**BCSH - Project**

**Evandro Gomez Quintino**

2960774

Submitted in partial fulfillment for the degree of

B.Sc. in Computing Science

Griffith College Dublin

06, 2021

Under the supervision of Supervisor’s Name

**Disclaimer**

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the Degree of Bachelor of Science in Computing at Griffith College Dublin, is entirely my own work and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

**Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Acknowledgements

# 

# I dedicate this project to my wife, who always supported my dream to become a game designer and who has been beside me during all the difficulties. And to myself, that beside all the odds, I managed to succeed all the barriers in order to become something that I only dreamed.

**Table of Contents**

[Acknowledgements iii](#_Toc297629416)

[List of Equations v](#_Toc297629417)

[List of Figures v](#_Toc297629418)

[List of Tables v](#_Toc297629419)

[Abstract vi](#_Toc297629420)

[Chapter 1. Introduction 7](#_Toc297629421)

[1.1 “Your area”, (or specific field of expertise, problem name, etc.) 3](#_Toc297629422)

[1.2 “Goals” (or something similar) 3](#_Toc297629423)

[1.3 “Overview of Approach” (or something similar) 3](#_Toc297629424)

[1.4 “Document Structure”, “Document Layout” (or something similar) 4](#_Toc297629425)

[Chapter 2. Background 5](#_Toc297629426)

[2.1 Literature Review 5](#_Toc297629427)

[2.2 Related Work 6](#_Toc297629428)

[Chapter 3. Methodology 9](#_Toc297629429)

[Chapter 4. System Design and Specifications 11](#_Toc297629430)

[Chapter 5. Implementation 13](#_Toc297629431)

[Chapter 6. Testing and Evaluation 15](#_Toc297629432)

[Chapter 7. Conclusions and Future Work 16](#_Toc297629433)

[References 17](#_Toc297629434)

[Appendex I 18](#_Toc297629435)

# List of Equations

Equation 3.1 [ 14](#_Toc297556829)

Equation 3.2 [ 14](#_Toc297556830)

# List of Figures

[Figure 5.1 An example figure. 13](#_Toc297556985)

# List of Tables

[Table 4.1 Breakdown of project marking 12](#_Toc297556987)

# Abstract

The abstract should be at most one page in length, giving a very high-level overview of your project. Your abstract should state the following:

* What your project is
* Why you chose it / what are your goals
* What your project does / what problem does it address
* How you did it / what did you use
* What does it accomplish / what goals were achieved.

This guide is designed to assist you in writing your project documentation. Different chapters contain instructions regarding all aspects of your document and project. In addition, this document is formatted as a template, and can be used as a starting point for your documentation. All of the font sizes, margins, etc. are correct, and the chapter layout presented is an example of what is expected, but will need to be modified to suit your specific project. You should consult your supervisor regarding this. General and overall project/documentation instructions and comments are found in most chapters below, marked with the tag “**General Material**”. Each chapter also contains chapter-specific instructions specific to that chapter. These instructions are marked with the tag “**Chapter Specific Material**”.

It is advised that you look at past project documentation in the library, however bear in mind that guidelines and regulations change over time. Thus you should not use past documentation as anything but to get a general feel of what project documentation looks and feels like. This guide is up to date and supersedes any formatting or parameters of past documentation.

# Chapter 1. Introduction

**General Material:**

First and foremost you must understand what your project is. **It is a major contribution to your degree.** Your project must address several important points, including your ability to:

* Understand the state-of-the-art of a specific area
  + You will need to learn, know and adapt to the most current technologies in your area
* Learn and apply new technologies on your own
  + The ability to work under your own inertia on new/changing technologies is important
* Develop and demonstrate a productive relationship with your supervisor
  + Collaboration and working productively with others is a very important aspect of your abilities. Co-workers, colleagues and bosses will always be there
* Demonstrate a comprehensive understanding of development lifecycle
  + You need to know what a project entails, top to bottom, start to finish

Your work should reflect on your ability to:

* + Take an idea, do background research, understand it thoroughly, and develop and demonstrate an application based on that idea
  + Describe your project in well structured and formatted documentation so that others can understand what you achieved
  + Demonstrate a working application, simulation results, etc.
  + Discuss your project with faculty examiners, answering any questions put to you regarding your project
  + Take something you have learned and push it further – to take it beyond the classroom

Your project should be critical and analytical. Your project and especially your documentation should be focused on not just “doing”, but explaining. Why are you doing this? What does it do? Why are you doing it this way? Why is it different? Why is it better? Why is it worse? Where are its strengths? Where are its weaknesses? Where is it going? Where is it not going? Notice that the word “how” has not yet been mentioned. How you complete your project is only one part of the whole. Equally important to the “hows” are these “whys” – the decisions you made along the way and the results and impacts of those decisions.

