CSC3002 Computer Science Project

Student Handbook 2018-19

IMPORTANT: This is a double module worth 40 CATS points; normally one of the conditions for the award of a BEng/BSc degree in Computer Science is that CSC3002 is passed.

ChangeLog

6 August 2018	Original Draft

1. *Important Dates (all subject to change)*

Project Selection by	Wednesday 3rd October 2018, 4pm
Initial Project Description and Work Plan emailed to supervisor by	Friday 19th October 2018, 4pm
Problem Description, Solution Approach and Work Plan via Turnitin	Tuesday 11th December 2018 4pm
Interim Demo to supervisor completed by	Friday 14th December 2018
Demonstration and Expo Day	Wednesday 1st May 2019
Final Dissertation submitted to TurnItIn	Monday 29th April 2019 4pm

2. *Module Co-ordinator*

The module coordinator is Dr Des Greer. Email: des.greer@qub.ac.uk

3. *Learning Outcomes*

On completion of this module, the successful student should demonstrate competency in the techniques necessary for a rigorous and disciplined approach to software construction. As part of this students should demonstrate:

- an ability to analyse a problem area using a range of sources and to synthesise the relevant information into a proposed solution approach;
- innovation, creativity as well as practical and analytical skills in providing a software solution to a problem area;
- competency in implementing a substantial and robust software product and to self-manage the project;
- ability in critical appraisal of their own work in relation to a wider context; and
- ability in documentation and report writing.

The marking scheme will reflect these learning outcomes. A student guide to the marking scheme is given in Appendices 1 and 2.

4. *Computer Science projects*

Projects are arranged with members of staff as supervisors. You will be allocated a supervisor near the start of the term. Computer science projects are completed as individual projects. Typically, projects are set in an area of interest to the member of staff's research, but alternatives can be arranged with individual members of staff. ALL projects must be approved in advance by the allocated supervisor.

The project is a means of utilising the skills and knowledge you have acquired from the modules that you have studied (as well as, possibly, your professional experience). Coding is only one aspect of a project! Problems that on the surface seem relatively straightforward often have hidden depths. Successful projects require organisational as well as technical skills. You may have to develop new skills and gain new knowledge in the project and the assessors should take this into account when marking.

5. Selection of project

A list of project areas will be published prior to and during the first week of term. Projects available will have a brief description of the area to be investigated and/or the software to be developed and other relevant information. Students will create a shortlist of 5 projects and then rank their project preference by Wednesday 3rd October 2018, 4pm. A URL link will be sent to by email for you to complete this online. The allocation will be done shortly after the selection deadline.

Please note that there is no guarantee that you will be allocated a project in the area of your choice (because of supervisor availability constraints).

6. Self-defined projects

Defining your own project is possible, but you must find a member of staff who is willing to supervise it. The project must be something you haven't worked on already and you must adhere to the rules of normal projects and the constraints for final year projects. If you wish to work with an outside company, be aware that this is often difficult, since the needs of the company may not match those of the final year project. The needs of the company are secondary and they should be made aware of this. Entrepreneurial projects are also possible but again you will need to find a willing supervisor and the needs and constraints of CSC3002 are the priority. In all cases the work must be new and the supervisor can insist on a given direction in the project. If you have arranged a project with a supervisor, please get the supervisor to email the module coordinator and await approval before proceeding (des.greer@qub.ac.uk).

7. Course Organisation and Timetabling

The dates and deadlines for the various stages of the project are given at the top of this document

Deadline extensions are not normally granted (e.g. a 2 day absence due to illness in the 1st semester will not result in an automatic extension of two days to the dissertation submission date). There is only one initial lecture in semester one and if needed another one in semester 2. All necessary documentation will be posted on QOL.

8. *Supervisor*

Your main point of contact for project advice is your project supervisor. You should arrange a meeting with your supervisor and agree a project with them as soon as possible.

9. *Course Background*

- The Computer Science Project is a project-based course. There is no examination. The work involved with a CSC3002 project should be similar to the work associated with two conventional modules (including time for examination preparation).
- Marks are allocated for project work submitted on or before the dissertation deadline.
- Project work is of special interest to employers and is often discussed during interviews. Project work is often cited in academic references.
- Group work is **not** permitted.

