A Short Guide to Writing Your Final Year Project Report

Based on

Cardiff University

School of Computer Science

and Informatics

Abstract

This guide is intended to help you produce a good final year project report. It gives advice on how to gather relevant material, how to organize it into a suitable form and how to then turn it into a written project. It also describes the conventions that should govern the structure and suggests some descriptive devices that you can use to make it more effective. A summary of the guidelines is given at the end. The appendix lists the rules governing presentational details, including print quality, font sizes, etc.

1 Introduction

This guide is meant to help you produce a good final year project report. A good report is one that presents your project work *concisely* and effectively. It should contain various materials relevant to the work you have undertaken in respect of your project; it should be organized into a logical framework; and it should be supported by written material that follows well-established academic conventions in a consistent fashion.

The purpose of the project is, in the context of the degree you are studying, to integrate various aspects of the taught material and to demonstrate your (academic) research skills and your (professional) analysis, design and implementation skills. It gives you the opportunity to conduct in-depth work on a substantial problem to show individual creativity and originality, to apply where appropriate knowledge, skills and techniques taught throughout the degree programme to further oral and written communication skills, and to practice investigative, problem-solving, management and other transferable skills. The management and execution of the project is your responsibility, but you should seek and take advantage of advice from your supervisor.

Your project supervisor will guide you on what it is reasonable to expect a project in your chosen topic to deliver. However, all projects are required to justify all decisions made at every stage of research and the development of appropriate deliverables, including the choice of approach.

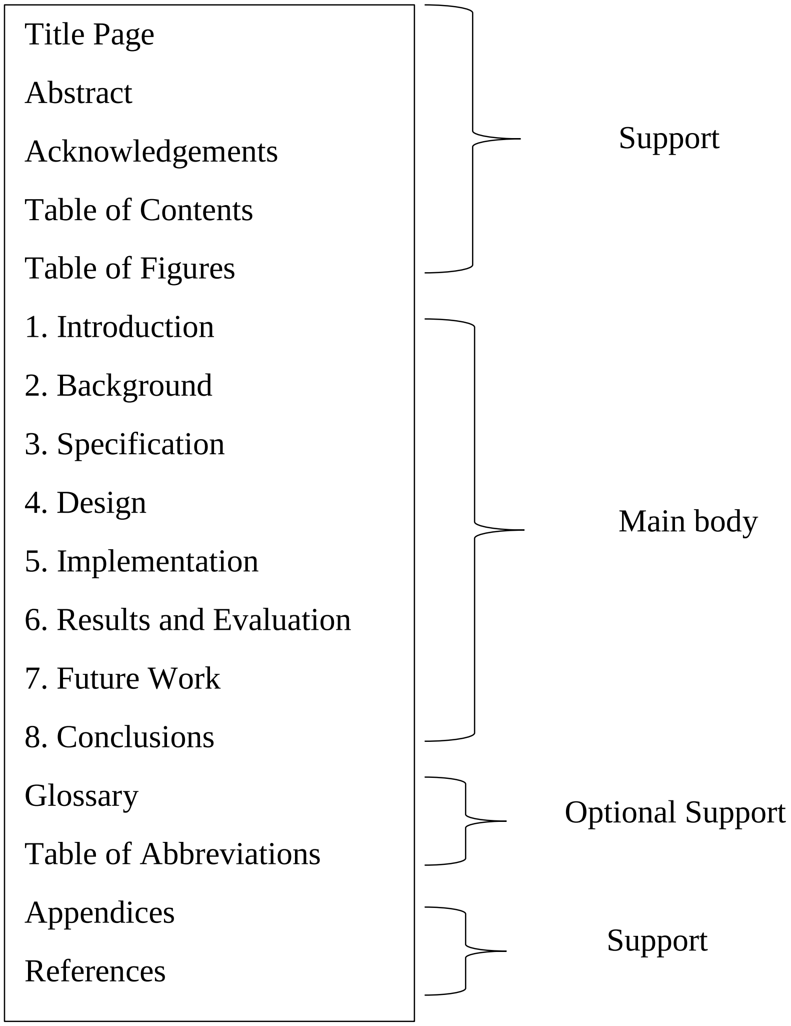
2. Arranging Material and Structuring the Project Report

You should consider, at the beginning of your project, what you need to do to solve the problem you have chosen to address. This will then inform choices about the structure of your report; your written report needs to be both a “narrative” (telling the story of your project) and an “argument” (providing a logical justification of the steps you have undertaken to solve your chosen problem). Once you have started to gather material you can begin to arrange it in a form which can then be refined into the final project report, though the outline chapter headings shown below will serve as a good guide in the early stages of your work.

