Department of Computer Science

BSc / MSci

Final Year Project Handbook

2025 – 2026

Authors of this revision: Edward Anstead, Hafizul Asad, Tracey Booth, Chris Child, Jacob Howe, and Vahid Rafe

Date: 29/09/2025

Version history:
v20250929_1410: First published version for 2025–2026

The department reserves the right to vary details of this project handbook.
Any changes or clarifications will be announced via Moodle.

Contents

1 Introduction

This document is the definitive handbook for students who are undertaking their final year BSc or MSci project in the Department of Computer Science at City, University of London.

The final project is a module (IN3007/INM450) like other modules in the course scheme, but worth 45 credits: this is equivalent to three standard modules. It has specified learning outcomes against which you will be tested. These learning outcomes are listed in the module specification, available on Moodle. Note that you are responsible for ensuring that you satisfy all requirements of the project module.

The project module is co-ordinated by the Project Team.

- Edward Anstead (Module Leader and Application development stream)
- Hafizul Asad (MSci projects stream)
- Tracey Booth (Application development stream, specialising in UI design)
- Christopher Child (Games development stream)
- Jacob Howe (Application development stream) and
- Vahid Rafe (Client-based projects stream and Ethics lead).

Administration is dealt with by the Programmes Office (ug.cs@citystgeorges.ac.uk).

2 The Essentials

2.1 What is the project?

The project is a substantial piece of individual work to solve a problem in computing. It must be performed autonomously and professionally by the student, with support from taught sessions, making use of drop-in sessions, while receiving feedback from academic project consultants[1].

All projects must produce a Project Definition Document (PDD) and a Final Project Submission (FPS). The PDD and its attached Ethics form require approval by the consultant for the project to be allowed to be carried out. The deadlines for these outputs and the key milestones of the project are set out in the Project Calendar, available in Moodle. Table 1 summarises these and other essential steps in the project.

Table 1. Project process and deliverables

Activity	Deliverable
Steps for project planning, preparation and approval	
Student chooses the project topic	-
Module team allocates students to consultants/supervisors	-
Student submits Project Definition Document (PDD) with its Research Ethics Form	PDD with its Research Ethics Form
Student receives feedback about PDD and Research Ethics Form:(approval or request for changes) from the consultant/supervisor	-
If feedback was a request for changes, student submits final Project Definition Document (PDD), with its Research Ethics Form	PDD including Research Ethics Form
Interaction during project execution	
Student-consultant meetings, according to schedule provided, for feedback on work reported and drafts of various report parts	Notes taken by the student, actions agreed, meeting logged.
Students who will involve human participants obtain consultant approval for ethics related materials	All materials (e.g. plans, consent forms, participant information sheets)
Student attends taught sessions (lectures) & drop-in sessions with module team and with specialists	-
Student may discuss draft project report with consultant for feedback	-
Student submits Final Project Submission (FPS)	Project report, video, code and other related outputs; separate deadlines for the product package and the rest of the submission
Student shows software product and discusses project in Final Project Review with the two markers	

[1] BSc students have consultants, whereas MSci students have supervisors due to the specialist demands of their projects. Where you read “consultant” in this document, for MSci students it means “supervisor”.

The Final Project Submission, FPS, is treated as assessed coursework with respect to submissions and deadlines. Late submissions will be penalised.

For extenuating circumstances, the standard rules apply: if you are affected by extenuating circumstances and wish to apply for an extension, you must apply to the Extenuating Circumstances Panel with appropriate evidence. You will not be granted an extension on the basis of other work commitments.

2.2 Expectations.

Any student who does not attend the sessions with their consultant/supervisor (or does not discuss their work as it progresses), or fails to report weekly progress on Moodle as required, may be summoned to a viva by the project team: attendance at this viva is mandatory

This module is worth 45 credits and is expected to take 450 hours of the student's work. To pass, and to get a mark that matches your ability, it is essential that you put in this total amount of effort.

You must meet with your consultant according to the pre-defined timetable; and you must keep records of these meetings and actions agreed at them.

You are also expected to attend the taught sessions to support your work: these cover aspects that are common to most student projects and are designed to help you succeed. Each taught session includes a presentation delivered by the module's team (about one hour) and a Q&A session for students to ask related questions. You may also attend the module drop-in sessions, to talk with a member of the module team, if you have questions specific to your own project, and with specialists for questions about specific aspects of project work.

It is essential that you obey the rules set out in this document and the announcements posted on Moodle by the project team. Violating rules risks penalties in your marks or large delays in processing them and therefore serious delays in graduating. If after reading rules or announcements you are unsure about what you should do, ask via the Moodle forum or at the module's drop-in sessions. Never rely on rumours from other students!

2.3 Where to get advice

The rules for projects are set out in this document and in announcements posted on Moodle module IN3007/INM450. Read these first.

For advice that is specific to your own project, attend the module's drop-in sessions and engage with the feedback provided by your consultant. Make sure that if you need clarifications about rules, you indicate what written rule exactly is unclear to you. If you need technical advice about aspects of your project, attend the regular "specialist sessions", where colleagues with specific expertise are available for advice: details are on Moodle.

For general advice about planning and running your project, see: Dawson, C. W. (2015). *Projects in Computing and Information Systems: A Student's Guide*. Pearson Education, 318 pp.

This book is available through the City Library both as print copies and e-Books (*note: The Library also has copies of earlier editions, which are just as good for general contents, except* *being less up to date regarding evolving areas like e.g. software engineering methods and research ethics.).*

2.4 Keeping records

You must maintain a personal record of all your project work, covering all phases of it. Weekly summaries will be required: details of how to record them will be provided through Moodle.

You must also keep all raw data collected at any stage of the project, e.g. notes from interviews and other meetings; data collected in experiments, interviews, testing of software; all the correspondence and email that you exchange during the project. Keep both paper and electronic documents. You should also keep a personal record of the meetings with your consultant: what was discussed, feedback you received and planned actions[2].

It is expected that you use a version control system[3] like City's GitLab service https://gitlab.city.ac.uk/ or a private GitHub repository, which gives you version control and a backup of your files. You will need to give read access to your markers and include a link to the repository in your report.

2.5 The academic project team and their responsibilities

The project team's responsibilities towards you include:

- Delivering a set of taught sessions to support the running of all projects; enabling you to learn about aspects that will support your work in your project;
- Holding drop-in sessions to answer project-specific questions, these include sessions in term 1 to help you define possible project topics; enabling you to seek appropriate help where needed and be more effective in your work;
- Defining who your academic project consultant/supervisor is; this enables you to receive individual feedback on your progress, and consequently improve the quality of your work;
- Monitoring that you are engaging with your project consultant/supervisor; enabling you to demonstrate your progress, receive feedback to improve the quality of your submission and demonstrate the authenticity of your work;
- Overseeing the marking process; ensuring that your work is fairly marked.

2.6 BSc academic consultants and their responsibilities

Your academic consultant's responsibilities towards you include:

- Meeting you at various pre-defined times, providing you with feedback and guidance to support your work;

[2] MSci students in Cybersecurity must, at the end of the project, submit the minutes of the supervisory meetings, as an appendix to their project report. This is good practice for all MSci students.

[3] Please give read-only access to your consultant/supervisor.

- Approving your PDD: ensuring that your project objective is clear and focused, and that it is feasible for you to achieve it and demonstrate that you did; as well as helping you ensure that your project work plan is complete and appropriate to your project objective; enabling you to demonstrate understanding of the relevant state of the art;
- Providing ethics approval for your work: ensuring that your project adheres to our professional codes of practice, as well as approving your Research Ethics Form and all your artefacts that are related to ethics (such as plans for interaction with participants, questionnaires and their corresponding consent forms): ensuring that your project is not endangering anyone involved (e.g., you, potential users, or participants of any of your project-related activities);
- Giving feedback on your work during the pre-defined meetings, allowing you to make improvements in good time for your submission;
- Supporting the assessment of your deliverables, including acting as second markers.

2.7 MSci academic supervisors and their responsibilities

Your academic supervisor's responsibilities towards you include:

- Helping you to ensure that your project objective is clear and focused, and that it is feasible for you to achieve it and to demonstrate that you did;
- Helping you ensure that your project work plan is complete and appropriate to your project objective; enabling you to demonstrate understanding of the relevant state of the art;
- Approving your PDD: confirming that your planned project fits your degree programme and satisfies the above points about objectives, suitability and feasibility.
- Providing ethics approval for your work: ensuring that your project adheres to our professional codes of practice, as well as approving your Research Ethics Form and all your artefacts that are related to ethics (such as plans for interaction with participants, questionnaires and their corresponding consent forms): ensuring that your project is not endangering anyone involved (e.g., you, potential users, or participants of any of your project-related activities);
- On request by you, providing advice on how to carry out your project work or referring you to a source that can provide such advice;
- Giving feedback on your work, including your write-up, when sought in good time by you, appropriate to the amount of time you have available before the relevant deadline(s);
- Supporting the assessment of your deliverables, including acting as second markers.

2.8 Your Intellectual Property Rights

About the ownership of Intellectual Property (IP) that might arise from a student project, the University has established a policy, which you can check online. The project module' process adopts and implements this policy. Some relevant rules are described here.

The initial presumption is that where a student has developed IP, the student owns the IP, that is, the student can for instance exploit it commercially. However, the student can also agree beforehand to share or cede the IP to someone else. This may be required by pre-existing agreements in cases where a student is sponsored by a company or other body; is performing their project as part of a larger sponsored project; is performing it for their employer; or is working on a project that derives from the IP of academic staff. In the last of these cases, if the University wishes to exploit a commercial opportunity, the student will be required to assign their IP to the University.

Regarding copyright in the project report and other submitted deliverables and results of project work, an exception to your rights as author is that the University retains the right to copy them freely for assessment or other internal purposes, including submission to plagiarism-detection services (Turnitin). Copies of **submissions or parts thereof** may be provided to other students as aids for their own projects or other parts of their studies; extracts of videos may be used to provide other students with guidance on their studies, as well as for promotional purposes. Authors may explicitly indicate on the report or video that they contain confidential material not to be disclosed, which will preclude these uses (see section on Submitting confidential material). In addition, if you pursue a client-based project, we may give the client copies, or show the client any material that you submit to us as part of your project.

IMPORTANT: to mark your project, the University requires the ability to examine your outputs and test your software product. Some clients may think that this conflicts with their wishes about intellectual property and/or confidentiality on either what you produce or their intellectual property (e.g. software with which your own software interacts, data that they provide to you). It is essential that you resolve these issues before you start your project. The project team will advise, if you ask, to help you reach a workable solution. Seek a clear agreement with the client and ensure that it is acceptable to City, by asking the project team. Students who neglect these precautions risk ending up with a high-quality project that fails because the client will not allow it to be marked. On the other hand, students who followed this advice have never experienced any late conflict of this kind.

3 Setting Up Your Project

3.1 Your responsibilities at the start of your project

At the beginning of the project, you must check that you are enrolled in the IN3007/INM450 Moodle module, read this Project Handbook document, attend the project sessions (the presentations with Q&As) and define your project topic. You must also keep track of announcements made via Moodle and direct email to you about your project. All email communication should be via your University email address.