10. *Project Roles*

- Student: The student is responsible for ensuring that the project is carried out effectively and completed on schedule. This is a double weighted module so it should have at least twice the effort of a normal module. It is also one that employers like to talk about at interviews. Good project management is important and there should be a record of this via planning documents, designs, commits, and meeting minutes. There is a template for this on QOL. It is the responsibility of the student, not the supervisor, to ensure that his/her project is completed satisfactorily. It is the responsibility of the student to arrange regular meetings with the supervisor during the project, and record minutes of meetings. Minutes should be uploaded to the repository in a top level subfolder named Minutes.
- Supervisor: The supervisor is normally responsible for providing technical advice and guidance to the student.

If you have a query regarding a technical aspect of a project, then it should be discussed with your supervisor. If you have a non-technical problem, then it should be discussed firstly with the supervisor and then with the Module Co-ordinator. Additional technical support may be available for some projects depending upon the hardware and software used. **The staff providing this assistance must not be used as surrogate supervisors.** You are expected to communicate with your supervisor and the Module Co-ordinator via e-mail as well as face to face.

11. *Organisation*

Planning is critical to the success of the project. Plans help you determine the scope of the project, track its progress and to support decisions. The plan does not have to be right first time, but should be corrected as you learn during the project. In any case, you should devise a plan (e.g. Gantt chart or table) which outlines major predicted milestones and deliverables from the start. This should be kept updated as the project progresses and the products evolve. It is your responsibility to organise meetings with your supervisor to keep him informed of progress. The choice of software process is open to discussion with your supervisor. Some projects suit a linear process better, while others will need to accommodate change and learning as the project processes. Experience shows that delivering valuable functionality in the project early in semester 1 and continually tends to indicate a good project outcome in semester 2. Students who do not demonstrate steady and consistent progress throughout the period of the project often produce inadequate deliverables.

The total amount of time available for completing the project and the associated dissertation is approximately six months. In order to complete the dissertation on time, the dissertation may need to be started in parallel with the later stages of the development work. You should aim to complete all development work on the project before the end of March or even earlier.

12. *Problem Description* *(Friday 19th October 2018, 4pm)*

Once you have been allocated a project, you have to develop an initial problem description and work plan. This should give a good overview of the problem to be solved. Goals and requirements should be stated clearly. A description of a software / hardware development environment may be given. Verifiable criteria against which the success of the project is to be judged should be identified (e.g. features of the software and experimental results). You may state these as acceptance tests, if preferred. A Gantt chart or table which outlines the expected project development plan should be given (with major milestones and deliverables highlighted). Appropriate references should be provided.

13. *GitLab*

You are expected to use the school GitLab facility to store your code and other artefacts as you develop them. Access details about this via the EEECS Self-Service web pages. Additionally, you must agree that staff have access to the project repositories, to view, to be able to clone them for private browsing and to view and summarise statistics on the repository activity and content.

It is the student's responsibility to become familiar with using git. Guidance can be found at https://selfservice.eeecs.qub.ac.uk/docs/git/ .

Please ensure the readme.md file contains a brief description of the projects (a few sentences will do) and an explanation of the project file structure and contents.

Please make sure you regularly commit code (and other artefacts as they are generated) via GitLab. Staff may monitor your activity to ensure progress is being made. **Staff have been instructed to reward**

students in the interim report and demo mark for showing good project management including maintaining a repository for managing project artefacts. Make sure the readme file is meaningfully populated and you can show a history of regular commits.

14. User Interface Design

If you have a system with a User Interface, it is recommended that early in the project you use a rapid wireframing tool such as Balsamiq to prepare sketches of screens.

15. Project Description, Work Plan and Interim System Demonstration

As the first semester progresses and you learn more about the problem area, you should update your problem description and work plan document. This should then be handed in via TurnitIn by **Tuesday 11th December 2018 4pm** . Before **Friday 14th December 2018**, you must arrange an interim demonstration in conjunction with your supervisor**.** At the demonstration you should illustrate how much progress you have made with respect to a subset of requirements. These should be agreed with your supervisor at the outset of the project and only changed by agreement. Typically, an interim demonstration consists of the execution of a significant part of your system and may include initial experimental results. Having some mocked up some screens or some minor development work is not considered acceptable. There should be demonstrable value in the system.

