Edge Hill University

The Department of Computer Science

CIS3140
Research & Development Project

Level 6

Coursework 2 – Project Report and Artefact
2021/2022

Module Leader: **Prof Ella Pereira**
☎ 01695 65 7639
Email: insert email

Administrators:
☎ 01695 65 7603

Coursework 2

Weighting: **75%**
Submission date: **Wednesday the 11th of May, 2022**

Learning Outcomes Assessed:

LO2	Articulate, demonstrate and reflect on the outcomes of the project clearly and in a professional manner including legal, social and ethical implications
LO3	Manage the project effectively demonstrating systematic and effective planning, progress monitoring, reflection, use of time and resources, and in relation to the chosen methodology
LO4	Critically evaluate and integrate knowledge from a range of sources to develop appropriate ideas demonstrating their practical computing skills which result in the creation of a final report and artefact that fulfills the project specification
LO5	Synthesize project outcomes in order to critically evaluate the methods and/or methodology undertaken to deliver the aim and the objectives of the project

Introduction

The practical assignment is a report of the work undertaken during the project. In this report you will be expected to build on your Project Proposal (completed in CIS3161 R&D Methods, Semester 1) by extending it and revising its content based on the work that has actually been completed.

Please note you should discuss your report layout with your supervisor.

It will take you longer than you think to write up your Final Report. You will also want to get your supervisor to comment on a draft of the write-up; to allow sufficient time for the process of reading and revising the draft report you should write sections of your report as you go along. This will need revision near the end, but it is far easier to revise work than write it from scratch.

A typical Final Project Report will contain approximately 8000 words, excluding appendices. You will not get extra credit just for producing a lot of writing.

General Guidance

Recommended Contents of the Report

This section briefly describes the different elements that make up a typical Final Report.

Title: should be short and indicate clearly what the Project is about.

Abstract: summarises very briefly the problem tackled, the method adopted and the results obtained; it should be no more than half a page long. It should enable a reader to decide whether to bother reading the rest of the Report; no detail is required.

Acknowledgements: are optional - there is little correlation between your final mark and the amount of praise lavished on your Supervisor.

Chapter 1 Introduction: this should describe the Project in greater detail: it will provide background information; the purpose and objectives of the project, any research hypothesis (if applicable), describe previous work, identify the need for the present work and outline the contents of the following chapters. Do not assume that the reader has specialised knowledge of the area that you are describing and be sure to explain any technical terms that you use. This chapter also should include a section on ethics.

Main Chapters: these are often produced while work on the Project is progressing. A good way of discovering whether your writing is at the right level is to get somebody else on your course to read a sample chapter or two: they will have general background knowledge of the subject, but will not be familiar with your special subject.

If the main purpose of the Project is to produce software, then you may consider to include

- Background and Literature Review (will include Rated Work when appropriate)
    - An in-depth investigation of the context and literature, and where appropriate, other similar products/related work
- Problem Analysis / System Requirements
- Design: where appropriate, a clear description of the stages of the life cycle under taken, a description of the use of tools (CASE tools) to support the development process
- Implementation
- Testing and Known Problems

If your project is more research focused and involves an investigative work then you may have:

- Background and Literature review
- Research Design and Methods
- Data gathering and analysis
    - Experiments/tests
- Validation of results
- Developed outcome OR Outcome and discussions

Final Chapter Conclusions: should begin with a summary of the aim and objectives of the project and how well you've achieved them and lessons learnt during the course of the project.

It must also contain the following sections:

Critical Evaluation: this is a very important part of your Project Report. The aim of the critical evaluation is that you adopt a detached view of your Project and give an objective evaluation of your work. You should evaluate all aspects of your work, including the rationale for any design/implementation decisions, an assessment of whether the aims of the Project, as stated in the Project Proposal Report, have been achieved. In particular you should look back and reflect on each of the main stages of the software development process:

- Requirements: Were they complete, i.e. were all aspects of the system's behaviour described? Was anything missing? Were they too tightly linked to a particular design or implementation?
- Design: Were alternatives considered at each stage? Were the correct alternatives chosen? Were the right sets of use cases, scenarios, classes, etc. chosen?
- Implementation: Was the correct programming language chosen? Were the correct data structures and algorithms chosen for each class' attributes and methods? Was it necessary to modify the classes that were produced as the result of the design?
- Review of your plan and an explanation of any deviations from it.
- In the event that the individual work is part of a group enterprise, a clear indication of the part played by the author in achieving the goals of the project and its effectiveness.