The title page, as in this template, is centered and not numbered. All other front matter (Disclaimer, Table of Contents, Acknowledgements, Abstract, etc.) each start on their own page (except for lists of equations/figures/tables if they are short) and are numbered in lower case Roman numerals (i, ii, iii, iv, v, vi, …). The Table of Contents should be automatically generated by MS Word or the word processing package you are using. The same goes for lists of equations/figures/tables. Do not forget to sign and date the Disclaimer of your final bound copy which you will turn in to the faculty.

Starting on the first page of Chapter 1, page numbering should start over (from 1) and count normally (1, 2, 3, etc.). Font should be 12 pt., fully justified, with 1.5 line spacing. Standard MS Word margins are used (as they are in this document). Consistency is the key to your documentation. All headings should have the same style, all margins should be the same, etc.

The exact chapter layout of your documentation should be discussed with your supervisor and made to fit your project. A traditional layout would be the following:

1. Introduction (*Page No 1 Starts here)*
2. Background
   1. Literature Review
   2. Similar work
3. Methodology
4. System Design and Specifications
5. Implementation
6. Testing and Evaluation
7. Conclusions and Future Work

(Un-numbered) Bibliography

(Un-numbered) Appendices

From *Introduction through Conclusion and Future Work* should be between 40 and 50 pages. Lengths outside these limits need to be approved by your supervisor. The subsections of Chapters 3-6 will be specific to your project, and additional chapters may be necessary. You should discuss this further with your supervisor. Remember that the 40-50 page guideline does not include front matter, references, or appendices. Chapters should always start on their own fresh page, never on a page where another chapter ends.

**Chapter Specific Material:**

This chapter is all about your project as a whole, not specifics. It is intended to give the reader a brief summary of the “what”, and the “why”, with a very high-level “how”. Technical details and procedures will be discussed in later chapters.

## 1.1 “Your area”, (or specific field of expertise, problem name, etc.)

In this subsection you briefly state what your project is. A good rule of thumb is to use about 3-4 paragraphs, expanding on what the “what” part of your abstract.

## 1.2 “Goals” (or something similar)

In this subsection you expand again on your abstract but explaining in more detail why you did this project, what you want to achieve, what your goals are, who will benefit from your work and why.

## “Overview of Approach” (or something similar)

In this subsection you briefly describe how you achieved your goals and accomplished your work. This should be done at an abstract level at this point. You can name the technologies involved and why you chose them, etc., but the details are for later.

## “Document Structure”, “Document Layout” (or something similar)

In this subsection (the last of Chapter 1) you describe your document layout. It should read similar to the following:

“The rest of this document is as follows. Chapter Two provides a literature review of the area of <your topic area> and the sources consulted in accomplishing my project, in addition to related work. Chapter Three describes the methodology and high–level design of my project structure. In Chapter Four, I discuss the system design, and requirements/specifications including any hardware and software used. Chapter Five provides the implementation details of my project/system/etc. In Chapter Six, details of the working prototype of my project/system/etc are provided, including my testing and evaluation technique. I also discuss results including any revisions to the overall design and implementation that were deemed necessary. Finally, Chapter Seven presents conclusions and future work.”

# Chapter 2. Background

**Chapter Specific Material:**

This chapter will acquaint the reader to the existing literature in your project area and to related work that has been done. This chapter should contain about 50% of your references.

## 2.1 Literature Review

In this section you present the findings of a review of up-to-the-date literature on your project *topic area and related areas*. This chapter will have many references to some (preferably all) of several of the following: scholarly journals, conference proceedings, books (text books or expert volumes), whitepapers / technical reports, software (including software documentation), etc.

For example, if your project was on parallel computing, you may refer to some of the following:

* Scholarly Journals
  + IEEE Transactions on Parallel and Distributed Systems (IEEE)
  + Journal of Parallel and Distributed Computing (Elsevier)
  + International Journal of Parallel Programming (Springer)
* Conference Proceedings
  + (Similar publishers to Scholarly Journals, above)
* Books
  + University-level texts
  + Other expert volumes
* Software Documentation
  + Product, Company, Version, Authors, etc
  + Only up-to-date, official documentation
* Online Material
  + Only *terminal references,* from official/reputable sources
  + No Wikipedia, random/unknown/un-reputable sites
  + Treated as endnotes, not a reference (more later)

## 2.2 Related Work

In this section you present the findings of a review of *specific works closely related to your project.* You should discuss the similarities and differences between your project and others like it that have been done before. A pointed and convincing argument should be presented as to *why* your approach/technique/etc. is an improvement/extension/etc upon previous work. You do not need to go into the specific details on *how* this is achieved here. This will be explained throughout the coming chapters. For now your job is to bring your reader up-to-speed with the current state-of-the-art, how your project fits into that, and why yours is better!

**General Material:**

References are a very important part of your work and must be done carefully and correctly. References are how readers of your work will connect and relate your work to the work of others and your topic area in general. Without proper referencing your documentation would only be a description of work, not a piece of work on its own, that is related to your field.

**Bibliography Style:**

* The Bibliography (or References) must contain a list of books, journal and conference articles and all other material cited in the main body of text (except websites).
* Entries in the bibliography must contain: author(s), title, conference/journal, publisher, date of publication and possibly other reference-specific information. Consult your supervisor.
* Each entry in the bibliography is numbered consecutively in order of appearance, such as [1], [2], etc. These citation numbers are included in the main body of text in square brackets.
* All bibliographical information is exclusively included in the list of “References” section at the end of the document, next to the respective citation number.
* Please see the following example.

**Bibliography Example:**

**Main body of text:**

“A prefix labelling technique presented in [1] seems to be appropriate for topologies similar to a Tree structure. Our research focuses on developing a prefix labelling technique of B-Tree topology [2]. Once a network is organized as a B-Tree the prefix can be calculated using a distributed process as suggested in [1]. The ultimate aim is to achieve load balancing for distributed systems [3], [4] organized as a B-Tree topology.”

**Bibliography (or References) (placed at end of documentation):**

1. C. Li and T.W. Ling. An Improved Prefix Labeling Scheme: A Binary String Approach for Dynamic Ordered XML. *10th International Conference on Database Systems for Advanced Applications, DASFAA 2005*, Beijing. Volume 3453*/*2005 , April 2005, pp.125 *−* 137.
2. D. Comer. Ubiquitous B-Tree. *ACM Computing Surveys (CSUR)*,Volume 11, Issue 2, June 1979, pp. 121*−*137.
3. Ka-Po Chow, Yu-Kwong Kwok. On Load Balancing for Distributed Multi-agent Computing. *IEEE Trans. on Parallel and Distributed Systems*, Volume 13, No 8, 2002. pp. 787 *−* 801.
4. M. H. Willebeek-LeMair, A. P. Reeves. Strategies for Load Balancing on Highly Parallel Computers. *IEEE Trans. on parallel and distributed systems,* Volume 4, No 10, Sep 1993.

**End of Example**

All referenced websites must be *terminal references*. A terminal reference is the “root” reference for a specific idea/theory/concept/product/technology/etc. For instance, *no* Wikipedia pages are terminal references. All Wikipedia pages have references to other (possibly terminal) references! Another way to view a terminal reference is as the reference where the “chain of reference” stops. A chain of reference is (for example) when someone discussing *concept A* refers to a website, which refers to a Wikipedia site on concept A, which refers to another website, which refers to a book, which refers to another book, which then refers to a scholarly journal article, which presented concept A for the first authoritative time. That journal article is the terminal reference for concept A. In some cases, websites may be terminal references, however this is rare. Another time that a website may be referenced is when the website is the official website or portal to a specific tool/technology/etc.

Websites *are not* included in the bibliography/references section. They are included as footnotes.

**Example:**

“Although the number of proteins with known structure continues to grow, the number of proteins with known sequence, but unknown structure is growing faster. Thus the gap between the number of proteins of known structure and the number of proteins of known sequence is growing. As protein structure dictates protein function, and proteins of similar sequences often have similar structure and therefore function, databases of protein sequence and structure information such as UniProt[[1]](#footnote-1) have become increasingly useful to both those who are sequencing proteins, and those that are predicting protein structure.”