3.2 Choosing a project

Your first task is to choose a topic. You may suggest your own topic or choose a topic offered by a client (subject to the restrictions set out in sections 3.3, 3.4 and 3.5). In either case, you should visit a module drop-in session to discuss your topic with a member of staff who will advise on the suitability of the project and the potential for gaining good marks.

To choose a project, you should attend the scheduled project sessions, check examples of good projects on Moodle, as well as investigate the possibility of pursuing a client-based project.

3.3 Project outputs

All projects must aim to deliver something of value to identifiable beneficiaries such that you can identify their requirements. Every project must include a substantial design-and-build part, but may produce other outputs as well. Projects will generally produce outputs belonging to these types:

- A software prototype to be used by someone other than the student;
- A working software system to be used by someone other than the student (e.g. a client);
- A substantial improvement or new feature for an existing software system, or a component to be part of a new system (e.g., in an open source project or a client's project);
- An evaluation of a working software or computer system, for a client;
- A software engineering method or technique, for use by a client or to be developed further by researchers;
- A detailed scientific result (e.g. from an experiment or study) to be used by a named academic researcher or a client.

Results of general surveys, literature reviews, business guidelines and discussion documents are **not sufficient as primary project outcomes.** Any project that produces mostly these outputs **will fail**.

3.4 Contents of project work

Choosing an appropriate project topic is crucially important. Good topics address issues in Computer Science and allow you to tackle challenges and demonstrate professional skills.

All projects must contain the design and construction of software or other digital products intended for users other than yourself. This component is **mandatory** and the design and construction must be non-trivial. The development should follow an established software engineering method and this must be reported in the final submission.

You can reuse existing software, provided that you have a right to use it (e.g., software made publicly available with explicit license for reuse, or software provided by your client) and that this reuse is clearly and completely acknowledged at all stages of your project. Reuse is a good software engineering practice. However, it is unacceptable to rely solely on such reuse; the project must include an amount of development work comparable to that required for projects in which no reuse takes place.

While all projects must develop some software, projects may have additional characteristics, belonging to one or more of the categories discussed below.

3.4.1 Research-oriented projects

You may pursue projects with a significant academic research component, that is, projects that aim to add new knowledge to what is already known about a subject. Note that these are challenging. They involve answering research questions of general scientific interest, by undertaking experiments, statistical studies, building and assessing novel systems, proving theorems, or other similar activities.   They also need to have a substantial design-and-build component, like all other projects.

For a research project to be successful, you need a good understanding, beyond casual reading, of the existing knowledge in that research area. Typically, successful research projects are those that are run as academic client projects (discussed further down).

If you propose a project with a significant research component to your consultant, the research question must be about computing, and it should be specific. Broad and general questions make bad research projects. For instance, asking what effects a certain new technology will have on business is not an appropriate topic: you would not be able even to read all the material already produced by professional researchers to answer the question, so you would not produce any useful new knowledge. However, a careful, professionally run study of the effects of that new technology on a specific company, or a small experiment to check whether a conjectured effect on a certain type of users does occur, could be a good project, adding new knowledge to what is already known about the subject.

Important: Academic research may include data gathering in the form of experiments, questionnaires, interviews and web-based forms. But projects that have data gathering as their main component are not acceptable unless they also involve building a product, and substantial effort in the design and building of this product and/or the rigorous analysis of the data and/or some other output, based on the data, that answers research questions and/or can be used as input for further work by a third party. Furthermore, academic research must be designed so that it is clear what factors affect the validity of the results and to what extent these results can be generalised. Projects involving only a literature review, and/or surveys of people and/or interviews are not acceptable.

3.4.2 Games development

This is design and construction of a game and/or games technology related software intended for users other than yourself. All BSc Games Technology students must undertake this type of project.

3.4.3 Client-based projects

There are two types of client-based projects, Academic Client projects and External Client projects.

Academic client projects are projects on a topic proposed by an academic who will act as both client and consultant. The academic, as client, will propose requirements for the project and evaluate the outcomes, and as the consultant they will review and give advice on your PDD and draft report. The project must produce outputs (e.g. computer products, evaluations, analyses, proposals) that are approved by the academic client. The project must, like all projects, include a significant design and build component.

External client projects are instead undertaken with an organisation, which is often external to the University. You will negotiate the aims and objectives with the client. The project must produce outputs (e.g. computer products, evaluations, analyses, proposals) that are approved by the client. The project must include a significant design and build component.

The Department will contact the client organisation at the beginning of the project to confirm the arrangements you have made, and again at the end of the project to confirm the assistance given to you, how they have evaluated your work and how satisfied they are with the outputs. You need to forewarn your client that the Department will require information at these two stages.

In planning a client-based projects it is important for you to have a clear agreement on what each party (you and your client) is entitled to expect from the other. This includes support they will give you (e.g. advice, IT resources, information), what intellectual property, if any, you will cede to them or share with them, what confidentiality obligations you accept. To plan these matters, check section 2.8 on intellectual property rights. In some cases, safeguarding the client's interest may require special arrangements on how to share the outputs of the project (e.g., demo at the client's premises). These require previous written approval by the module team. You should agree these details before finalising your PDD.

3.4.4 Application development

This category covers any other software development for a BSc project. It includes desktop applications, mobile apps and web apps. It can also produce a part of a software system, for instance some components or feature[s] for a system developed by others (e.g. a client or an open-source project). Note that a static website would not be an acceptable project.

3.4.5 Specialist MSci

MSci students must undertake projects in their specialist area, reflecting Level 7 learning outcomes, as defined in the Module Specification: these projects must demonstrate a deeper understanding of issues in their respective areas, have a higher degree of complexity and demonstrate a more mature professional knowledge than BSc projects. If you are a MSci student, please read Chapter 12 carefully.

3.5 Project choice: the five rules

Your project must comply with the following six rules:

1. *The Consultant Approval Rule:*

   Your choice of project topic and its related ethical aspects requires approval by your consultant.

2. *The Deliver Something of Value Rule:*

   Your project must solve a problem, that is, aim to deliver one or more outputs that are of potential value to identifiable beneficiaries other than yourself.

3. *The Build Rule:*

   Your project must include a substantial design-and-build element.

4. *The IT Work Rule:*

   Every project must address a topic in computing and must entail work that is not just a literature review or a survey.

5. *The Specialist Degree Rule:*

   If you are in a specialist degree (MSci degrees and BSc with Games technology), then you must undertake a project in your specialism.

3.6 What makes a good project?

It is important, when setting up your project, that you think ahead about what will be involved. Think about what work you will need to plan and undertake. Make sure there is scope to gain good marks: your marks will be influenced by how much work you do, how well you do it and how challenging it was. It is important that you try and envision how the project results will look, even at this stage, as it will guide you throughout the project.

All projects must solve some IT-related problem. Projects are more educational for the student, and earn more marks if professionally executed, if they deliver solutions that are actually useful for other people: for instance, a prototype of an application that solves a problem for which no application existed; or provides some function that other applications already provide, but demonstrates a better way of doing it; or a new feature added to an existing open-source project. Projects where the problem is defined by the student alone, or that implement some product that is not competitive from any viewpoint with products already in broad use, generally receive lower marks, and teach students less new knowledge or skills, because they are less challenging.

3.7 British Computer Society Accreditation

Membership of the British Computer Society (MBCS) is an important professional qualification that can lead to the Chartered IT Professional (CITP) Qualification. Our BSc Honours Degrees exempt students from parts of the BCS membership exams on condition that they pass a practical problem-solving project that satisfies the project criteria set out by them (https://www.bcs.org/media/1209/accreditation-guidelines.pdf, item 2.5). All our individual projects are **required** to satisfy these criteria. Your project consultant will help you to check that the plans you describe in your PDD are suitable.

4 The Project Definition Document (PDD)

4.1 Overview

The Project Definition Document (PDD) is the first deliverable that you will submit for the module. It is the specification of your project. It describes the problem that your project will address, what the project will achieve, the work plan for conducting the project and monitoring progress. It is the basis on which your project is planned, conducted and assessed.

Your consultant will approve the PDD, or request improvements. If they request improvements, typically you will revise parts of the document, and possibly of your plans, so that they can then approve. In rare cases, the consultant will have to request a change of project topic or objectives. In any case, the changes require the consultant's approval. If your PDD fails to obtain approval, this will amount to failing the module.

Once your (possibly updated) PDD has been approved, your project must then continue according to the approved PDD. Small changes of plan may be needed along the way, but changes in the topic and general objectives of the project are ***not allowed***, except in special cases beyond the students control **and only if authorised** by the project team.

You should start work on the PDD as soon as possible; if possible, discuss (part of) its contents during the module's drop-in sessions, for feedback.

4.2 Proper conduct

Important note: your project is not valid until you have your Project Definition Document with its related Research Ethics Form approved by your consultant. If you perform work that is unethical, or without obtaining the relevant ethics approval, you will be called before the Academic Misconduct Panel.

As with all your work, any material authored by others (text or figures) that you include, any ideas due to others (including your consultant or client) must be correctly attributed. See the sections of this handbook about "Good Professional Practice"**, and check in particular the instructions about correct referencing, about plagiarism, and about use of generative AI.**

4.3 Submission of PDD

You must submit your PDD as a single pdf or Word file via Moodle , and then meet your consultant to discuss it. The consultant will set a meeting according to the Project Calendar which is available on Moodle.[4]

At the meeting, have a copy of your PDD at hand for discussion with your consultant.

[4] Communication failures may occur. Do check your incoming Email and Outlook calendar. If you do not receive Email from your consultant by the start of the prescribed time window for a meeting, you are required to Email them to check and avoid unnecessary delays.

The consultant will record a decision whether the PDD is adequate or needs changes, using the Department's project mark tool, which will send you the decision and any notes added by the consultant.

4.4 Resubmission of PDD in case changes are required.

If the consultant's decision has been that changes are required, you must submit your improved PDD via Moodle. Inform your consultant when you have done so. The consultant will again record a decision and notes, and you will be notified by Email of whether the PDD is now accepted and you are allowed to proceed.

4.5 PDD structure

All PDDs must be divided into the parts listed below:

- Cover sheet
- Proposal, divided into:
  - Problem to be solved
  - Project Objectives
  - Project Beneficiaries
  - Project plan
  - Risks affecting the project
  - Legal, social, ethical and professional considerations
  - References
- Research Ethics Checklist
- Client information sheet (only for External Client projects, and mandatory for them)
- If you used AI to produce any parts of the PDD, an appendix (no limits of length) reporting exactly what you did (see section 9.7, Rules on use of generative AI)
- For some projects (category 3 in section 4.5.7 " Legal, Social, Ethical and Professional Issues (LSEPI)", an extra appendix on LSEPI issues.

Format the PDD to be well readable: font Times New Roman 12pt or a larger font, page margins of 2.5 cm. The PDD must print well on A4 paper. Do not add your name, userid, student number and/or email address as a footer or header.

