

CO7201 MSc Individual Project

Study Guide 2016/17

Convenor: Dr Richard Craggs (CA7201)/ Dr Nir Piterman (CB7201,CC7201)
Department of Computer Science
University of Leicester

November 28, 2016

Foreword from the MSc Director

The project is, for many students, the highlight of their MSc Course. So far much has been taught through courses with associated exams; in that sense it is probably been similar to your undergraduate studies (though hopefully a little bit harder and more challenging!). At the stage of reading this, you are embarking on the challenge of a large-scale full-time project, possibly for the first time in your career.

Projects start with the selection of an appropriate topic, this requires thinking about what you would enjoy doing and discussion with members of staff. There are no right or wrong projects; neither are there harder or easier projects. It is important that you find something that you will enjoy doing; of course this ties in with the skills and techniques that you learnt during the taught part of your MSc programme. We endeavour to publish project topics very early, so that you have plenty of time to talk to staff members and think about this before having to commit to a choice.

The bulk of the project occupies 3 months of full-time work cumulating in a written final report, and possibly some software or research results. We know that 3 months seem a long time when you set out to start on a project, especially with no predefined idea as to how you will spend this time. Rest assured that this will be quite a busy time for you; we can only encourage you not to leave things for later, thinking that the deadline is quite far away - you will find that time flies. You will see that we encourage you to be careful in planning your project and we have given you some intermediate milestones to achieve.

We hope that you will find this study guide a useful resource at all stages in the project - it is meant to be more than just rules and regulations: it is a guide to help you get the most out of the project, covering aspects ranging from what a project is all the way through to what individual pieces of work should be contained. Study it carefully at the onset of the project, and come back to it whenever you need some advice.

We hope that this last building block towards your MSc Degree will be an enjoyable experience.

Contents

1	Module's Aims, Learning Outcomes and Useful Resources	3
2	Project Phases	4
2.1	Characterisation of Projects	5
2.2	Project Workload	7
3	Project Organization	7
3.1	Administration and Communication	7
3.2	The Role of the Student	7
3.3	The Role of the Supervisor and Second Marker	8
3.4	Progression checking and Submissions	8
4	Important Dates and Deliverables	9
4.1	Project Description	10
4.2	Supervisor Meetings	10
4.3	Preliminary Report	11
4.4	Interim Report	11
4.5	Interview	11
4.6	Final-report Template	12
4.7	Final Report	12
4.8	Viva	13
5	Format and Structure of Documents	14
6	Plagiarism	15
7	Feedback	16
8	Late Submission of Deliverables	16
9	Mitigating Circumstances	17
10	Notification of Illness	17

1 Module's Aims, Learning Outcomes and Useful Resources

The aim of this module is to demonstrate a student's ability to undertake a substantial investigation of a technical problem and its domain, by evaluating tools and methods, and by developing a professional information technology project; also, it intends to give students the opportunity to

- show individual creativity and originality;
- manage their own time and their progress towards completion of the project;
- analyse information from a critical point of view;
- apply, where appropriate, and go beyond, where necessary, the knowledge and skills taught throughout the course;
- investigate/solve new and/or intellectually demanding problems (from specification through implementation and critical evaluation of results);
- conduct and sustain a complex argument in a coherent and lucid fashion.

Among other transferable skills, students will improve the following skills: solving practical and abstract problems, communication, writing, editing, searching/gathering/evaluating information, developing evaluation strategies and managing time.

The main learning outcomes of this module are to initiate, plan, manage and deliver a substantial information technology project. Upon successful completion of the project, students should be able to:

- select suitable methods and tools for analysing a substantial problem and for developing a computer-based solution for it, within known constraints;
- access, retrieve and organize information relevant to the problem under study by making use of resources such as the internet and textbooks, but also of scholarly articles published in journals and conferences;
- prepare a project plan, follow it and conduct regular reviews of the plan and update it;
- present a properly referenced, well-structured final report, in a format suitable for professional dissemination;
- communicate effectively in a presentation environment;
- perform a critical reflection of the achievements in the project after its completion.

The following book describes the main elements that are involved in the development of an information technology project. It provides resources to develop the aforementioned skills:

Christian W. Dawson. Projects in Computing and Information Systems: a Student's Guide. Addison-Wesley. 2nd edition. 2009.

The resources referenced through this study guide and relevant for the project are collected below.

1. Calendar: <https://campus.cs.le.ac.uk/teaching/resources/C07201/#calendar>
2. SVN Local Pages: <https://campus.cs.le.ac.uk/demonstrators/info/VCS/>
3. MSc Course handbook: <http://www.cs.le.ac.uk/admissions/masters/MScCourseHandbook.pdf>
4. Information about plagiarism: <https://campus.cs.le.ac.uk/ForStudents/plagiarism/>
5. Student Learning Centre: <http://www2.le.ac.uk/offices/careers/ld>

6. Enhancing writing: <http://www2.le.ac.uk/offices/careers/ld/resources/writing/index>
7. Presentation skills: <http://www2.le.ac.uk/offices/careers/ld/resources/presentation/index>
8. Logos of the University of Leicester: <http://www.le.ac.uk/corporateid/new/logos/>
Important: The University of Leicester does not allow such logos to be modified (cut, resized, reshaped, change colour etc.). Therefore, if you want to include logos in any of your documents (presentations or deliverables), do not modify them.

