**IMAT3451 Final Year Project: Writing the Final Deliverable**

Remember that the final deliverable is supposed to be both an overview and a critical evaluation of your project. Relegate any low-level detail to an Appendix, for example:

* If the product was a game, you would need to provide the objective of the game, a summary of how it operates and the basic rules in the main body of text but the detailed rules would be in an Appendix.
* If the product was a comms protocol, you would need an outline of the operation in the main body of text but the precise protocol formats and rules, together with any supporting RFCs should be in an Appendix.
* If the project was a piece of research, you would need a description of the research background, context and methods you used, with a summary and discussion of the findings and conclusions, but any actual questionnaire, consent letter or other survey instrument would be in an Appendix.

Remember that the final deliverable may be read by a moderator or by an External Examiner; they need to know not only what you did but also why you did it the way you did and what you learned from the project. Include appropriate illustrations to support your descriptions and put lots of focus on “why” as well as “how” in your deliverable.

The project documentation and lectures you received provided general guidelines on how to format the final deliverable. These guidelines are meant to supplement rather than replace them.

The **Final Deliverable** (main report) is typically structured into the following sections:

* Introduction
* Body
* Conclusions

The **Introduction and the Conclusions** should each occupy about 15% of the final deliverable overall. The quality of these sections is normally the main discriminator between a good deliverable and an average one.

**The Introduction** provides the context for the rest of the deliverable. Some supervisors may ask you to write this first; others may require that you write this at the end when you have a clear picture of what you did, how you did and what you’ve learned. Generally, it should cover about 3-4 pages:

* What is the product? What are the main functions it performs? Provide a diagram or screen shot so a moderator has some idea what it looks like, for example if your project was to provide a newsfeed reader, give some background on how newsgroups are organised, how the relevant protocol operates and a screen shot of an existing client program
* If your project is a piece of research, this becomes What is the research? What question were you trying to answer? How did you intend to investigate? Give some background on the research context and other related published work.
* Why is such a product needed or why is it worth doing? Your own personal skill development is a perfectly valid reason for doing the product
* If your project is a research project, this becomes Why is your research question relevant or worth looking at? Your own personal skill development is a perfectly valid reason for doing the research
* What did you hope to get out of doing it (academic objectives)

Some supervisors may ask you to write **the Conclusions** after the Introduction, or vice versa. The Conclusions section should cover 3 things (in about 5-6 pages):

* Product evaluation:
  + How much does it do? How good is it?
  + Are there any bits you are particularly proud of?
  + What isn’t implemented? How would you extend it given more time?
  + You may use first person active voice for this section
* Or, research evaluation:
  + How much of the research question did you answer? Is it a complete answer?
  + Are there any aspects you are particularly proud of?
  + What did you not get to research? What further research would you do if given more time?
  + You may use first person active voice for this section
* Evaluation of your approach:
  + In terms of project management approach and in terms of development methods or research methods used
  + How successful was the approach you adopted?
  + What techniques did you use that worked well? Why?
  + What did you learn by doing the project – did you met your academic objectives?
  + How would you do it better if you did the project again?
* Evaluation of tools used:
  + What languages, libraries, environments etc did you use for the development? Or, what libraries, methods, techniques did you use for the research?
  + Were they suitable? What were their main plus and negative points?
* And a rounding-off paragraph to end on a positive note

Now write the bits in the middle. Avoid a narrative (“I did this, then I did that...”) style and concentrate on highlighting the major decisions you made at each development stage and the reasons for your choices.

Why do it in this order? So that you focus on the overview for both the introduction and the conclusions. Then you can delve into the detail for the main body.

**Examples of Final Deliverable Structures**

There are two common ways of structuring the main body of the deliverable.

**1. by Activity Type 2. by Product Component**

e.g. e.g.