All good project reports whatever their subject, follow certain well-established conventions and have a similar overall shape. They generally consist of a main body surrounded by other information (presented in appropriate formats) that support it in various ways. Some of these are mandatory, others are optional.

Figure 2.1 shows an example of the layout we suggest for a project which implements a piece of software. You should vary the titles of the sections if these are inappropriate for your project – your supervisor is the best person to guide you on this. For the moment we will concentrate on the main body of the report and leave the supporting information until later. We recommend that you do the same when writing your report, though you should have a plan for your final report which will guide you on what material you should be retaining for eventual inclusion.

Project reports describing projects whose aim has been to develop a particular software system tend to have a main body with a characteristic structure as illustrated above.



Title Page
Abstract
Acknowledgements
Table of Contents
Table of Figures

Support

1. Introduction
2. Background
3. Specification
4. Design
5. Implementation
6. Results and Evaluation
7. Future Work
8. Conclusions

Main body

Glossary
Table of Abbreviations

Optional Support

Appendices
References

Support

Figure 2.1: Suggested report structure for a project which implements a piece of software.

We look at each of the general sections of the report structure in more detail below. You can use this characteristic structure as a rough template for organizing the material. However, often it may be of advantage to adjust the suggested structure to your particular project instead of sticking to the template. Consult your supervisor for advice. It is also a good idea at this stage to plan roughly how long each part should be, to make sure that the length and overall balance are about right. You can then construct each part to produce a first draft of the main body.

Chapter 01

The "Introduction"

A good introduction should tell the reader what the project is about without assuming special knowledge and without introducing any specific material that might obscure the overview. It should anticipate and combine main points described in more detail in the rest of the project report. Also, importantly, it should enthuse the reader about the project, to encourage them to read the whole report. Normally it should include such things as:

- the aim(s) or goal(s) of the project;
- the intended audience or "beneficiaries" of the work done;
- the scope of the project;
- the approach used in carrying out the project;
- assumptions on which the work is based; and
- a broad summary of important outcomes.

Chapter 02

The "Background" (or Literature Review)

The purpose of the Background section is to provide the typical reader with information that they cannot be expected to know, but which they will need to know in order to fully understand and appreciate the rest of the report (see Section 4.1 for details of who a typical reader might be). It should explain why the project is addressing the problem described in the report, indicate an awareness of other work relevant to this problem and show clearly that the problem has not been solved by anyone else. This section may describe such things as:

- ✓ the wider context of the project;
- ✓ the problem that has been identified;

- ✓ likely stakeholders within the problem area;
- ✓ any theory associated with the problem area;
- ✓ any constraints on the approach to be adopted;
- ✓ existing solutions relevant to the problem area, and why these are unsuitable or insufficient in this particular case;
- ✓ methods and tools that your solution may be based on or use to solve the problem;
- ✓ and so on.

The wider context of the project includes such things as its non-computing aspects. So, for example, if you are producing software or any other products, including business recommendations, for a specific organization then you should describe aspects of that organization's business that are relevant to the project.

Relevant existing products, documents or artefacts that you should mention could be ones that, for example,

- ✓ are similar to the one you are proposing;
- ✓ support your project;
- ✓ your project aims to extend or replace;
- ✓ demonstrate the "deficiencies" your project intends to address.

You need only describe things that will be unfamiliar to the potential reader, or are unique to the organization or topic your project addresses. Your project, if it involves software development, will almost certainly use all kinds of existing software such as language compilers, subroutine libraries, etc., but you can assume that the reader will be fully acquainted with, for example, general purpose programming languages such as Java, C/C++, Fortran, Pascal, Python, PHP, etc., Also, it may involve the better known specialized packages such as MySQL, ORACLE, OpenGL, etc. You should mention the particular variety and possibly version number, e.g. Java SE 6, but you need say nothing more than that.