2 Project Phases

To fulfill the requirements of the MSc degree, an individual project must be undertaken full-time by the students who passed the taught component of the MSc programme. Please refer to the MSc handbook for further details of progression rules.

The project is taken full-time for a period of 12 weeks (Summer) or 13 weeks (Autumn and Spring). Students who finish the taught part of the MSc programme are expected to undertake her/his project. Students will be taking the project in one of the following three periods according to their own individual circumstances:

- from October to January – for students who are returning from an industrial placement too late for a June start.
- from June to September – for students who started the previous year in October or coming back from an industrial placement.
- from February to May – for students who started the previous year in January or coming back from an industrial placement.

Students who have to resit several of their taught modules may be told to take the project in a period that matches the schedule of their resits.

Projects are carried out under the supervision of a member of the academic staff.

The project consists of the following phases:

- **Announcement of topics:** A list of possible topics will be made available before the start of the project. For each topic, the information will include a provisional title, a short description, the name of the supervisor and a list of MSc programmes for which the topic is suitable. The student has to take a topic in the field of her/his chosen MSc course. Before selecting a topic from the list, a student should discuss its details with the corresponding supervisor.

Students can propose their own topics and seek a suitable member of staff to supervise them, provided that the MSc project convenor approves the topic. The project must be clearly at post-graduate level (cf. Section 2.1). The proposal should take a similar format to the topics proposed by staff.

- **Topic registration:** As mentioned above, there are two types of topic registration:
 - Staff topics: Students may choose up to four projects through the **project registration system**. Chosen projects should be ordered by preference, with the student's first choice appearing first in the list of selected projects. Only two topics with the same supervisor can be chosen and at least three different supervisors. We will make every effort to ensure that you are allocated one of your choices. However, students do get allocated their fourth choice. Choose wisely!
 - Student proposal: A student can propose a project to a supervisor. After an agreement with a supervisor and the module convenor, the student has to give a **signed registration form** to his/her supervisor. In this case, students need to contact a supervisor **early before the registration deadline** in order to start the discussion of the project. Students should reach

an agreement and submit the signed proposal by the corresponding registration deadline. In this type of registration students shall not select projects through the registration system.

Choosing a topic implies committing to it so that a choice cannot be changed after the deadline. Bear in mind that a topic can be allocated to a different supervisor from the one stated in the proposal.

- **Topic allocation:** Topics are allocated to students by taking into account students' preferences and supervisors' availability. Students who do not discuss topics with supervisors will have lower priority in the allocation process of the corresponding topics. Being first or last to talk with a supervisor does not affect your chances of being allocated that supervisor. Supervisors' consent to supervise you does not mean that they actually will.
- **Project description:** A project description form has to be submitted once topics have been allocated.
- **Official start of the project (week 1):** The student then arranges for regular meetings (usually held on a weekly basis) with her/his supervisor. Weekly meetings take place until the end of the project.
- **Preliminary Report (week 2):** The student submits a preliminary report.
- **Interim report (week 6):** Students submit an interim report.
- **Interview (week 7):** An interview with the second marker takes place during week 7 and has to be arranged by the student. The student should explain what has been studied and developed until that point. During this week, a meeting with the supervisor usually does not take place.
- **Final Report Template (week 8/9 (Summer/Autumn and Spring)):** Students submit a template of the final report. The template must include a detailed literature survey relevant to the project and a detailed outline of the final report.
- **Completion (week 12/13):** Final report and all project materials are submitted.
- **Viva (week 13/14):** Students give an oral presentation, show their software and answer questions.

Each phase is described and assessed as detailed in Section 4. Deadlines are reported on the Google calendar (1).

2.1 Characterisation of Projects

This section specifies the requirements that an MSc project must have. Although a project does not need to yield substantially original results (e.g., design of a new algorithm, or state and prove new theorems), it is expected to contain some element of original work (e.g., study a known algorithm under more restrictive hypothesis, specify a new validation technique, perform a comparison among techniques or tools). The project may include the development of a substantial computer-based system, but may also be purely theoretical.

A project must include practical and/or theoretical results obtained by the student and cannot be mainly a review of literature. For example, converting an application available in C++ to Java or replicating an existing network simulation experiment are unacceptable topics for a project.

Projects have been categorised into three main types, **software development projects**, **technical projects** and **theoretical projects**. Please keep in mind that these classes are not disjoint. Namely, it is possible that a project lies in the intersection of two types or all of them. The following descriptors are meant to guide the characterisation of the projects.

I. Software development projects. These are projects where the main contribution is developing a significant software system. Typically, the project requires the student to use her/his creativity

in modelling/developing a non-trivial software system. Although a software product is the main result of projects of this type, the development process (software development tools/techniques, appropriate methodologies/architectures) is relevant.

