COM1030 Crossover Project Stage 2: Systems Specification

Your task in this stage is to produce a formal specification for your system. Note that there are some specific requirements for how you must divide this work up between members of your group, as described below.

Format of Stage 2 Document
This document should be set out as follows:

1. **COMMENTS on STAGE 1:** In brief, what do you think of the stage 1 document? (You might even wish to email this section to the previous group! Be critical, but fair).

   SCOPE: The stage 1 document should detail which parts of the overall system should be specified. If you have good reason to disagree with any of this, here is the chance to say why, and what YOUR view of the system is.

2. **DATA SPECIFICATION:** A formal specification, **complete with appropriate explanatory comments**, should be given for all the data that is required within the system (for instance, as indicated by the class model and the accompanying class descriptions). You need to describe the basic types and to define them, where practical, in terms of primitive types and constructors and predicate conditions. You also need to define the INPUT and OUTPUT types together with the MEMORY types that will be used.

3. **OPERATIONAL SPECIFICATION**: Functions and other operators should be defined carefully to ensure that they are completely specified. Each X-machine function will need defining in the correct way using input, output and memory parameters. Where necessay any subsidiary functions need to be defined also.

   Specifications which include no explanatory comments will also be given significantly less credit.

   It is a requirement for this stage that you organise the work so that each member of the group will write some part of the data specification, and as far as possible you should do this so that each person will contribute a roughly equal share of this. In addition to the usual kind of information that should appear in them, the explanatory comments in the specification must make it clear who has written which parts. The practicalities of dividing the work up in this way will be discussed in the next lecture.

4. **SYSTEM SPECIFICATIONS:** An X-machine specification, **again complete with appropriate explanatory comments**, should be given to describe how all the functions that have been defined in the data and operational specification above are used in the machine or machines. There should be an X-machine for each of the main functions that the system is to be able to perform (for instance, as indicated by the use cases). Ideally these specifications should be robust, but ones that are not will be acceptable, although they will not gain as much credit. Also, ideally common dialogues

should be specified as well, but a specification that does not include these will be acceptable, although again it will not gain as much credit. The overall, top level, X-machine should be given – this will describe the user interface and may involve subsidiary X-machines. Make sure that your X-machines are complete.

As with the data specification, it is a requirement for this stage that you organise the work so that each member of the group will write some part of these operation specifications, with each person contributing a roughly equal share, and that the explanatory comments in the specification must make it clear who has written which parts.

5. **LIMITATIONS**: The correspondence between the formal specifications and the requirements should be clearly stated. How much of the system is specified? Where, if anywhere, does the specification need more work? Have any decisions about the design of the system had to be made as part of producing the specification, and if so what? What flexibility for design is left?

   INFORMATION USEFUL for FUTURE STAGES: Did the specification suggest anything worth stating about the design? What test cases suggest themselves?

   OTHER COMMENTS: How satisfied are you with the result? What problems did you have? How (if at all) did you overcome them? What different models did you use and reject? Why?

6. **GROUP ORGANISATION:** The requirements for what is to go into this section of the report are as for stage 1, including the requirement for submission of minutes of group meetings.

   AMOUNT OF WORK FOR THIS STAGE: How much time did you have to spend on the various activities that made up this stage? How would you measure the size of the problem that you had to tackle? How did these measurements of the actual time compare with any estimates that had been made in stage 1? Again, the requirements for what is to go into this section of the report are as for stage 1, including the requirement for submission of timesheets.

   AMOUNT OF WORK FOR THE FUTURE: Given the specification that you have produced, what estimates can you make for the amount of work that might be involved in subsequent stages of the project? How do these compare with any estimates that were made in stage 1?

COM1030 Crossover Project
Marking Scheme for Stage 2: Systems Specification

The instructions for stage 2 of the crossover project required the production of a document containing the following six sections. For each of these sections a grade will be allocated on a three-point scale (poor, average and good), or a five-point scale for the two main sections (ie adding very poor or very good), and an overall mark computed from these, as follows.
All sections average - mark 60
Any sections poor - deduct 4 marks each, and another 4 for very poor (so minimum theoretical mark is 24)
Any sections good - add 4 marks each, and another 4 for very good (so maximum theoretical mark is 96).
Note that the two main sections are assessed on an individual basis, and individuals not contributing may have their marks for any sections reduced, in extreme cases to zero.

1. **COMMENTS on STAGE 1:** this should be assessed along with the next section.
   SCOPE: this should discuss any departures from the stage 1 document's conclusions as to which parts of the overall system should be specified.
   Poor - an unfair, weak or uncritical evaluation of the previous stage.
   Average - some reasonable discussion of the previous stage, but not comprehensive.
   Good - a comprehensive critical review of the previous stage.

2. **DATA SPECIFICATION:** this should give a formal specification for all the data that is required within the system.
   Very Poor - inadequate data model (eg omitting significant classes or no English description), and obvious weaknesses in the mathematics.
   Poor - one of the major faults above.
   Average - data model that matches the class model reasonably well, has some English description and has no obvious major errors (but maybe some minor ones).
   Good - either good data model (as below) or typechecked successfully, but not both.
   Very Good - good data model that matches the class model well and has comprehensive English descriptions, and evidence that the mathematics has been checked thoroughly.

3. **OPERATION SPECIFICATIONS:** this should give a formal specification for each of the main system functions.
   Very Poor - inadequate specification (eg omitting significant use cases or ignoring robustness or with no English description), and obvious weaknesses in the maths.
   Poor - one of the major faults above.
   Average - specification that is largely mathematically correct, has English descriptions to demonstrate that it matches the use cases reasonably well, has made some attempt at robustness, and has no obvious major errors.
   Good - good robust specification in places.

Very Good - good robust specification that matches the use cases well and has comprehensive English descriptions, and evidence that the mathematics has been checked thoroughly.

4. **SYSTEM SPECIFICATIONS:** this should contain a top level X-machine and lower level machines where appropriate.
Very Poor - inadequate machine diagrams – neither the states nor the functions are correctly defined.
Poor - one of the major faults above.
Average – X-machines that are largely complete or largely correct and such that some attempt has been made some at robustness, and have no obvious major errors.
Good - good set of fully defined machines which are largely both complete and robust.
Very Good - good robust set of X-machines specification that matches the use cases well and has comprehensive English descriptions, and evidence that the mathematics has been checked thoroughly.

5. **LIMITATIONS**: this should clearly state the correspondence between the formal specifications and the requirements, indicating where there might be gaps, or where the specification might need more work. It should be assessed along with the next section.
INFORMATION USEFUL for FUTURE STAGES: this should indicate where flexibility for design has been left, or other aspects of the design, and how test cases might be derived from the specification.
Poor - little discussion of these points.
Average - a reasonable discussion of them, but not necessarily very conclusive.
Good - a well-written discussion which reaches sensible conclusions.

6. **OTHER COMMENTS:** this and the following two sections are assessed together.
GROUP ORGANISATION: this should describe how the group set about the task and organised itself, and should include minutes of group meetings.
AMOUNT OF WORK: this should indicate (by providing timesheets) how much time the group had to spend on the various activities that made up this stage (with comparisons with the estimates from the previous stage), and their estimates for the size of the problem that had to be tackled, and for the amount of work that might be involved in subsequent stages of the project (again, with comparisons with the estimates from the previous stage).
Poor - little discussion of these points, or no timesheets.
Average - a reasonable discussion of them, with timesheets or minutes, but not necessarily very conclusive.
Good - a well-written discussion, with both timesheets and minutes, which reaches sensible conclusions.