PDDs for BSc projects can be up to a maximum of 1700 words in length; PDDs for MSci projects can be up to a maximum of 2700 words in length. This excludes the cover sheet, the references section of the proposal, the ethics checklist and the client information sheet.

4.5.1 Cover sheet

The first page of the PDD must be a cover sheet with the following information:

- The title of your degree programme;
- A short, informative project title;
- Your name and email address;

- The name of your consultant and name(s) of any client(s);
- Who proposed the project (you, your client or someone else);
- A brief outline of all arrangements for proprietary interests in your project work and/or outside help. Attach documentation of agreements. If there are no such arrangements, this must be clearly stated;
- Any other promises you are making in order to secure acceptance of the project (e.g. by your consultant);
- Word count[5] for the document.

4.5.2 Proposal - Problem to be solved

This section must give a clear and concise statement of the problem to be solved by the project. Remember that all projects must solve an IT-related problem and involve substantial" design and build" work in creating an IT product. The problem might be for instance the lack of a software system to fulfil a purpose, a complex or difficult design, an implementation problem to overcome, or an unexplored opportunity that has arisen from advances in information technologies; a commercial problem that requires an IT solution, a failure with a current IT system, a need for some new IT solution. For an academic client project, the problem might be a research problem, identified by your client, that requires a specific solution, an evaluation of new technologies, a standalone experiment in a larger research project in which your client is involved, the development of new software engineering methods or techniques, etc.

All projects must be based on existing knowledge about the topic. Therefore, this section must refer to existing relevant work (e.g. software products, research findings reported in books, academic papers or other documents). For details of how to write citations and the References section of your PDD, check section 9.5 "References and referencing style" of this handbook.

4.5.3 Proposal - Project Objectives

A successful project will have well-defined objective(s) and, where relevant, well-defined research question(s) and/or hypotheses. The objectives are what you aim to achieve through the project. There should be a single main objective – this will give the project a clear focus. It needs to be explicit and unambiguous. This will then be broken down into sub-objectives that have to be met for the overall objective to be met.

You should define the objectives with a subject and verb such as: *"This project shall …"*.

Objectives should be "testable": you need a way to check whether by the end of the project they have been met. It is important at this stage to identify, for each objective or sub-objective, how you will know, at the appropriate time during your project, whether, or how well, you achieved it. Briefly state this "test" , for each objective, in this section of your PDD. In your Project Report you will be required to demonstrate application of this test (or these tests) and report the result (objective is met, not met, or what parts of it are met).

[5] Use the word count facility of the text editor that you are using.

In the case of research projects, you also need to specify research questions and/or hypotheses. A research question is a question for which your project is meant to find an answer: the main objective of the project is to answer these questions. By the end of the project, you should have obtained data and results to enable you to answer each research question. For example: *“Will the XXX method improve ....?"* or *“By how much will the use of the YYY software tool reduce ...?"*. For some research projects, it will be appropriate to define *hypotheses* that your research will test. A hypothesis is a tentative statement that your investigation will try to support or reject. Hypotheses propose potential answers to research questions.

4.5.4 Proposal - Project Beneficiaries

You must state who will benefit from the project and how they will benefit. This makes sure that you have a consistent view: whose problems you are going to solve, and thus, to whose needs you need to pay attention in your work. This is typically a very short section, as who are the beneficiaries may require very little explanation. E.g. if you are writing software for a client, it is usually meant to benefit directly that client and their customers.

Projects that are mostly about building a product will typically benefit stakeholders such as further developers and/or end-users of the software. For projects that are run for an external client, beneficiaries might be the client organisation, its stakeholders and/or its customers. The results of academic client projects will usually benefit members or parts of the scientific and technical community first (e.g. a specific academic or team or research project, or broader communities of researchers), and hopefully end-users and/or other people later.

Whatever the project, specify the expected benefits clearly, and how at the end of the project you will be able to demonstrate whether you have actually delivered the intended benefits. Some benefits may be meant to be realised after your project ends; in this case, be clear about what parts of your objectives you *will* be able to judge by the end of your project. For instance, a product may be meant to improve a client's business results, by a certain amount after being in use for one year. At the end of the project, you will not know yet whether it will indeed produce that benefit; but you can know whether the product works reliably, and how well the client judges that it matches what he/she expected, or other indications of how well the longer-term benefit is likely to be realised.

4.5.5 Proposal - Work Plan

You will need to undertake a systematic piece of work, based on appropriate methods, to achieve your project objectives and/or answer your research questions. You must produce a work plan for this. The work plan is an outline of the work you will do to produce the intended outputs and intermediate products of your project, as well as when the various stages of work will happen and produce their results.

Produce a simple work plan divided into activities with their outputs, resources, start dates and end dates. It may be useful to cite further sources to justify your choice of methods. Try to be specific, e.g. “This product will be programmed in C++, using the libraries available from company so-and-so, and tested using the tool available from my client”; for parts of your method that are undecided, say how and when you will finalise them. This will help you to reflect on how best to plan your work and will enable your consultant to give you useful advice. Consider whether your initial work plan is feasible, and correct it otherwise. Include a simple diagram of the work plan with a time axis (e.g. a Gantt chart).

It is important for you to relate the planned work to the outputs that you intend the project to deliver. Projects in this module must deliver 2 different types of documents and artefacts:

- *Deliverables* that you are required by the module's rules to produce for assessment, with fixed deadlines: the Project Definition Document and Final Project Submission.

- Project outputs specific to your project, and needed by the immediate beneficiaries of your project. This is what brings marks and your Final Project Submission has the purpose of documenting them. Think carefully about what outputs these people will need (usually a piece of software, but also reports and documents of various types), and define the work plan so that it delivers them.

During your project, use the work plan to control progress: keep track of your work and compare it against the plan. Deviations will indicate that you need to change something, to ensure the success of the project. When reporting progress to your consultant, refer to this plan so that they can advise about responding to deviations.

Note: the various Kinds of Risks

The PDD must demonstrate that you understand the main risks that are associated with your project and you have appropriate plans to mitigate these risks. “Risks" means harmful events that are not sure to happen, but could plausibly happen and, if they happen, will cause harm.

For events that may cause serious harm, you need to have an idea of how you can (1) make them less likely to happen, and/or (2) limit damage if they do happen. Likewise, a car must have good brakes to avoid collisions; but also seat belts to reduce harm if the collision does happen; and owners need insurance to provide for *recovery from the harm done*, e.g., mending the car and compensating for damage.

Three kinds of risks are important: (1) risks *to* your project, which may cause it to fail, or fail in part, and reduce your mark; (2) risks arising *from* your project, which may harm you or others, and are treated in your PDD as part of "Legal, social, ethical and professional considerations"; and (3) risks to people *involved in* your project, that are covered by the “research ethics” policy and explained in the handbook section about the Research Ethics Checklist.

All projects must conform to the ethics code related to our profession. IT professionals (including Computer Science students) must not undertake work that can endanger individuals, governments and/or organisations. Our work must conform to the laws of the land and also not provide means for others to conduct illegal or dangerous activities.

4.5.6 Proposal - Risks to your project

Project risks can arise in many forms. Examples include:

- Lack of resources, such as the absence of your consultant, client or important resources;

- Changes in the requirements of the project client or sponsor at key stages in the project;

- Serious failures in the technology, including programming problems that cannot be resolved, hardware failures and incorrect assumptions you may have made about what can be achieved.

While some risks are common to all projects (e.g. disk failure or theft), you should *focus on identifying risks that are more specific to your own project* (e.g. if it requires you to become familiar with an especially complex technology, or there is a concrete risk of a late change in your client's requirements, etc.) and outlining your strategies and fall-back plans regarding them. We note that often students focus on how to avoid the bad events happening, for instance "to avoid programming problems, I will study carefully the languages and libraries to be used"; but it is also important to plan what you can do if those problems happen anyway; for instance, have a plan about which non-essential feature of one's product one may drop if necessary, so that a delay due to programming problems does not sink the whole project.

A category of risks has to do with legal matters. For instance, all projects need to use items (code, data) produced and owned by others: before using them, you need to know that you do have this right (see also chapter on Good Professional Practice), and if you have not yet checked or not yet obtained these rights, you must consider the risk of not obtaining them or obtaining them too late.

4.5.7 Legal, Social, Ethical and Professional issues (LSEPI)

In this section of your PDD you need to demonstrate that you have thought properly about this set of concerns and how you need to take them into account during your project. The most important concerns are the risks that the outputs of your project (an IT product, in all projects, but also in some cases other outputs, like answers to research questions) may create for others, and your legal obligations. You need also to reflect about the more general effect (benefit, harm or just changes) that what you produce may have on various people and society as a whole. All this is part of your duties of professional and social ethics.

You were given bases for these reflections in your module IN1012 "Computer Science, Ethics & Society": both some knowledge of general issues and approaches for examining them, like the FATP "framework". You must also read in full and apply the advice given in this handbook.

Ideally, your PDD will show how you already know that there is no serious risk of harm to others and that what you do and what you produce will be fully compliant with legal requirements. If you still lack the information to confirm all this, you must show that you have planned how to ensure that such problems will be solved before the end of the project. For instance, you may not know whether the owner of a certain data set will grant permission to use it, but you must then show that you are aware of the need, you have already planned when the permission needs to arrive (this is a date in your workplan) and what is your fall-back plan if it is not granted (this is listed in your "risks to the project"). Or, you might need to state that whether a product will present excessive risk for users can only be found by some empirical measurement on that product, which you have planned, and you have also planned how to reduce that risk if the measurement detects it.

To organise legal considerations clearly, one way is to list first items (e.g. data or code) that you plan to use and whether you have already verified your right to use them the way you plan; secondly whether your outputs may incur legal problems (e.g. if it is a social media service, could it violate privacy, or allow display of forbidden material, allow unlawful use by minors, etc.).

To what extent and depth must you discuss Legal, Social, Ethical and Professional Issues?

From this viewpoint, projects fall into three broad categories which require different depth of this discussion in the PDD:

1. Many student projects carry no direct risk to others or direct social consequences because they are not meant to be distributed. For instance, most software development projects by students are meant as "proof of concept" prototypes: they allow you to demonstrate a new idea and evaluate it, but they are not meant to be deployed as products. For such projects, the PDD must in this section (a) explain why the project is of this kind, that in itself does not pose risks or concerns and (b) give, however, a summary of these concerns as they would apply if the outputs (e.g. the software product) were for actual use by third parties or the public under your responsibility. Even for this kind of projects, you may want to plan work about how to limit risks, as part of the new ideas that you want to demonstrate (e.g. if your project deals with better ways of introducing security features), or as a way of assessing the practicality of your ideas and product by showing how to ensure that they do not cause undue risk.
2. Some projects have outputs that are meant for practical use, but the responsibility for their use or distribution and for appropriate precautions is taken by others, not by the student. A typical case is that a client takes responsibility for assessing and eliminating any undue risk. In this case, this section of the PDD must (a) document this third-party' responsibility but also (b) give a summary of these concerns as they would apply if the outputs (e.g. the software product) were for actual use by third parties of the public under your responsibility. You must also (c) **describe how your outputs for the client will support their decisions regarding these concerns**. For instance, specify whether you will deliver measurements of response time, or usability testing, and also whether you will add your own assessment of what decisions they would require regarding deployment or improvements to the product; and that you have agreed that the client will be responsible for assessing the validity of the measurements and for deciding whether and how to deploy, use or distribute the product.
3. Some rare projects are meant for use of the outputs under the direct responsibility of the student. The PDD must then document what risks are possible and how in practice you will take care of the concern in point (b) above. This typically requires either a type of outputs that is inherently incapable of causing harm (and such that this can be proved without special skills) or more professional qualifications than you have acquired in your studies. **In this last case you may need extra space for describing your plan to properly control risk and your ability to implement this plan, and you are allowed an extra appendix of up to 800 words.**

For all three categories of projects:

- Consultants are required to disallow any project that poses excessive risk, and also to fail any PDD that does not contain the discussions outlined above.
- In your report, in the final submission, you will be required to include fuller discussions of these issues. This is both to document that in running your project you took the appropriate precautions as planned and/or as the need arose, and because the British Computer Society requires that we assess your ability, in your future professional life, to reflect on the consequences of your work and take a responsible professional stance.

