client >>

From:

CC:

BCC:

Date: 19/07/2027

Subject: Perfect Policies Quiz Project – Third Party Library utilised

-------------------------------------------------------------

Hi <Client>,

In the Perfect Policies Quiz project, we have required the utilisation of a following Third Party Library: << Name of the required Third Party Library >>

A Third Party Library is... <<Relevant explanation of the concept of a Third Party Library for the client.>>

In relation to cost… <<Initial and ongoing>>

Licence detail:

Web / Server Link to Licence:

Regards,

<Lead Developer>

----------------------------------------------

< Lead Developer >

Lead Developer@UptownIT.com.au

-----------------------------------------------

To: Project Manager / Manager

From:

CC:

BCC:

Date: 19/07/2027

Subject: Perfect Policies Quiz - Project Completion

-------------------------------------------------------------

Hi <Project Manager>,

The Perfect Policies Quiz Project has been completed and I would like to offer the following comments:

1. How did the project go?

2. Any changes or adjustments required?

3. Any project risks identified?

4. Any adjustments required to be made to the team’s coding standards?  
(UptownIT\_Java\_Coding\_Standards\_Template.docx)

5. The two (or more) optimisation techniques utilised were:

* Optimisation Technique 1…
* Optimisation Technique 2…

6. …

I have attached the program documentation for your review and signoff.

|  |  |
| --- | --- |
| **PROGRAMMING PORTFOLIO - SIGNOFF**  Signing off on this document signifies that the submitted code and documentation comply with the Client Business requirements. | |
| Project Manager  Signature:  Date: | Programmer  Signature:  Date: |
| PROGRAMMING PORTFOLIO NOT APPROVED  Please provide feedback on the changes needed. | |

Regards,

<Lead Developer>

----------------------------------------------

< Lead Developer >

Lead Developer@UptownIT.com.au

-----------------------------------------------

To: << Client >>

From:

CC:

BCC:

Date: 19/07/2027

Subject: Perfect Policies Quiz - Project ready for deployment and training

-------------------------------------------------------------

Hi <Client>,

The coding, testing and review for the Perfect Policies Quiz Project has been completed and we would like to advise:

1. In relation to the deployment phase…

2. In relation to training for your staff…

3. In relation to ongoing support…

4. Other aspects …

Regards,

<Lead Developer>

----------------------------------------------

< Lead Developer >

Lead Developer@UptownIT.com.au

-----------------------------------------------

# 11 Contingency and advanced programming concepts

1 **Contingency task**. You are the project manager for a new large-scale application project. Development is scheduled to start in 10 days and today the lead programmer assigned to the project has handed in his resignation. The organisation has two other programmers with similar expertise but they are fully utilised in other projects.

Outline a strategy to find a programmer (expert Java developer) to start the project.

|  |
| --- |
|  |

2 A client project requires an undo and redo facility for a small and finite set of data entry sets that are added to and removed quite quickly. At a meeting with the project’s systems analyst, you discuss what you believe would be the best-suited data structure for this facility.

After some discussion, you suggest to use a Doubly Linked List of Objects. Justify your recommendation.

|  |
| --- |
|  |

3 A client project requires the storage of an object and its associated index value. The ability to efficiently locate a stored object is of highest priority in relation to the project’s required functionality.

Propose the most appropriate data structure for the task and explain your selection.

|  |
| --- |
|  |

4 Quick Sort is a very efficient sorting algorithm and is commonly utilised within program libraries for sorting sets of data.

Nevertheless, it is not always possible to utilise a Quick Sort from a program library, potentially because:

* The project’s data set is being stored in a complex data structure not supported by the Quick Sort facility provided.
* A Quick Sort does not suit the required sort.

Outline four (4) conditions that the subsequent selection of a sort algorithm in a specific situation may depend on.

|  |
| --- |
|  |
|  |
|  |
|  |

5 In creating an indexing system for a collection of data objects, a mathematical function (a hashing technique) could be utilised. The hashing technique or algorithm would be applied to the key of each data object providing for a set of individual numerical addresses. This allows for the efficient retrieval of a given data object once a search parameter has been provided, and has had the same hashing algorithm applied.

Identify and explain two (2) advantages of using a hashing technique.

|  |
| --- |
|  |
|  |

6 One of the programming concepts that can be utilised in both inter-process communication and operating system signals is Threads.

Analyse the above statement and justify the use of Threads.

|  |
| --- |
|  |