16. Project System Demonstration

Each student will be required to give a demonstration of their completed system to their Supervisor and to their Assessor on **Wednesday 1st May 2019** (demo/expo day). The supervisor and assessor will assign a mark to the system produced, based on how it has met the project goals and on the quality of the product. If a system is incomplete but partially demonstrable then marks may be awarded for the completed portion of the system. Local employers from industry may attend system demonstrations. Demonstrations provide an opportunity for you to bring your work and skills to the attention of employers.

17. Dissertation Structure

The dissertation structure should be guided by the advice of your supervisor. A suitable template is provided in appendix 3 of this document. There may be circumstances where a variation in the structure is appropriate; such changes should always be discussed with your supervisor.

---

18. Copyright of Material Supplied

Material supplied at the outset or during the project may be the subject of copyright (or other industrial property rights) and other such material may be created during the project. An example of this would be if, at the outset of your project, you were supplied with software by your supervisor that you incorporated in your final project you could not then 'market' the software that you produce in your project as your own. The University's stance on this is

---

Queens do not lay any call on IP generated by undergraduate students. The only exception is where the undergraduate has had input from a member of QUB staff and they have shared ownership rights in the invention - in such cases we will seek to protect the return to our member of staff.

If you have any queries on this then contact me and I will put you in touch with the relevant body within Queen's that deals with IP.

19. Allocation of Marks

Problem Description, Solution approach and Work Plan (Semester 1)	10%
Interim Demonstration (Semester 1)	10%
Dissertation	40%
System Produced (including demonstration)	40%

20. What You are Required to Submit: A Summary

1. A **Problem Description** (max: 4 pages) should be emailed to your supervisor by Friday 19th October 2018, 4pm.
2. An **Updated Problem Description, Solution Approach and Work Plan** to contain the updated problem description, a solution approach including verifiable criteria for success of the project and a work plan (max: 8 pages) by Tuesday 11th December 2018 4pm .
3. An **Interim Demonstration**. This normally comprises a demonstration of a partially working system which satisfies a *substantial* number of the requirements agreed in advance with your supervisor. Demonstrations must be complete by Friday 14th December 2018 as arranged with your supervisor.
4. An electronic submission of your dissertation as a pdf via Turnitin (max. 45 pages for main text). Details of how to do that by email/qol) . Deadline Monday 29th April 2019 4pm.
5. The Demonstration Day to Supervisors and Assessors will be held on Wednesday 1st May 2019. **All students are required to attend demonstration day.** Details of the schedule will appear closer to the time.

21. Dissertation and System Assessment

Each project dissertation will be assessed by the supervisor and up to two assessors not associated with the supervision of the project.

22. Late Submissions

All pieces of work must be submitted before the stated deadline.

University regulation: <u>a 5% penalty is incurred for each day that a submission is late.</u>

23. Hardware/Software Platform to be Used

The projects must be developed on the hardware/software platform agreed with the project supervisor. You are responsible for ensuring that adequate precautions are made to prevent the loss of data.

24. Dissertation Submission Format

A PDF version should be submitted via TurnitIn..

Document Layout – Documents not conforming to this will be returned.

Documents must:

- be prepared using a word processor.
- Use Times Roman 12pt font.
- Have a page number on each page
- Have a Justified Layout with 2.5 cm (one inch) margin on all sides
- Must have a line spacing: 1.5 lines

Please read your submissions carefully and check document spelling using spell checking tools. The main text **should not** exceed 45 pages. Most assessors prefer a passive voice is used in the text e.g. Rather than writing "I developed the system using Java", use "The system was developed using Java".

You are required to submit a pdf version via the Turnitin web service. More details of how to do this will appear closer to the deadline. This will check for plagiarism. Be aware that this service detects and highlights any text that is similar to existing sources. You will have a chance to review and correct the document in TurnItIn before finally submitting.

25. Background Reading

Supervisors may provide references to suitable background material for the project. However, it is the student's responsibility to read around the area using books, articles and web based material. There are various search tools available via the library site (Use the Article search to obtain research papers). Google Scholar is also very useful.