Typical challenges are that: technologies used are novel or just emerging (just being new to the student is not enough!); novel development methodologies are employed; or technologies are used in an innovative way.

II. Technical projects. These are projects with an experimental flavour where the main contribution is the use/extension of existing techniques or tools for studying/analysing a substantial problem. These projects may also involve software development. However, their main focus is on the use of existing tools/techniques. Typical challenges of these projects are the complexity of the problem to study. For instance, studying the correctness of the authentication protocol Kerberos by formalising it and its properties in FDR (a CSP model checker). This project is challenging because it requires understanding how to model protocols with process algebras for verification purposes.

III. Theoretical projects. Their main contribution is of theoretical nature, e.g., extending a formal modelling language, enhancing an existing algorithm and proving its time and space complexity, etc. Such projects need not include any software development. However, they require an accurate and abstract understanding on how software is designed so that strong mathematical skills are recommended. Challenges in projects of this type usually involve the interpretation of a software-related problem as a mathematical, abstract problem and the application of suitable mathematical theories and tools to solve them. For example, developing a more efficient network routing algorithm or a modelling language that allows to model new classes of software using an algebraic approach.

The following examples give a description for each class of project:

I. *Course Information Manager*

The department runs a number of modules, each convened by a member of staff. Throughout the year, certain tasks have to be conducted for managing courses. Such tasks include the generation of study guides, module forms and websites, as well as providing material on websites as the course progresses. Each of these tasks might be small, but they do take a considerable amount of time.

The project requires the student to develop a system that automates these tasks so that staff can manage them more efficiently. This project will make use of service oriented computing to obtain a high degree of loose coupling and extensibility.

II. *Model-based Testing of Service Oriented Middleware*

Usually, services-oriented middlewares are informally specified through textual descriptions. This leads to incomplete/ambiguous semantics that often lead to inconsistent implementations.

In this project UML and graph transformation techniques are combined for modelling middleware platforms starting from their informal description. Moreover, the project requires the student to develop concepts for testing implementations of these platforms against their executable specification. As a typical use case, a core part of the BPEL4WS execution engine should be modelled with graph transformations. Such a model will then be used to test an existing Java implementation of this standard.

III. *Task Allocation Problem*

A task is an activity that an agent can perform in a finite amount of time. Given a list of tasks and a list of agents such that each agent is able to undertake each task in a finite amount of time (different agents perform the same task within different times), the problem is to find an optimal allocation of tasks to agents so that the completion of all the tasks is performed in the minimum amount of time.

In this project an algorithmic solution for a task allocation problem will be developed. The specific allocation problem will be identified with the student. There are certainly various ways of tackling these problems: classic exhaustive search based on heuristics, or probabilistic approach.

2.2 Project Workload

Working on the project is like doing a challenging full-time job for 3 months. 450 hours of self study are allocated to this module since it carries 60 credits. This corresponds to the amount of effort that has to be put in four taught modules, including private study and preparation for exams. Students should put the right amount of effort in the project from the beginning to ensure they will be able to achieve its objectives. If a student sees that (s)he is failing to spend the amount of hours needed, the student should take an action to bring her/his project back on track.

Students must reside in Leicester or within easy commuting distance of the city for the development of the project. Students are required to attend lectures and practicals as specified in the module timetable (Google calendar) or as communicated by the MSc project convenor.

In case of any mitigating circumstances, the normal departmental procedures should be followed and the appropriate form must be submitted. Further information about illness and mitigating circumstances is in Sections 9 and 10.

Requests for absences of more than one week must be explicitly approved by the MSc Director, and will only be granted if the Department is in agreement with the proposal, and if the student concerned takes full responsibility for the completion of outstanding academic work. This procedure also applies if the absence is required for religious reasons, but as students are required to notify the Registry at the beginning of each academic year if there are likely to be religious reasons for any absence during that year, academic departments and administrative offices are expected to utilise this information pro-actively, so that any specific religious needs can be anticipated and, where practicable, met.

3 Project Organization

This section describes how the project is organized/run and how students, project supervisors and the module convenor communicate. Progress monitoring mechanisms are described as well.

3.1 Administration and Communication

This study guide is the main reference for the project. Students will also receive information about the project module in other ways:

- The convenor may send emails on logistics (e.g., where/when lectures take place) and deadlines.
- Deadlines and dates are reported on the Google calendar (1)
- Supervisors give guidance and solve specific questions about the project during meetings.

3.2 The Role of the Student

The project is required to be entirely the students' own work, and it is their responsibility to ensure its quality.

Students are responsible for:

- Reading and understanding the study guide.
- Checking University email on a regular basis.
- Engaging with the project in a pro-active and reflective way, planning tasks on a regular basis and discussing ideas, queries and plans with the supervisor.
- Attending meetings on a regular basis.
- Meeting deadlines (consult the Google calendar and take note of the deadlines).

- Keeping back-up copies of all work/code/software related to the project on a regular basis by using the SVN repository as explained in Section 3.4.