**Note the footnote at the bottom of this page, corresponding to the 1 above. This is how websites should be cited.**

**End of Example**

All references other than websites should be added in MS Word as a “citation” – usually Insert > Citation, or something similar. This will give you options on how to present your references, and allow you to automatically generate your bibliography, similar to your table of contents.

# Chapter 3. Methodology

**Chapter Specific Material:**

This chapter *begins* to explain how you accomplished your project and your objectives. Explain here what was needed to implement your project, both in technology and effort. Discuss the high-level decisions you made. These are your design decisions. Why did you choose technology x instead of technology y? Why did you choose a top-down approach instead of a bottom-up? If you chose a divide-and-conquer paradigm to a specific problem, why did you, and why did you not choose a brute-force or greedy approach? The subsections of this chapter will be specific to your project and should be discussed with your supervisor.

This chapter does not need specific details *about* your chosen technologies. For example, you do not need to explain what java is or what cloud computing is. What you do need to discuss is *why* you chose java, or *why* you chose a cloud platform. If you did choose java over C#, why? If you did choose cloud computing, why did you choose PaaS instead of SaaS or IaaS?

**General Material:**

The role of your supervisor includes the following:

* Guiding you in the right direction relating to your project idea
* Identifying and suggesting the technologies that might be useful to implement your idea
* Identifying and setting up project milestones and deadlines
* Monitoring progress, milestones and deadlines

Remember that you were told about your project almost a year ago, at the start of your Research Methods module. This is when your project started! The summer period is for implementing, not starting your project idea! You should check with the faculty and your supervisor regarding the demonstration and documentation submission dates. You should aim to be done well before these deadlines. You will find that once you are “done” and have removed some pressure from yourself, you will go back and make many important changes that you wouldn’t if you were rushing to meet the deadline. You will also give a much better presentation if your finished project has had some time to mature.

You are highly advised to keep a “progress log” while you are working on your implementation. This will be an invaluable help when you begin your documentation proper. Document everything! Document the bad decisions, the mistakes, the things that didn’t work, as well as those that did. They will all contribute to your documentation and help you answer and explain the big questions, what, why, how.

# Chapter 4. System Design and Specifications

**Chapter Specific Material:**

This is the first chapter where you can describe specifics. What technologies did you use? What vendor/version/etc.? What features of these technologies made you decide to use them? How were they helpful? How were they difficult? What might have been better? Did you have compatibility issues? Did you have any other issues?

You will also need to discuss and present *your* system architecture/model. How did you use your different technologies/platforms and how did they work together? Did your architecture use a tiered system? How many tiers? How are they separated both logically and in implementation?

In this chapter you can use architecture diagrams, code snippets, UML diagrams, formal diagrams, etc. All of these can be included as figures (see Chapter 5).

A note on code snippets: Code snippets should be just that – snippets. Snippets are small, core pieces of code that are integral and unique to your project. Typically a snippet is 10-15 lines of code. *Long segments of code, general structure, headers, etc. should not be included in documentation*.

You can also describe your process modelling (software) and data modelling here. Data modelling can include the type of input/information your system needs, the use and processing of that information, and what your system generates (output).

**General Material:**

Assessment and Evaluation:

*The project will be evaluated on its quality of thought, interpretation and insight as well as the contribution it makes to the field of study and the writer’s own professional development. An essential ingredient will be the student’s ability to master a technical body of knowledge and apply it to a given problem domain. The ability to think and reason with the material at issue is crucial. The design, layout, quality of expression, structure and coherence of all documentation will be taken into account when grading the finished work. The ability of the student to present and defend the material is also of significant importance.*

Marking:

Your project is marked according to Table 4.1.

|  |  |
| --- | --- |
|  | **Marks** |
| **Development Process** | 20% |
| **Interaction with supervisor** | 10% |
| **Product application (demonstration)** | 40% |
| **Documentation** | 30% |
| **Total** | 100% |

Table 4.1 Breakdown of project marking.

# Chapter 5. Implementation

**Chapter Specific Material:**

This chapter discusses specifically how you implemented the working version of your system. The specifics of this chapter should be discussed with your supervisor. Screen shots *may* be appropriate in certain circumstances in this chapter and subsequent chapters. See the note on screen shots in General Material, below.

