Group Software Project

Contents

This set of notes are meant to provide students with guidance for their work on their second year group project module. The notes will be complemented by lectures. These will look at various aspect of the module and provide additional details on what is expected from the students.

1 Course Aims and Objectives

The main objective of this module is to give all students an opportunity to work in small groups to produce a reasonably sized software system. A number of deliverables and working methods will be prescribed to facilitate such task. We hope that this

Aims and objectives

- will provide a useful team working experience;
- will provide experience on all aspects of the development of a moderately sized software system;
- will prepare students for their individual projects in the third year (and will also be a valuable experience for those students that will soon embark in an industry placement as part of their degree);
- will consolidate material from previous parts of the programme.

For COMP208, the software to be produced could focus on a database application, the domain being chosen by each team themselves. For COMP215/216, it is hoped that the various teams will focus on applications involving a web-based or mobile user-interface. For COMP214, the project should make use of AI methodologies and techniques to solve a difficult computational problem. More details will be provided further on. In each case a precise methodology should be followed prescribing the steps to be taken and the documentation to be produced during the software development process.

Final product, module specific requirements

The skills acquired or enhanced by the end of the module are:

Learning outcomes

LO1. an awareness of the issues involved in working as part of a team;

LO2. personal, interpersonal and communication skills;

LO3. an understanding of the software development process;

LO4. the ability to specify (fully, and precisely) the requirements of a software system;

LO5. demonstrate some experience in the design of a software system;

LO6. an experience in the implementation and testing of a moderately sized software system;

LO7. an awareness of the typical project management issues;

LO8. an understanding of the process and role of software documentation;

LO9. an experience in the writing of a sizeable report on a software project.

2 Course Mechanics

Teaching and Learning Strategy

The module is run by Michele Zito, in collaboration with Bakhtiar Amen, Michael Bane, Sebastian Coope, Keith Dures, Floriana Grasso, Rida Laraki, and Shagufta Scanlon. Dr. Zito is the main module co-ordinator. He will deliver all lectures and have a number of academic, pastoral, and administrative responsibilities in relation to this module, including that of setting up and managing the assessment process. All other members of the teaching team will participate in the assessment, arranging an running the various reviews (see Section 3), and then marking the student work. Dr. Zito will act as moderator, assessing about 10% of the class work and controlling the completeness and consistency of the whole assessment process as well as the quality of the feedback returned to the students.

The next sections explain how the module will be organized.

2.1 General Overview

All students will work *as part of a team* in order *to produce a substantial piece of software*. All significant software developments - whether commercial or academic - involve teams of people. Working as part of a team requires a number of skills including the ability to plan the work, communicate with one another, cope with incoming issues, and co-ordinating the group activities. Being able to work as an effective part of a team is a quality that is required in all walks of life. Exercises and assignments in other modules give students some experience of software development, but often these only provide partial understanding of the process involved. Seeing a project through from conception to completion, and gaining experience of all the stages in the development process, is essential to understanding what is required to produce real software systems. For instance, Connolly and Begg [1] identify the following stages in a database application lifecycle:

1. Planning
2. System Definition
3. Requirements Collection and Analysis
4. Data Design, typically including Logical design; and Physical design
5. Applications Design:
6. Implementation
7. Testing
8. Operation and maintenance

It is expected that, using previously gained software engineering skills, all teams will be able to agree on a precise development process and complete their work following the stages of such process.

A fully integrated learning environment is set up to facilitate the learning experience, including lectures, an electronic repository, a comprehensive reading list, tutors and project monitors.

2.2 The Software Required

All teams will work on a substantial sotware development project. While some general guidance was provided in Section 1, the teams have complete freedom as to the project they choose. A list of possible projects is available at

`http://www.csc.liv.ac.uk/~comp208/projects.html`

The full list of project titles for the academic years 2012-13 and 2013-14 is contained in the Appendix at the end of this document.

Further, module dependent, guidelines are given below. When it comes to choosing a project the most important piece of advice is

> BE INVENTIVE, BE BOLD. USE THE GROUP PROJECT AS A VALUABLE EXPERIENCE.

The application should be of reasonable complexity, bearing in mind that typically five to six people will be working on it for approximately 120 hours each. “Complex” does not necessarily mean large. It might include a complex data model, some non-trivial processing, or the ability to be accessed by several different types of users.

This module gives a unique chance to practice team work in software development, in a controlled setting. Use it!

2.2.1 COMP 208

For students enrolled in COMP208, the required software will typically (but not necessarily) be a database application, of the sort exemplified by the case studies in [1]. However, alternative projects with a more significant algorithmic component are also encouraged.

The team will be responsible for choosing the domain of the project. Ideally this will reflect the interests of at least one team member: sports records, hobby data, or a database for a small club or even a small business all provide sensible subjects for a database application, and there are many other possibilities. Application domains resembling the ones used in the examples on Connolly and Begg, and other “trite” scenarios (e.g. libraries, simple on-line shops, gyms) are not allowed.

2.2.2 COMP 214

For students enrolled in COMP 214 typically the project will involve some non-trivial application of AI techniques studied in other parts of the curriculum. Here’s a couple of examples:

- you are required to build a search engine for a peer-to-peer (P2P) network. Search is one of the main areas of investigation of Artificial Intelligence, because many forms of machine intelligence involve search across some space of possible actions or knowledge. Information (such as music files) in a peer-to-peer network is distributed and may be difficult to find if there is no central catalog of which peers have which files. In such networks, some search algorithm is needed to locate peers with desired information. You will need to define the P2P network structure (eg, flat, cellular, hierarchical) and develop alternative algorithms for searching the network. Alternative search algorithms could include query flooding or more intelligent methods, including ones using heuristics (such as records of previous searches).