Important: Since the SVN repository has to be used on a regular basis, a loss of data due to your laptop being lost or stolen, a hard disk failure or similar circumstances, cannot be used as a justification for either a deadline extension or a mitigating circumstances form.

Students should also actively request feedback from their supervisor on their general progress and should not feel afraid to ask whether their work on the project seems suitable for achieving the grade they are aiming for. However, supervisors are not supposed to indicate specific tasks to achieve this mark.

3.3 The Role of the Supervisor and Second Marker

The project is subject to both continuous monitoring by the project supervisor and occasional monitoring by the second marker.

Students will meet their supervisor on a regular basis for up to half an hour to discuss the progress of the project, starting in week 1. This meeting usually takes place on a weekly basis. The primary function of supervisors is to advise students on:

- general aims and objectives of their project,
- methods to be used throughout the project,
- preparation of deliverables,
- preparation for the Viva.

Supervisors **are not expected** to give students instructions on how to solve problems to be undertaken in the project or to do any of their work. If the supervisor is away for more than two weeks, (s)he will make alternative arrangements for achieving a satisfactory supervision. For example, (s)he may brief a substitute supervisor (often the second marker) on the progress of the project. Should this happen, the student would be notified and (s)he would have to arrange a weekly meeting with the substitute supervisor.

The second marker of the project is announced when projects are allocated. (S)he is actively involved in assisting the supervisor in assessment, feedback and advice. The second marker is involved in supervision occasionally, mainly in the interview with the second marker, as explained in Section 4.5.

3.4 Progression checking and Submissions

Subversion (SVN) is an open-source revision control system. It is frequently used for safely managing code and digital resources in a distributed working environment. SVN is used to maintain files in a way that allows you to get back to earlier versions in a seamless and convenient way. The way to work with an SVN repository is to have a local copy stored on your computer (or multiple copies on all the computers you work on) and frequently commit them to the repository. Then, the committed version can be updated to all local copies of the repository. It is also possible to upload an earlier version of a file or a set of files. So it is possible to go back to a previous version of your work. More information on SVN can be found at <https://campus.cs.le.ac.uk/demonstrators/info/VCS/>.

All deliverables will be handed in through Subversion (SVN). Students' SVN repositories can be accessed (by the student and supervisor) from any computer, either on-campus or off-campus.

Students **must submit all deliverables** through SVN and failure to use it may result in a Departmental Warning Letter or further action if recommendations in previous warning letters have been taken to their full extent. The usual deadline for submitting project deliverables is before midnight – 23:59:59 – on the corresponding deadline, as indicated in the Google calendar (1). SVN must be **mandatorily** used for:

- Safety backup of students' work. Students' work-in-progress must be submitted on SVN regularly.
- Storing and submitting project deliverables, and software related to the project.
 - Every non-trivial change should be committed separately, with a meaningful comment summarising the change.
 - An absence of commits will be interpreted as a lack of work by the student on the project. Any work must be reflected in SVN activity.
- Project progression monitoring.
- Retrieving and marking students' work. Assessment will only be based on project deliverables, not on intermediate versions of documents/software.

At the end of the project the following must be submitted **on SVN**:

1. An electronic copy of the final report in the format of the text editor that was used *and* a PDF version. The PDF version must be named `username_final_report.pdf`, e.g., `np183_final_report.pdf`. *All in lower case letters!*
2. A copy of the source code of the software developed together with all of the following:
 - A setup program for installing the software.
 - All the code necessary for compiling and running the project.
 - The binaries for executing the software developed in the project, including third-party software components necessary for executing the program (provided that this is allowed by their license).
 - A copy of any technical documentation and/or user manual, if any.
 - A copy of any other project artifacts that might have been produced.
 - A README file that: locates the main software artifacts of the project in the repository; explains how a user can install the software from the original code; and explains how the executable itself can be run. Details like the operating system or other programs required for compilation should also be stated in the README file.

Finally, **before the viva** an electronic copy of the presentation developed for the viva must be submitted **on SVN** as well. The student must submit the sources for preparing the presentation (acceptable formats are: MS PowerPoint, Keynote, Open Office, and latex) and a PDF version of the presentation. This is the only change that is allowed to the SVN repository after the project submission deadline.

4 Important Dates and Deliverables

Important dates and deadlines for the project are in the Google calendar (1) **Please, consult the calendar and write down the deadlines in your agenda as soon as possible. Check for updates periodically.**

Students progression is checked at two different levels: on a regular basis during the supervisor meeting and on certain deadlines (interviews and deliverables).

The assessed elements of the project (with their respective deadlines) are:

1. Project description (pass/fail): at most one side of A4. **(CA7201 10/2/2016)**
2. Dedication to the project (marked: 5%): The supervisor will assess the student's diligence and commitment to the project.
3. Preliminary report (marked: 5%): (1500 words about 4-6 pages). **(CA7201 24/2/2016)**
4. Interim report (unmarked): (up-to 2 pages). **(CA7201 24/3/2016)** Following the interim report the student will get feedback from the supervisor on progress so far and on the report itself.

