Mark Management System

by
Kamil Zak

In partial fulfilment of the
requirements for the degree of
Bachelor of Science
in
CS408: Individual Project

University of
Strathclyde
Glasgow

Department of
Computer and Information Sciences

August, 2021

Abstract

The abstract is a bird's eye view of the project.

It should not exceed one page. Mention the scope and objectives of the project, the methodology used,the main findings, and significance of your results.

Acknowledgements

I would like to thank the participants who evaluated my project, my supervisor etc.

Contents

1 Introduction

This is a concise and clear overview of your dissertation(more or less 2-3 pages). You can start off the project description provided of the project that was allocated to you and flesh it out.

Include (1) the problem you have tackled, (2) why this problem is worth addressing, (3) what you did to address it - in broad terms. Detail will come later.

i.e., issue(s) on which the research will focus, shall be clearly identified and described. You shall refer to past research work relevant to the topic and objectives, i.e, of the study. You shall outline where applicable the potential research output with respect to research transfer and uptake by the community.

The introduction says: this is an overview of the project. This is why I did it (the problem) and how I tackled it. It is the *runway* into the project. It lets the marker know what to expect of your report.

If you're doing a research project, this would be the place to include the research questions you plan to address. For example:

RQ1: Where did James Bond come from?

RQ2: Why are pumpkins orange?

If you're doing a project of type 1, include a list of objectives. For example:

Objective 1: Provide software to allow James Bond to become invisible.

Objective 2: Provide software to keep track of all loyalty points in one place.

Provide a 'map' of your dissertation. For example: Section 2 reviews the background literature that was reviewed to inform this project. Then Section 3.....

2 Background Literature

This section should provide the grounding for your project. You will be able to refer back to this in Section 3, where you report on design decisions and methodology.

There should be a strong link to your research questions (for project types 2 and 3). There will be a strong link to the software design decisions for project type 1.

Don't just include a paragraph for each paper you read. Synthesize it and create a story line.

Review between 5 and 10 authoritative sources. For research based projects, some of these should be from peer-reviewed journals or conferences.

This section will reference literature like this [1]. Sloppy referencing will (1) take time and effort, and (2) lose you marks if you don't do it.

3 Specification & Design

Describe all details of the design and procedures used to achieve the project objectives. Do this chronologically.

It should be detailed enough to allow for an assessment of the rigour of your process, and, in the case of research projects, in terms of how well grounded your research is in the research literature. In these cases, refer back to relevant sections in the previous chapter.

Say which software lifecycle approach you used e.g., Waterfall, Spiral, Agile.

How did you gather user requirements?

3.1 Methodology

Which methodology did you choose?

3.2 Analysis

How did you decide on the particular software artifact you decided to develop?

3.3 Requirements

Here you explain what the functional and non-functional requirements are. Explain how you prioritised them. See `https://www.nuclino.com/articles/functional-requirements` for more information.

Functional requirement: "The system must **do** [requirement]."

Non-functional requirement: "The system shall **be** [requirement]."

Well-written functional requirements typically have the following characteristics:

Necessary: Although functional requirements may have different priority, every one of them needs to relate to a particular business goal or user requirement.

Concise: Use simple and easy-to-understand language without any unnecessary jargon to prevent confusion or misinterpretations.

Attainable: All requirements you include need to be realistic within the time and budget constraints set in the business requirements document.

Granular: Do not try to combine many requirements within one. The more precise and granular your requirements are, the easier it is to manage them.

Consistent: Make sure the requirements do not contradict each other and use consistent terminology.

Verifiable: It should be possible to determine whether the requirement has been met at the end of the project.

3.3.1 Functional Requirements

This is the **WHAT** of your artifact.

Functional requirements are product features that developers must implement to enable the users to achieve their goals. They define the basic system behavior under specific conditions.

Functional requirements need to be clear, simple, and unambiguous.Examples:

- The system must send a confirmation email whenever an order is placed.
- The system must allow blog visitors to sign up for the newsletter by leaving their email.
- The system must allow users to verify their accounts using their phone number.