Further Work: possible directions for further work should be given, and any outstanding problems should be described.

References: In Harvard style.

Appendices: In general, any large amount of technical material which would interrupt the flow of the Report can be assigned to an appendix. Examples of the types of material that should be placed in separate appendices are:

- Requirements Specification- only include a sample of this in the main body of the Report, the bulk should be put in the appendix if it is too wordy
- Design Methodology material: only include a sample of this in the main body of the Report. If any design stage produces a large number of diagrams, tables, etc., they should be put in the appendix.
- Test data: only include a sample of this in the main body of the Report, the bulk should be put in the appendix.
- User manual for software.

Glossary: explanation of any technical terms that you use. This is optional.

Report Layout Requirements

Your aim with report layout should be to make it as easy as possible for somebody to read and understand your Report - it is obviously in your own interests to ensure that this is the case.

The Final Report must be on A4 paper, using both sides. It should be 1.5 line spaced, make sure the Left hand margin is big enough for a spiral binding.

If possible submit the Final Report as a single volume, however if this would be greater than 200 pages, consult your Supervisor;

- All pages, including pages containing diagrams or illustrations, should be numbered. Conventionally, the preliminary elements (e.g. Abstract, Acknowledgements Table of Contents, List of Figures, List of Tables) in a report are numbered in lowercase Roman numerals (i, ii, iii, iv…), with no number on the title page. The main body of a report (e.g. from Chapter 1) is numbered in normal Arabic numerals.
- The title page should provide sufficient information to indicate the contents of the Report. It must contain the following information:

  Title of the project

  Student's name.

  Supervisor's name.

  Date of Submission.

  *The statement: **'This Report is submitted in partial fulfilment of the requirements for the BSc Honours Computing (or Web Systems Development, Computer Science) Degree at Edge Hill University.'***
- **Abstract**: The abstract should be no longer than 200 words, and it can be single spaced.
- **Acknowledgements** (optional)
- The **table of contents** comes next. It should include the following items (where applicable):acknowledgements, list of figures, list of tables, contents, chapters, only the main numbered sections of each chapter, References and Appendices;
- The main text should normally be no longer than 8,000 words long. This is roughly 40 pages of solid writing; allowing for diagrams and illustrations, therefore, this gives an ideal length of perhaps 60 to 70 pages;
- The main chapters, starting with the 'introduction' and ending with the 'conclusions', should be numbered, given a title, e.g. 'Chapter 1 Introduction', and **start on a new page**;
- Each table and figure should be clearly labelled with a number and caption.
- References should be in Harvard style in alphabetical order by author. Do NOT separate out Web, book, journal references, simply put them all in alphabetical order of author.
- The appendices come last

Hints on Style

The Final Report should be a description of what you have achieved, not a story. The Report should be written in the **third person**, so **'I' should not be used**.

Structure is all-important; the people marking your project have a lot to read in a short time, and you can make it easier for them (and thereby create a good impression) by making your Report well-structured and easy to follow.

General Guidelines for Developing Software

Many projects involve the development of software. The following stages should be covered in a Report which describes the production of software.

You will not be expected to cover all of the following stages to the same depth, which elements you concentrate on should be discussed with your Supervisor. If a stage is to be omitted this decision must be justified.

One of the most important aspects of the software development process is the consideration of alternative ways of developing the software. This is particularly important during the design and implementation phases of the Project. Therefore, it is essential that in your Final Report you document the alternative designs, algorithms and data structures, etc. that you considered, along with the final decisions you made and the reasons for the decisions.

Requirements Analysis: a precise statement of what the software should and should not do. Normally this is in two parts – functional requirements, i.e. a description of the list of features; and the non-functional, i.e. software performance. As a rough guide, the statement of requirements of the system should be of sufficient detail so that, if they were given to another student doing this module, they would be able to produce the system that you envisage developing. In other words, all aspects of the planned behaviour of the system should be described;

Design: the design process should be recorded. The design should preferably follow a recognised design method. For example UML for object oriented style produced software, ER diagrams and normalisation should be discussed in the case of database projects. . A very important aspect of the design process is to consider alternatives at each stage. So it is essential that within the Design chapter you include (sensible) alternative approaches along with the final decisions you made and the reasons for the decisions. It is very important to include this type of material in the Final Report. Assuming that UML is used for the design, then the Design chapter would normally start with an outline of which parts of UML are going to be used, and which are not, with reasons. This would then be followed by sections describing the application of the design stages used.