26. Use of other resources

- You must not make any use of any projects which are available online unless approved by your supervisor (any such projects should be referenced clearly). Project submissions will be checked for plagiarism (via Turnitin).
- You are required to complete the short tutorial on plagiarism at http://www.qub.ac.uk/cite2write/introduction5.html
- DO NOT copy text from other sources unless placed in quotes and cited.

- Even if you write something in your own words but it is based on an existing source, please place a citation in the document to the source.

27. Level of Project Difficulty

The project is meant to stretch your ability. On the one hand, if you are given a problem which is beyond you, you may not be able to deliver anything much at all; on the other hand, if you are given a problem which is not challenging, it will not give you scope to do yourself justice, or the chance to gain high marks. You should try to realise your maximum potential.

While we will take your stated preferences into account in allocating projects, we cannot guarantee a project which meets all your preferences. Part of the training in this module is being able to take any task which is given to you, and complete it to the best of your ability.

28. Dissertation Contents and Marking

Supervisors will advise on this but the dissertation should normally have the flowing layout.

Title Page

This should have the following format:

Title of Dissertation
A dissertation submitted in partial fulfilment of
the requirements for the degree of
BACHELOR OF *SCIENCE/ENGINEERING** in Computer Science
in
The Queen's University of Belfast
by
'Student Name'
'Date of Submission'

* as appropriate

Declaration Cover Sheet (Declaration of Academic Integrity)

Attached to the end of this document

Acknowledgements

To those who have helped the author during the project and the preparation of the dissertation and to anybody who has given financial support.

Abstract

A summary (100 words) which provides an outline of the subject matter and the results, findings and/or conclusions of the dissertation.

Contents

A complete list of chapters, sections, appendices etc. with page numbers.

Main Text (see below)

The main body of the dissertation as described below organised as a sequence of chapters each normally containing several sections. The main text should not normally exceed 45 pages (it may be less).

References

A list of references to documents (books, papers, web pages etc.) which are referred to in the main body of the text. Use the IEEE citation style as detailed here https://www.ieee.org/documents/ieeecitationref.pdf. There is some guidance on referencing at http://www.qub.ac.uk/cite2write/home.html.

The first citation should be the URL to the software code repository which should contain the code and any other resource required to run the software.

Appendices

These should include as appropriate:

(a) A User manual giving details on how to use the software, including details of input data, output formats and error messages.

(b) Test results, if appropriate.

(c) Other information which is not convenient or appropriate to include in the main body of the dissertation.

The Main Text

The Examiners will be looking for quality rather than quantity in your dissertation. You should try to keep the main text of your dissertation as concise as possible. Spelling should be correct, sentences grammatical, and formulae, figures and tables accurate. All figures, tables and appendices should be given numbers and headings. Your writing should be precise, concise and fluent. Avoid the first person (i.e. say what was done, rather than that you did it). Some dissertations may be organised differently from that headings given below. In particular, some parts might have more emphasis than others for some projects Further advice will be available from your supervisor. Below is a sample of what might be expected.

1.0 Introduction and Problem Area

Background material should be given which introduces the problem area, its context and background. You should identify the particular problem under consideration along with information about the problem area that enables the reader to understand the problem scope and nature. If your project involves a particular domain, algorithm, method theory etc., you may describe it in the introduction (alternatively or additionally, it may be described later, if appropriate). For best marks the student should show that they have systematically researched and fully analysed the problem, synthesising the relevant information. The very best dissertations will be of a standard suitable for part of a peer reviewed or professional publication.

2.0 Solution Description and System Requirements

You should provide a precise description of the system developed. Note that this is likely to be different from those that you started with, since you now have considerably more knowledge and understanding. The final dissertation should therefore contain an updated set of requirements matching the final system delivered. You can list these as a *requirements definition* from the domain perspective but you should also derive a *specification* for the software. The software requirements specification establishes the basis for what the software product is to do (and is not expected to do). You should list any assumptions made about the problem and any system constraints. Overall your requirements, functional AND non-functional should be correct, complete, consistent, clear, feasible and objectively verifiable. Content depends on your project but could include:

- A complete set of function definitions (as use cases if preferred), as far as possible written so as to be testable
- Measurable and testable non-functional requirements
- Description of interfaces required such as with other software or systems
- Any specific user interface requirement
- User characteristics

The target to aim for here is to describe a solution that satisfactorily solves the problem. Ideally your solution will be convincing and creative. Your requirements could be the basis for a contract or handing to external developers to complete. The best dissertations will show outstanding work, approaching that of the best professionals.