5. Interview (pass/fail). **(CA7201 27/3-7/4/2016)**
6. Final-report Template (unmarked): a template of the final report including a complete literature survey and detailed outline of the rest of the report. **(CA7201 21/4/2016)**
7. Project conclusion (marked: 90%): consisting of three parts.
 - (a) Final Report: 10000-12000 words (40-60 pages). **(CA7201 12/5/2016)**
 - (b) Project core: depending on the type of project. Usually includes code on the svn repository of the student. **(CA7201 12/5/2016)**
 - (c) Viva: presentation of the project in front of supervisor and second marker. **(CA7201 15-19/5/2013)**

All three parts are required. Failure to supply, attend, or perform satisfactorily in one of these three components could lead to the student being awarded a mark of 0 in the project conclusion. This would imply a failure in the project as a whole.

The word limit excludes appendices (if any) and bibliography, but includes indexes (table of contents, table of figures) and footnotes. The final report should not exceed the upper word limit by more than 10% (13200 words); exceeding the word limit by more than 10% may result in a deduction of marks. It is our experience that final reports that are under the lower word limit by more than 10% (9000 words) have usually failed to explain important aspects of an MSc project. Short final reports are also sometimes indicative of a project where the overall achievement is not at MSc level (i.e. the student hasn't done enough that (s)he can write about).

In some unusual cases it may be the case that a good project is adequately explained in fewer than 9000 words, due to the nature of the project. You should consult your supervisor, who will advise you if your project comes into this category.

Each deliverable is described in the following sections.

4.1 Project Description

A project description has to be submitted through SVN after the topic allocation. The project description is a document of at most one side of A4 with the following contents:

- Title of the project.
- Brief description including motivation and main challenges.
- Requirements of the project, classified in three categories: essential, recommended and optional. Essential requirements are both core to the project and mandatory to get a minimal working prototype. Recommended requirements are extensions to achieve a better and more complete product. Optional requirements are defined to envisage challenging features that would stretch the student's ability to complete in the duration of the project.

As feedback, your supervisor should discuss with you how realistic these requirements are and how realistic is your evaluation of the partition to essential, recommended and optional. Please note that requirements set at such an early stage of the project are likely to evolve as the project progresses and are not directly related to the grade of the project.

4.2 Supervisor Meetings

Starting from week 1, students should meet with their supervisor on a regular basis.

During a meeting, the student should: provide a summary of the outcomes from the previous meeting; discuss achievements, ideas and queries with the supervisor; propose tasks to be done during the week ahead and discuss them with the supervisor. The student should participate actively and take notes during a meeting, as necessary.

4.3 Preliminary Report

By the end of week 2, students have to submit a preliminary report of 1500 words approx. (5-6 pages long). The word count of the document has to be included on the front cover.

The preliminary report is a reference for the project's aims, objectives, challenges and the time plan. The student should define/describe them by investigating background material and related work/literature. Project's aims, objectives and challenges must be clearly stated and presented. It is also necessary to design a viable work plan, which might be updated as work progresses.

The preliminary report must include:

1. a brief discussion on project relevance/motivations (why is it worth doing it?) and its main challenges;
2. list of requirements, including high-level requirements or aims, and detailed requirements or objectives;
3. technical specifications of the project;
4. background material (including a reading list);
5. a detailed timetable and plan for achieving the objectives of the project, including the milestones of the project and a risk plan.

4.4 Interim Report

By the end of week 6, students have to submit an interim report. The interim report should include a short update on the current status of the project in reference to the preliminary report. It should make it clear what parts of the requirements have been completed, started, and pending. It should also include an update of the plan that was submitted in the preliminary report. The recommended length of this deliverable is up-to 2 pages (excluding cover page). The word count of the document has to be included on the front cover.

The interim report must include:

1. A clear description of the progress that has been made with reference to the requirements that have been set in the preliminary report.
2. An update of the plan that was submitted with the preliminary report.

The interim report *must* be submitted as a *PDF* file in the directory docs/3_interim_report of your SVN repository. It *must* be named username_interim_report.pdf, e.g., np183_interim_report.pdf.

4.5 Interview

During week 7 the students will have an interview with the second marker. This interview allows students to get feedback on the project from a different point of view. The project has to be presented in a precise, informative and self-contained way.

During the interview, the student will start describing the project as follows:

1. Introduction and aims of the project.
2. Description of the work performed until that stage. Depending on the project type, this step may cover:
 - I. Requirements of the system, design and demo of a prototype.

- II. Presentation (with demo if possible) of the tools/techniques that are being used and description of experiments.
 - III. Analysis of existing techniques, languages or theories, identification of the main problems that need to be solved, and proposed solution.
3. Main challenges and contributions.
 4. Work that remains to be done and schedule.

The second marker might ask questions while the student is presenting the work. There is no need to prepare slides for presentation.

The interview takes place during week 7 and has to be arranged by the student. The interview usually takes place in the second marker's office. This interview may take up to 30-45 minutes. During this week, a meeting with the supervisor is not requested.