Implementation: a discussion of the issues involved in the decisions about - algorithm choice/design, data structures and language, use of predefined software, use of software tools, operating system and hardware restrictions, etc. For Object-Oriented development one way to structure the Implementation chapter is to consider each class and describe:

- the data structures used to implement its attributes, providing examples and diagrammatic representations of the complex ones. However little needs to be said about those that use built-in types.
- the algorithms used in the methods, including worked examples of complex ones. However little needs to be said about the set and get methods.

Software Testing & Known Problems: this should be done systematically using a test plan. All aspects of the software should be tested, e.g. data input validation, functionality and GUI. The list of the known problems with your software, i.e. bugs, and suggested bug fixes.

Documentation: a user manual for the software may be produced (discuss this with your supervisor), including installation instructions. This should be included as an appendix.

What you should submit

- You must submit your report in PDF, saved as: surname_ID_cw2.pdf. e.g. pereira_1234567_cw2 as via Turnitin.
- You must submit your artefact in a separate dorpbox set up for artefact submission.

Assessment Criteria – (Project Report & artefact) – 75%

CW3	LO2: Articulate, demonstrate and reflect on the outcomes of the project clearly and in a professional manner including legal, social and ethical implications LO3: Manage the project effectively demonstrating systematic and effective planning, progress monitoring, reflection, use of time and resources, and in relation to the chosen methodology LO4: Critically evaluate and integrate knowledge from a range of sources to develop appropriate ideas demonstrating their practical computing skills which result in the creation of a final report and artefact that fulfills the project specification LO5: Synthesize project outcomes in order to critically evaluate the methods and/or methodology undertaken to deliver the aim and the objectives of the project			
	Project Report (70%)			**Project Artefact (30%)**
Mark	**Process documentation (40%)**	**Critical evaluation (20%)**	**Structuring and use of language (10%)**	
FAIL (0-29)	Inadequate level of engagement with the task to demonstrate appropriate knowledge or skills.	No , or irrelevant evaluation	Inappropriate structure and/or demonstrates extensive mistakes in spelling/grammar.	No deliverable or not recognisably meeting what was planned
Narrow Fail (30-39)	Identifies some recognisably relevant content, including problem analysis and methods of solution.	Evaluation *describes* some problems and actions taken.	Recognisable structure and some appropriate content.	Deliverable(s) produced, showing some relevance to planned
Pass (40-49)	Report provides analysis of problem, context and describes methods of solution, with some recognised tools and techniques, though level of detail is likely to be variable.	Evaluation describes some of: problems actions taken, learning achieved and what would be done differently, but with variable level of detail.	**Mostly** structured appropriately, omissions or inappropriate content in some sections. Written in generally correct English, with few spelling and grammar errors. Some references resembling Harvard.	Deliverable(s) produced, meeting **some** of the requirements.
Satisfactory (50-59)	Report provides analysis of problem, context and describes methods of solution, with some recognised tools and techniques, with **appropriate detail and depth in some areas**.	Evaluation s details some problems actions taken, learning achieved and what would be done differently **with clear links**	As Pass and **all** sections structured appropriately containing mostly **relevant content. Most refs in Harvard**	Deliverable(s) produced, meeting **most** of the requirements.
Very Good (60-69)	As satisfactory and **most** aspects **are appropriately detailed**.	A **balanced** and **detailed** reasoned evaluation of problems, actions, learning and alternative actions	As satisfactory pass and consistent **relevant content throughout.**	As satisfactory and deliverables meet almost all requirements (maybe a few major weaknesses).
Excellent (70-84)	As VG pass and **all** aspects are clear and appropriately detailed. Demonstrates a range of problem-solving approaches.	As VG and clear justification of conclusions.	As VG pass and consistent appropriate academic style throughout.	As VG pass and deliverables meet all requirements (may be minor weaknesses)
Outstanding (85-100)	As excellent pass and shows systematic approach with initiative in some aspect.	A systematic and critical evaluation with clear and evidence-based justification of conclusions.	As excellent pass, using a ***consistent and appropriate level of detai***l suitable for a professional specification.	As excellent and deliverable(s) exceed requirements in some way, demonstrating initiative.