We give here additional advice that you need in order to put these requirements into practice.

An essential part of professional ethics is "do no harm"[6]. So, you must consider whether your project poses risks to others. If you are producing software for others to use, one risk for them is that you do not deliver, and you will have considered this already as a risk *to* your project. But if you deliver a product that seems to work and instead has flaws and misbehaves, especially in ways that users do not notice, or that can be misused, the harm to others may be even greater. For instance, any E-commerce product puts its users at risk of attack. Then, you need to be clear about whether your product will have all the necessary security (and how you will ensure that it does) or instead it will be an evaluation prototype, and your possible users or client will be warned that it is not ready for commercial use.

This E-commerce example indicates that we need to think of *what harm* a project can do; and of how to *reduce this risk* to acceptable levels. Where potential for harm exists, professional organisations have specialised methods and skills for controlling the risk; a student cannot emulate this capability. Ways that you can avoid risk to others are:

- if your product and other outputs of your project have no, or negligible, potential for harm;
- if you work for a client organisation that will apply to your project the appropriate professional methods for controlling risk;
- if you set effective protections and warnings; for instance, for a flight simulator game, making it deliver clear warnings that it is not suitable for actually learning to fly.

So, the first requirement is to *identify* how a project might harm people. This is sometimes hard to do; we easily imagine the good things that we *want* to achieve, not the bad things that we may cause by accident. We give two examples:

- An innocuous game asks users to set up a password for access; some users choose a password that they also use for other, important accounts; the game software then becomes a way for criminals to steal passwords and enter people's bank accounts;
- A product presents to users information from the Web, say about some consumer products, or the stock market, or sights to see in a city, or self-help advice. If the Web sources of information used are incomplete or erroneous, become outdated, or are changed by criminals, this may harm users through for instance erroneous purchases, dangerous health-related decisions, etc.

To orient yourself before the start of your project, there are two complementary methods for identifying possible risk (use both):

[6] You can read codes of conduct for IT professionals, for instance, at https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/ or https://www.acm.org/code-of-ethics
Other parts that are very important for student projects are explained in the present handbook in the section on Good Professional Practice

- Consider what your project output (e.g. your software) will be *capable* of doing. For instance, a satnav system gives directions for going from one place to another: wrong instructions may cause traffic jams or accidents. Or an app may store private information about its user; if the information is not properly protected, criminals might steal it and thus harm the user;
- Consider the steps through which the product will operate and users interact with it; think of how each step could deviate, by accident or intentional misuse, from what you intend, what the consequences might then be, and whether you can prevent either the deviation or the subsequent harm.

Last but not least, you need to consider the limits of your skills. The student project is an occasion for you to learn; your product will not normally have the reliability or the security of a commercial product. Typically, student products are only useful as demonstrations of design concepts, or prototypes that can be redeveloped into full-fledged products; or components to be integrated by a client organisation into larger products, after applying necessary quality controls. Without these controls applied by a capable organisation, we expect student project products ***not*** to be ready for distribution to general users; and if you plan to distribute them, you must seek advice about ethical advisability and necessary validation steps.

This issue links to the professional principle that the British Computer Society summarises as “Show what you know, learn what you don't”, with detailed rules like “only undertake work that is within your professional competence”; “[do] NOT claim any level of competence that you do not possess”; “develop your professional knowledge, skills and competence[...]”; “ensure that you have the knowledge and understanding of legislation and that you comply with such legislation [...]” (you can find the whole code online: https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/).

For a student project, these principles mean that you must: learn what is necessary for serving the user’s need as well as possible, within the limits of a 450-hour project; but make users aware that they cannot trust your product as they would trust a professional product; be especially concerned about naive users, who might over-trust your product.

Your PDD needs to briefly state the potential for risk to others that you have identified and how you plan to reduce any risk to acceptable levels.

4.5.8 PDD: Research Ethics Checklist

This section concerns risks covered by the Research Ethics Policy of the University, that is, affecting people that are involved in your project (even though they will not be affected by your project's final outputs).

All projects must conform to the ethics code related to our profession. IT professionals (including Computer Science students) must not undertake work that can endanger individuals, governments and/or organisations. Our work must conform to the laws of the land and also not provide means for others to conduct illegal or dangerous activities.

All projects must undergo an ethics approval process, **governed by City’s guidelines for research ethics**. Without ethics approval, a project is not approved, hence cannot be undertaken (let alone pass the module).

From the Research Ethics viewpoint, projects can be high, medium or low risk:

- **You are not allowed to propose high risk projects (please review the advice given in the research ethics checklist), except in the case that the project is proposed by a client who *already* obtained ethical approval for it.**
- **Medium risk projects require approval from the Computer Science Research Ethics Committee (CSREC) but may be forwarded to the Senate Research Ethics Committee (SREC) depending on the level of risk in your application.**
- Lower risk projects can receive ethics approval from the consultant or supervisor. This process is supported using a research ethics checklist, concerning parts of your ethical obligations.

All students must complete a Research Ethics Checklist and include it as section 3 of the PDD (see also section 9 on “Good Professional Practice”). The checklist helps to determine whether you need approval by an Ethics Committee.

Detailed guidance for forms to be used will be available on Moodle. It is your responsibility to pay attention to ethical issues that may arise from your project. These concerns are ***not limited*** to any specific type of project. Projects must be planned and executed with adequate safeguards for the rights of all people, and projects must not proceed if these guarantees are not possible or are not provided.

A specific set of rules applies when a project studies people, or data about people. More precisely, quoting from the Research Ethics Policy document:

> *“all projects involving human participation [...] should undergo ethical review. In REP [research ethics policy], ‘human participation’ covers direct data collection from people, for example surveys, observation and physiological measurement. It also includes retrieving data from individual people records. Research involving anonymised secondary data, such as reviews of published or otherwise publicly available documents and analysis of aggregated data, does not require ethics committee approval[7]. Research involving animals requires ethical approval too, but it is not covered by REP and should be submitted for consideration by the Research Ethics Committee of the Senate”*

Every student must assess their project in view of ethical principles and safeguard the rights of all involved. Guidance applying the current policy will be in the IN3007/INM450 module on Moodle, with reference to policy documents, forms to be used and document templates to follow. You must read this document to understand the possible research ethics issues, and follow the instructions it contains.

You must consider early on the possibility of these issues arising in your project and discuss them with your consultant/supervisor. Otherwise, you risk planning a project that cannot be allowed to continue, or that will be delayed heavily. Consider that research ethics issues,

[7] Note, however, that some forms of publicly available data are protected by law. E.g. forms of "data scraping" from social media are illegal (https://ico.org.uk/media/about-the-ico/documents/4026232/joint-statement-data-scraping-202308.pdf). If you consider a project that involves "scraping" of personal information from the web or social media, check what rules apply and seek advice as needed from the project module team.

dealing with people's participation in the project and collection of data about them, may not be obvious.

Research ethics approval is required for the final version of your work plan and for artefacts that you produce during your project (typically, questionnaires, participant information sheets, procedures for ensuring confidentiality). Usually, your PDD will at most contain drafts of these items, because the details of how you will run experiments (or any activities involving participants) will only become clear later in the project. So, **remember that you will need approval later** (from your consultant/supervisor or the appropriate ethics committee) for your work plan, questionnaires, consent forms etc., once they are in their final form and before using them.

Please note that you MUST obtain approval from your consultant/supervisor before you recruit participants for your project. Failure to obtain approval is regarded as severe academic misconduct by the university and will result in an academic misconduct investigation with implications for project e.g. reduction in marks or failure.

Note that if you plan to use human participants you need to explain (in your original workplan or in the material attached to the checklist or in the final workplan submitted according to the above paragraph) what you intend them to do, for which purpose, and how you will recruit and select them. Without these details it would be impossible to give approval.

4.5.9 PDD: Client information sheet (for External Client projects)

For all projects run for an external client, the PDD must also include a "client information sheet" containing the information specified in Appendix A of this Handbook.

The client information sheet must be included as an appendix at the end of the PDD. The Department will ask the client to confirm the project and that the details are correct.

In some circumstances you will also be required to provide documentation of a sponsorship arrangement, in which you get some benefit from assigning ownership of. or a licence to use, your intellectual property.

4.6 Approval of the PDD

Projects are only valid once the consultant has approved the PDD and Ethics-related artefacts. If your consultant determines that your submission cannot be approved in its current form, then it is your responsibility to make the changes required, and resubmit by the deadline for resubmissions, so that the material can be approved.

5 Meetings with Your Consultant

You must meet with your consultant as described in the timeline found on Moodle[8]. These are meetings with specified scope, meant to ensure that you are on the right track to succeed in your project and provide you feedback to help you make the most of your work. They also provide us with reassurances about your progress and the authorship of your work.

The first meeting is a group session where you will meet you consultant with their other students. This will be an opportunity to get feedback on your initial ideas and to plan your PDD. In your second session you will meet one-on-one to discuss and approve (or not) the PDD, as discussed in Section 4. The other meetings are for you to report progress to your consultant, showing various drafts of what you have written, and to obtain feedback to help you (1) improve your work and (2) improve the way that you are reporting on the work undertaken. You will have short meetings to focus on: what you read and found about the project's context (literature etc); methods used; and results obtained. You will also be offered the opportunity to have a longer meeting to show your consultant a draft of your full report.

Your consultant will discuss your drafts with you, focusing on the areas that you have identified as points for discussion and provide you with verbal feedback. You should make notes on the feedback received at every meeting with your consultant and keep them for reference. It is general good practice to share the meeting notes with your consultant, to get an agreement that your understanding of what was discussed is correct.

You should engage with the feedback that you receive at these meetings: it will help you improve your chances of getting a good mark in your project.

Last but not least, a mandatory Final Project Review will take place, See the related part of this handbook.

[8] If you don’t engage with your consultant, you will be called to a viva with the consultant and the module team.

6 The Draft Project Report

The Draft Project Report is a draft of your final project report that you can show directly to your consultant to get feedback. During your meeting, your consultant will discuss with you the aspects of the document that you request feedback on.