3.3.2 Non-Functional Requirements

This is the **HOW** of your artifact. Example non-functional requirement: "When the submit button is pressed, the confirmation screen must load within 2 seconds."

3.4 Design

3.4.1 Interface Design

Explain how you used wireframes, and how you tested these to design the user interface.

3.4.2 System Design

Show how you designed your database (if appropriate) and how you designed your system architecture, and the individual parts. Use UML and an Entity Relationship diagram

4 Product

4.1 Implementation

Provide implementation details: language used, architecture (e.g. server and client, or hub and spoke). Explain how you secured personal details.

4.2 Verification & Validation

How did you verify that your software was debugged and worked correctly?

How did you validate that you had implemented all the functional and non-functional requirements?

5 Results & Evaluation

Results are often presented in tables, figures and other relevant illustrations. Include text that refers to these figures/tables.

5.1 Evaluation Process

If you involved humans in the evaluation, how many did you have? What can you say about the demographics of your participants? (If you did collect these)

In terms of the *user interface*, how did you carry out a usability evaluation, how did you go about doing this? How did you recruit participants?

In terms of delivering *functionality*, did you carry out user acceptance testing? (see attached guidance).

5.2 Results of Evaluation

This Section includes a direct interpretation of the gathered data and evaluation processes.

5.3 Returning to the Research Questions

Return to research questions or objectives as appropriate.

5.3.1 RQ1

It is clear from our findings that James Bond was born in Wigtown in Scotland. However, he grew up in Diss, in Norfolk. We know this because ....

5.3.2 RQ2

We were not able to answer this question from our studies, although some suggestions were made. These could not be proven.

5.3.3 Objective 1

5.3.4 Objective 2

6 Discussion & Reflection

6.1 Interpreting the Results

Here you will discuss your findings. This is especially relevant for research projects. You might interpret what the data and evaluation implies, both for future research and for practice (if appropriate).

The discussion is **not** a review of literature. You should try to compare research findings with previous work, provide explanations for your findings, discuss research findings, in terms of their contribution.

6.2 Reflection

Look back and think about what you would do differently if you were going to start the project with the knowledge you have now. Be honest about your mistakes or missteps.

6.3 Challenges

This is not the place to mention personal circumstances but rather challenges related to the work involved in the project.

6.4 Limitations

Acknowledge things like: small number of participants, software wasn't completely debugged, or whatever else went wrong and affected your project. *Include as appropriate*

6.5 Future Work

If someone else wanted to build on your project's product, what would be cool to do next?

7 Conclusion

The conclusion is similar to when a plane lands. You don't rewrite the introduction. You say something like - I addressed the problem outlined in the introduction, and I built some software to do this. I tested the software like this ADD FEW WORDS.

Summarize main findings drawn from the project work. Mention the objectives or research questions. Do not repeat points raised in the discussion and reflection Section. If applicable, you can make recommendations. The conclusion should NOT contain any references.

References

[1] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

A Appendix

This is where you can include your documentation.

Remember that the marker is not required to read this, but might well check to ensure that you have included product documentation, and ethical approval, as required.

A.1 Ethical Approval Form

If your project required you to do any evaluation with humans, you MUST include this. It can be downloaded from the Ethics system.

`https://local.cis.strath.ac.uk/wp/extras/ethics/index.php`

A.2 Participant Information Sheet

If your project required you to do any evaluation with humans, you MUST include this