3.0 Design

This section should describe the design of your proposed system. Normally this several parts, depending on your project:

(i) Architectural Description of the system – textual and/or diagrammatic. This could be a simple diagram showing the components and how they relate or it could describe the choice of architectural style or pattern used.

(ii) User Interface Design (if applicable). Show sketches of the design or screenshots with explanations of choices made, if necessary.

(iii) Software System Design. The role of each component and the interfaces between components should be described. There should be a clear correlation between your design and your specification.

(iv) Where applicable give a critical discussion of key design decisions/styles/patterns used; data model; UI design, external Interfaces, other important issues e.g. concurrency, event handling, data persistence, error and exception handling, fault tolerance, security, distribution of components.

The design should be linked to requirements and, where applicable give a critical discussion of key design decisions/styles/patterns used. There might be a data model, a UI design, details of external interfaces, and of other important issues e.g. concurrency, event handling, error and exception handling, security, data persistence. No particular notation or tool is mandated. A satisfactory design will show a grasp of the main design issues. For top marks aim for outstanding design documentation approaching that of the best professionals. Prove that you have a very strong grasp of the design issues and aim for documentation that could be passed on to a developer without the need for further explanation

4.0 Implementation

You should describe any languages, packages, and libraries etc. that are used in the development of your system. There is no need to describe your code in detail. You may highlight data types and implementation techniques that are of special interest. If appropriate, you may provide:

(a) Choice of implementation language(s)/ development environment(s)

(b) Use of software libraries;

(c) Key implementation decisions

(d) A description of how some important functions and algorithms were implemented.

(e) A description of how each component is implemented.

Program code can be accessed by the assessors via the git repository **so there is no need to include code listings**. It is recommended that you comment code appropriately (not excessively). Programs should be written in a clear style with good program structure and well-defined data structures. The program code should reflect its design and show an understanding of relevant implementation issues.

5.0 Testing

This section will be judged in tandem with other evidence including evidence of unit tests and/or test documentation on the Repo. There should be a discussion of Test Approach e.g. unit testing, system testing, regression testing etc; Test cases should be described and justified; Include Testing tools used and provide evidence that testing coverage was complete. Provide proof that testing was completed, either showing sample test history and/or describing automated tests.

6.0 System Evaluation and Experimental Results

Provide a summary evaluation of the success of the project with respect to criteria identified in the introduction. Different projects will have a different emphasis. In all cases you are expected to provide empirical results and to draw conclusions from those results. You may use your software to generate experimental results. Be sure to describe the methodology of your evaluation or experimentation. An experiment is typically described in terms of its goals, the hypotheses being tested, the subject of the experiment, what is being measured and what is controlled, the results obtained and the analysis and interpretation of those results. A discussion of the significance of your experimental results may be appropriate or why the new system you have developed improves on what was already there. Do your results agree with other previous work or ideas? How does your system compare with similar ones?

Alternatively (or additionally), you can assess the product in terms of how it compares with other similar products and/or in terms of user feedback (e.g. via a survey or interviews) or some measurable quality aspect such performance efficiency or reliability.

Draw conclusions on the *process* used in the project as well. What went well? What did not go well? What are the strengths of your solution or conclusions? What are the weaknesses? Suggestions for further work should also be discussed. You can be critical and draw a negative conclusion. Not all projects will be successful. A well-explained failure is as an acceptable an outcome as a spectacular success. Assessors are looking for excellence in a critical appraisal of the work and a convincing argument for the significance of contribution in the context of wider work. This section should be objective, fair and comprehensive.

In all cases, societal implications and commercial and economic aspects should be evaluated. Has your project an outcome that potentially could improve some community or group of people? Perhaps your project can impact on the lives of others for example In education, employment, health, public policy or services, security, the environment, general wellbeing etc. There may be commercial opportunities arising from your product or findings. Describe these and include how the project could eventually brought to deployment and to deliver value. Discuss the feasibility of doing that. It may be that your project could make some process more efficient. Try to quantify the savings or improvements, generally or in one or more scenarios. You should be realistic though and include the risks and any negative impacts of your work and the potential impact as well.