If your project depends on any specialist or uncommon software such as specialized subroutine packages or a more obscure or specialized programming language, you should describe them briefly and discuss whatever features are relevant to your project. Often this can be done by comparing it to some well-established piece of software, for example

The Descartes language is like a restricted version of Pascal but with the following extra features…

Again, long descriptions of details are to be avoided and references to suitable sources of detailed information should be given instead.

Other background information could consist of the sequence of events leading up to the present situation or the results of earlier investigations. You could also discuss such things as any cost or time constraints imposed on the project.

Your background section should end with a clear statement of the research questions problem your project is trying to answer. These will reflect the aim of your project, but will be different in that they explain the problem you are attempting to solve, e.g.,

Example 1:

Aim:

The aim of this project is to develop software for the improved planning of the routing of delivery vehicles to customer locations, that reflects the forecast availability of each customer to receive goods.

***Research question(s)*:**

In order to demonstrate the achievement of the stated aim, this project will identify route planning software currently in use and the underpinning algorithms, define appropriate performance metrics, determine how to express constraints on an alternative algorithm, develop an improved algorithm and demonstrate on what basis it is judged an improvement, and implement the improved algorithm in a usable and robust software package.

Example 2:

Aim:

The aim of this project is to develop a business strategy for organization X that will improve the survivability of X in the face of increasing global competition.

***Research question(s)*:**

In order to develop a business strategy, it will be necessary to identify key stakeholders and determine their vision for the organization at the end of the strategic planning timeframe, assess the likely outcome, in terms of the organization's survivability, of maintaining the current strategy, and develop and assess an alternative set of activities to achieve the stated vision.

Chapter 03

The "Specification & Design" (or Methodology)

The purpose of the Specification and Design sections is to give the reader a clear picture of the system you plan to create, in terms of the capability required. A specification should tell the reader what the software system is *required* to do. The design then gives the top-level details of how the software system meets the requirement. It will also identify constraints on the software solution, that are important in guiding decision making throughout the development process.

Describing what a software system does (specification) and how it does so (design) effectively usually means describing it from more than one viewpoint. Each viewpoint will convey some information about the system that other viewpoints omit. (You would use the same technique when describing any complicated construction such as a building, an aircraft, a novel or a painting). Possible viewpoints might be:

- the business model the software supports;
- the user interface;
- the dynamic behavior of the system;
- how data flows through the system;
- what data types are implemented in the system;
- what algorithms are implemented in the system;
- the static architecture of the system, i.e. how the code is partitioned into modules, etc.

A common approach is to first define the user or business requirements, then describe the static architecture, identify modules and groups of closely connected modules, and then to apply other views to each of these groups. Fine details, specifically details of code, should be left out.

We strongly recommend that you make extensive use of diagrams, such as entity-relationship diagrams, UML diagrams, state charts, or other pictorial techniques.

As well as describing the system, it is important that you *justify* its design, for example, by discussing the implications of constraints on your solution and different design choices, and then giving reasons for making the choices you did. Typically, these implications will relate to the aims of the project and to aspects of it discussed in the Background section.

The design of the system will almost certainly have evolved while you were developing it. Obviously you should describe its final state but often there are good reasons for describing intermediate states, too; for example, if you want to discuss the details of the design method used or to highlight learning that you later refer to in the Reflection section. If you do this, take special care to make sure the reader does not get confused between different stages of the design.

Chapter 04

The "Implementation"

The Implementation section is similar to the Specification and Design section in that it describes the system, but it does so at a finer level of detail, down to the code level. This section is about the realization of the concepts and ideas developed earlier. It can also describe any problems that may have arisen during implementation and how you dealt with them.

Do *not* attempt to describe all the code in the system, and do *not* include large pieces of code in this section. Complete source code should be provided separately (see Appendix B and submission guidelines). Instead pick out and describe just the pieces of code which, for example:

- are especially critical to the operation of the system;
- you feel might be of particular interest to the reader for some reason;
- illustrate a non-standard or innovative way of implementing an algorithm, data structure, etc.

You should also mention any unforeseen problems you encountered when implementing the system and how and to what extent you overcame them. Common problems are:

- difficulties involving existing software, because of, e.g.,
  - its complexity,
  - lack of documentation;
- lack of suitable supporting software;

- over-ambitious project aims.

A seemingly disproportionate amount of project time can be taken up in dealing with such problems. The Implementation section gives you the opportunity to show where that time has gone.