`https://www.strath.ac.uk/ethics/informationsheetandconsentform/`

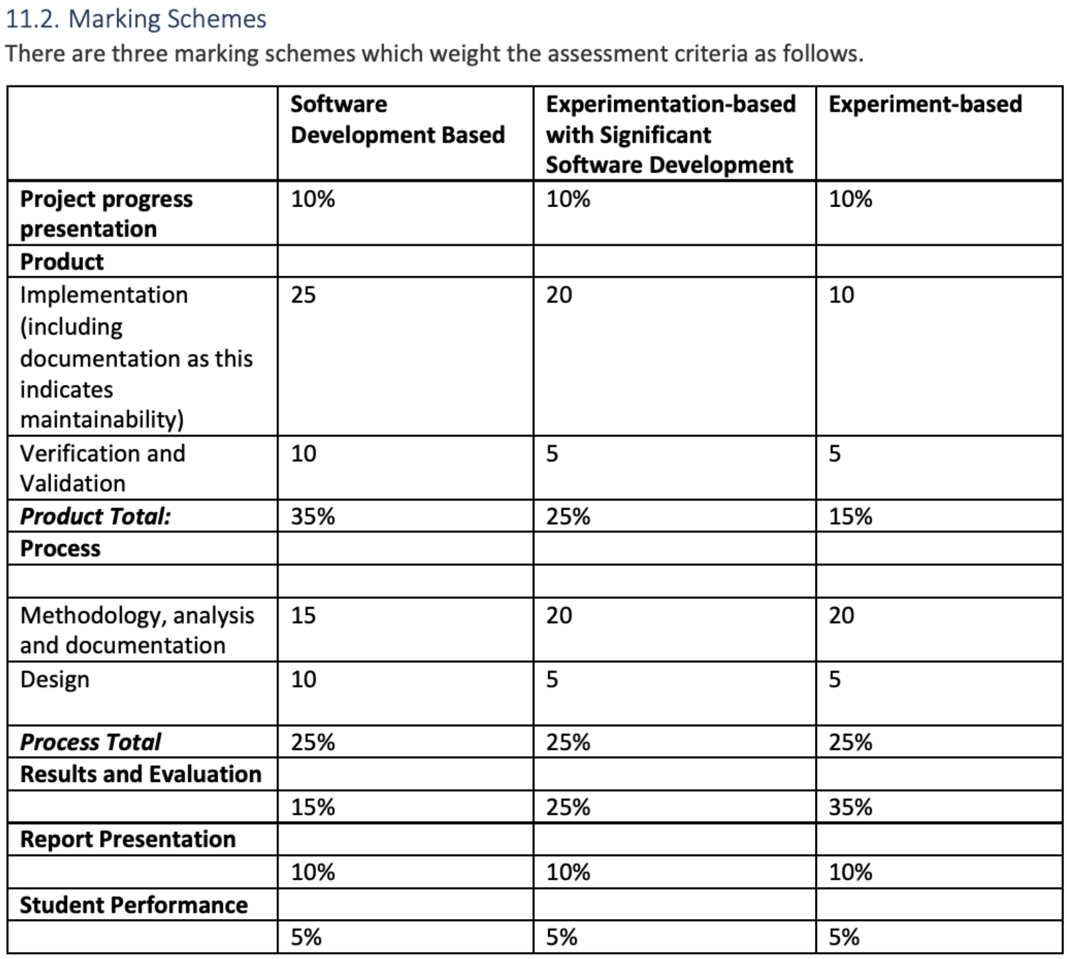
A.3 Consent Form

If your project required you to do any evaluation with humans, you MUST include this.

`https://www.strath.ac.uk/ethics/informationsheetandconsentform/`

A.4 Marking Scheme

REMEMBER TO DELETE THIS. IT IS ONLY INCLUDED FOR your INFORMATION.



11.2. Marking Schemes

There are three marking schemes which weight the assessment criteria as follows.

	Software Development Based	**Experimentation-based with Significant Software Development**	**Experiment-based**
Project progress presentation	10%	10%	10%
Product			
Implementation (including documentation as this indicates maintainability)	25	20	10
Verification and Validation	10	5	5
Product Total:	35%	25%	15%
Process			
Methodology, analysis and documentation	15	20	20
Design	10	5	5
Process Total	25%	25%	25%
Results and Evaluation			
	15%	25%	35%
Report Presentation			
	10%	10%	10%
Student Performance			
	5%	5%	5%