Course Project: 2012

1 Outline

In this project you will be required to work as part of a software development team that has been commissioned to build and release a software application. Starting with a loose specification, you will need to plan the scope of development for the application, propose a detailed design, implement the design and perform adequate testing of the final product. This must all be performed in groups of three to four with both individual and group contributions.

2 Background

No specific problem will be given and groups will be required to choose and agree on their own problem to solve.

3 Requirements

Using a development platform of your choice and other appropriate tools, you are required to develop a software system. You will be required to formulate the exact specifications yourselves – however, the following general requirements must be satisfied:

- The application must be developed by a group of 4–5 students and each member must make a substantial contribution to the execution of the project.
- You must ensure that the design and scope of your solution to the project can be completed, including documentation, with a commitment of 4.5 hours work per student per week over seven weeks.
- Your solution must provide an intellectual challenge, while at the same time not be so extensive that there is no time to do justice to the application.
- Each student within the group is encouraged to use at least one new development tool, technique or programming language during the project.
- Your solution must contain aspects of information storage, processing, calculation and visualisation.
- Appropriately licensed code from any source may be used as long as it is correctly referenced and acknowledged. However, as a guideline, the completed application should contain between 30–40% original source code.

Specific requirements are:

- The completed application must take advantage of the additional memory / compute power available in a small cluster using `MPI/OpenMP`.
- The completed application must provide a graphical user facing front-end, such as a web interface or mobile application to create, monitor and view *jobs*.
- The application must not depend on a fixed cluster size, and should scalable (up and down) to take advantage of available nodes.
- A source code control system **must** be used during development.
- A modular design must be implemented to allow for extensions to be easily added to improve functionality.

Additional requirements will be discussed during lectures.

4 Arrangements

Students will be required to divide into groups of either four or five of their own choosing. Successful completion of the project will require considerable team work as well as contributions as individuals.

4.1 Deliverables

1. **Requirements and High-level Design Document - Group Report**

    - Each *group* will be required to submit one *concise* report that focuses on the problem definition.
    - Requirement engineering techniques (such as use cases) should be used to elaborate on requirements.
    - An iterative development approach must be followed, consisting of at least two iterations. Requirements must be prioritised such that key features are delivered early.
    - The application must be decomposed into parts that can be developed independently. From the component decomposition, appropriate work units should be allocated to each team member.
    - Where appropriate, suitable interfaces between parts (components) should be defined. No further detail is required for the high level design.
    - A project plan detailing the work units of each team member should be provided.
    - System test cases should be proposed that demonstrate that the system meets specification.
    - Preparation and management of this report will be a group effort.
    - The report must include a page immediately after the front cover that contains a declaration of the contribution made by each member and the number of hours spent on the project thus far.

2. **Low-level Design, Implementation and Testing Report - Individual Report**

    - Each member of the group will also have to submit an individual report (no longer than 12 typed pages) that focuses on the individual's work items completed.
    - The first section of the report should provide an overview of the system and describe the implemented high-level design. A comparison to the initial design should be given, focusing on why deviations occurred.
    - The second section of the report should document the individual's contribution. It should cover aspects of design, implementation and testing. Class and sequence diagrams may be used in the discussion of design.
    - The report must discuss lessons learnt from the project. Specific attention should be given to teamwork, design and implementation challenges and any other noteworthy events.
    - A critical analysis should be provided as to whether the realised software is extensible and maintainable.
    - The report must include an appendix that contains a comprehensive project timesheet. This timesheet must include both a complete breakdown of the original **estimated** time, as well as the **actual** time spent on the project.

3. **Presentation** - Each group will be required to give a *20 minute* demonstration of the example use of their application. After the demonstration, each group member will be required to make a *5 minute* presentation summarising:

    - their role;
    - the work items they designed, implemented and tested;
    - Documentation produced at code level e.g. *Javadoc*, *RDoc*, *Doxygen* ;
    - Noteworthy challenges and lessons learnt.

The completed documentation and presentations must be concise and of a high standard, in accordance with accepted guidelines for report writing and presentation delivering that apply within the School of Electrical and Information Engineering. All submissions must be in strict accordance with the guidelines contained in the Blue Book and the rules contained in the Red Book. No exceptions will be considered.

NOTE: Each group is also required to upload the following to the course `Moodle` page:

- The group report in Portable Document Format (PDF).
- The individual reports as PDF documents.
- The finished application (source code and documentation).

4.2 Components of the Assessment

The course project has three assessment components:

- Group Report
- Individual Report
- Demonstration & Presentation

The form used during the assessment process will be distributed to the students as a separate document.

The course project contributes 35% towards the final course mark.

To map the assessed outcomes to a percentage mark an algorithm is used. The key weighting factors used in the algorithm are summarised in the table below. Note that the specific outcome areas within each component have equal weightings. There is one overriding rule: if any outcome category is rated as *Unacceptable*, then the maximum mark for that component cannot exceed 40% (note, however, that the mark can be **lower** than 40%).

Course Component	**Weights**
Group Report	0.30
Individual Report	0.55
Demo & Presentation	0.15

Outcome Rating	**Weights**
Unacceptable	0.00
Poor	0.25
Acceptable	0.60
Good	0.75
Excellent	1.00

4.3 Submission

The individual report is the course project submission and as such must be submitted by *07h50* on *Monday 21st May 2012*. The documentation and source code must be uploaded to `Moodle` by the same time.

The Schools policy on timely submission of projects and assignments will be enforced. The policy must be read and understood by the student.

All other submission dates will be specified on the course homepage.

5 Important Information

Please note the following important information:

- The cluster will be set up to run on Linux. It is recommended that you attempt some development & your testing using a Linux installation.

- There is no real restriction on the choice of 3rd party API's or design philosophy, other than the fact that the relevant software should be produced under a suitable licence scheme. Any choices made must be fully justified.

- It is recommended that open-source software is used wherever possible during design, implementation and testing. Furthermore, each group should consider releasing their application as open-source under a licence such as the GNU General Public License.

- The front cover of the group and individual reports must contain the following information:
  - Application name;
  - Members names and student numbers.
  - Author(s) of the report.
  - Course name and code.
  - Date of report.

- It is highly recommended that all documentation within a group adheres to a single style and document template.

- To provide comprehensive project timesheets of both estimated and actual time spent, students will obviously have to make use of some sort of time tracking tool or system to successfully meet this requirement.