Project

Dissertation

School of Computing & Information Engineering

|  |  |
| --- | --- |
| Name: |  |
| Project Title: |  |
|  |  |
| Supervisor: |  |
| Second Marker: |  |
|  |  |
| Submission Date: |  |



Plagiarism Statement

I declare that this is my own work and that any material I have referred to has been accurately and consistently referenced. I have read the University’s policy on plagiarism and understand the definition of plagiarism as given in the Project Handbook. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions in accordance with the policies and procedures of the University. A mark of zero may be awarded and the reason for that mark will be recorded on my file.

[your signature]

Acknowledgements

You should provide a statement of thanks to anyone who contributed directly or indirectly to the project, in any way.

Table of Contents

Abstract 1

Chapter 1: Introduction 2

1.1 The Problem 2

1.2 The Solution 2

1.3 Dissertation Structure 2

Chapter 2: Analysis 3

Chapter 3: Design 4

Chapter 4: Implementation, Testing and Evaluation 5

Chapter 5: Conclusions 6

References 7

Appendices 8

A1. Code 8

A2. Test Suite 8

A3. Questionnaire Results 8

# Abstract

The abstract is a short overview of the completed project. Typically, it has three parts: (i) a summary of the problem that has been tackled; (ii) a summary of the solution developed; and (iii) a summary of the work completed. It normally appears on a page by itself.

It is recommended that you try to produce the abstract before the rest of the dissertation to make clear what you intend to cover.

Length: ~300 words

**Some notes on formatting / presentation:**

* Use sections / sub-sections to structure your document and number each section heading.
* You should also make use of Word features such as heading styles – for example the headings and sub-headings in this template are formatted using ‘Heading 1’ and ‘Heading 2’ styles. By doing so the table of contents can be generated / updated automatically.
* Use 12-pt font for the main text and use 1.5 line spacing.
* The length of the dissertation should be **no more than 60 pages**. The page count is from the Abstract to the References – i.e. the Abstract is numbered page 1 and the reference should finish on page number 60 or less. Nothing after page 60 will be marked! So, if your references appear after this you will lose marks allocated for referencing.
* All figures and tables should be numbered and captioned – the captions for **figures** should appear **below** the figure; the captions for **tables** should appear **above** the table. Captions can be in a font size a little less than 12. All figures and tables should be referred to from within the main text.
* Code fragments should be presented in a fixed width font (e.g. Courier) but should not dominate the surrounding text (in this case you may need to use a Courier font size less than 12 so that it isn’t larger than the main text).
* You can use appendices to provide *secondary* material in support of the description in the body of the dissertation. In principle, the reader need not look at the appendices and no specific marks are awarded for this section.

# 

# Chapter 1: Introduction

The Introduction is effectively an expanded version of the abstract. It again summarises the problem, aims and objectives, solution and work undertaken, but this time in more detail. The chapter should conclude with a summary of the remainder of the dissertation.

The chapter(s) can be split into sub-sections with each sub-section being numbered.

Note – the following *are only suggestions* …

## 1.1 The Problem

Describe the problem …

## 1.2 The Solution

Describe the solution …

## 1.3 Dissertation Structure

Describe the structure of the dissertation …

Chapter length: ~4 pages

# 

# Chapter 2: Analysis

The Analysis Chapter documents your work in finding a suitable solution to the problem being tackled. The nature of this investigation will vary from project to project, but may involve the use of questionnaires, interviewing stakeholders, researching relevant background material and solution options, including the creation of experimental prototypes, professional issues, and so on.

It is valuable to include a discussion of any ideas that were investigated but subsequently dropped, to illustrate the problem-solving process involved.

Material that you previously prepared for the Analysis and Design Report in the first semester should be a good starting point for this chapter.

Length: ~12 pages

# 

# Chapter 3: Design

The investigation of the problem and its potential solution will have led to the identification of a set of system requirements, which should be presented in this chapter. The requirements should be documented using whatever descriptive techniques are appropriate for the type of system and type of requirements involved. For example, functional requirements might be documented as user stories or use cases.

Design also covers the user interface, software architecture, data definitions, algorithms and other high level descriptions of the system you have created. Ideally, a good system design document is one that can be passed to someone else to implement.

Material that you previously prepared for the Analysis and Design Report in the first semester should be a good starting point for this chapter.

You are encouraged to use descriptions and models suitable for your own circumstances. For example,

* In describing how the user will interact with your system you may want to present a block diagram identifying key parts of the user interface in addition to showing screen shots. If you considered HCI guidelines, you should explain how these influenced your design
* If you are using web pages, you may want to include a map of your site, indicating how users navigate through it
* If you are using a database, you will need to describe the design schema, including details of any normalisation involved. You may also want to include an entity-relationship diagram
* For a Java program, you will need to start with the class design, identifying your main classes and indicating their purpose
* For some processing it may be necessary to make use of complex algorithms, which should be described and illustrated appropriately

Length: ~18 pages

# 

# Chapter 4: Implementation, Testing and Evaluation

The content of the implementation, testing and evaluation chapter is largely self-explanatory.

First, under implementation, you should describe the approach you took to development, indicating what was learned from the prototype and successive versions produced, identifying any other incremental steps followed to achieve the final system.

Secondly, you should describe any specific tools used to support the development process, including details of any new languages that you needed to learn.

Also under implementation, you should include details of any significant aspects of the code you have produced, including the use of specific algorithms. If you have used code from other sources, this should also be identified, indicating how it was integrated into your system.

Your approach to testing should then be described, identifying the test cases that you have used to verify the correctness of the software. These should demonstrate that your testing has been appropriate and thorough.

Under evaluation, you should assess the perceived value of your system to its intended users against the specified requirements. You are not expected to create a ‘perfect’ system so marks are awarded more for the thoroughness of your validation than any praise obtained. Indeed criticism gives you an opportunity to explain the issue raised and suggest improvement.

Note that when planning your project, sufficient time should be included after implementation for testing and evaluation as these activities have a significant impact on the final quality of your system and the write up in your dissertation.

Length: ~20 pages

# 

# Chapter 5: Conclusions

In the final chapter you should summarise your project work overall and assess it critically. This should indicate what lessons you have learned and so clarify what you might do differently if faced with the same situation again. In particular, you should identify and discuss how the project plan evolved as the project progressed.

The limited time available for implementation means that you are likely to have ideas for further work. These should also be included in the conclusions.

The Conclusions chapter, like the Introduction, should be freestanding, allowing the reader to understand what the project has achieved without studying other chapters.

Length: ~4 pages

# 

# References

Present the references used in the dissertation in Harvard format.

# 

# Appendices

The appendices are an opportunity to provide secondary material in support of the description in the body of the dissertation. In principle, the reader need not look at the appendices and no specific marks are awarded for this section.

Sample content:

## A1. Code

Code developed through the project – or samples of code for the main algorithms etc.

## A2. Test Suite

Full set of tests applied to the software

## A3. Questionnaire Results

Results of questionnaires used to evaluate the software or identify requirements