Your supervisor can guide you on what is appropriate, but typically the very best projects have shown results derived using scientific method, that could be publishable with little or no work or show an exemplary empirically based evaluation of a software product. Those projects will also fairly and honestly assess the potential impact of the work socially or economically.

Any publication of results of the student's work is left to the discretion of the supervisor, but you can expect appropriate credit to be given to your work.

7.0 Appendices

Appendices will not be marked but may be referred to by the assessor to aid their understanding. They are useful if there is something that helps in understanding earlier parts of the dissertation, but if included inline might break the flow or readability of the document. For example, there may be large tables of data, design documents, evidence of testing etc etc.

---

29. System Marking

The system will be marked based on what is demonstrated on demo day, what is in the dissertation and what is in your Repo. The following categories are considered.

Functionality/Achievement: A passing project should demonstrate a good range of functionality demonstrating ability in implementation, even if it is a straightforward piece of work. The very best projects will show that you have a clear mastery of advanced techniques and demonstrate that major challenges have been overcome.

Output: Your product and results should be usable. The highest marked projects will show an exceptional product that could be used as is in the real world or target community.

Implementation Competency: The work should demonstrate that you are a competent developer, both at the demo and from the codebase.

Discussion/Defence of System: At the demo you will describe your system as you show it working. You will also be asked questions about the product, the process you used and the results obtained. At a minimum you should be able to discuss your system satisfactorily and the very best projects will show high levels of insight and give confidence in the work.

30. Resubmission of Projects

In the event that you do not submit or do not pass CSC3002, you will have the opportunity to resubmit your project work at a date specified by the examinations board. It is your responsibility to initiate contact with your supervisor, as soon as possible after the examination results are published, in order to discuss how your work needs to be revised.

For failed projects, the final mark is capped at 40% and will be based on just the dissertation and system components.

Appendix 1 – Interim Report Marking Guidance

QUEEN'S UNIVERSITY BELFAST

CSC3002 Computer Science Project
Problem Description, Work Plan and
Interim Demonstration – Marks Guide

School of Electronics, Electrical
Engineering and Computer
Science

2018/ 2019

Problem Description, Work Plan and Project Management (10%)

Excellent 1st		Fail
Understanding of problem		
Student has systematically researched the problem and fully analysed and synthesised the relevant information.	←	Superficial grasp of broad ideas and concepts. Major shortfalls are apparent in some key areas.
Verifiable Criteria (from point of view of end user) / Acceptance Tests		
Verifiable criteria comprehensively addressed according to the goals. Every criterion can be tested.	←	Verifiable criteria are poorly stated and not useful for assessing achievements later
Updated Work Plan		
Exceptionally well thought out work plan.	←	Work plan contains major omissions and is not what was discussed with supervisor
Project Management – i.e. Concepts developed early, Requirements elicited and prioritised early, (re)Planning carried out, gitLab in use, readMe in place, meaningful regular commits, regular communication/meetings, etc		
Exceptional Project Management: near perfect project conception, planning, execution and communication	←	Project Management poor–many major weaknesses

Interim System Demonstration (Note that substantial work is expected) (10%)

Excellent 1st		Fail
Requirements met/ Progress		
Exceptional progress with a large proportion of valuable requirements demonstrably met.	←	Progress unsatisfactory & prototype/system, is of only limited value but could be built on
Quality of System/ Results so far		
Exceptionally high quality system. Robust and working well.	←	Inadequate system quality but some evidence of work towards an acceptable system

Appendix 3

CSC3002 Computer Science Project

Dissertation and System - Marks Guide

School of Electronics and Electrical Engineering and Computer Science

2018/2019

Dissertation (40%)