4.6 Final-report Template

A document in the format resembling the final report is to be submitted. The template consists of a detailed survey of the literature relevant to the project as well as the outline of the rest of the final report. The word count will be included in the front cover.

The template must include a table of contents. Each section that will appear in the final report should be included in the template, along with the subsections. Each section/subsection should include a written sketch of the main ideas that will eventually be included in the final report.

The final-report template *must* be submitted as a *PDF* file in the directory docs/4_report_draft of your SVN repository. It *must* be named username_final_report_template.pdf, e.g., np183_final_report_template.pdf.

4.7 Final Report

The final report is the final and possibly the most complex among the project deliverables. The student must give a detailed account of the project in terms of

- achieved results and their evaluation;
- a critical comparison of the student's work with respect to related work;
- concluding remarks and a reflective analysis of the project (e.g., a discussion of non-achieved or modified objectives, and a description of possible further developments).

Typically, a final report contains 10000 to 12000 words (40-60 pages). A rule of thumb is that theoretical and technical projects tend to produce shorter final reports than software development projects.

Given its complexity, it is important that students prepare this deliverable carefully. A final report must be written in good English. Students must organise information in a consistent way and should use cross-references to allow the reader to easily access the document (see Section 5 for information on the format of the final report).

The content and structure of the final report must be discussed with the supervisor (who will be able to guide the student to identify the structure). Generally, each final report is expected to include a description of the following aspects of the project:

1. Abstract: max 300 words describing aims, objectives and results of the project.
2. Introductory section: an account of aims, objectives and the original results of the project. A rule of thumb for writing a good introduction is to write it such that a reader can understand the contribution and the context of the final report *without* having to read it.

3. Background section: gives a thorough bibliographic account of related work. A good background section not only describes relevant work related to the project, but also tries to put the project in its appropriate context and to contrast the original approaches/methodologies/technologies proposed by the student with related approaches.
4. Contribution of the project: several sections that give details on the project results and their evaluation. For example, the structure of the body sections in the final report could be, depending on the project type:

I. Requirements, Design, Implementation and Test, Evaluation.

II. Problem description and problem statement, Techniques and/or Tools, Approach, Evaluation and Results.

III. Problem description and problem statement, Theory, Methods, Results.

For projects of type I and II, part of your contribution is the code you have written. For these projects, the contribution section *must* contain 3 excerpts of code that *you have written yourself*, with references to the files in your SVN repository where the excerpts appear. You should explain what each piece of code does. You should also state what the technical contribution of each piece of code is. For example, a piece of code that implements a complex algorithm is a bigger technical contribution than a piece of code that simply sums an array of integers. In the rare case that you cannot fulfill this requirement, you should ask your supervisor for guidance.

5. Conclusions: a critical appraisal of the project results (what has been done? does it compare to other similar proposals?). This discussion should cover aims that were part of the project but that have not been achieved (where applicable) and, in such a case, a detailed justification should be developed.

The supervisor can advise on technical adequacy and style of presentation. The Student Learning Centre (<http://www2.le.ac.uk/offices/careers/ld>) can offer help with writing and language skills.

The final report *must* be submitted in the format of the text editor that was used *and* as a *PDF* file in the directory docs/5_final_report of your SVN repository. The PDF version must be named user-name_final_report.pdf, e.g., np183_final_report.pdf.

4.8 Viva

The main goal of the Viva is for the student to show a clear evidence of the work that has been performed in the project. The supervisor and the second marker will try to establish that whatever is claimed to be a student's achievement in the project is actually her/his own work. As such, questions may be asked at all stages of the Viva and students will be examined thoroughly and rigorously on all aspects of the project. Students are not expected to be able to answer all questions in depth, but they should be able to provide sensible and competent responses. During the Viva, if the project involves the development of software, the student must demonstrate that the developed prototype satisfies the specification included in the final report.

During the last week of the project students have to contact their supervisors to arrange the Viva if they have not been contacted already with this regard. The Viva is conducted by the student's supervisor and second marker and has two main parts:

- Formal presentation with slides (PowerPoint or similar) explaining the motivation, aims, contributions and results of the project. This part will usually take 10-15 minutes.
- Demo (if any) and interview with the supervisor and second marker. The structure of this second part is flexible and it is meant to be quite interactive.

Both parts together usually take up to 45 minutes. Attendance at a scheduled Viva is mandatory. Absence is allowed only in the light of suitable medical evidence (in which case a new Viva will be scheduled). Typically, the Viva takes place in the student's supervisor's office.

There is one mark for the entire conclusion of the project including the final report, project core, and Viva. It follows that performance in the Viva impacts the entire project conclusion. Failure in the Viva might result in failing the project as a whole.

5 Format and Structure of Documents

All documents that students have to deliver during the project must respect a few typographical conventions (summarised at the end of the section). Moreover, it is important (especially for the final report) that students take care of how the material is organised and written.