For this project, you only need to simulate the P2P network, not necessarily construct such a distributed network.

- Timetabling. A school runs a number of classes. Pupils in one class have to take a number of different modules. Teachers teach modules but can have a given maximum working load. You will have to represent a system of this type and then define a search strategy to solve the entailed allocation problem.

2.2.3 COMP 215/216

For students enrolled in COMP 215 or COMP 216, the task is again to develop a reasonably sized application. In particular it is envisaged that the system will have a sofisticated graphical user interface (e.g. web-based or mobile) and particular care should be spent in designing and implementing such component.

For the database aspects of this project, see the instructions above for COMP 208 students. Lecture materials from other modules offered by this Department and help from the Computer Science help-desk team should provide the required support on the networking aspects of your work.

For the COMP215 students application areas of particular interest should have a business component (e.g. payroll systems, business activities management systems, accounting systems). For the COMP216 students an interesting application area is that of social networks.

2.2.4 Intellectual Property

Under the terms of the University's IP policy (available through the module electronic portal), sections 2.9 - 2.13 state that

> "as part of the registration process students assign to the University any commercially-exploitable IP which they generate as a consequence of their studies or research, or which is created using University facilities."

Students should contact a member of the module team if they foresee the possibility of commercial exploitation of their system or they need further information in relation to Intellectual Property.

2.3 Reading Lists

Students will find that the most valuable source of information for this course is the full collection of lecture material used since the beginning of their course, as the work for this module will lead students to use various tools and techniques previously encountered in other parts of their degree course.

On top of that, two books are recommended for this course:

- Christian W. Dawson: Computing Projects: A Student's Guide [2].

  This book gives good advice on all aspects of how to set about computing projects, both team and individual.

- Thomas Connolly and Carolyn Begg: Database Solutions [1].

  This book gives a step by step guide to developing a database application, the sort of software application you will produce on this project. It is required that you follow the methodology represented by this step by step guide on your project, so that this book can serve as a ”project handbook”. The book includes two case studies: these represent the kind of thing which you will be trying to produce. It is strongly recommended that each team has access to a copy of this book.

Additional, module dependent, reading material might include:

- Russell, S. and Norvig, P.: Artificial Intelligence, a Modern Approach [4]
- Pressman, R. S.: Software Engineering [3]
- Stevens, P. Larman, C. and Pooley, R.: Using UML, Software Engineering with Objects and Components [5].

Students are strongly advised to contact the module lecturer for tailor-made advice on additional reading material.

2.4 Teams

Students on the relevant programmes will be put into teams of roughly equal size, typically with a minimum of four members, and thereafter will be expected to work largely autonomously. Teams will be expected to hold regular project meetings, the minutes of which will be monitored by staff (see Section 3.5).

Students have significant freedom in the choice of their team partners: any choice is allowed provided all people in the same team are registered for the same module. Thus the following is an example of acceptable team for COMP208 (all team members from the same programme):

```
NAME                    PROGRAMME
Gerrard, Steven         G50A
Sterling, Raheem        G50A
Coutinho, Philippe      G50A
Balotelli, Mario        G50A
Jones, Brad             G50A
Sakho, Mamadou          G50A
```

... but the following is also allowed: a team of people from different programmes, sharing the same module (COMP208)

```
NAME                    PROGRAMME
Baines, Leighton        G40A
Howard, Tim             G610
Costa, Diego            G50A
Terry, John             G50A
Matic, Nemanja          G610
Mourinho, Jose          G403
```

whereas the next one is NOT (Sanchez and Pellè belong to programmes that do NOT contain COMP208)

```
NAME                    PROGRAMME
Mata, Juan              G502
Toure, Yaya             G610
Silva, David            G40A
Sanchez, Alexis         N300
Nasri, Samir            G610
Pelle`, Graziano         G700
```

We encorage students to set up teams with people they do NOT know (it should be remembered that in a professional environment we hardly ever choose our colleagues).

Team (Self-)registration will be done through VITAL by the end of the first week of the teaching term.

> IMPORTANT: Please bear in mind that although every effort will be made to accommodate student preferences, the module lecturer might have to re-arrange the team members due to the need to allocate all students.

The team becomes operational as soon as the lecturer approves it. Upon approval each team is assigned a group space on the module portal in VITAL. As soon as such space becomes available the teams can start their work.

Anyone who is not part of a registered team by the end of WEEK 1 will be assigned to a team on a random basis.

> One golden rule:
>
> STUDENTS ARE STRONGLY ADVISED TO BE ACCOMMODATING ABOUT TEAM MEMBERS.
>
> Working with friends can be good fun, but can lead to tensions and stress. Furthermore, in a real working environment, we hardly ever choose the people we work with. Be positive about it! TIME IS REALLY TIGHT: the most important thing is to get going as soon as possible.

2.5 Planning

Any (human) activity needs a plan, so that time can be managed effectively, and progress can be monitored. When there is a team of people involved an agreed plan becomes even more important, because development becomes a complex process articulated in a number of related tasks. These need to be clearly defined, assigned to team members, and carefully managed.

There is a good discussion of software project planning techniques in [2, Chapter 3].