Excellent 1st		Fail
Understanding of problem, context and background ((Judged across the whole dissertation, updated and extended from semester 1 work)		
Student has systematically researched and fully analysed the problem, synthesising the relevant information to a standard suitable for part of a high quality, peer reviewed or professional publication.	←	Superficial grasp of broad ideas and concepts. Major shortfalls are apparent in key areas
Solution Description – creativity and innovation (judged across the whole dissertation)		
The solution described solves the problem convincingly in an exceptionally innovative and creative way.	←	The problem remains largely unsolved/ partially solved and there is very little, if any, demonstration of creativity and innovation
Requirements: complete; clear; consistent; accurate; functional AND non-functional; feasible; objectively verifiable.		
Outstanding work, approaching that of the best professionals. Provides the basis for a contract or for handing to external developers.	←	Requirements are incomplete, inconsistent or incorrect in many cases and/or poorly stated.
Design. Linked to requirements; Contains architectural design and detailed functional/object design. Where applicable gives: critical discussion of key design decisions/styles/patterns used; data model; UI design, external Interfaces, other important issues e.g. concurrency, event handling, error and exception handling, security, data persistence.		
Outstanding design documentation to a standard expected of the best professionals. Outstanding grasp of the design issues. Could be passed on to a developer without the need for further explanation	←	Lacks many areas of design. Difficult to follow.
Implementation: Choice of implementation language(s)/ development environment(s); Use of software libraries etc.; Key implementation decisions.		
Outstanding, academic and professional understanding of relevant implementation issues.	←	Poor discussion of implementation issues. Difficult to follow
Software Testing. Largely judged from the dissertation but backed up by evidence of testing e.g. documentation/unit tests in repo.		
Exemplary testing and evaluation of the software product. Testing is complete. The evidence gives full confidence in the software.	←	Testing attempted but is not sufficiently complete or reliable.

Product Evaluation and/or Experimentation: This should be objective, fair and comprehensive. Where projects are mainly research-based, assess the scientific method and experimentation as well as a comparison with related work. Where product-based, there should be feedback from users and/or comparison with related products. In both cases, societal implications and commercial and economic aspects should be evaluated.		
Exemplary objective, fair and comprehensive evaluation/ experimentation. Standard appropriate to competition-winning or high quality publication standard.	←	Incomplete evaluation and/or poor scientific approach. Unreliable or of insufficient quality.
Dissertation format and structure: Consistent formatting, comprehensive full citations, professional or academic writing style.		
Outstanding report approaching that of the best professionals	←	Difficult to read. Lacks a logical train of argument. Very poor organisation.

System (40%)

Excellent 1st		Fail
Functionality/ Achievement (judged from software demo, dissertation and code)		
Excellent functionality, achieves the goals of the project. Clear mastery of advanced methods and tools. Overcame major challenges.	←	Functionality is insufficient and/or the work may be unchallenging or performed poorly
Output: A usable product has been described. Judged over dissertation, code and demo.		
Exceptional Software product could be used as it is in the real world or target community	←	Output is not complete or reliable or of sufficient quality.
Implementation Competency Demonstrated (judged from software demo, dissertation and code)		
The system demonstrates an exceptional level of competency in implementation.	←	The system does not demonstrate sufficient competency in implementation
Discussion/ Defence of System in Demo		
Discussion shows very high levels of insight.	←	Unable to maintain a discussion of the work done.

Appendix 3: **Cover Sheet for the beginning of the dissertation**

SCHOOL OF ELECTRONICS, ELECTRICAL ENGINEERING and COMPUTER SCIENCE

CSC3002 – COMPUTER SCIENCE PROJECT

Dissertation Cover Sheet

A signed and completed cover sheet must accompany the submission of the Software Engineering dissertation submitted for assessment.

Work submitted without a cover sheet will **NOT** be marked.

Student Name: Student Number:

Project Title:

Supervisor:

Declaration of Academic Integrity

Before submitting your dissertation please check that the submission:

1. Has a full bibliography attached laid out according to the guidelines specified in the Student Project Handbook
2. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
3. Does not exceed the specified page limit
4. Is clearly presented and proof-read
5. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.

By submitting your dissertation you declare that you have completed the tutorial on plagiarism at http://www.qub.ac.uk/cite2write/introduction5.html and are aware that it is an academic offence to plagiarise. You declare that the submission is your own original work. No part of it has been submitted for any other assignment and you have acknowledged all written and electronic sources used.

Student's signature *Date of submission*

[type your name here instead if submitting electronically]