The Draft Project Report must include a full table of contents (covering any sections that are yet to be written) and a clear indication of how you plan to evaluate your work. You are advised to discuss with your consultant parts that you still intend to include in the document, especially if you do not to have a full draft ready by the meeting. Your consultant can advise you about prioritising work on what you submit.

Discussing your draft project report with your consultant is not a mandatory step. However, when doing so your consultant will give you feedback, e.g. regarding major omissions, defects of clarity or relevance, inadequate arguments or unsupported claims. It is very important that you act on this feedback to produce a good project report. There are no mark penalties for skipping this step, but you risk missing vital advice.

Note that your consultants' tasks do not include proofreading your writing, or warning you about signs of poor practice or misconduct. Proper writing and professional behaviour are your responsibility.

7 The Final Project Review

7.1 Final Project Review Overview

Students are required to present all the outputs of their project to their markers (the student's consultant/supervisor and another, independent, marker), after the final submission. This is called the Final Project Review and is mandatory. You will be asked questions about your work (e.g. about methods, source code, data collected): the ability to explain your results and method orally is one of the learning outcomes of this module. The review carries no marks but passing your project is dependent on attending it.

The project review serves several purposes:

- To communicate information to the markers about the project and its results, introducing further details to the independent marker to the project;
- To demonstrate the student's ability to explain their work;
- To protect against academic misconduct by asking the student to explain the project, including materials, methods and results of the project.

Not attending the review will lead to a 0 mark on the project, regardless of its quality. Students that are unable to demonstrate they understand their projects content and implementation maybe considered for academic misconduct.

7.2 Organisation

Every student must be available for the Final Project Review during the period specified in the project calendar and must attend in person, on university premises, at their individual set time. The time period for the final project reviews is chosen not to interfere with other scheduled activities.

7.3 Contents

A typical review will last approximately 30 minutes and will have the following structure:

- You start the review by describing your problem and objectives briefly. The independent marker might need a basic explanation of what your project is about. Be ready to answer additional questions.
- Then you answer questions, demonstrating and explaining **all** outputs from your project and how they were produced. For the software that you developed, you will likely be asked to demonstrate parts of how it works, beyond the demo covered in your submitted video, and to explain parts, chosen by the markers, of the code and the design documentation. The markers may ask you about any aspect of your project; they may also ask you to refer to the material you submitted and explain or clarify any part of it or how you produced it. If you have produced specifications, analyses, reports for clients, manuals, data sets and analyses, algorithm descriptions, you may be asked to show and explain these. It is a good idea to refer to your project diary when you want to recall what happened earlier in the project.

Important: arrive early at your review with all material ready to show and demonstrate. E.g., if for demonstrating your software you will use your laptop or other device, arrive with the battery charged and having checked that you can connect on Wi-Fi to all the sites needed for the demo.

8 The Final Project Submission

At the end of your project, you must submit a final project report and all project outputs for assessment. The project outputs include a short video demonstrating what you have developed, as well as all code and other digital and non-digital artefacts that you have produced.

The Project Report is the principal basis for assessment of your project by the markers. This is where you must report all the work you have done. It consists of a main report plus appendices, documenting all the work done and all the materials produced. You should start work on the Project Report as soon as possible, typically just after your PDD has been approved.

The next few sections separately explain the structure of the report and what other material you must submit. Instructions on how to submit will be detailed in a separate document.

In this version of this document, updates have been made that cover in more detail reporting on Legal, Social, Ethical and Professional issues.

8.1 Project Report: Basics

8.1.1 Structure

The report must consist of a title page, table of contents, 100-to-250 word abstract, main body, references and appendices. Optionally, a glossary may also be included. The main body must be structured as set out in section 8.2.

8.1.2 Writing style

This is a formal report, not a diary of your work. Write simply, formally and clearly. Avoid excessive use of the first person singular. Be factual. Do not exaggerate. Do not use informal abbreviations (e.g. *"don't"*, *"u"*). Write in the past tense, because the report is a retrospective account of what you did and what you found. Use correct spelling and grammar. Use diagrams to aid communication, but ensure that you explain their meaning and provide a key for any symbols.

You should assume that your reader has studied computing but not the specific field(s) that your topic addresses. Define any specialist terms when they first appear. A separate glossary can also be useful. Try to avoid technical jargon.

8.1.3 Formatting

All prose text must be in one of these fonts: Times 12pt, Times New Roman 12pt, Cambria 11pt, Arial 11pt, Aptos 11pt or Calibri 11pt. Program code must be provided using a monospaced / fixed width font (for example, Courier New or Menlo 11pt). Don't use screenshots of your code editor for including code.

Use page margins of 2.5cm. Start each chapter and main appendix on a new page. Use informative headings. Prepare the report so that it will print well on A4 paper. Do not add your name, userid, student number and/or email address as a footer or header.

8.1.4 Main report word limit

The main body of the report, excluding the title page, table of contents, abstract, glossary, references and appendices, **must not exceed 12,000 words for BSc projects and 15,000 words for MSci projects.**

8.1.5 Appendices

The main body of the report must be a stand-alone document, that is, readers must be able to read through it and find the most important information about the project without ever going to the appendices. Any additional details that do not fit into the main body of the report, but are required to document your work, must be put in appendices. Label the appendices as Appendix A, Appendix B etc. A common mistake that you should avoid is to write the main report as a sequence of references to diagrams, software code and text that are in appendices.

8.1.6 Page numbering

All pages must be numbered.

8.2 Structure

The main body of the report must consist of the following chapters. In addition to the specifications here, the weekly lectures answered questions raised by students for clarification.

8.2.1 Cover sheet

The cover sheet must contain the same information required for the PDD (see section 4.5.1).

8.2.2 Chapter 1: Introduction

Chapter 1 is a concise introduction to your project and a summary of what readers should expect in the rest of the report. It must:

- Describe the problem you tried to solve;
- State the project objectives, sub-objectives and research questions (if relevant);
- State the anticipated beneficiaries and how they are expected to benefit;
- Outline the work you performed to meet the objectives and/or answer the research questions;
- State any assumptions that you may have made and, where relevant, how you limited the scope of the project.

Use your Project Definition Document when writing the Introduction chapter, updating the information as appropriate to reflect details that have changed from your PDD and writing in the past tense. The Introduction must explain how things really went as opposed to what you had planned in the PDD. If there were substantial changes from the PDD, you should mention them here.

8.2.3 Chapter 2: Output Summary

Chapter 2 specifies the outputs from your project and can be a maximum of **2 pages** long. All projects must produce at least one output (deliverable) for use by a named individual, group of individuals or organisation. It is mandatory to specify these outputs in Chapter 2 and include them in one or more of the appendices. Note that “outputs” are *not* the internal deliverables of the project, such as the PDD. They are the results produced by your project. For each output, you must provide:

- An overview description. For example: *“Video description of how to use the XX application.”*
- A statement of what type of output it is (e.g. software code, a system specification, an evaluation report), its size (if relevant) and any materials it re-uses. Be brief and informative, e.g. you might state *“the deliverable is 10,000 lines of C code (not counting comments), of which 3,000 were written by me and the rest re-used”*.
- The intended recipient(s) and/or end-user(s) of the output.
- How these recipients will use the output and benefit from it.
- Links to the sections of the Results chapter, identifying how the results described in that chapter contribute to each output.
- Link to one or more Appendices where the complete output is reported**.**

For instance, this summary may contain information like:

> *“the first output is a software product that classifies gene mutations as prescribed by the standard classification prescribed in reference (Xnjkls, 1999).”* or
>
> *“an analysis and diagnosis of the malfunctions in the automatic management of security patches on BlaxR2 servers at company SplatBfec, with a proposed solution”*
>
> *“This product is built as 33 Java classes, amounting to 10,300 lines of Java code (not counting comments), of which 10 classes (3,200 lines plus comments) were written by me and the rest are re-used from the sources indicated in the Results section. The complete software plus its design documentation can be found in the OneDrive folder https::/..... and can be used on the web at the URL: https://....”*
>
> *“this product was delivered to my client, whose research team will use it to ...” or “this software will be used by my client CompanyX in their business for...”*
>
> *“a prototype of the web service is hosted at http:... so that the markers can test it”*

8.2.4 Chapter 3: Context

Chapter 3 summarises the basis of knowledge on which your product is based: readings that you undertook to enable you to do the project work, existing products that you intend to improve or from which you learned, etc.. It must summarise the main topics that you read about and provide full and correct citations to the documents that you read. This Chapter will usually be at most **1,000 words** for BSc projects and at most **2,000 words** for MSci projects**.**

You should organise this chapter by topic. For example, for the design and build parts of a project, these topics might be: the application domain of the software; similar software products that already exist; other software products that have similar elements, from which you took inspiration of how to implement or not to implement your own; the type of algorithms and architecture required. If there is a client, there may be documents describing their specific domain, the current state of technology in that area of business, etc. For academic client projects with a research focus, you would normally have read previous research publications, describing the state of knowledge before your project and helping you to refine the method for running your own project. You will have read several sources for each topic: briefly report for each source its relevance for your project; make sure to list the source itself in the References section and cite it in your text using the required format for in-text citations.

In the other chapters, you must show how you used this information: explain your decisions, with reference to the sources of information and/or advice that affected them (therefore, make sure to have in-text citations of the sources listed in your References section, wherever in the report you give these explanations). **In this Context chapter**, instead, we are looking for evidence that you have read an appropriate choice of material, studied existing products and other information, etc, in the domain of your project. We will award marks for this and for how well your project decisions used the knowledge you found.

8.2.5 Chapter 4: Method

The Method chapter must provide a detailed and objective report of the work you have undertaken. It should describe the analysis, design, implementation and evaluation activities of your project in detail. Be prepared to state the obvious (no detail is too small for the Method chapter and the relevant Appendices). You should give an accurate description of your work, set in the context of a clear picture of the project plan and general approach.

Do not report general information about software engineering methods and life cycles: e.g. do not explain the alternative lifecycle models (waterfall etc), but state which one you followed and detail what was specific to your own project. Readers are interested in how *you* applied the standard methods, how you deviated from them and why; and the history of your project, with the main decisions and changes of plans. Do not use the Method chapter as an opportunity to include information that could not fit into the Literature Review chapter. At best this will give you no extra marks and, in the worst case, this will result in marks being deducted for poor presentation.

Do not report in the Method chapter results arising from the application of the method (this is what the Results chapter is for).

You must also be clear about any materials that you reused during your project. E.g. if you reused code or consent forms or questionnaires or material from online tutorials, you must state this clearly in this chapter.

The recommended length of the Method chapter is **2,500 words** for BSc and **3,500 words** for MSci projects.

8.2.6 Chapter 5: Results

The Results chapter must present **ALL** of the results produced during your project. This includes outputs from analysis, design, prototypes, experiment, evaluation and other activities. Every activity reported in the Method chapter should deliver results that are reported in the Results chapter. Be prepared to state the obvious (no detail is too small for the Results chapter and the relevant Appendices). It is useful to divide the presentation of results into appropriate sections: e.g., for a software development project there may be sections for detailed requirements, software architecture, specific algorithms and coding decisions, verification and evaluation results; for a research project there may be separate sections about the data collected and the analyses that were conducted. Be very clear in identifying what data or other resources you had at the beginning of the project, which ones you collected or created and which you derived in your further analysis. The results should be traceable to descriptions of each output in the Output Summary chapter.