Students are strongly encouraged to consult the facilities that the University of Leicester provides for improving their skills. Advices on writing, avoiding plagiarism and organising material in documents can be found at <http://www2.le.ac.uk/offices/careers/ld/resources/writing/index>. Moreover, it is possible find how to cite references in scientific documents at <http://www2.le.ac.uk/library/help/citing/managinginformation>. The reference list should contain a mixture of books, research papers (if appropriate) and internet resources, and should not consist only (or mainly) of Internet resources. An example of a proper citation is

In [1] general citation standards are presented, but the interested reader is referred to [2] for a more in depth discussion. Note that references to Internet sites should include the **full** URL including *http*, etc. These references should also include the date on which the site was accessed. Software like Zotero [3] can be useful to keep local copies of web resources and access dates. Bear in mind that, though useful, Wikipedia should not be used by itself for primary research [4].

...

Bibliography

- [1] Student Learning Centre of the University of Leicester. "*Referencing & Bibliography*". <http://www.le.ac.uk/teaching/pdf/writingskills/referencingandbibliographies.pdf> (accessed 10/6/2009)
- [2] Julia Johns, Sarah Keller. *Cite it Right* (1st Ed.), 2005. SourceAid, LLC. Ostervill, MA
- [3] Zotero website: <http://www.zotero.org/> (accessed 10/12/2009).
- [4] Wikipedia. *Researching with Wikipedia*. http://en.wikipedia.org/wiki/Wikipedia:Researching_with_Wikipedia (accessed 10/9/2009)

The format and structure of the documents submitted must comply with the following conventions:

1. All delivered documents must start with a title page containing the following information:
 - (a) "Department of Computer Science, University of Leicester"
 - (b) "CO7201 Individual Project"
 - (c) The full title of the project
 - (d) The full name of the author
 - (e) Student email and university ID
 - (f) Document name (Project Description, Preliminary Report, Interim Report, Final-report Draft, Final report)
 - (g) Name of both the project supervisor and the second marker
 - (h) The date of submission
2. All documents must have all pages numbered, apart from the title page.

In *all* documents, the page immediately following the title page must contain only the abstract and the following declaration.

DECLARATION

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s). Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s). I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name:

Date:

The student must add their name in the appropriate place and date the declaration to indicate that the declaration holds for their work.

Your work will not be assessed without this declaration and your final report will not be assessed until the declaration has been named and dated.

The final report must be formatted on A4 paper and we recommend 12pt traditional serified font (e.g., "Times" or "Bookman") for the main text, and 10pt fixed width font (e.g., "Courier") for program code segments and computer outputs.

6 Plagiarism

The issue of plagiarism is very important. You MUST read the University's statement and the departmental regulations concerning plagiarism. Those can be found either in the University Regulations or on the Information for Students web pages at

<https://campus.cs.le.ac.uk/ForStudents/plagiarism/>

You will note that the rules regarding plagiarism for *project* modules are different to those that apply to non-project modules.

While you may certainly ask your supervisor for advice if you are having difficulties with your project, and while you may discuss your project with other students in general terms, **the work you hand in must be your own. Copying other people's work, or asking other people (e.g. via websites or similar means) to undertake work for you is a form of plagiarism, is strictly forbidden**, and will result in **ALL the students involved being severely penalized**. It is also strictly forbidden to **let people other than your supervisors have access to your work**. For example, if you use a laptop to do your work, you should not lend it to anyone else while you are doing the project.

In project modules it is acceptable to include quoted text written by other people, provided that the sources of information are clearly referenced in the bibliography, and are cited properly within the project report. Students must be in a position to understand such quotations, and to explain them to supervisors and examiners, especially at the time of interview and the viva voce examination. It is also acceptable to make use of software components to support the student's own original work, but again their use must be very clearly indicated, the sources must be referenced, and students must expect to be able to explain what role they play in the project.

Note that according to the departmental regulations, if a student has committed gross academic dishonesty, full details of the case will be forwarded to the Registrar for consideration under the Code of Student Discipline. The module mark may be reduced (even to zero), which would mean that the student may not be awarded the MSc degree and that the case may constitute grounds for course termination. Any student who undertakes their studies genuinely should

have no concerns, and students should understand that the role of such penalties is to try to ensure that only true and genuine effort is appropriately rewarded.

Note also the **declaration that you must include in all your reports and sign in your Final Report** (see Section 5).

7 Feedback

The department, in line with University policies, attempts to mark assessed coursework within ten working days. This policy applies to the following deliverables: Preliminary Report, Interim Report and the template of the Final Report. Students will be informed to see information about their own grades and, more importantly, about the markers feedback on their achievements. The feedback of the aforementioned assessments will be available online through the Student Feedback Webpage. Depending on the time between submission of project description and start of project feedback on project description will be given either verbally during the first meeting with the supervisor or through the Student Feedback Webpage.

On the other hand, given the nature of the assessment for the project, the turnaround time for marking the final report on this module may well be more than ten working days. In this case, after all involved markers finish their marking, a moderation meeting will be carried out to guarantee assessment consistency. After the final meeting of the Board of Examiners in June (for Spring projects), in October (for Summer projects), and in February (for Autumn projects) students will be informed of the results of the degree. At this point, students may contact their supervisor for more detailed feedback on the project mark itself.