Chapter 05

The “Results and Evaluation”

In this section you should describe to what extent you achieved your goals.

You should describe how you demonstrated that the system works as intended (or not, as the case may be). Include comprehensible summaries of the results of all critical tests that were carried out. You might not have had the time to carry out any full rigorous tests – you may not even got as far as producing a testable system. However, you should try to indicate how confident you are about whatever you have produced, and also suggest what tests would be required to gain further confidence.

This is also the place to describe the reasoning behind the tests to evaluate your results, what tests to execute, what the results show and why to execute these tests. It may also contain a discussion of how you are designing your experiments to verify the hypothesis of a more scientifically oriented project. E.g., describe how you compare the performance of your algorithm to other algorithms to indicate better performance and why this is a sound approach. Then summarize the results of the tests or experiments.

You must also critically evaluate your results in the light of these tests, describing its strengths and weaknesses. Ideas for improving it can be carried over into the Future Work section. Remember: no project is perfect, and even a project that has failed to deliver what was intended can achieve a good pass mark, if it is clear that you have learned from the mistakes and difficulties.

This section also gives you an opportunity to present a critical appraisal of the project as a whole. This could include, for example, whether the methodology you have chosen and the programming language used were appropriate.

Chapter 06

The “Future Work”

It is quite likely that by the end of your project you will not have achieved all that you planned at the start; and in any case, your ideas will have grown during the course of the project beyond what you could hope to do within the available time. The Future Work section is for expressing your unrealized ideas. It is a way of recording that „I have thought about this‟, and it is also a way of stating what you would like to have done if only you had not run out of time1. A good Future Work section should provide a starting point for someone else to continue the work which you have begun.

Chapter 07

The “Conclusions”

The Conclusions section should be a summary of the aims of project and a restatement of its main results, i.e. what has been learnt and what it has achieved. An effective set of conclusions should not introduce new material. Instead it should briefly draw out, summarize, combine and reiterate the main points that have been made in the body of the project report and present opinions based on them.

The Conclusions section marks the end of the project report proper. Be honest and objective in your conclusions.

Chapter 08

The “Reflection”

We believe in the concept of “lifelong learning”. One of the principles applied throughout the assessment during your studies is that of the value of reflection. We believe that it is important that we reflect upon our performance in order to identify “transferable learning”, that can be carried over into future activities. Reflection should focus on what Argyris calls “double loop learning”; this is where we identify, not relatively “simple skills”, such as the mastery of a new programming language, but the impact of what we have done on the assumptions, concepts and ideas we used to make decisions about our work. For example, a “reflective practitioner” would try to identify the characteristics of the problem that has been addressed, and consider whether assumptions or decisions about the relevant approach to solving that problem had been appropriate, in order to make a better decision in relation to problems that might be encountered in the future.

The "References"

In Section 2 we said that you should relate your work to that of other people. Other work explicitly cited should be listed in the Reference section and referred to in the text using some kind of key. It is important that you give proper credit to all work that is not strictly your own, and that you do not violate copyright restrictions.

It *may* be desirable to provide a Bibliography section separately from the reference section. In general, references are those documents/sources cited within the text. The bibliography lists documents which have informed the text or are otherwise relevant but have not been explicitly cited.

University of Moratuwa allows two standards of referencing, namely IEEE and APA. Guidelines for using those methods can be found at the respective URLs given below.

http://www.lib.mrt.ac.lk/images/User%20Education_%20Guidelines_IEEE.pdf

http://www.lib.mrt.ac.lk/images/User%20Education_%20Guidelines_APA.pdf

you have included material from other sources; in other words, you need to critically assess the work of others, whether it is supporting your position or not:

- ✓ If the material you are citing from another source supports your position, you must explain why it should be trusted. For example, material from a published journal will, normally, have been peer-reviewed and can therefore be considered to have some validity, according to subject matter experts. Much of what is published on the Internet cannot be regarded in the same way, however.
- ✓ You will often find that there are conflicting views in the published material; in such cases you must explain which view you favour and why, before relying on the material to support your position.
- ✓ If other writers have taken a different position to the one you support; you must explain why the reader should accept your ideas rather than those proposed elsewhere.