Should you include pieces of source code, design, specification, requirements diagrams, in the Results chapter? This is a frequently asked question. The body of the report must be such that readers can understand the main characteristics of the work done, without the help of the appendices. So, among all the material that you *could* include in the Results, you must choose on the basis of importance. Those items, or selected parts of those items, that are most important for understanding the work will usually be in the main body of the report. They may also be repeated in full in the appendices, so that readers can see the complete output. Examples of what might be appropriate to include in the main body of a report might include:

- a specific sequence diagram, because it clarifies the way a certain essential protocol works;
- a specific class diagram, because it clarifies some important relationships among data items in the state of the software;
- selected requirements that are important and distinctive for your software;
- snippets of source code[9] that are important for understanding your software, or represent important parts of your original work (e.g., algorithms you invented).

The reported results must be a significant portion of your report. We recommend that the Results chapter should be at least **6,000 words** for BSc and **7,000 words** for MSci projects.

8.2.7 Conclusions and Discussion

The Conclusions and Discussion chapter(s) tie up and discuss all of the project themes. It must include a section on legal, social, ethical, and professional practice issues (LSEPI). Students that do not include this section will have marks deducted.

The first thing that you want to do in this chapter is to revisit the project's objectives and, if applicable, research questions and hypotheses. You will want to demonstrate to what extent each of the objectives has been met, the project's research questions (if any) answered, and the research hypotheses (if any) accepted or refuted. You will then want to draw more general conclusions from the above.

You need to assess not only to what extent you have achieved the various sub-objectives set out in your plan, but to what extent the whole project satisfied its overall goal. If the latter (for instance, an intended benefit for a category of users) cannot be assessed yet, or only partially, give an objective statement of what conclusions can be drawn by the time you write your dissertation. In general, do not claim achievements that are hoped but not yet demonstrated. As in the rest of your submission, claims that are not proved will cause poor marks and are poor academic practice or academic misconduct.

[9] Snippets of source code should be provided in text, not as a figure!

A good project report should also discuss the conclusions drawn from the objectives achieved or not, research questions and research hypotheses. This might happen in 2 parts. Firstly, each conclusion is discussed in its own right. Secondly you should link each conclusion to the state of things before your project, outlined in the Context chapter, presenting the contributions that your project work has made to solving the problem and/or improving knowledge , and comparing this to previous work by other people.

The chapter should also discuss any implications of your results for future work in the area (perhaps in terms of a future project).

Finally, you should also discuss your own progress in the project, including the management and control of the project, what you have learned, the challenges that you faced and how you reacted to them, as well as what you might do differently if you had your time again.

In this chapter we expect you to revisit the legal, social, ethical and professional issues concerning your project. Include: what more you have learned (if anything) during the project about these issues for your project, compared to what you wrote in your PDD; how these concerns should affect any future, follow-up work on or with your product and/or your research results, etc.). If you did some work towards investigating and/or solving any problems in these areas (e.g.: studying potential social effects if your product were used; effects of this study on your specifications; development of security or safety features; etc) you must report this work in your Method and Results chapters, but also recap here, in the Discussion/Conclusions chapter, what you achieved, as you do for any other parts of your work. Refer to which category of project your is, among the three identified in section 4.5.7, "Legal, Social, Ethical and Professional issues (LSEPI)", with the different allocations of responsibilities to you and others that they implied; and ensure that the reader sees how these responsibilities were or will be fulfilled. Even for projects that were limited to assessing or demonstrating some possible new idea, without any product being passed to others for use, in this chapter you need to demonstrate briefly that you are aware of the concerns that could arise in case your results or product were used by others. Any more in-depth discussion may bring higher marks depending on its quality.

8.2.8 Glossary

You may need to define specialist terms, unlikely to be known by your intended audience, in a project glossary appended at the end of the project report.

8.2.9 Reference List

You must include a references section with full details of all sources referenced in your project. As described in sections 9.2 and 9.5, we recommend a standard format for specifying all references and for citing them in the text. This is the **Harvard** style.

8.2.10 Appendices and other supporting material

Any material that would interrupt reading of the report should be presented in the Appendices or anyway submitted in appropriate submission areas (see next section). These must include all data and material used in producing the results of your project and all outputs produced that are not part of the main report. The following must be included:

- Appendix A: Project Definition Document;
- Appendix B: **Reuse Summary** (this must always be Appendix B; see Section 9.6, "Re-using software and how to document re-use");

- (mandatory for MSci students in Cybersecurity, good practice for all MSci students) Appendix C: **Minutes** of **Supervisory Meetings** (this must always be Appendix C);
- Complete records of each interview; questionnaires and complete questionnaire replies;
- Requirements;
- Routine design documentation;
- Source code, with instructions for building an executable version (as part of the “product package” submission; see next section);
- Test plans and test results; output listings; displays etc.;
- Any software installation guides and user guides produced (as part of the “product package” submission; see next section);
- An executable version of your software with installation instructions, or a URL if it is a web-based application that can be tested online (as part of the “product package” submission; see next section);
- All the reports produced for a client;
- Additional screen shots of your product running that are useful for understanding the product.

Rule: complete submission. All material you produced and data you used in your project **must be included** in the submission. Failure to do so will raise concerns about the validity of the results reported: you may be called before the Academic Misconduct Panel to prove that you really did the work described in your report. Therefore, include all raw and processed data in your submission (Appendices of the report or other submission areas – see next section), and be careful not to delete or lose your own copies of any material until after you graduate.

8.3 Video submission

Two videos should be submitted—one mandatory, one optional:

1. **Project demonstration video (15 min, mandatory)**: A summary of your work, with a brief introduction of its purpose, and including footage of your software product in use, structured as a software demonstration, with a voice-over describing the content. With this video, aim to help your markers by: (a) introducing the aims and results of your project, before they read the details in your dissertation, (b) showing how your product works in practice, as far as you can in the 15 minutes, and at least in its essential features, and (c) describing the implementation of a meaningful part of your code.
2. **Portfolio-ready demonstration video (5 min, optional):** A short demo video for inclusion in your online portfolio and/or as a demo on YouTube. Although not mandatory, this will be useful for promoting your work/skills.

Quality: Your video will help your marker understand what is most important about the work you have done. Observe the guidance and tips relayed in the dedicated lecture session on video production and submission. Pay attention to *quality*—ensure that video and audio content can be seen and heard clearly, without problems.

Confidentiality: Videos can be treated as confidential or non-confidential—upload your video(s) to the relevant submission point(s), accordingly (see section 8.5 How to submit confidential material)

Video filenames: Include your name in any video filename.

8.3.1 Project demonstration video (mandatory; 15 minutes)

You must produce and submit a 15-minute video that demonstrates what you have developed. This video is part of the evidence by which the markers assess your project: make sure it is informative and clear.

The structure of this video should be as follows:

1. *Introduction***:** First introduce the project idea, outlining the problem that your project addresses, your motivation for addressing it, and a brief overview of the solution. Assume no prior knowledge on the part of the viewer—your marker will not have seen your project before.

- *Demonstration:* Then present a narrated demo of what you have developed, showing the software running, and demonstrating key features. Aim for this to take up approximately a third of the total video time.

2. *Code walkthrough:* You must then provide a narrated code walkthrough, stepping through a section of your code that implements one of more key features, in a debugger, explaining how it works. This should also account for roughly a third of the total video time.

3. *Additional content*: Finally, show any other outputs or results, if required/relevant.

8.3.2 Portfolio-ready demonstration video (optional; 5 minutes)

We encourage you to produce and submit a shorter demo video that you would be happy to include as part of your online portfolio or as a demo of your work on YouTube. It should take the form of an advert for your work. You can use the same footage as the 15-minute version, but when deciding upon the content, you should consider what would work best to advertise/promote your work and skills. Although not assessed, we strongly encourage you to create and submit this video.

8.4 Penalties

Late submission

The Final Project Submission will be subject to a 10% deduction to the agreed mark if it is submitted up to 48 hours late. Submissions later than this will receive a mark of 0. This is in accordance with the university late submission policy. For example, a project submitted 42 hours late that receives 60% will be adjusted to 50%.

For further details of the late submission policy, visit https://www.citystgeorges.ac.uk/about/governance/policies/student-policies-and-regulations

Non-submission

If you fail to submit, you will receive a mark of 0 for your project.

Video non- submission: A penalty mark will be applied to all project submissions that lack the mandatory video submission. The penalty will be 10% of the awarded project report mark. For example, if the Final Project Submission receives a mark of 50%, the penalty applied will be 10/100 * 50 = 5%. As a result, the final mark awarded to the **Project** will be 50-5 = 45%.

Other omissions

Other omissions will be penalised according to the seriousness of the poor practice. Omission may also trigger misconduct investigation that may delay your degree: make sure your submission is complete.

Excess length

A mark penalty will be applied to all project reports that exceed the allowed word count. The penalty will be 1% of the awarded project report mark for each 500 words by which the report exceeds the word limit. For example, if a mark of 60% is awarded for the final project submission, but the report exceeds the word limit by 2300 words, then a penalty of 5% of the 60% will be applied: overall penalty = 5/100 * 60 = 3%. Therefore, the project mark will be 60-3% = 57%.

Poor Practice

Penalties will be applied to marks for all project reports or code where poor academic practice is detected. The scale of the penalty will depend on the scale of the poor practice. See also chapter 11 on Academic Misconduct.

8.5 How to submit confidential material

It is possible, especially in a client-based project, that your project will use information that is confidential. In other cases, you may find you have to collect information about individuals (for instance, respondents to a questionnaire; see also the section about Research Ethics). In these cases, you have a duty to protect the confidentiality of information. This section explains how to submit this material. The basic principles are:

- The university will honour confidentiality requirements about your project and the material you submit;
- But it is your responsibility to inform us about these requirements, in the ways we require.
- Confidential submission areas are only accessible to staff involved in the module. For especially sensitive information, a client may require more protection. In those special cases, you may submit the material in encrypted form. Use a public encryption scheme approved by your client and provide the decryption key to the Course Officers via a separate channel (typically an envelope, with indication to deliver to the IN3007/INM450 module team). Only the staff directly involved in marking your project will receive the key to read this material;
- Rules to follow for submission of confidential information:
- If you have to submit company-confidential material, inform your Course Officer before submission;

- For any confidential document that you submit, you must mark it as "confidential" on the cover page;
- For any confidential document that you submit online, you must submit it in a submission area labelled “confidential”;
- Submit confidential documents in their own files, and non-confidential ones in other files; likewise for anything that you submit in hard copy, submit confidential documents as stapled or bound items that are separate from any for non-confidential ones;
- For any confidential items to be submitted to the office in an envelope, contact the Module Team for instructions.