The Examiners may deem a project report to be satisfactory provided that some minor amendments are made. In such a case, the Examiners may require the candidate to make the specified amendments within a specified amount of time (at most three months but usually not longer than one month). The final result will depend upon the Examiners being satisfied with the amendments that have been carried out.

8 Late Submission of Deliverables

Please note that each deliverable in this module has to be handed in by a specific deadline, as reported in the Google calendar. **The deadline for each deliverable is strict and no extensions are allowed. Work that is submitted late will be penalized.** We need you to meet these deadlines, since it is in your interest that we keep to the prearranged timetable for the marking so that you receive constructive feedback on your progress in time. A deliverable, except for the final report, that is submitted late will get a **mark of zero**. In the case of a late submission of the final report:

- A penalty of 10% of the available marks will be imposed upon the expiry of the deadline (day 0). Work handed in on day 0 after the deadline will get a mark of $M - \frac{M}{10}$, where M is the student's mark before penalty.
- A penalty of 5% of the available marks will be imposed on each of the ten subsequent working days. Thus if work is handed in on day d the mark awarded will be $M - \frac{M}{10} - (d * \frac{M}{20})$. Work handed in after this will receive no marks.
- "Available marks" in this context means the maximum marks available for the piece of work (for example, 100 would be the available mark in a percentage marking scheme, 20 would be the available mark in a 1-20 marking scheme).
- "Working day" in this context means a period of twenty four hours or part thereof from Monday to Friday inclusive.

In the event of you being unable to work on the project because of illness or other bona fide reason, allowance will be made provided that a medical certificate or other adequate documentary evidence is produced (see Sections 9 and 10).

In view of the importance of handing in work on time, you need to make a conscious effort to organize your time effectively. Note in particular that when we allocate, say, two weeks for a document, we mean that it will take you two weeks (allocating the correct proportion of your time to the module) to carry out the work. You will not be able to meet the deadline if you spend one week and a half on something else and then try to do all the work in the last three days.

9 Mitigating Circumstances

It is the responsibility of students to inform their department(s) of any matters (whether of an academic, personal, medical or other nature) which may be relevant to their academic performance, and to supply substantiating evidence, for example, a medical certificate. Such information should be submitted before the expiry of the following departmental deadlines governing the submission of evidence of special circumstances.

The reporting of mitigating circumstances on time is crucial for the correct assessment of the project. Even though there are no extensions on the project, valid and timely submission of mitigating circumstances may be used to assess individual cases and spare unnecessary complications.

In order for the Project Convenors and the Exam Board to consider mitigating circumstances the relevant Mitigating Circumstances forms (MCFs) must be submitted as soon as possible and at the latest by 12 noon of the deadline for submission of template of final report and one week after the submission of the project.

Appeals grounded on new mitigating circumstance unknown to the department are likely to be rejected if not supported by extremely good reasons for the delayed notification to the department. In general, the appeal panel may not be favourable to mitigating circumstances that were not reported promptly to the department.

In general terms, the presentation of medical or other special circumstances does not of itself guarantee that academic concessions will be granted. Cases are considered on their merits in the light of the extent to which the adverse circumstances might reasonably be deemed to have affected a student's performance or justified a failure to meet deadlines.

The filled mitigating circumstances form has to include an evaluation of the negative effects that the circumstances had on the project and an estimation of the amount of time lost due to these circumstances. Submitted forms must be accompanied by supporting documents.

10 Notification of Illness

It is students' responsibility to make all reasonable efforts to submit all deliverables on time. In extreme circumstances, late submission may be considered as long as the last modification date of the relevant files is for the submission date (or earlier). Late submission of final report and project will be heavily penalized if not promptly justified by relevant mitigating circumstances (see Section 9).

Students who suffer a minor illness lasting less than seven days during the run of the project are required to report this to the department.

Students must self-certify their illness using a **Mitigating Circumstances** form (MCF) available at

<https://campus.cs.le.ac.uk/ForStudents/welfare/Welfare.html>.

Normally, Notification of Illness forms which are returned after the deadline for the template of the final report submission and more than one week after the submission of the project might not be considered promptly (for example, for the purpose of extensions). However, all such forms will be considered at the subsequent Board of Examiners meeting. **Self-certification is suspended for the project submission and the viva.**

Where the illness is of more than seven days' duration or it is of a non-minor nature, medical advice should be sought and a medical certificate submitted to the Office. Students are responsible for collecting medical certificates from the Freeman's Common Health Centre and supplying a copy to the Department.

It is the responsibility of students who are required to produce medical evidence to acquire such evidence.

Freeman's Common Health Centre now charges the University for providing medical certificates and reports. Students and tutors may be asked to complete an application form before a letter is written (this request form is submitted to Freeman's Common Health Centre through the Student Welfare Service for audit purposes). Other general practices may charge for providing reports and such charges must be met by the student concerned.