Peer Review Sample

In this paper, I respond to the subMIT system with observations regarding the design and presentation of the system as outlined in the preliminary report. By focusing upon design priorities, central trade-offs, and an evaluation of grading the process, I hope to offer valuable insights to the team regarding their work. The following sections will summarize the system, establish design priorities and important trade-offs, and then look at subMIT's grade handling procedures. It is worth noting upfront that submit identifies its core priorities as simplicity, security, and utility (p. 2).

While the subMIT architecture features the central modules requested by the design specification, it is the file system that first offers a novel intervention to the problem of redesigning the course website. Prioritizing simplicity, the design offers two directories, one for a*ll_students* and for *recitation_X* (pp. 3-4). This two-grouping design allows for individual assignment and class assignment of permissions and content, so long as recitations and tutorials overlap. This overlap of recitation and tutorial is a condition supported by the design spec even though the additional flexibility of edge case semesters where tutorial and recitation might not overlap is perhaps worth considering.

In support of the subMIT system's focus on security, the file system only allows instructor access to their allotted section with the lead lecturer granted access to all sections. This choice does support the security priority of the system, but questions arise about what happens in the event that someone must take over for a lecturer, thus pushing a larger burden on the lead instructor to work around permission issues within subMIT.

The grading component for subMIT features a database for storing assignments and a separate database for storing grades and comments (pp.10-11). The process for returning grades on assignments allows for either a grade or a grade and comment. Key features of the grading process are that instructors have read and write permissions to the tables with their students and that grades can only be published in bulk. In other words, the entire recitation must be published at once (p. 12). Bulk publishing is justified by the priority of simplicity to save users the hassle of posting each grade individually. The reasoning for this decision is clearly stated, though tradeoffs could have been more robustly in the explanation.

In terms of evaluation, one edge case related to correctness that the paper considers is Late Drop cases. In this situation, the system includes a special function for removing late drops from the *all_students* directory (p. 16). This process also revokes read and write permissions for that students' ID within the system. An additional evaluation calculates that the system can robustly handle 400 100kb PDF uploads in about 6.4 seconds. Calculations for larger video or slides submissions were not included, however.

All in all, the system's design choices are stated explicitly and clearly. The properties are regularly used as justifications, and details for how the system is implemented are quite transparent and accessible. Tradeoffs and limitations are less explicit in these design choices, raising questions about what edge cases and concerns were actively considered in the design process. More details about comparative alternatives and evaluations about the limits of the system would have been insightful.