Typical cases are:

- A project involving confidential information of a client. You need to inform the Course Officer and Project Team about this, stating what the client's confidentiality requirements are. You must then use the confidential submission area(s). and label any relevant offline submission as “confidential”.
- A project for which some appendices contain personal information, e.g., names of people participating in a study. The best option is to collect all personal information into a single file. This file must then be submitted to Moodle’s confidential submission area. When that is not possible, print it out to submit ***only offline.***
- Your PDDs often contain addresses etc for specific people. Please either ***remove*** these if you wish to keep the PDD in the same file as your report, or you can omit the PDD (appendix A: just indicate in the list of appendices "Appendix A: in Moodle PDD submission area"), since we have it on file.

All material submitted to Moodle will be processed by the Turnitin plagiarism detection tool; but anything in the “confidential” submission areas will ***not*** be deposited in Turnitin's repository that is accessible to other Turnitin clients.

It is most convenient for you if any confidential material is in separate appendices (in one or more, confidential files), rather than in the main body of the report: this will make it easier for the School to handle the project report, and you will be able to show the main body of your report, for instance to a possible employer, without violating the confidentiality agreements you made.

If you have any doubts or questions, the module team can advise about the general rules.

9 Good Professional Practice

A good project requires not just hard work, but also adherence to professional standards. There are basic rules of sensible and ethical behaviour among professionals: one needs to learn them, and violating these rules is not just a blunder, but a major offence, in the University as well as in professional life. First, you must apply the general rules that apply to everyday life: you need to be truthful in what you say and write, deal honestly with clients and anyone else involved in your project, etc. In this section, we call your attention to aspects of good conduct that are very important in professional life and that are particularly relevant to the conduct of your project.

Make sure that you read, understand and apply the rules and advice explained in this section. If you violate these rules, you risk behaving dishonourably, and being punished heavily for it.

While you should follow all the rules in this handbook, in this chapter we have underlined some text about rules that if violated can **cost you your degree.** It is *your* responsibility to read, understand and follow these rules; if you have any doubt about what they require in your specific circumstances, you must ask for clarifications.

9.1 Use other people's results with proper acknowledgment

All good professionals use the product of other people's work. They read reference books and research articles, they re-use software and take advice from experts. Without this, we would be forever reinventing the wheel. So, in your project you need to read about the methods and technologies you need to use, going back to modules studied in the university as well as reading more specialised literature concerning your project topic; seek advice from your consultant, the project team, colleagues at a client company, etc. Note that one ***must always acknowledge*** these contributions. Not doing so is discourteous and is a form of cheating, by letting others believe that you discovered facts, invented designs, wrote material that you have not done. Cheating in one's project carries heavy penalties: see section 10 on Academic Misconduct.

9.2 Reproducing material from your sources: when and how

You often need to report what someone else said that is pertinent to your topic. Note:

- Usually, it is best to report in your own words the concepts that are pertinent. In these cases, always make it clear *whose* ideas and results these are. There is a prescribed way of doing so, by an in-text citation pointing to an entry in your reference section. These are explained below, under section 9.5 "References and referencing style".
- Sometimes one wishes to quote *exactly* what someone else said or wrote. In this case, it is essential to identify clearly **all** the material that is copied word for word. This is done, conventionally, by enclosing it between double quotes. Occasionally you need to reproduce whole paragraphs, in which case you can also typeset them differently from the rest of the text, e.g. indent them with respect to the rest of the text, and must write explicitly that the indented text is taken from your source, giving a reference to the source. Examples are below in section 9.5 "References and referencing style". Omitting the double quotes and the other indications, even by

mistake, is a serious violation of professional behaviour and thus also of the rules for the project. In some cases it is also a violation of copyright law.

- Never copy material and then change words here and there, so that it is more difficult to identify the material as someone else's. This is forbidden and is easily detected by the markers. If you really need the other person's own text rather than your own summary, copy it verbatim between double quotes.
- In general, using a large amount of text from other authors, even if properly acknowledged, is not a good idea. You have a limited number of words to report *your* work. Don't waste words on copies of other people's text.

9.3 Use of copyrighted material

You must not use material (e.g. graphics, background music) taken from other sources that you do not have permission from the copyright holder to store or use. If you do this in a web site or in software or other material that you make available to others, you risk being prosecuted for breach of copyright. If you wish to re-use any such material, read carefully whether this use is allowed by the copyright owners, and under what conditions, and respect these conditions. Quoting short passages of a work in a dissertation is generally allowed by U.K. copyright law under the "fair dealing" regime. You can check details at https://libguides.citystgeorges.ac.uk/friendly.php?s=copyright/fairdealing - s-lg-box-11429697 and the sources cited there.

9.4 Justifying one's decisions, statements and claims

Your project report must document the decisions you took and ideas you developed during your project. However, *it is not enough to say what you thought or felt*. You need to document on which basis you reached a conclusion or took a decision. So, throughout your report, you must support any assertion that you make, either by clear reference to your own work documented in the report (e.g. measurements taken or interview results), or by a reference to indicate its source in the literature. You often also need to explain why this support information (evidence) leads to your conclusion; e.g., why the measurement data imply the judgement that you reached.

9.5 References and referencing style

You must list all references at the end of your report and you must cite them clearly in the other sections. We ***strongly recommend*** a standard format for specifying all references and for citing them in the text. This is the **Harvard** style. This style is effective, reader-friendly and easy to use. You can read more about the Harvard style at:

- https://www.citethemrightonline.com/category-list?docid=CTRHarvard (log in using your City St George's credentials)
- https://guides.lib.monash.edu/citing-referencing/harvard2020-intro

The following information is taken from the second of the above sources:

> *"Referencing is a standardized method of acknowledging sources of information and ideas that you have used in your assignment in a way that uniquely identifies their source. Direct quotations, facts and figures, as well as ideas and theories, from both published and unpublished works must be referenced. Referencing is*

> *necessary to avoid plagiarism, to verify quotations, and to enable readers to follow-up and read more fully the cited author's arguments.*
>
> *When citing references within the text of an assignment, use only the name of the author, followed by the year of publication.* ***In general, page numbers should be included in all in-text citations, as many schools insist on this practice****:*
>
> > *The theory was first propounded in 1993 (Hamilton 1994, p. 58)*
>
> OR
>
> > *The theory was first propounded by Hamilton (1994, p. 58)."*

Summary and reminder: all text that you reproduce from your sources must be identified as such, typically between double quotes, so that your reader has no doubt about which text is originally yours and which is not. Any time you use such text, or use ideas from other people, you must also insert a proper citation and a corresponding reference entry in your bibliography (or "References") section**. *Violating these rules, even by accident, amounts to plagiarism and carries serious penalties.***

Reproducing figures and other non-text material: just as copied text, any other material that you reproduce requires an explicit indication that it is reused, and acknowledgment of the source. E.g., if you copy a figure from one of your sources, you must indicate this in your caption for the figure, e.g. *"reproduced from (Cunningham 2006, p. 32)"*. If you took the figure and then changed it, indicate, e.g., *"adapted from (Cunningham 2006, p. 32)"*.

Also remember, for any material (text or anything else) that you copy from sources and reproduce or adapt in your submission, to check that you have a right to do so under copyright law, either because you are explicitly allowed by the owner (and owners may demand that you cite which kind of license they granted, or set other conditions: you must obey these conditions) or because it is "fair dealing" under U.K. copyright law (see section 9.3).

9.6 Re-using software and how to document re-use

Re-using software, that is, including existing software in new products, is an important part of software engineering. To do this ethically, you ***must***

- Ensure that you are authorised to re-use it (by checking its licence), and obey all conditions under which you are granted this permission[10]
- Clearly indicate what you re-used and where you used it

So, you **must**:

- In the Method chapter, indicate what kinds of software you re-used and how re-use affected your development process.

[10] Apart from specific conditions in the license, software is subject to copyright. Useful guidance is at https://libguides.citystgeorges.ac.uk/friendly.php?s=copyright/software

- In the Results chapter, list the software you reused and explain what parts of your own software they form.
- In Appendix B, list the software libraries used, and flag which parts of the code are your own and which parts are re-used code and/or code that was automatically generated. Identify tutorials and other online sources where you obtained (ideas for) parts of your code. Specifically, include a list of all the source code files, indicating exactly which ones are reused (and the source from which you copied them: give the URL or other way for finding the original code), which ones are completely written by you, and which ones are reused with changes. For these latter files, where you made changes, you must provide, in addition to your file, a file or printout in which all your changes are clearly visible (that is, one can identify the previous text and your text that replaces it). Where there are complete directories automatically generated or reused by you without changes, you do not need to list the individual files; however, you must list the directories. The same is true for directories that only contain code created by you.
- In your conclusions, recall and discuss the role of re-used software in your process and in your product.

In describing the project's achievements in Chapter 2, you should summarise the size of the product produced (often by more than one measure, e.g., lines of code, number of classes, bytes in executable software, number of web pages, number of database tables, ...) and how it is divided between subsystems. You ***must*** also indicate how much of this is produced by you and how much is re-used.

9.7 Rules on use of generative AI

Many students use generative AI (e.g. ChatGPT, CoPilot, Bard, DALL-E) for various purposes[11].

For IN3007/INM450, using these tools is allowed as per the School of Science and Technology policy; it may or may not be useful for you. If you use them,

it is mandatory:

- for both the PDD and the final submission (and the INM450 peer review), to add an Appendix in which you specify how and to what extent you have used generative AI tools. This Appendix does not count towards word limits. It must list details of precise parts of the work where each AI tool was used and how it was used (including what prompts/inputs were given to the tool, what came back, and what you did with that output)
- not to use AI tools for plagiarism or cheating. Always properly credit the original sources when using information generated or assisted by AI. Follow the University's guidance:

[11] For example, help with grammar (sometimes useful), getting ideas for new code (sometimes useful), literature summaries (dangerous), or even instead of doing work themselves (academic misconduct).

https://studentcommunications.newsweaver.com/1a439l3pi5/frcc1atnwhw3lqfua0ym7o

- In particular:
    - if you insert AI-generated images, text or code, identify it clearly (between quotation marks for text, in comments for code, in the caption for images), and give a reference as specified at https://studenthub.citystgeorges.ac.uk/academic-resources/learning-and-study-support/online-learning-tools/using-generative-artificial-intelligence-ai-for-learning .
    - do not waste your time paraphrasing AI-generated material if the material is good enough as it is.
    - if AI paraphrased original ideas from other people, you must still give credit to the original authors with a citation.
    - if you use an AI tool to check grammar or language, you do not need quotation marks, but still specify in the appendix on what text you applied it and how.

In any case **YOU**, not the AI, are responsible for submitting material that is both correct and ethical.

It is highly recommended:

- Before you use any generative AI tool, take the time to understand how it works, its limitations (especially its inability to discriminate between truth and falsehood) and potential biases.
- Watch the videos posted about generative AI: https://mediaspace.city.ac.uk/channel/Generative+AI+-+Video+Discussions+and+ReYd+Material/316306582
- Verify the accuracy of AI-generated results. Do NOT unthinkingly rely on AI outputs (e.g. code, text, alleged data or summaries of facts). Check whether they make logical sense, cross-check all information using other sources, examine, correct as necessary, and test any code.

Use of generative AI tools will be considered **Academic Misconduct (AM)**, in cases like (but not limited to):

- Plagiarism – where material used in the assessment is the direct output from an AI tool and has not been acknowledged properly. Plagiarised material that has been amended using further AI tools (e.g. paraphrasing) remains plagiarised.
- Incomplete, vague and/or misleading reporting of how you used AI.
- Falsification or fabrication – where your submission refers to sources or data that do not exist. Generative AI often produces such material.
- Contract cheating – where generative AI tools have been used to produce the whole or large parts of your assessment.

By approaching generative AI tools with a **responsible, cautious and ethical mindset**, you can use the benefits of these technologies while minimising potential risks and negative consequences. Also keep in mind that:

1. generative AI tools may produce not only wrong statements but an excess of vague, uninformative text; while you have word limits, within which you must fit as much **useful** information as possible.
2. when assessing your PDD, source code, results, and reporting, academics mostly care about the content rather than the syntax mistakes (and will react with suspicion to unnecessarily sophisticated use of language); this work is not an English literature assessment. Your text needs to be clear, it does not need to be beautiful.

10 Project Outcomes

There are a few possible outcomes at the end of the project process. In most cases, a student will fulfil all of the requirements of the project and pass the module to obtain 45 credits.

Failure to attain the pass mark will result in one of two different recommendations being made by the Assessment Board:

Minor changes: The student will be given 4 weeks (PP students will get 6 weeks) to bring the project up to the required standard, with minimal supervision. The consultant will provide the student with a list of changes to be made; the student will continue to work with the same consultant on the same topic. The student will receive a new mark for the Final Project Submission. The module mark will be capped at the module pass mark (40% for BSc projects; 50% for MSci projects).

New project: In some circumstances, in particular if there has been academic misconduct in the project, the Assessment Board will require the student to undertake a new project. The student will resit the module, usually with a new consultant, following the project calendar for the next academic year. They will be required to submit all deliverables for the new project, from the Project Definition Document to the Project Report. New marks will be awarded for the new submission. The module mark will be capped at the module pass mark (40% for BSc projects; 50% for MSci projects).

If you have to do a new project, make sure you follow the rules that apply that year, including those concerning how to choose a project, the deliverables to produce and the deadlines.

If you failed due to poor practice or misconduct, we will examine your resubmission carefully for such behaviour; and if you repeat any offence, the penalty will be heavier than the previous time. If your offence was that you used other people's text without acknowledgment, do ***more*** than just removing the passages that we found the first time around. Ask yourself how you wrote your report, and make sure that what you submit as your resubmission is ***all*** your own text, or properly acknowledged quotes.

If your **second** attempt (resubmission or new project) fails, you will **fail** the IN3007/INM450 module.

10.1 The marking process

Your project will be marked by two independent individuals and marks are moderated by the module team. The module team and the Programmes Office have close oversight of the marking process. In the event of discrepancies of judgement between individuals in the marking process the module team gets involved to have any issues identified and resolved, at times involving further academics in the discussion.

All marks and supporting comments are shared with the External Examiner to further support our quality control. The full process is overseen and discussed by our Assessment Board, who has the authority to decide the final module marks.

10.2 Release of marks and feedback to you

Marks can only be released to you after they have been approved by the Assessment Board. Hence, you will receive them with your letter confirming your results.

Once these results have been released you will receive your individual feedback by means of an email automatically sent to you via the marking tool.

11 Academic Misconduct

Any kind of academic misconduct, including plagiarism and the fabrication of results, is unacceptable. City St George's, University of London takes such behaviour very seriously and will punish any student found guilty of academic misconduct[12]. The punishments vary (e.g., a student may receive a mark of zero for the affected submission, may be given an Ordinary degree instead of an Honours degree, or may be ***expelled from the University without a degree)***. In all cases, a record of academic misconduct remains on the student's file. Every year we catch students in this kind of unprofessional or clearly fraudulent behaviour. Please do not be one of these students.

Academic misconduct includes plagiarism such as:

- Consistent failure to correctly cite the work of others;
- Presenting the results or work of others as your own, either deliberately or unintentionally;
- Unacknowledged reuse of work from another source in the completion of your project
- Illicit use of AI aids. Note that using AI without reporting it is misconduct, even if the use is one that, if reported, would be considered proper conduct.

It also includes fabrication of results and other outputs, for example:

- Presenting fictitious data and claiming that it was gathered from questionnaires, interviews, evaluations, etc;
- Reporting a software solution as fully functional when in reality, the functionality of the software is defective or incomplete.

Consultants, markers and project team members use a variety of tools to check for academic misconduct throughout the project and especially when marking the final reports and checking other outputs. All cases of suspected academic misconduct will be referred to the project team. Minor cases of poor practice will be penalised by the project team; all other cases will be referred to the Academic Misconduct Panel (AMP).

11.1 Results from AMP

The AMP will reach one of the following decisions for each case brought before it.

No Academic Misconduct Has Taken Place

The AMP may decide that no misconduct has taken place, in which case the student is informed in writing, and the allegation will not be recorded on the student's file.

[12] We follow the regulations of City St Goerge's, University of London. You can find further information on the policies and guidance related to Academic Misconduct cases on the University's web pages under the Quality Manual.

Poor Practice Has Taken Place

The AMP may decide that no misconduct has taken place but there is evidence of poor practice. In this case, a penalty will be applied by the project team.

Typical examples of poor practice include minor cases of poor referencing, inadequate reporting of the work undertaken, accidental minor misrepresentation of the work undertaken ("method") or of its outcomes ("results").

The penalty for incorrect referencing, when it does not amount to misconduct, is typically a deduction of 5% from the mark for **each separate instance** of the incorrect referencing identified in the report. This deduction might well impact on the final class of degree that is awarded, or cause failure in the project. Furthermore, the Project Team may require the student to fix the referencing and citations in the submitted document. If this occurs, the delay might well cause a delay in the awarding of the final degree, with possible implications for your graduate employment.

Penalties for failing to report all of the work and results, or misrepresentation of the work and results reported, can be more severe depending on the nature of the offence, and can lead to a mark of 0 being awarded for the relevant aspects of the submission.

The student will be informed in writing. For poor practice that does not amount to misconduct, the allegation of misconduct will not be recorded on file.

Academic Misconduct Has Taken Place

The AMP may decide that misconduct has taken place. The usual penalty recommended to the Assessment Board will be a mark of zero for the submitted work. If this is a second finding of academic misconduct for the same student, the Panel may refer the case to the Academic Registrar as specified in the Assessment Regulations. The student will be informed in writing that the AMP has decided that misconduct has taken place, with a note of the penalty to be recommended to the Assessment Board. This is a recommendation only and final decisions will be made by the Assessment Board. There is no right of appeal against this judgment by the AMP.

Passing the project module is a pre-requisite for passing the final year of your degree course. **Misconduct may bring late graduation, lower marks, or even no degree**.

12 MSci Projects

MSci (INM450) projects follow the same process and timelines as BSc (IN3007) projects. MSci students are required to submit a PDD, discuss various drafts with their supervisors and make a Final Project Submission.

However, as the MSci is a Master's level program, there are some important differences between MSci and BSc projects. MSci projects must demonstrate a more *advanced and critical approach* to the work than is required for success at BSc level. Specifically, these differences are:

- The module pass mark is 50%.
- MSci projects have an academic as *supervisor* instead of consultant . The supervisor may interact with the student in more ways than the set of prescheduled meetings. Supervisor and student must agree early on how they will communicate during the project. Students should create minutes of the meetings with their supervisors and check these minutes with their supervisors to confirm that they have properly understood the discussions that took place. For students in the Cybersecurity course, these meeting minutes are **mandatory** and **must** be submitted as the Appendix C of the final report.
- MSci projects must demonstrate a better-informed choice of objectives and/or research questions than UG projects, established in discussions with the project supervisor when choosing your project topic, and a well-reasoned approach to any decisions needed during the project.
- MSci project reports may be between 12,000 and 15,000 words in length. This longer report allows MSci students to:
    - Outline in the Introduction chapter a more in-depth motivation for the work and a more detailed set of objectives as needed. This should include a clear articulation of how the project contributes to solving the problem addressed or to the knowledge within the area of study;
    - Conduct a more extensive literature review (the Context chapter should be at most 2,000 words);
    - Report in the Results chapter on the analysis of alternative options justifying decisions taken during the project, and a more in-depth evaluation of the work;
    - Provide in the Conclusions and Discussion chapter a more complete and critical reflection on their work, covering the extent to which objectives were met and/or research questions answered, the highlights and limitations of the project and their reasons, the scope, impact and generalizability of the work and an outline of any possible short- and long-term future work.
- MSci students must also perform a peer review activity: early in the project schedule (see timetable on Moodle) groups of MSci students will review one another's project plans, hold a meeting (called by an academic as facilitator) and give constructive feedback in writing.

Appendix A: Client Information Sheet (External Client Projects)

If you have a client for your project, you must submit a client information sheet with your Project Definition Document.

The Department will contact your client and ask them to confirm that the information is correct and that they have agreed to be your client in this project. This helps us to be sure that you have a committed client and protects you by having the company confirm their agreement with you in writing to the University.

The information required is:

- The complete name, address and telephone numbers of the client organisation. We want landline numbers for the company's official premises.
- The name, address, telephone number and e-mail address of a contact person at the client organisation. We require company email addresses (not Gmail, for instance),
- Any relationship that you have with the client organisation or the contact person (e.g. if the owner or your contact person is a relative, or you are employed by the company).
- The client problem to be addressed. One or two lines are usually enough.
- The resources that the client will make available to you (e.g. interviews with staff; supervision by a member of staff; software to re-use; use of company premises or computers. Be specific. You need to have a clear agreement with your client on these points.)
- The outputs required by the client.
- How the client will review these outputs.
- The intended use of the outputs.
- Your agreement with the client about ownership and responsibility for any data collected as part of the project, concerning Data Protection regulations and Intellectual Property Rights, if applicable.
- All other agreements concerning responsibilities for Legal, Social, Ethical, Professional issues
- Any other obligations agreed between you and the client, including:
    - Any special agreements made between you and the client to protect the client's confidentiality or intellectual property needs during the University's assessment of your project. You must make sure these agreement have been **approved by the project team in writing** before you submit your PDD; and state here what the agreements are and the date of the Email with the project team's approval.
    - Confirmation that the client will provide a letter at the end of the project, upon request by the University, to confirm what input they provided and to

indicate their assessment of the project outputs. In other words, you must have informed the client of this requirement and they must have accepted it.

All these details are required because starting a project without clear arrangements between you and the client **would be a serious risk to your project**. If details of your interactions with the client depend on how the project proceeds (e.g., you may not know yet whether you will need access to some data that belong to the client), ensure that you have agreed the principles that will govern these details (e.g., that the client will grant access to any data required to complete the project and have it assessed by the University; and what confidentiality guarantees you will give about these data).

If any of this information changes during the year, you need to inform your project supervisor or consultant and your course officer.