2.5.1 Creating a Plan

Some events, such as the various project reviews, happen on fixed dates, and the group plan must accommodate these. It is also important to identify other milestones - key elements of the project and a date by which they should be completed. To this end one should identify what needs to be done for each milestone, and make an estimate of how long each task will take. Decomposing large tasks into smaller tasks can make these estimates easier. It is also useful to identify what activities depend on other activities, and this will impose a sequence on the activities. When all this is complete, the team will have a list of tasks that must be done before other tasks and others that can be done in parallel. Plan creation

Tasks can then be allocated to the team members. The allocation should be fair. There should be an even spread across people and an even spread throughout the project. The situation where everyone is waiting for one person to complete a crucial task should be avoided. Task allocation

It is also important to allow some slack in the plan for when things take longer than expected. For instance, note that there is not much time between the demonstration and the deadline for the final portfolio. It is therefore essential that the work on the portfolio is started in parallel with work on the implementation.

The plan is best recorded diagramatically as an activity diagram or a Gantt chart.

2.5.2 Using the Plan

Once the plan is drawn up, it should be used, by allocating tasks and then regularly checking their progress. Inevitably, some activities will take more time or less time than you estimated. If this happens the plan should be adjusted, putting tasks back if possible, or allocating extra resources to unexpectedly difficult tasks. Re-plan is non a problem on itw own: it is better to adjust the plan when problems become evident than to stick to a plan until it becomes impossible to meet the milestones.

2.6 Support

As mentioned before, teams are expected to work largely independently, at the development of a reasonably sized software project. However the learning environment includes resources to discuss issues and problems as soon as they arise.

The module co-ordinator should be the first point of contact to discuss any arising matter related to the group project work.

As part of the team formation process, each team is also assigned a member of staff from the module teaching team and their academic *project monitor*. The project monitor provides additional support for questions or difficulties related to the group project. (S)he will also periodically check that team work is progressing well, by periodically inspecting the team meeting record on VITAL. The module co-ordinator and the project monitors will advertise times at which they will be available for a meeting or a chat. Meetings with the co-ordinator or the project monitors should be minuted and the minutes should be copied to the electronic module environment (see Section 4). Keeping the monitor informed about the group progress, raising issues and problems, events etc is usually expedient to the success of the project.

2.7 Professional Issues

When carrying out a project should be aware of the professional issues involved. The British Computer Society issue a Code of Good Practice and a Code of Conduct. These can be found here:

`http://www.bcs.org/upload/pdf/cop.pdf`
`http://www.bcs.org/upload/pdf/conduct.pdf`

These documents set out guidelines for the proper conduct for a software professionals undertaking a project. Of course, these were written with a large multi-person development in a commercial environment in mind, but you should be aware of these codes, and apply their principles, where appropriate, as you carry out your project. The Code of Conduct is probably the more relevant.

2.8 Lectures

There will be a number of lectures on this course, providing additional material on the course requirements and expectations, material on some relevant technical aspects, and information about transferable skills developed during the course. Most lectures will be given by Dr. Zito. We also aim each year to have some guest lectures on topics such as Group Working and Large Software Systems Development. Details about these will be communicated during the term

Copies of slides will be made available during the term through the VITAL module portal.

3 Assessment

This module is assessed by coursework only. During the course of the project there will be a number of reviews which will contribute to the assessment for this module. Poor performance or failure in one review may be compensated by better performance in others. All reviews will assess the first three skills mentioned in Section 1 as well as the level of awareness of the typical project management issues and an understanding of the importance and the role of software documentation. Furthermore, different reviews will focus more specifically on some the module skills:

Some reviews require the teams to produce a written document and to discuss such document at a meeting with the project assessor. Others will be carried out based solely on the content of written reports. It will always be assumed that all written documents have been produced by all members of a team. Under normal circumstances, each of the four reviews will result in the same mark being awarded to each team member for her work. However failure to turn up for a review meeting will result in ZERO mark for that component of the particular review.

Along with the team project portfolio (whose format and submission process is described in Section 3.4) each member of a team will submit an individual statement of learning outcomes and a peer group assessment form (see details in Section 3.4.1). These individual submissions, together with the individual participation as recorded in the meeting notes, will be assessed and used to further differentiate the marks for each member of a team.

Under normal circumstances late submission is allowed for all written reports, and treated according to standard University late submission policies. Since the oral reviews are set up by joint agreement of the academic assessor and the teams involved, as described in the forthcoming subsections, failure to arrange or turn up for such meetings will result in ZERO mark awarded to all truants for that component of the particular review.

Separate reviews are run for each team. They must be attended by all members of that team and take place in front of a reviewer, normally in the reviewer's office, or in one of the Department's meeting rooms, as mutually agreed by all parties involved. Reviewers will be members of the module teaching team. The full list of reviewers, for each team, and each phase of the assessment process are published on the module electronic learning environment at the beginning of the second week of term, soon after the teams are finalized.

Additional details on each assessment task follow.

3.1 Requirements Walkthrough

The first review focuses on the *requirement specification*. The purpose of this component is to ensure that an appropriate and feasible system is planned for implementation by the team for the module, and that the team is working following a reasonably well-defined plan. The first review assesses the general learning outcoumes **LO1**, **LO2**, and **LO3** as well as **LO4**, as described in Section 1.

Each team will be required to submit a specification document by the end of Week 3. In the academic year 2018/19 the submission deadline is 12 noon on February 15th, 2019. No bound is set on the size of this document, but the report must contain a precise specification of the chosen system.