In summary, you need to ensure that you have clearly assessed the relevance of referenced material to the development of your position, or your argument, and demonstrated that you are justified in taking this material to be authoritative.

University of Moratuwa uses automated plagiarism detection and you must not have more than 10% similarity index for your report.

Appendices

You may have several appendices to refer to further details related to chapters

- ✓ SRS
- ✓ Use Case Diagrams
- ✓ Activity Diagrams
- ✓ ERD (as necessary according to your project)
- ✓ Software test plan, procedure and test reports
- ✓ Software user Manuel etc.

General Guidelines for Preparation of Final Project Report

Your dissertation must be prepared according to the following instructions. This includes general guidelines, assembling pre- pages, and structuring of the body of the text. It should be noted that the final Draft of the report and the final project report should have the same format. The Draft should be submitted in the form of Spiral binding while the Final Project Report should be in the **hard-bound form**.

1. General Guidelines

Submission: Three (3) hard-bound (Spiral binding for the Draft) copies of the dissertation must be submitted. In addition to that better to keep a copy for your personal use.

Colour: Dark Blue (Applicable to final Dissertation only)

Paper: The dissertation should be printed on good quality A4 size paper, on single side

Lettering: Times New Roman. Size will change as per chapters/sections, etc. Cover and Spine should be in GOLD lettering.

Printing: A laser quality printing, do not use colours.

Spacing: 1.5-line spacing must be used for the body of the text and the list of Reference entries. Only footnotes, quotations, table and figure captions may be single spaced.
Margins: Left, 1-1/2 inches; top, bottom, and right, 1 inch.

Chapters and Sections: All chapters must be numbered (Arabic). Chapter names and the headings of the chapters must be 18 in font size and bold. Each chapter should have a section called Introduction at the beginning and a Summary at the end. Sections through Introduction to Summary must be structured with subsections, if necessary, and numbered accordingly. Headings of sections and subsections must be 12 in font size and bold.

Tables and Figures: All Tables and Figures must be named with captions and cited inside the text.

Reference and Citations: The list of Reference must be arranged in an alphabetical order of the name of the first author, and numbered in square brackets. The corresponding number of a reference must be used to cite the particular work inside the text. All items in the list of reference must be cited inside the text.

Appendices: Appendices must be named in alphabetical order (Appendix A, Appendix B, ….) and also a name should be given for each appendix. All Appendices must be cited inside the text by its name.

Cover page: Print the following information by leaving appropriate line spacing

- Title of the Project (16 font, bold, centered)
- Your name (12 font, centered)
- Index Number (12 font, centered)
- Faculty of Information Technology (12 font, centered)
- University of Moratuwa (12 font, centered)
- Month Year (12 font, centered)

Spine of the dissertation: The following information should be printed from top to the bottom of the spine.

The Degree must be Bachelor of Information Technology (BIT) as appropriate.

Title Degree Year

Page numbering: All page numbers must be centered at the bottom of each page. Roman numbering must be used for pre-pages. There should not be any page numbers for the title page. Arabic numbers must be used for the body of the dissertation.

Title Page

- Title of the Project (16 font, bold, centered)
- Your name (12 font, centered)
- Index Number (12 font, centered)
- Type the following (12 font, centered)

  Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka in partial fulfillment of the requirements of the Degree of Bachelor of Information Technology(External) in Information Technology.
- Month Year (12 font, bold, centered)

2. Assembling of Pre pages

The pre pages include: Declaration, Dedication, Acknowledgements, Abstract, Table of Contents, List of Figures and List of Tables. These headings must be centered. Use the Roman numbers to number the pre pages.

Declaration: Type the information given in the sample below (*Appendix D*).

Dedication: Optional, but nice to have it

Acknowledgements: Thank those who helped (supervisor, friends, parents, etc.) you.

Abstract:

This should be one-page description that comprises of the problem that you address, your approach (users, input, output, and process), analysis & design, implementation, evaluation and conclusion. Write more about your work. Do not use citations, abbreviations and further works in an abstract.

Table of Contents

Here you should list headings of chapters/sections/subsections with page numbers.

List of Figures/Tables

Here you should list Figures/Tables captions followed by names, with page numbers.

Sample for Declaration page

Declaration

We declare that this report is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student: Signature of Student:

Date:

Supervised by

Name of Supervisor(s) Signature of Supervisor(s)

Date: