



Queen's University Belfast

School of Electronics, Electrical Engineering and Computer Science

CSC7058

Individual Software Development Project

Delivery 2024-2025

E-MAIL
j.bustard@qub.ac.uk

WEB
www.qub.ac.uk/eeecs



Introduction

Aim of CSC7058

The general aim of the CSC7058 triple module is to enable you to demonstrate that you can develop a sound, effective software system to an identified problem and record this through a Master's level dissertation. A secondary aim of the individual software development project is to assist you in developing the skills necessary to be an effective software developer.

Learning Outcomes

The general learning outcomes which you should achieve for this module are that on completion you should be able to:

Demonstrate the ability to understand a problem which could be addressed by the creation of a software system and communicate this understanding as part of a project dissertation.

Design and develop a software system and communicate its structure in the project dissertation in such a way that other developers can build upon it.

Test the developed system to determine whether the functionality works reliably and document this testing in the project dissertation.

Evaluate to what extent the system helps address the problem that the system is designed to improve. Reflect on the methodology of the project and suggest improvements. Include this evaluation and reflection in the project dissertation.

What you have to submit

On completion of the module you are required to deliver two key components although other interim elements are required (and are detailed later in this handbook).

Firstly, a working project, which should demonstrate your skills in creating an effective software system as a solution to your approved problem.

Secondly, your project should be described in the dissertation which is an academic document and for Master's level students is expected to be no more than fifty pages in length for the main text which does not include table of contents or appendices. Some students may be concerned that their work is going to be too short or too long but you can seek clarification from your academic supervisor.



Overview

You should recognise that the Individual Software Development Project represents a major piece of work, and this is reflected in both the weighting in CATS points and the amount of support that is available. The specification for the Individual Software Development Project module indicates that it should represent enough work to occupy a student full-time for at least 12 weeks which equates to at least 24 weeks part-time.

For many students on the MSc pathway, writing a dissertation is a challenging experience. It is, almost by definition, a self-directed process. In the other modules of the course you have been set targets and objectives through the coursework assignments and deadlines but in this phase of the course; you will need to set your own deadlines, and manage the experience. You will need perseverance, endurance and determination. Initially this can seem intimidating, but it is also an opportunity for you to learn and grow. Finishing the Individual Software Development Project on time is largely about having the will to do it. The majority of the skills that you will need, you already have gained. However, the Individual Software Development Project is designed to be a learning experience and it is highly likely that you will extend your existing programming skills or that you will have to learn some new skills.

Writing at Master's level and developing a software system are specialised skills, but ones which will develop your intellectual, critical and problem-solving capabilities.

The dissertation is the academic work which marks the transition between being a student and a software developer. Writing your dissertation will enable you to develop valuable technical writing skills. Regardless of the career path you have chosen, the processes of thinking analytically, assembling a well-reasoned software solution, writing professionally and managing your time will all serve you well in the future.

The major requirements for your dissertation are that it:

- is literate. That is it uses correct spelling and punctuation and contains no (or very few) grammatical errors.
- clearly expresses your thoughts
- develops your arguments in a manner that can be understood by someone with a passing knowledge of software development
- is critical and reflective
- demonstrates awareness of state of the art techniques for software development and their literature
- is focused on your project



Assessment

The assessment for this module has several elements:

Problem Specification / Project Plan	
Dissertation / Technical Documentation	30%
Developed Software / Testing / Installation Files	60%
Project Demonstration (and video)	
Adherence to a professional development method.	10%
	100%

In order to pass the module, you are required to attain an overall score of 50%.

There is provision in the regulations for students to attend a viva voce examination if required to do so by the Board of Examiners.

The completed software project, with all its files (software, documents and demo video), must be submitted via Canvas by the specified date.

These submissions are used for the assessment. **NOTE:** If the code and demo video are not received then the project will not be considered by the Board of Examiners as completed.

Each student's work is assessed by two academics. One of these will be the academic supervisor, and the other is usually someone who has not had direct contact with the student during this module. If there is a significant degree of disagreement between the two academics on the student's grade, a third opinion will be sought. The external examiner will moderate the marking of the projects.

The grade will correspond to:

- greater than 70% (distinction)
- 60 to 69% (commendation)
- 50 to 59% (pass)
- below 50% (fail)

After the meeting of the Internal Examiners, and following the meeting of the Board of Examiners with the External Examiner, your grade will be published on QSiS.

A student who scores less than 50% will be referred to the School Students' Support Committee (SSSC) which is likely to require the student to accept the Graduate Diploma in Software Development.

Students who pass the module may then register for graduation by clicking the 'Register' button on their QSiS homepage. The closing date for registration is usually posted on QSiS. Students who intend to graduate in absentia are also required to register and you should be aware that there is a fee for graduation.

The Assessment Criteria



Because the nature of the work is so individual, it is not possible to provide a definitive list of criteria for a high-scoring Software Development Project. However, the following guidelines will be used by your company/employer and academic supervisor to grade your work:

Problem Specification / Project Plan

The student has to demonstrate that they understood the problem and described a sensible approach to solving it and that they clearly specified the objectives of the project. There should be an analysis of the problem domain and an indication of the proposed interface design and content of the project. You could, for example, invent a fictional user/s (the more realistic the better) and use this to explain how they would use the application and the features which they might need or want to use. A timetable for the work should be included along with notes on potential problem areas. The primary purpose of this deliverable is to ensure that the project is suitable/feasible and that the student has a clear plan and is engaging correctly with the project. This document should be no more than five pages in length. A copy of the Problem Specification/Project Plan should be uploaded to Canvas CSC7058 by the deadline for submission as outlined in the Project Schedule table in this document.

Dissertation (worth 30%)

The main priority for the dissertation is that it provides valuable information for another person wanting to run, maintain or build upon the project in the future. It should outline the reasoning behind language, framework, architecture and ui decisions. It should also provide a high level overview of the code, testing and experimentation to make it easier for the reader to understand the associated code files, testing plans/scripts and results.

Writing Style

Overall, the main text of the dissertation should be clear, concise, fluent and well organised. You should concentrate on the quality of the dissertation rather than the quantity. Spelling should be correct, sentences grammatical, and formulae, figures and tables accurate. All figures, tables, and appendices should be given numbers and headings. As is usual in academic / technical writing, do not describe things in the first person (say what was done, rather than you did it).

Title Page

The title page should have the following format:

‘Title of Dissertation’
A dissertation submitted in partial fulfilment of
The requirement for the degree of
MASTER OF SCIENCE in Software Development
in
The Queen’s University of Belfast
By
‘Student Name’
‘Date of Submission’

Declaration Cover Sheet (Declaration of Academic Integrity)

This is available at the back of this document; it must be signed and submitted with the document.

**Acknowledgements**

This is a section written to those who have helped you, as the author, during the preparation to completion of the project and dissertation. The student should also thank anyone who has given you financial support.

Abstract

This is a summary (100 words) outlining the subject matter and conclusions of the dissertation.

Contents

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

Main Body of the Dissertation

The following section outlines the sequence of chapters that should be included in the dissertation. Some dissertations may be organized differently from the chapter headings given below. However, you should attempt to address each of the following aspects in your dissertation. Further advice will be available from your supervisor.

Introduction

This should outline the structure and key sections of the rest of the document.

Chapter 1: Understanding the Problem

The student must clearly describe the perceived problem and the target audience. It should be obvious from the chapter that the student has a thorough understanding of the problem domain and current applications used (if available) to address the problem at present. This section should demonstrate a good understanding of possible languages, libraries and frameworks that could be used to develop the project. This chapter should explain and justify the process by which the requirements have been elicited. It should then also clearly identify the requirements of your project, which can be later tested. Depending on the chosen development strategy this chapter may be written retrospectively. The student's academic supervisor will provide additional advice on this where required. The development strategy must be clearly described, adequately justified in terms of the problem and appropriate to the project.

Chapter 2: User interface design

This chapter should describe, illustrate and justify the user interface design of your proposed system. Not all projects will have a significant user interface component, for example if they are back end algorithms or experimental projects. For projects without a GUI a short overview of the interfaces to the software should be outlined. For projects that have more substantial graphical user interfaces there should be an explanation for how the design has been developed including any feedback that shaped the design. The goal being to ensure that anyone building on the system understands the reasoning behind the UI and the feedback of users that led to its design.

***Chapter 3: Architecture design and algorithm explanation***

A high level overview of the architecture of the code should be provided in this section. The overview should be designed to help another person seeking to adapt and maintain the software and should refer to the source files used in the project. Where relevant this section should also explain the design of any api interfaces that have been designed for the project to enable others to easily interface with the project. This chapter should also include a description of any complex algorithms that may be hard to understand simply by reading the code and its comments. Where algorithms and architecture are based on other work this should be clearly explained as well as any references to external explanations of algorithms or architecture used in developing the software.

Chapter 4: Experimentation (if relevant)

This chapter should be included for projects that include experimentation, such as projects with a machine learning component. This section should outline how the experiments conducted in the project have been designed to accurately measure likely performance in a realistic usage of the completed system. It should also highlight why each experiment was run, visually show the results e.g. in graph, discuss what was learnt from each experiment and how the project was adapted based on what was discovered. Appropriate experimental scripts, data and result files should be included with the developed software and referenced in the text so that the experiments can be reproduced and the results can be easily analysed.

Chapter 5: Testing

This section should include a justification for the approaches used to test the resulting system. These may include such approaches as unit tests, manual and/or automated end 2 end testing, performance testing etc. The section should explain what parts of the project are at greatest risk of having uncaught bugs within them, for example because of their complexity, and how the approach to testing has been developed to analyse those areas in detail. For particularly complex algorithms for example this section can outline the design of the test cases to ensure good test coverage. The section should clearly reference any automated testing code, manual testing plans and test results included with the project that provides clear evidence that testing has taken place and can be easily used by anyone further developing the project.

Chapter 6: Evaluation and Conclusion

This section should include a general evaluation of the success of the project measured against the criteria stated in the introduction and/or requirements. An evaluation of the hardware/software environment and language used may also be presented. Suggestions for further work should also be discussed. Do not be afraid to be critical or to draw a negative conclusion; not all projects will be successful. This section should provide a thorough and honest reflection on the process followed in the project and the results of that process. To do this well, the student should not leave any blind spots in their reflection and should identify the most and least successful aspects of the project. It should be written in such a way as to be helpful to a person seeking to adapt the project or to create a similar project in the future.



**References/Bibliography**

A list of relevant references to other documents (books, papers, web-sites, etc.) which are cited in the main body of the text.

For books, record:

- The author's name or the editor's name
- The year the book was published
- The title of the book
- The edition of the book
- The name of the publisher
- The ISBN of the book

For electronic resources, record:

- The author's name or the editor's name where available
- The year the article was published where available
- The title of the article
- The date the source of the information was accessed
- The electronic address or email
- The type of electronic resource (email, discussion forum, WWW page, etc)

Appendices

These may include:

- Testing code, datasets and results.
- Printout of code for section/sub-sections of the application developed in relationship to the submission.
- In most cases this information should be present within the code submission of the project



Developed Software (worth 60%)

The academic supervisor should have access to a Github (or similar) repository to ensure the amount of individual contribution from the student is acceptable. Access to a repository of code applies in all project cases whether individual, academic or company based. The details of the repository should be agreed at the outset of the project.

If a company project is undertaken and it contains confidential / sensitive information then there must be an agreement that the academic supervisor and assessors can view and assess the project code. Therefore if confidentiality / non-disclosure forms have to be signed they should be submitted to the supervisor at the start of the project.

Functionality

The total amount of useful work that has been completed on the project.

Addresses requirements

The degree to which the project addresses the problem being solved.

User focused design

The degree to which the project has been adapted to take into consideration the needs of any users of the system. For example, iterations in the user interface to reflect user feedback, or APIs to enable integration with users' systems. Projects that demonstrate significant effort on the part of the student to deliver value to the user will do well. Projects that are designed based on the limited technical knowledge of the student will limit the value of this section.

The interface should be effective and 'polished' and appropriate to the target audience. It should meet its required purpose and be intuitive and easy to use. Where possible, facilities should be provided to help and guide the user.

Some projects may not need a user interface. This should be discussed with your supervisor at the outset of your project and an agreement should be made on how to assess the project in the absence of an interface.

Ease of maintenance and future development

This section assesses the efforts that have gone into ensuring that the resulting project can be maintained and built upon by others. For example, the presence of installation instructions or scripts, documented apis, an understandable readme which may include a code overview. Students should seek to present their project in a similar way to a high quality open source github project.

Code Quality

The code must be written using good programming practices, coding standards and should be appropriately commented. It is desirable that the code demonstrates good programming skills, appropriate and effective use of the chosen language and system features provided within the language.

Testing

The degree to which the student has performed testing which can include unit, end 2 end, user testing, a manual testing plan with evidence of having performed it, automated testing, performance measurement/optimisation if relevant.

Academic judgement of adherence to a professional development method (10%)

This is a general perception of the student's professionalism through above and beyond extra effort as revealed through all parts of the submission, including code, dissertation and demo.

Organisation

The student should demonstrate skills of managing the project, setting clear and sensible objectives with the aid of a project plan and proceeding in a logical manner. The work should be completed on time and to the standard agreed with the supervisor or client.

Effort

The student should show evidence of having worked diligently and independently throughout the project and have the ability to cope with difficulties and setbacks.

Supervision

The student should demonstrate skills involving working independently and seeking assistance from their academic supervisor as and when appropriate. The student should exhibit initiative and ingenuity in the completion of the project.

Project Demonstration

The student should arrange in advance a suitable time and date with their supervisor to present/demonstrate their project in a professional manner after the deadline for the submission of the dissertation. This may include communication with companies, if present for employer based projects, and other academic assessors. The student should bring all required software and hardware to the demonstration. As the demonstration may require setting up of hardware and/or connecting to servers, demonstration of the software and the answering of questions posed by assessors, the student should ensure that they provide themselves ample preparation time for their demonstration.

System Demo Video (Project cannot be marked without Video)

The system should be demonstrated by recording screen captures of it in use with supporting appropriate audio commentary for viewing by the External Examiner. It must cover the main functions of the system developed. The video must **not exceed ten minutes** in runtime. At the start of the video you should show the title of your project along with your name, student number and image of yourself.

The video should be saved in **MP4 format**. Students must ensure that the video plays on VLC on the lab computers in the Computer Science Building(CSB) labs. You should test your video on the CSB lab computers before uploading your video through Canvas.

All text on screen in the video must be clearly legible and audio commentary must be clearly audible.

The System Demo Video will be made available to the External Examiner as well as been reviewed by the other assessors. It may be made available to future students of the course.

Please Note: If the system demonstration video is not submitted as part of the final submission or cannot be played in full by the assessors, then the project submission may not be marked.

Supervision

One of the key differences between the Software Development Project and all the other assignments you have completed as part of the MSc in Software Development is in the way you will be tutored. Each student on this module is assigned to a supervisor for the duration of the project. Your supervisor is responsible for providing advice and guidance to permit you to complete the project in the time available.

The Software Development Project should be a period of self-directed study, so your academic supervisor cannot get in the way of this goal by becoming too involved. They are not expected to tell you what to do, chase you for work, structure your studying, or design your project. They will offer advice and support, check that you are on the right lines and point you in the direction of relevant materials for both your project and dissertation.

At the beginning of the project you and your supervisor should agree on a project timetable and arrange a series of meetings where progress, problems and writing can be discussed. You should make sure that your supervisor is kept up to date with the progress of the project as well as the dissertation. Your supervisor will offer general guidance and support. You may submit a draft of each chapter of your dissertation for your supervisor to comment and provide feedback on the draft. Your supervisor may provide feedback on the content and structure of the draft however it is not their role to proof read the document for spelling and grammatical errors. Upon receiving comments and feedback on a draft chapter, it is not acceptable to resubmit a revised draft for further feedback. It is important that if you wish to receive feedback on the draft work, that it is submitted to the supervisor in good time to allow this to happen.



Individual Software Development Project Guide

Responsibilities

You are expected to take responsibility for your own studies, and it is up to you to arrange access to whatever equipment and resources (e.g.) that you need.

It is your responsibility that:

- your problem specification and project plan are submitted on time
- your project is completed satisfactorily and on time
- problems are sorted out at an early stage
- your project is "debugged" and working as specified
- your finished project system and code is delivered through Canvas on time with all associated files and is working correctly
- your dissertation is submitted on time
- regular meetings are held with your academic supervisor (don't hide from them because you haven't done as much work as expected)
- advice and guidance is sought from your academic supervisor at an early stage
- all your written work is proofread and corrected before submission

It is in your own interest to maintain regular contact with your academic supervisor for the duration of the Project, and to have a good working relationship with them. It's fairly certain that at some period during the Project your academic supervisor will be unavailable due to holidays or work commitments. You should take account of your academic supervisor's schedule in your planning, and not expect them to be always available when you need advice and guidance. You should also give them ample time to reflect on draft work which you are submitting. It is wise to send regular updates and draft pages to your supervisor rather than sizable pieces of work.

Although most students quickly establish a good working relationship with their academic supervisor, there are occasionally cases where this does not happen. If you are concerned about the level of supervision you are receiving, you should in the first instance raise this with the module tutor. If this action does not lead to a satisfactory conclusion, or you would rather not approach the tutor, then you should acquaint the Course Director for the MSc in Software Development with your concerns. Note that the School and University expect you to raise any difficulties which you have with your supervision in a timely manner.

Minimum Contact

Normally a frequency of one meeting per two weeks should be regarded as an effective level of engagement with your supervisor/company mentor.



Recording Meetings

It is a university Master's level requirement that you keep minutes of meetings with your academic supervisor. The minutes should be sent to your supervisor within 48 hours of the meeting and before the next meeting. The minutes should contain a brief summary of the meeting, the matters discussed and any action agreed between you and the supervisor. At the next meeting, the record of the minutes from the previous meeting should be reviewed and discussed.

If Things Go Wrong

Your academic supervisor also has a pastoral role. This is especially important if you are experiencing difficulties with the work or in your life. The default position of all the academic supervisors on the course is that we will assume things are going smoothly unless you tell us otherwise.

For some students, this phase of the course can be a lonely and frustrating time. Some students experience a crisis of confidence mid-way through the work. One way of overcoming this is to break the work down (for all the components - project and dissertation) into small steps and tackle them one at a time. Motivation may be maintained and improved as each step is "ticked off". Making consistent progress (even if it is only a small piece every day) is a better strategy than trying to "blitz" the work as you approach the deadline. If you are experiencing a crisis of confidence, talk things over with your academic supervisor. You may be surprised at how helpful such a strategy can be.

It is not uncommon for students to experience personal problems ranging from ill health to family bereavement. There are procedures available to help students overcome such difficulties. It is rare for the School to be confronted with something entirely new, and often we can provide tangible help very quickly. Advise your academic supervisor or speak to the Course Director if you are having personal problems which are impacting on your work. Some of these difficulties may be categorised as Exceptional Circumstances and how the School may be officially informed of these circumstances may be found at <http://www.qub.ac.uk/directorates/AcademicAffairs/StudentGuidance/ExceptionalCircumstances-AStudentGuide/>

Some students may decide during their project that they do not wish to continue with the work. Any student on this phase of the course is, by definition, entitled to ask to transfer to the Postgraduate Diploma pathway, and to accept the Diploma. A request to transfer from the MSc to the Postgraduate Diploma pathway has to be in writing. Please do not take such a decision lightly. If you think that things are getting too much for you, make an appointment to meet the Course Director.

Starting the work

Use the early days of the Software Development Project as a time for research and experimentation. See what is available in your project domain and what ideas you can borrow. Look for unusual and effective ways of solving your problem and see if you can incorporate some of these into your proposed design.

You should also use the early weeks to consider what functionality and content you will incorporate into your project, and discuss your ideas with your supervisor, and think about the design and overall user experience.



Avoiding Plagiarism

During the course of the MSc in Software Development you will have explored some of the issues relating to academic plagiarism. Doing original work, especially on something as protracted as a project and dissertation at Master's level is not easy. It is usual practice in academic writing and in software development to build upon the ideas of others. The use of another's work does not constitute plagiarism unless and until that work is presented as being your own. In the pressure to complete the dissertation on time there is always the temptation to take a short-cut without realising that these may lead to suggestions that the submitted work is not original. The satisfaction in completing the Project comes from mastering the complex material which underpins the field of Software Development.

You must always distinguish your own ideas, words, comments and contributions in the Project from those of others. To do so requires the high level of intellectual honesty which is expected from you and your peers.

When you restate another person's words, thoughts or ideas you are paraphrasing them. In such a case you are still obliged to reference the original work. Paraphrased work does not have to be enclosed in quotation marks but the source still needs to be fully referenced.

A summary is a looser form of paraphrasing, but this also should be referenced. A summary involves condensing and rephrasing other authors' work and is an important facet of academic writing. Using appropriate referencing should make clear to the reader which ideas come from which author.

Originality of Computer Code

The Project is your chance to demonstrate what you have learnt from the taught modules, and to show your skill in design and programming. As part of this process you will inevitably have to spend some time either revising or learning new skills in coding.

You will only be given credit for code which you have written and which is original to the project. One of the aims of the module is to determine students' level of skill and this would be compromised if "cutting and pasting" code from other developers was accepted. Code which you have developed for previous assignments which is recycled for the project will not attract marks.

Almost every project borrows ideas from elsewhere, and you are encouraged to reuse the exercises used in the practical classes as springboards for work on your project. However, you will not receive any credit for directly using exemplars and practical exercises from class, textbooks, web or other sources.



Specification for the Dissertation

The Body of the Work

Text should be well-spaced (1½ line spacing) and must be printed on both sides of the page. For in-text emphasis, italics or bold typeface should be used (instead of underlining or upper case).

Font: Garamond, size 11 point or Arial, size 10 point

Margins:

- top and bottom 2.0 cm
- left 2.5 cm
- right 2.5 cm

Alignment: left justified, ragged right edge

Paragraphs: No indentation; Space before 10 or 11 pt

Page numbers: Centred at the bottom of page;

Use lower case Roman Numerals for numbering abstract, acknowledgements page, contents page, list of diagrams etc.

Use continuous Arabic Numerals for the remainder of the dissertation, but not including appendices

Headings

Font size: 11, 12 or 14 pt, Bold

Alignment: Left justified at the top of a new page; single line Space after

Use headings and subheadings to organise the content of your chapters, however, these should be used carefully. Having too many subheadings may result in the chapter appearing fragmented and (paradoxically) disorganised. If you use consistent formatting in headings by adjusting the 'heading 1' 'heading 2' tags etc you can use these to automatically generate a table of contents.

Header and Footer

The Header and Footers should commence on page one of chapter one with the header containing your name and student number and the footer containing the centred page number.

Submitting Your Finished Work

As your work nears completion you should look at the dissertation "in the whole". Check that the separate chapters link together and that any changes made in the design during the implementation of the project are reflected in the dissertation. Give your supervisor a chance to see and comment on your project as you are working on it. It is unwise to keep the project from your supervisor and then spring it on them as a fait accompli a few days before submission.

You should submit your project (software files, documentation files and system demonstration video) via Canvas as part of the final submission.

Sample Project

Sample projects are available to view in the Computer Science Reception. Each student has one twenty-minute opportunity to view the projects so that they may observe the layout of the project. No



photographs or notes may be taken whilst observing the project and the project may not be removed from the Computing General Office.

The format of dissertations and code submissions have changed this year. Students should be aware of the different format that is being asked of students this year.

Project Schedule

The schedule for the project and milestones is available on Canvas

Regulations

The completed project and dissertation has to be submitted by the deadline for submission date. Standard university regulations on late submissions apply.

See, <https://www.qub.ac.uk/directorates/AcademicStudentAffairs/AcademicAffairs/GeneralRegulations/StudyRegulations/StudyRegulationsforPostgraduateTaughtProgrammes/>

The project and all ancillary files must be provided to the School via Canvas by the deadline for submission date. Both components (code and dissertation) constitute the submission, late submissions apply to the whole submission.

A student may not be given credit for any work submitted in whole or in part for another assignment or part of an assignment within the modules associated with the MSc in Software Development. Materials submitted as part of the assessment for the MSc in Software Development may not be concurrently submitted as part of another degree at The Queen's University of Belfast or any other university.

A signed and dated declaration sheet asserting the originality of the work must accompany each copy of the dissertation and should be inserted after the title page of each copy of the dissertation submitted.

The length of the **main text of the dissertation must not exceed fifty pages** which does not include the table of contents or the appendices. Submissions which fall outside these parameters may be accepted at the discretion of the Course Director for the MSc in Software Development.

The written component of the Project should be submitted in English. The spelling should be correct, sentences grammatical and formulae, figures and tables accurate.

All figures and tables included in the dissertation should be labelled and referenced from the main text.

Significant example sections of Computer code should be presented in A4 format and include comments. As far as possible all work for the project must be original. All non-original material must be acknowledged in the appendix. It is acceptable to use computer code from other people or which is in the public domain, provided that the source is made absolutely clear by comments within the code and noting the source in the appendix.

In some cases, and at the discretion of the Board of Examiners for the MSc in Software Development students may be required to attend a viva voce examination before their Project is accepted.

An instance where work from another student's submissions or unpublished work from this or other institutions has evidently been incorporated into the Project will be considered a serious breach of academic regulations.



If work submitted by a student is considered not to be original, and is not properly attributed to the original author, the examiner is obliged to bring the matter to the attention of the Chair of the Board of Examiners. If the Chair of the Board of Examiners is satisfied that unattributed and non-original work has been presented as that of the student, the work will be brought to the attention of the School Director.

The Director will cause an investigation to be brought into the work. At this stage, the student will be invited to explain the submission of apparently non-original work. If the Investigation Panel is not satisfied by the student's explanation, then the work will be brought to the School Board of Examiners with a recommendation that:

- the submission should not be accepted (in which case the student will be shown as failing the Project and Dissertation through non-submission), or
- a mark of zero is recorded for the work (in which case the student fails the Project), or
- the grade or mark should be substantially reduced.
- disciplinary action may be taken by the Board of Examiners.

The publication of any results of the student's work is left to the discretion of the supervisor. Material supplied at the outset of the project may be copyrighted solely for educational use.



Types and Choice of Project

In stating your project preferences, you should consider among other things the skills you would like to develop through the project, the experience you would like to gain, and the usefulness of your project in discussing it with potential employers. You should also aim to match your ability to the project, to maximise your potential achievement.

There are two types of project: -

a) **Student Specified Project:**

If you wish you may propose your own project problem (please complete the Project Preference Form). The aim is to enhance your motivation and enthusiasm for the project.

Constraints

There will naturally have to be certain constraints on the problem, so that the project is suitable for achieving the learning outcomes of the module. In particular:

- a project must involve the development of a significant piece of software;
- a simple web development problem is not suitable. However, a web-based system which involves developing a database, and providing interfaces for dynamically accessing and updating the database, would be acceptable, provided it requires a sufficient degree of data modelling and processing, and provided the code is not generated by a tool. In particular, a project must not consist of stitching together pieces of code generated by a tool.
- the project must be a completely new piece of work;
- the academic supervisor (and, if necessary, the Module Co-ordinator) must approve the project proposal, and may suggest modifications to meet the above constraints;
- you must of course not make any use of any projects which are available online unless approved by your academic supervisor, and you must reference them clearly in the normal recommended manner in your dissertation. Project submissions are frequently checked for plagiarism by submitting them to the national JISC Service.

b) **Academic Project:** standard software development or research-related project

Standard Software Development projects may normally use Java or C#, typically with some GUI development. Examples are: games, graphical editors, animation of algorithms for teaching, or data management systems.

Research-related projects typically involve developing a software tool, or developing and implementing or evaluating algorithms for problems in a research area.



Scope and Level of Difficulty

The project is meant to bring out the best in you, and 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 the most challenging, it will not give you scope to do yourself justice, or the chance to gain very high marks. We want you to realise your maximum potential. Student-specified projects must also meet these constraints, by agreement with the supervisor.

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.

**SCHOOL OF ELECTRONICS, ELECTRICAL ENGINEERING and COMPUTER SCIENCE****CSC7058 – INDIVIDUAL SOFTWARE DEVELOPMENT PROJECT**

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

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

Declaration of Academic Integrity

Before signing the declaration below 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.
6. Software and files are submitted via Canvas.

I certify that that the submission is my own work, all sources are correctly attributed, and the contribution of any AI technologies is fully acknowledged.

I declare that I have read both the University and the School of Electronics, Electrical Engineering and Computer Science guidelines on plagiarism - <http://www.qub.ac.uk/schools/eeecs/Education/StudentStudyInformation/Plagiarism/> - and that the attached submission is my own original work. No part of it has been submitted for any other assignment and I have acknowledged in my notes and bibliography all written and electronic sources used. I am aware of the disciplinary consequences of failing to abide and follow the School and Queen's University Regulations on Plagiarism.

Name: (BLOCK CAPITALS)

Student Number:

Student's signature

Date of submission