**General Material:**

A note on Screenshots: Screenshots should be used carefully and sparingly. Only use those that are very explanatory in nature. A good rule of thumb is that if it would take many lines of text (with a footprint larger than that of the screenshot, caption, and brief text explanation combined) to explain what you want, use the screenshot/caption/brief text explanation. Otherwise just describe in text.

Tables, figures, and equations may appear throughout your documentation. All tables and figures should be centered and have captions. Captions should be inserted in MS Word with “Caption…” after selecting or right-clicking the table/figure. This will allow you to generate a list of tables and/or figures for your front-matter if desired, similar to your table of contents. For an example, see Figure 5.1.



Figure 5.1 An example figure.

Equations may also appear throughout your documentation. If you refer back to a particular equation in your documentation more than once, you should center and number the equation, allowing you to refer to it by number, and to allow you to generate a list of equations, similar to your lists of tables/figures.

**Example:**

Einstein’s famous mass-energy equivalence is given by Equation (3.1),

 **3.1**

Where *E* is energy, *m* is mass, and *c* is the speed of light. Equation 3.1 can be rearranged to an expression for the speed of light as in Equation 3.2.

 **3.2**

Equations 3.1 and 3.2 are unit independent and dimensionally consistent.

**End of Example**

Equations should be inserted into word with: (Insert > Equation, or Insert > Object > Microsoft Equation 3.0), depending on your version of Word.

# Chapter 6. Testing and Evaluation

**Chapter Specific Material:**

This chapter should include a description of the process or processes you used to test and evaluate your system. You can use things such as user experience reports, attempts by yourself or others to break your own code, graphs/charts of outputs or performance, etc. Include discussions of why things work, and why and when they don’t work. You can also include any refinements made to your implementation as a result of your testing.

**General Material:**

Your project demonstration will occur at the end of the project period. Check with your supervisor for the exact date. You will be examined by two teams, each consisting of two faculty members. You should prepare 5-8 slides briefly outlining your project idea. You do not need to go into implementation specifics in your slides. It should take no longer than 3-4 minutes to go through your slides (approx 30 seconds per slide).

Your first slide should contain the following information for the examiners:

* Name
* Student Number
* Project Title
* Supervisor Name

After your slides, you should answer any questions the examiners may have for you, and then show them your project in action. This is when you should walk through your application from a user’s point of view. After this you should be ready to have the examiners ask you to see code, explain how certain functionalities are implemented, etc. Your entire demonstration should be no longer than 20 minutes, including examiner questions. **A good recommendation for the timeline of your demonstration is the following: 5 minutes – Intro/Slides, 5 minutes – Demonstration from user’s point of view, 10 minutes – Questions from examiners, TOTAL – 20 mintues.**

# Chapter 7. Conclusions and Future Work

**Chapter Specific Material:**

The first part of this chapter should present your conclusions. A good way to do this is to take your abstract and address all of the goals and objectives stated there. Tell the reader what you achieved, and very briefly how and with what. Be sure to highlight major successes and to note limitations.

The second part is future work. This should include a discussion on what you would do if you had more time and how you would address the current limitations of your system.

This chapter should be no longer than three pages in length.

# References

This section should list your references as outlined in Chapter 2, General Material. The references section should be page-numbered, but is not a chapter and therefore should not have a number itself. It should be listed in your table of contents, and be the last thing in your document unless you have appendices.

General Material:

**Do not forget, the last things you need to do:**

1. **Spell/grammar check**
2. **Update Table of Contents, all lists, etc. (Right click on TOC or list and select “update field”, then select “update entire table”**
3. **Sign and date your declaration**
4. **Get two copies bound at a bookbinder – one hard bound and one soft bound. These are for the faculty. You can order more for yourself if you wish**
5. **Burn all documentation, code, etc. to a CD/DVD and turn this in to the faculty or supervisor with one hard bound and one soft bound copy at your demonstration.**

# Appendex I

One or more appendices may be necessary but should be approved by your supervisor. Appendices should be used for material you would like to refer to such as figures/diagrams/code/etc, but are deemed too large and bulky for the main text, or outside the “flow” of any particular chapter. Appendices should be page-numbered, and numbered with capital Roman numerals (I, II, III, …). This is not a place for large pieces of code. (There is no place for large pieces of code!) Appendices are the last section(s) in your document.

1. http://www.uniprot.org [↑](#footnote-ref-1)