6.033 Spring 2019
DP Final Report Assignment

Due 5/6 at 11:59pm

Having now had two rounds of feedback on your design, it's time to write your final report. Unlike the proposal document, the report should contain enough detail that it could feasibly be turned over to Facilities for implementation. It should also contain an evaluation of your design.

Outline of the Report

Your team's report should be approximately 6000 words[1] and follow the basic outline below:

- **Title page:** Give your report a title that reflects the subject and scope of your project. Include your names, email address, recitation instructor, section time(s), and the date on the title page.
- **Introduction:** Summarize the problem to be solved and what your design is intended to achieve. When summarizing the problem, you should extrapolate and highlight the technical challenges that make this issue a hard systems problem from the design description. Outline your design and briefly outline why your design meets the requirements.
- **System Overview:** Provide a high-level description of your system's modules, behaviors, and innovating techniques or strategies. This should include a system diagram and serve to introduce definitions for key terms used in the Design section. The system overview also provides an opportunity to prioritize the main system objective(s).
- **Design:** Explain your design. Identify your design's main components and workflows. You should sub-divide the design, with corresponding subsections in the text, so that the reader can focus on and understand one piece at a time. Explain why your design makes sense as well as explaining how it works. Use diagrams, pseudo-code, and worked examples as appropriate.

  It should be clear from this section that your design meets the specifications of the assignment (e.g., that it does not exceed the storage available on the servers nor the capacity of the network, etc.). Leave any major calculations to the evaluation section, though it's fine to reference those calculations beforehand (e.g., "Our design results in a communication overhead of fewer than 1Kbit/sec; see Section 3.1 for an analysis.").
- **Evaluation:** Evaluate your design. There are more details about this section below.
- **Conclusion:** Briefly summarize your design, highlighting the novelty or specific focus of your system, and provide recommendations for further actions and a list of any problems that must be resolved before the design can be implemented.
- **Author contributions:** A brief statement (typically 1-3 sentences long) describing the contributions of each author. These contributions could include designing specific components of the system, research or investigation related to the design problem, qualitative or quantitive evaluation, writing the

[1] As always, use this word count as a guideline. If you are writing significantly more than 6000 words, you're giving us too much detail. If you are writing significantly fewer than 6000 words, you are giving us too little. We care much more about the content of your paper than the exact word count.

text of the report, editing the report, creating figures, etc.

- **Acknowledgments and references:** Give credit to individuals whom you consulted in developing your design. Provide a list of references if appropriate.

Evaluation

A good evaluation will do more than just calculate metrics relevant to your system; it will also use calculations to justify design decisions. For example, “Our method for transmitting data fro the server results in an average upload time of two minutes, compared to a design without this method, which results in an average upload time of ten minutes.”

At a minimum, your evaluation section should address the following questions:

- **What is the communication overhead of your system?**

  Typically this will be a measure of the amount of traffic sent between client devices and the server, and Gradescope and the server.

- **On average, how long does it take a student to upload an assignment to the server? In the worst case, how long can this take?**

- **On average, how long does it take for Gradescope grades to be transferred to the server? In the worst case, how long can this take?**

  That is, how long is it from the time a grade is available on Gradescope to the time it’s reflected on the server?

- **How much data are you storing on the server? How long will it take before the server can’t store any additional data?**

- **What parts of your system limit scale, and what are those limits?**

  Could your system handle the addition of 500 new students to 6.033? Could it handle being used by every class in EECS simultaneously?

- **How long does it take your system to deliver all student grades to the Course Lecturer?**

- **How long does it take for a file transfer to be killed, if requested? How long does it take for the user to be notified that the transfer was killed?**

- **How long does it take to create all student accounts at the beginning of the semester?**

- **How usable will users find your system?**

  For example, think about how responsive your system is to particular user activities, and how much tolerance users will have to different functions as they execute them under different circumstances. (As an example, if a student is submitting an assignment an hour before a deadline, they may be more tolerant of a five-minute delay than they would be if they were submitting two minutes before a deadline.)

In answering those questions, you should provide the appropriate numbers as well as some context for them. How do the values that you calculated affect users or other entities in the system? If your system takes X seconds to respond to a failure, is that good or bad? