An Explanation to the Course Project Grading Policy

Concerning the project requirement

There were two main sources of confusion:

1. Storage requirement

   We didn’t stipulate that persistent database is required in the project, but we do require persistent storage for the bug report information, which should not be an unexpected requirement that need to be specified explicitly for a bug reporting system. There are other solutions for persistent storage than using a database, of which we prohibited none. We only require that you can justify your design choice. Here is my answer to the related question on the forum earlier:

   + Persistent storage of the data IS REQUIRED for the project.
   + We don’t constrain which specific technique you should use.
   + However, whichever technique you choose, scalability should be one of your major concerns in making such decision.
   + One most obvious solution for this would be using a database,
   + and Eiffel has provided a library for this purpose: EiffelStore.
   + To know more about EiffelStore, you may find http://docs.eiffel.com/book/solutions/eiffelstore to be helpful,
   + where you may find relevant class references and examples.
   + If you decide to choose other solutions, please be prepared to justify your choice and convince the others both in your report and during the demo.

2. "minimal" set of functions and "maximal" grade

   Since it has been explicitly stated in the project description that the functions listed are the "minimal" set of functions needed in the system, I did not expect someone to understand it as a criterion for "maximal" grade. However, even if you only implement the "minimal" set of functions, you still have a chance to get up to 65 (25+15+15+10) points by doing the other parts well, as explained below.

Concerning the grading policy

The final grade was supposed to consist of two parts, 30 points from final exam and 70 points from course project. The total 70 points of project were actually split into three parts: software system 25-40 points, project report 10-15 points, demonstration 10-15 points. This implies that if you finish the 'minimal' set of functions, hand in the report and participate in the demonstration, you will get 45-55 points. (if you only do the minimum job in all parts, you can just pass) 10 possible points from providing also the software artifacts are actually the “bonus” for good software engineering practice.

Compared with the project description announced in the beginning of semester, which says "Your result would be assessed in the following aspects: software system (40%), artifacts (20%), project report (20%), and demo (20%)", the project report and demonstration weight almost the same, the artifacts part is merged into the software system part under the assumption that a successful system has to be designed somehow – maybe not in a well-organized form – and a bonus is offered for tangible documentations. Therefore, I would say the grading policy is consistent with, and clarifying, the method of assessment described earlier.

The system demonstration is almost as important as the final exam, which I do not think you want to miss, and the deduction of 30 points includes actually a punishment for not doing it, i.e. you will not only lose the 15 points of the demo, but also be subject to a deduction of 15 points from your final grade. For those who have time conflicts, we have also suggested they should “contact the course assistant as soon as possible”, so we can work on that, e.g. choosing another time. Honestly, if you have done the work, I cannot think of anything preventing you from giving the demo.

Admittedly, the Xebra framework does have some problem with it. However, firstly, there are successful examples of finishing the project among you, which shows the project is doable; secondly, this situation has been taken into account in our grading policy so that you will not take all the blames for failing to deliver certain functions. Although the functions required by the "minimal" set are simple enough already, a deduction of only 2 points for an unusable function will be made if the errors are minor or there are reasonable explanations for them. This is also part of the reasons why we restricted that everyone should do the demo. During the demo you can explain your design, and perhaps also errors, so that we can "evaluate your work more properly".