Report. The report should be structured as follows:

1. Project Description

   This section describes what the project is about. This could include:

   - Who the project is being done for (if indeed such character exists!): this could be a (group of) friend(s), a customer, etc;
   - A mission statement for the application (what the aim of the project is, what it is intended to achieve);
   - A description of the mission detailed objectives

2. Statement of Deliverables

   This section describes what will be produced in the project. In some cases it may be useful to identify some deliverables as essential and others as desirable. As appropriate this will include:

   - Description of anticipated documentation;
   - Description of anticipated software;
   - Description of any anticipated experiments;
   - Description of methods for evaluation of the work.

   The focus should be on the description of the anticipated software. A detailed description of the features / functionalities of the software should be given (again, possibly distinguishing essential and desirable features). A clear system boundary diagram should be drawn, drafting the scope of the sought system. A description of the different user views, and their functional requirements should be included. For database applications it is also appropriate to include details of the transaction requirements. Any constraint on the sw/hw should also be specified here.

3. Conduct of the Project and Plan

   This section describes how the project will be carried out and describes the activities involved in different stages. This should include, where appropriate:

   - Preparation

     Background research: what information will be used to fully understand the problem and derive its solution, and provide a context for the project (you should state clearly what information you have already absorbed and what is yet to be read during the early stage of the design stage);

     Data required: what data will be need to be acquired for the project and where it will be obtained;

   - Design stage

     What design methods will be used and what the design documentation will consist of;

   - Implementation stage

     What hardware and software will be used;

     What testing will be carried out;

   In addition, a plan in the form of a time-tabled schedule of project activities and outputs will be given. This should include internal milestones as well as external assessments and reviews. The plan should both state progress to date and indicate future activities. A diagrammatic presentation of the plan is usually best, and there are standard techniques, such as Gantt Charts, which you can use.

   The stages of the project should be broken down into an appropriate level of detail, e.g., ”design 5 weeks” is too vague: you need to indicate what tasks will go into your design and how long you expect each of these tasks to take.

   Also included should be a risk assessment containing the following and how they might affect the plan.

   - What are the major challenges in carrying out the project;
   - What new skills will be required and how these will be acquired;

4. Bibliography

   An outline bibliography showing what reference material has been and will be used. These references should be cited wherever appropriate.

   The lecture notes provide additional details on bibliographies and referencing.

Examples of the relevant documentation related to the “StayHome” case study are in Connolly and Begg [1].

Review Meeting. The requirement review takes place in Week 4 of the teaching term. In the academic year 2018/19 this will be between February 18th, 2018 and February 22nd, 2019. The contact person from each team should arrange the meeting by contacting the reviewer assigned to the team for the requirement review. The review takes the form of a walkthrough of the requirements specification run during a half hour meeting with a project reviewer. Each team should prepare a story to present their documentation in the walkthrough. This can be in the form of a presentation using visual aids, but it does not need to be. It can just be a structured conversation, telling the project reviewer about the work which has been done and about the work which is planned. Each team should decide beforehand which student talks when. The project reviewer will ask questions during the walkthrough in order to clarify aspects of the documentation, and make suggestions where the material seems flawed or has potential for extension.

Marking Scheme. The requirement review contributes a maximum of 12% to the team mark. The quality of the specification document accounts for 60% of the available mark. The other 40% is awarded based on the team performance during the review meeting. Unjustified absence from the review meeting will result in a zero mark awarded that part of the assessment.

Written feedback will normally be returned through the electronic portal as a file left in each team file space. A blank copy of the feedback form can be found in the folder "Forms" in the electronic portal (see Section 4).

3.2 Design Walkthrough

The second review focuses on the *design* of the particular application. The review assesses the general learning outcoumes **LO1**, **LO2**, and **LO3**, as well as **LO5**.

Each team will be required to submit a design report by the end of Week 7. In the academic year 2018/19 the submission deadline is 12 noon on March 15th, 2019. No size requirement is set, but the document must be complete, clear, and concise.

Report. The report should be structured as follows:

1. Summary of Proposal

   A brief statement of the background, aims and objectives of the project , including any necessary changes to the original proposal or specification, based on new information or understanding.

   A summary of the research and analysis carried out so far should also be included.

2. Design

   Although designs will vary according to the needs of particular projects a typical design document will comprise

   - a description of the anticipated components of the system and how they are to be organised;
   - a description of data structures to be used by the system;
   - algorithms to manipulate these data structures;
   - a design of the intended interfaces; and
   - a description of the evaluation of the system.

   If following an object-oriented design methodology one might include:

   - Use-case diagrams;
   - An interaction chart (sometimes called an event trace);
   - The objects to be used in the system;
   - Pseudo-code for the key methods;
   - Interface design;
   - Evaluation design. what criteria will be used to evaluate whether the system is suceesful; how to assess these criteria; who will be involved in the evaluation; what testing will be carried out; what kind of conclusion do you expect from evaluation.

   If following a more traditional design methodology, and developing a DB application one might include some of the following:

- Data dictionaries;
- System boundary diagrams;
- Entity-relationship diagrams;
- Logical table structures;
- Physical table structures;
- Transaction matrix;
- Pseudo-code for the key methods;
- Interface design;
- Evaluation design. what criteria will be used to evaluate whether the system is suceesful; how to assess these criteria; who will be involved in the evaluation; what testing will be carried out; what kind of conclusion do you expect from evaluation.

Additional details are provided in Connolly and Begg [1, Chapter 7 - 17].