Analysis of Requirements Underlying Data Structures & Algorithms

Design Considerations User Interface

Implementation A.I. Interfaces

Networking Aspects

If you choose the first structure

* Analysis of Requirements should cover:
  + What were the major components of the project; typically you can break it down into a number of elements e.g. the basic functionality, the interface, comms elements, etc. Use diagrams or illustrations to highlight this if appropriate.
  + What did you identify as the most important requirements for each of these aspects?
  + Why are these the most important? How did you identify them (e.g. surveyed existing products in the same area)
  + If you do some research and have a literature review, this is where you put it – because your research is part of analysing the requirements
* Design Considerations should consider:
  + What were the main design decisions you had to take?
  + What alternatives did you consider and why did you make the choice you did
  + Use diagrams/ screen shots / code fragments etc to illustrate any significant data structures, GUI structure, background algorithms etc
  + If you have learned new concepts from your research, this is where you are likely to demonstrate how you applied them to your project, with brief mentions or referring back to the literature or literature review
* Implementation should consider:
  + What were the main problems you encountered in implementing your design?
  + How did you solve them, or if you couldn’t, how did you work around them? Again, use diagrams / screen shots to illustrate.
  + How did you test your implementation? How did you maintain the project code / documentation whilst you were developing it?

If you choose the second structure, you should cover the same material but discuss the requirements, design considerations and implementation techniques for each component in turn.

If you did a research project, you are likely to use a structure similar to what you find in published journal articles:

* Introduction or background
* Literature review
* Research methods used
* Findings or results
* Analysis and discussion
* Recommendations / conclusions

Remember that there is a **word count** for the Final Deliverable (main report). For research projects 15,000, for development projects 10,000 and for hybrid projects 12.500 (excluding references and appendices). You can go below these counts but not above.

Naturally, the actual deliverable may be a little shorter or a little longer, so there is a **10% allowance** but be careful to use the amount of words wisely; your manuscript needs to be structured, concise and interesting, but not too short so that integral information is missing.

In all cases, there may be an issue if you go significantly over the word count. We will be adopting the recommended Faculty policy for maximum word counts, which is as follows:

* Deliverables can exceed the word limit by up to 10% without penalty.
* A penalty of 20% of the marks is then applied for deliverables that exceed the word limit by between 11% and 30%. The part of any deliverable that exceeds the word limit by over 30% is not marked and hence will not contribute to the final mark.
* The penalty to be used is to stop marking the work once the 30% limit has been breached, and tutors will clearly indicate on the student’s work where the marking has stopped.
* All work that is subject to penalty because it breaches the word limit will be second marked and/or moderated, and the reduction must be consequently approved.

**The Appendices**

When deciding what to put in the Appendices, ask yourself these questions:

* What would a user need to run this application?
* What would a programmer need to maintain this application?
* What does another researcher need to replicate my study?

That implies that the appendices may contain any of the following:

* *Detailed context setting information:* This is the background information which either a user or a programmer may need to understand exactly what the application is trying to achieve. For example, if the application is a well-known game, the user won’t need a user manual on how to play it but they will need a complete definition of the rules that have been implemented and possibly screen shots illustrating different aspects of the game. If it is a novel strategy game, they may well need a user manual to explain how it is played. If the application is a complier, this part of the Appendices should contain a complete definition of the grammar. If the application supports some business process, for example a helpdesk or an appointments diary, the user may need an overview of the functions available to them and what part of the business process each function supports.

The other type of information that a programmer needs is an explanation of any special techniques or utilities that have been used in constructing the application. If the application relies on the use of AI mechanisms or mathematical algorithms, the necessary background and references should be provided. If it supports a specialist application area (e.g. music), some description of this area will be necessary. Similarly, if the project incorporates special purpose utilities, (e.g. MIDI, parsers, graphics libraries, etc.), an outline of how they are used should be provided.

* *Detailed technical information:* This is the background information that explains any non-obvious aspects of how the application was implemented.

1. As an absolute minimum, this section should contain the source code **in electronic format** and release notes detailing how to build the application and any constraints on the platform e.g., a particular operating system or library version. It will also contain details of any test plans.
2. It should also contain detailed structural information such as class/module/data diagrams with documentation explaining the function and usage of each major element. The documentation techniques you use for this should be those which you consider appropriate.
3. There should also be an explanation of any complex algorithms involved or novel coding techniques used.

* *Historical status information:* The appendices may also include historical status documentation for example things used for project management, such as Terms of Reference/Project Contract, Ethical Review Form, Project Plans, Progress Reports, Personal Diary notes etc.

The appendices need not all be electronically produced. In particular, graphical information may be better produced manually, and then scanned and inserted in the electronic file as a picture.