Applied Project & Minor Dissertation

Structure and Elements of Dissertation

This module…

- *Design, develop and deploy a project, either individually or in a group environment, delivering a piece of software in a timely and standards-driven manner.*
  - Experience working on a project similar to one encountered in the software industry, in collaboration with academic supervisors and technical staff.
  - Topic selected from a list or proposed by the student subject to suitability /availability of resources. Project must deal with a task or problem deemed to be of sufficient technical challenge and depth.
  - Research Methodologies module provides the appropriate skills and knowledge to undertake a dissertation.
- *The final dissertation should be in the order of 10,000 - 15,000 words, excluding appendices.*
    - Circa 450 words / page (Times New Roman, 1.5 spacing)
    - Approximately 35 pages (excluding diagrams, abstract, TOCs, references, appendices etc..).

Learning Outcomes

- Demonstrate the application of *appropriate research methodologies* and techniques related to software development.
- Demonstrate an awareness of the *present state of the art* in a specialist computing area including the ability to evaluate the literature base.
- Integrate disparate technologies and principles to successfully develop and *deliver an appropriately integrated solution* to a computer-based project.
- Apply research and *critical thinking skills* to a challenging computer-based problem.
- Evaluate, select and apply standard and customised *research tools and methodologies of enquiry*.
- Design and *implement a computing solution* that requires preliminary research.
- *Critically evaluate* the work and research and reflect on the strength, weaknesses and future potential of such work.

How many of these outcomes are evidenced in the:

- Software?
- Dissertation?

Document Structure

- *Latex template available on Moodle. You must used the structure specified.*
  - Document must be free from typos and grammatical errors.
- **Basic Structure**
  1. Cover Sheet
  2. Table of Contents
  3. Table of Illustrations
  4. Introduction
  5. Methodology
  6. Technology Review
  7. System Design
  8. System Evaluation
  9. Conclusion
  10. References
  11. Appendices

- You should start writing as soon as possible.
- Items 1-3 can be started today.
- Items 4 and 5 can be completed by Christmas!

Cover Sheet

- *The latex template already has the following required elements:*
  - College logo
  - Title of project
  - Title of degree programme
  - Name of team members
  - Name of supervisor / academic title
  - Name of Industrial Supervisor / company (Industrial Projects Only)
  - Submitted Date
- *Latex will automatically generate the Table of Contents and Table of Illustrations.*

Introduction (5 Pages)

- *Provide a clear context for your project.*
  - What is it about? Is it at the right level (8)?
  - Is the scope correct?
  - Do not assume that the reader knows anything about the domain.
  - Why should a reader care or be interested?
- Set out the objectives of the project clearly.
  - You will have to address each of these in the evaluation / conclusion.
  - The metrics by which success or failure is measured.
- Briefly list each chapter / section and provide a brief description of what each section contains.
  - List the resource URL (GitHub address) for the project and provide a brief list and description of the main elements at the URL.
- *After reading the introduction, a reader should be 100% certain of what the project is all about and why it is relevant.*

Methodology (3 -5 Pages)

- *Describe the way you went about your project. Was your approach to the problem valid?*
  - Software development v/s Research methodology.
- Agile / incremental and iterative approach to development.
  - Planning. Did you storyboard? How did you determine the requirements for the project?
  - Meetings. Frequency, structure, checks & balances, feedback.
- What about validation and testing?
  - Junit or some other framework.
- If team based, did you use GitHub during the development process? What about other development tools?
  - Selection criteria for algorithms, languages, platforms and technologies.
- Was an empirical approach used? How were problems solved?
  - Was any research undertaken first?

Technology Review (10+ Pages)

- *The "literature review" part of the dissertation. Should be tightly coupled to the context and objective from the introduction.*
  - Proves that you researched what you were doing!
- Describe each of the technologies you used at a conceptual level.
  - Standards, Database Model (e.g. MongoDB, CouchDB), XMl, WSDL, JSON, JAXP.
  - Use references (IEEE format, e.g. [1]), Books, Papers, URLs (timestamp)
  - Sources should be authoritative!
- A technology review that includes a lot of *de facto* or *de jure* standards supports the methodology!
  - Each chapter buttresses some other aspect of the dissertation.

System Design (n...m pages)

- *Provide a detailed explanation of the overall system architecture. The **HOW** of the project.*
  - System designed should be informed by the technology review, i.e. you applied the knowledge that you learned doing the research…
  - Standards-based where possible. How are components coupled?
  - Cloud hosted – IaaS / PaaS / SaaS.
- Use diagrams to augment an explanation of the architecture used.
  - Provide a comprehensive overview of the different components of the system and how they work together.
  - UML class, sequence and interaction diagrams.
  - Course and fine grain.
  - Use screen shots of forms or other UI components.
- Page count range difficult to state as varies significantly between projects.

System Evaluation (n...m pages)

- Evaluate your project against the objective set out in the introduction.
  - Prove that your software is robust. How?
    - Unit / acceptance testing for robustness / behaviour.
    - Stability metrics for structure.
  - Any tables / graphs of results belong here.
    - Provide an accompanying discussion.
  - Use performance benchmarks (space and time) if algorithmic.
  - Measure the outcomes / outputs of your system / software against the objectives from the Introduction.
- Highlight any limitations or opportunities in your approach or technologies used.
  - Identifying limitations is not a sign of weakness. It is proof of insight.

Conclusion (3 - 5 Pages)

- *Briefly summarise your context and objectives.*
  - Remind the reader about the overall rationale and goals of the project.
- Highlight your findings from the System Evaluation chapter.
  - List out the outcomes of the project in a bulleted list.
  - Serendipity – did you gain any tangential or even unrelated insights by happenchance during the project?
    - Lots of discoveries have been made this way, e.g. Flemming and antibiotics.
  - State any opportunities identified for future investigation.
- Finish on an upbeat note.

References and Appendices

- *References should all be in IEEE format and correctly collated.*
  - Use Google Scholar's settings to specify Bibtex as the "Import into…" option.
  - Can build the bibliography as a separate Latex file or as an inline document.
- *The appendices should contain the URL of the GitHub repo and a full set of installation instructions if applicable.*