For a project involving the empirical investigation of some hypothesis one would normally expect to see in addition things such as:

- A statement of the hypotheses to be tested;
- A description of the test data to be used;
- An experiment design, the experiments to be performed, any control to be used;
- A description of how the results will be analysed, including any statistical techniques that will be used;
- Anticipated conclusions.

For a project attempting to devise new algorithms one would normally expect to see in addition things such as:

- A description of the problem to be solved;
- A description of the existing algorithms of related problem and a critical evaluation of them (e.g., why they are not applicable in your project);
- A description of the approach used to solve the problem; A description of how the new algorithms will be analysed, including mathematical and experimental analysis;
- For the mathematical analysis to be carried out A description of the mathematical model to be used; A description of the performance metric the algorithm will be measured, ranging from correctness, running time, optimally or approximibility for some objective functions; A description of the control against which your algorithm is compared, e.g., the optimal algorithm;
- For the experimental analysis to be carried out An experiment design, the experiments to be performed, any control to be used; A description of how the results will be analysed, including any statistical techniques that will be used;

Alternative techniques, such as data flow diagram; navigation path diagrams; storyboards; functional descriptions or components, can be used. The important thing is that the documentation clearly shows that a design methodology to have been followed, and that the design has been carried out with sufficient attention to detail to inspire confidence that it can be realised, tested and evaluated in the time remaining for the project.

3. Review against Plan

   This is the plan produced as part of the specification, showing what has been completed, and the progress to date. Any necessary changes to the plan should be indicated also. (You are expected to include the Gantt Chart again.)

Note that the design work should inform the implementation stage both in terms of WHAT will needs to be done during the implementation stage (which algorithms or queries need to be implemented) nd in HOW is it to be achieved (which languages or libraries will be used). A clear description of these features is crucial to a successful design, as these can affect greatly the scope of the planned system.

Problem Case. Here's an email excerpt from long ago:

```
> Hi Michele,
> I'm emailing as I have a question about the code for our assignment.
>
> If we were to find code online that performs a function required in our
> project, would we be able to use that so long as we referenced it in our
> documentation? Or would we have to use the code as reference to develop
> our own solution?
> In other words. would this class as plagiarism, even though we aren't
> claiming that the code is ours, and we give the correct references?

Interesting question, thank you.

You can use anything you like but you must reference it correctly.
Definitely do not re-invent the wheel. Integrating pre-existing solutions
can be an interesting and rewarding exercise.

Note, of course, that this could reduce the implementation part of your
project to something very simple. This, really, should not be the case, and
measures should have been put in place earlier on use the time saved in the
most efficient way. In other words:

     saving programming effort is a plus but working less is not!

[..]
```

Email sent a week before the project demonstrations! This is, really, a design question. The best strategy to achieve the right balance between system scope and programming effort should have been picked earlier on in the development process.

Review Meeting. In the academic year 2018/19 the design reviews will be between March 18th, 2018 and March 22nd, 2019. The contact person from each team should arrange the meeting by contacting the reviewer assigned to the team for the design review. The review will then take the form of a walkthrough of the design document run during a half hour meeting with a project reviewer. The project reviewer will ask questions during the walkthrough in order to clarify aspects of the documentation, and make suggestions where the material seems flawed or has potential for extension.

The purpose of this coursework component is to ensure that an appropriate and feasible design is developed for the system to be implemented, and that the design achieves the system aims.

Marking Scheme. The design review accounts for 15% of the team mark. Half of the available mark will be awarded for the performance during the review, the other half for the overall progress in the project and the quality of the documentation. Unjustified absence from the review meeting will result in a zero mark awarded that part of the assessment.

Written feedback will normally be returned through the electronic portal as a file left in each team file space. A blank copy of the feedback form can be found in the folder “Forms” in the electronic portal (see Section 4).

3.3 Demonstration

The third review will take place in Week 11. In the academic year 2018/19 this will be between April 29th, 2018 and May 3rd, 2019. As for the first two reviews, the contact person from each team should arrange the meeting by contacting the reviewer assigned to the team for the project final demonstration. The review will take the form of a demonstration of the application to the reviewer. The purpose of this coursework component is to assess the work undertaken in developing the system specified at the beginning of term. This demonstration will assess the general learning outcoumes **LO1**, **LO2**, and **LO3**, as well as **LO6**.

Each team will present a demonstration of their completed software system. The demonstration is intended to show the complete application in action, so the team should prepare a demonstration that shows the various functions of their application. For COMP208 students, this could include: showing how the database is maintained, and showing how some sample queries and reports can be performed. If multiple user views are supported, this should also be shown. Students in COMP214/215/216 need to demonstrate the application, including undertaking the transactions enabled by the application.

The demonstration is also an opportunity to show any specially interesting or unusual features of the application at hand. The team should also be prepared to talk about the internals of the applications, and to describe how the design was implemented. Each demonstration should take around 30 minutes.

Supporting Material. Each team is required to

1. provide an electronic copy of the specification document detailing the set of requirements that have been implemented. This should be based on the specification document submitted for the first review back in February. However it is well possible that the original document might have been improved following reviewer's feedback or subsequent team discussion.

2. provide an electronic copy of any slide used during the demonstration.

3. provide user manuals, for ALL different type of user, with complete instructions on how to use the system.

4. make the implemented system freely available, even after the demonstration and at least until the end of the current academic year.

   In particular all teams who developed an application with a web-interface should have their system correctly installed and "live" either on Departmental (recommended) or external disk space. The main web-address of the application must be provided to the reviewer (for instance as part of the demonstration slides).

   Teams who developed a non web-interface application should submit a CD (or other type of memory support) including (a) a fully executable version of the system, (b) instructions on how to use the disk and (c) all relevant source code.

   Mobile clients should be provided in a CD or easily downloadable.

Failure to comply with these instructions will be penalized.

Marking Scheme. The assessment of the demonstration will contribute 15% to the team mark. Ten percent of the available mark will be allocated for the performance during the demonstration (focusing on content, organization, and quality of presentation in particular). The remaining ninety percent will be awarded to the quality of the demonstrated software.

Written feedback will normally be returned through the electronic portal as a file left in each team file space. A blank copy of the feedback form can be found in the folder "Forms" in the electronic portal (see Section 4).

3.4 Portfolio

The final review will be on the basis of a written portfolio, submitted at the end of Week 12. In the academic year 2018/19 the submission deadline is 12 noon on May 10th, 2019. The purpose of this component is to bring together all the work undertaken during the semester into a comprehensive technical report. The portfolio review assesses the general learning outcomes **LO1**, **LO2**, and **LO3**, mentioned at the beginning of the section, and **LO9**.

The portfolio brings together all the materials produced by each team on the project, together with a covering report. The portfolio should contain:

- A project report, including the following information
  - Details of the team members and a summary of their roles on the project
  - An overview of the application: what it does, who is intended to use it; why they might want to use it;
  - A description of the extent to which the specified requirements were met
  - An evaluation of the strengths and weaknesses of the project
  - Suggestions for future developments
  - A one page discussion of how your project related to the codes of practice and conduct issued by the British Computer Society. Such codes are available here:

    `http://www.bcs.org/upload/pdf/cop.pdf`
    `http://www.bcs.org/upload/pdf/conduct.pdf`

    The discussion should be evaluative and evidence-based.
  - A bibliography of materials used on the project

  The report must not exceed 10 pages, min font size 11pt, margins 1 inch.
- Design Documentation (what was presented at the design review, *modified as necessary* to fix mistakes and issues uncovered during the review)
- Test Documentation, including:
  - Testing strategies used
  - Analysis of test results
- Some sample screen shots of your application to indicate the look and feel of your system

Marking Scheme. The assessment of the project portfolio contributes at most 30% to the team mark. A maximum of 40% of the available mark will be allocated for the report. Design documentation attract a maximum of 30% of the available mark each. Testing and screen shot are worth at most 15% each.

Precise guidelines have been set for the length of the report and its structure (see above). Reports that do not adhere to the stated guidelines will not be able to attract a distinction mark. Reports that diverge significantly from the stated guidelines will not be awarded more than 60% of the available mark.

Written feedback will normally be returned through the electronic portal as a file left in each team file space. A blank copy of the feedback form can be found in the folder "Forms" in the electronic portal (see Section 4).

3.4.1 Individual Submission

This is the only document that ALL students registered for a group project module must submit. Due at the end of Week 12 as the final portfolio, this document asks students to provide individual informative feedback on the whole group project experience. It should comprise

- A personal statement of what the student has learnt during the project. Here students should reflect on their experience and say what they think they have learned, both in terms of technical & software-engineering skills and in terms of team-working & inter-personal skills. The statement should be written in simple personal terms, and not simply repeating material from the lectures or from textbooks.

  This document will show what was learnt from the experience of working to complete a team project; learning is possible even when events have not gone according to plan, perhaps even more so than when all goes well.

- A peer group assessment. Using a predefined form, each student should evaluate her/his own contribution and that of the members of the team to the project.

  Typically the feedback from all team members will provide a truthful reflection of the group dynamics, strengths and weaknesses.

The individual submission, which accounts for 20% of the total mark, is used to (further) differentiate the marks of different group members. Through the peer group assessment form each person assigns a score to all other people in his/her team over four different issues: attendance to meetings, contribution, carried out work, worked fairly. Then the sum of the scores obtained by each team member is compared with the team average score and the resulting discrepancies are used to refine each student mark. So, for instance, if we have

Smith, John 21 18 23 20 21

This means that Smith belongs to a group of five people and these people gave him scores of 21, 18, 23 ... respectively. Assuming that the average mark in that group is 23 we then discover that

Mr. Smith has slightly under-performed (his average score is 20.6). This implies that Mr. Smith final mark for the group project will be his average team mark reduced by a few percent points depending on the difference 23 minus 20.6

Typically the individual submissions will result in mark fluctuations in a range of $\pm 15\%$ percentage points.

3.5 Meetings

The fifth component of the group project modules assessment will look at the teams activities. These will be checked by accessing a set of meetings notes that should be made available by the team through VITAL (within the group File Exchange area). This component assesses the general learning outcoumes **LO1**, **LO2**, and **LO3**, as described in Section 1.

Each team is expected to hold regular formal meetings, typically at a rate of one per week. The purpose of these meeting is to report progress, and to plan forthcoming work. One member of the team will be the chair of the meeting, and another member will act as secretary. Different team members may perform these roles in different weeks.

The chair is responsible for producing an agenda for the meeting. The agenda will set out the business to be discussed at the meeting.

The secretary is responsible for producing the meeting minutes. The minutes should not attempt to record all the discussion, but should clearly set out major items worthy of reporting, and the agreed actions.

A typical minute should contain:

- The time, date and place of the meeting.
- Who was present
- Who was chairman and who was secretary
- For each agenda item:
  - What was reported.
  - What was agreed
  - What actions were decided upon and who was allocated the task
  - Date and time of next meeting.

A copy of all meeting minutes must be made available in the group file space within three days of each meeting.

Marking Scheme. The meetings minutes contribute a maximum of 8% to the team mark for this module. A distinction mark will be awarded to a complete set of minutes (as mentioned above this should include material from at least 10 meetings). Incomplete collections, uninformative reports, or documents submitted long after the meetings will be proportionally penalised.

Written feedback on this part of the assessment will be contained in the document providing feedback for the final portfolio.

3.6 Resit

The resitting mechanism for the group project modules is quite peculiar.

Because there is several assessment components, students may choose to resit all of them, or just the ones they have failed.

Typically, all students resitting all components will have to form of a new group and work together during the summer term. The application area must be different from any of the ones the failing students worked during the spring term.

Work Plan. Each resitting team should submit (by email sent to the module coordinator) a one page description of the planned system by the first half of July (exact dates will be communicated in due course to the people involved). A syntetic design document should be submitted, by email, by the end of July. A demo session will be arranged usually during the last week of August and by that date the team will also be asked to submit an overall project report (having identical structure to the one that was required for the regular assessment back in May). Precise dates depend on the resit exams time-table and will be agreed with the module coordinator at the beginning of July.

The resitting arrangements for students resitting only some components will be set up on an individual base soon after the students are notified of the May exams results.

4 VITAL

All material relevant to this group project module is accessible through the VITAL portal:

http://vital.liv.ac.uk/

Students can log in this portal using their University username and password.

Each student registered to one of the Group Project modules will have a corresponding entry in the **Modules** section of VITAL's home page. The main Group Project module web-page can be reached by clicking on the appropriate link (e.g. *201415-COMP208 - GROUP SOFTWARE PROJECT*).

In each case, the project home page shows a number of links under the title "About This Module":

- a link to the module Departmental web-page (for instance, this is

  `https://www.csc.liv.ac.uk/teaching/modules/module.php?code=comp208`

  for COMP208);

- a link labelled "Project Ideas" to a web-page displaying a list of available projects (the purpose of such web-page is to provide additional guidance to the teams in the choice of their project);

- a pointer, labelled "Schedule", to the current plan of lectures, and deadlines relevant to the group project modules.

- a folder, labelled "Forms", containing a blank copy of all feedback forms mentioned in Section 3;

- a pointer, labelled "Reviewer List", to a file describing the full allocation of reviewers to the different assessment reviews.

It also contains links *About This Module* (pointing to the group project home-page as described above), *Module Staff* (giving contact details for all academics involved in the group project modules), *Learning Resources* (pointing to the lecture slides and other material relevant to the module), and the *Group Setting* process (except COMP216) within a panel on the left-hand side. In that same area, there is a link to information relevant to each student's team. This component could be used to store documents relevant to the team's activities. In particular it is expected that the material relevant to the team's meetings (e.g. agendas, and minutes) will be stored in this area. Also, team specific feedback from the various reviews will be submitted in this area (details given in the previous sections).

A lecture during the first week of term will be devoted to explaining the use of the VITAL module environment.

5 Circumstances Affecting Your Work

(Mostly taken from University policies)

> "Students sometimes perform more poorly in assessments (whether examinations or other types of assessments) than their previous performance or achievements would have predicted. Sometimes this poor performance can be attributed, or partially attributed, to particular circumstances beyond the control of the student. These circumstances are described as *mitigating circumstances* if they are accepted in mitigation of the poorer than expected performance. When a Board of Examiners accepts that there have been mitigating circumstances, it will usually not regard the students poorer than expected performance at its face value in making decisions about the students progress in studies or final degree classification.
>
> Mitigating circumstances might include:
>
> - Illness affecting the student.
> - Bereavement.
> - serious illness affecting a close family member.
> - unforeseeable or unpreventable events.
>
> Independent documentary evidence, such as medical certificates, must be provided in all cases to verify mitigating circumstances.
>
> The following will not be regarded as mitigating circumstances:
>
> - Failure to attend an examination due to misreading the examination timetable.
> - Events such as holidays and weddings.
> - Inadequate planning and time management.
> - Having more than one examination on the same day.
> - Examination clashes arising from incorrect registration by the student, i.e. examinations scheduled to take place at the same time. (Students are responsible for reporting any examination clashes which occur in their examination timetable to their departmental Examinations Officer and the Student Administration and Support Division so that alternative arrangements can be made.)
> - Pressures from paid employment. (Students are reminded that the University recommends that they should be employed for no more than 15 hours per week.)
> - Any event that could reasonably have been expected or anticipated."

In the case of the group project modules, like for any other 100% CA modules, the decision about specific cases rests on the module coordinator. Individual cases will be dealt with according to the following rules:

M1. It is recognized that exceptional events can be of two types: *instantaneous* (typically lasting only for a short period of time) or *long lasting*. Such events will also typically be *unforeseen*, but could occasionally be *predictable*.

M2. Typically, predictable events will not lead to mitigating circumstance as their negative consequences on the student progress could often be minimized by appropriate planning. The team-working nature of the group projects also offers additional ways to cope with such exceptional situation.

The module coordinator or the group monitors could be point of contact if any such circumstance arises and will be happy to provide further advice as appropriate in each case.

M3. In the case of unforeseen events, the student is recommended to complete appropriate forms

`http://intranet.csc.liv.ac.uk/department/ltas/AbsenceForm.pdf`

`http://intranet.csc.liv.ac.uk/department/ltas/MitigatingCircumstancesCoursework.pdf`

and return them to the student office along with any appropriate supporting documentation as soon as possible. The module coordinator, once informed by the student office about the case, will decide on the best course of action.

While it is difficult to provide a list of settling actions that could be taken when exceptional circumstances arise it should be noted that:

- because of the particular nature of the group project modules, often short term exceptional circumstances affecting a single member of a team will be compensated by work done by her colleagues.
- no re-run of a review meeting or resubmission of written material can be arranged.
- Compensating actions might include:
  1. waiving the penalty for not attending a review meeting;
  2. waiving a late submission penalty

Finally, note that the defection (or scarce involvement) of any (number of) team member(s) is an unlikely but otherwise predictable event that may affect a team's progress. As such it is NOT a circumstance that needs mitigating in any particular way. The module assessment mechanisms are quite fit to cope similar situations and no case should be made to threat them in any special way.

References

[1] T. Connolly and C. Begg. *Database Solutions*. Addison Wesley, 2003.

Available in electronic format from our library.

.

[2] C. Dawson. *Projects in Computing and Information Systems: A Student Guide*. Addison Wesley, 2009.

Available in electronic format from our library.

.

[3] R. S. Pressman. *Software Engineering: a Practitioner's Approach*. McGraw-Hill Higher Education, 2009.

[4] S. Russell and P. Norvig. *Artificial Intelligence, a Modern Approach*. Pearson Education, 2010.

[5] P. Stevens, C. Larman, and R. Pooley. *Using UML: Software Engineering with Objects and Components*. Addison-Wesley, 2006.

Available in electronic format from our library.

.

Appendix: List of Projects From Previous Years

2014

```
COMP208   Music Search Engine
COMP208   Software application with the ability to take an order for delivery or
            collection in a restaurant
COMP208   To create a web application that students can subscribe to in order to
            enhance their studies.
COMP208   Recipe Database system. Advised to do menus or nutritional personalized
            stuff.
COMP208   Airline Booking System
COMP208   Compile, organise and remotely access their music collection.
COMP208   Payroll System
COMP208   Computer Game Shop Management System
COMP208   TurtleNet: a simple, privacy oriented social network, which demands
            zero security or technical knowledge on behalf of its users
COMP208   Air Traffic Control system
COMP208   Dark Tempus: Live Action Role Play Game
COMP208   Restaurant game. Registered restaurants allow potential customers to
            play a game
COMP208   Internet Dating Site


COMP214   Im Lost! is a competitive casual game and that can also be useful for
            systems which make use of the search algorithm.


COMP215   Bargain Hunter web-site
COMP215   Healthy Menu
COMP215   Home Match (web-site to match housing requirements with flat/house
            availability)
COMP215   Skin test and skincare product suggestion & Management system
COMP215   Photosky (A Photographic DB?)
COMP215   Hot Event register (a-la Liverpool Local)
COMP215   Payroll system
COMP215   SharIT: A Second Hand Online Trade Platform
COMP215   Enterprise Payroll Management and Query System
COMP215   Dream House: a system to collect, store, manage, and control access
            to the data that supports the money collection business for those
            who have achievable dreams but lack of money, and to present each
            transaction of projects that donators make
COMP215   Recycling web-site. Advised to extend to disposing process
COMP215   Travelling Database system
COMP215   Managing event web-site
COMP215   The purpose of the CCM (clothing company management) database system
            is to collect, store, manage, and control access to the data that
            supports the clothing company business for managers, staffs and clients;
            and to promote marketing, improve corporate management and coordinate
            the company information and resource.
COMP215   Premier League Player market value and transfer database
COMP215   Restaurant management  (???)
COMP215   Hotel Booking
COMP215   Windy Film Booking
COMP215   Buy/Delivery Food on-line. Advised to extend to many providers
            and include logistics
COMP215   Cinema Booking System
COMP215   Payroll System
COMP215   Taxi Booking System
COMP215   Discount Voucher System


COMP216   Social Network
```

2013

```
COMP208   an online pub sales and stock control system information system for
            the use of staff and students at the University of Liverpool
COMP208   The project is for a computer configuration system
COMP208   a system to help promote and organise activities for students around
            the University campus
COMP208   Online IT helpdesk
COMP208   A restaurant management system
COMP208   a sports centre booking system
COMP208   a hotel management system
COMP208   a super hero game
COMP208   a text-based adventure game
COMP208   Message Board System
COMP208   player assistant application for the online game called EVE, which
            is a space flight simulation game for a hotel room and event booking system
COMP208   Gym booking system
COMP208   an information system for a garage service department providing support
            for the daily operations of repair and service of motor vehicles.
COMP208   An online recipe system
COMP208   An airline booking system


COMP214   A timetabling system


COMP215   A tour booking system
COMP215   Management of a computer sale system (changed).
COMP215   A hotel reservation system
COMP215   a marketing system, which is meant to help promote an organisations sales
COMP215   Restaurant
COMP215   a system to enable an online marketplace
COMP215   a voucher discount website, this would provide vouchers which people could
            download into their mobile phones and then redeem at a retail outlet in
            the city centre
COMP215   A hotel booking system
COMP215   an online restaurant system
COMP215   a system to help promote and provide ticket sales for a cinema


COMP216   a system to run a student information system for the University,
            this would be to replace services such as Vital and Tulip and the staff
            notes on homepages
```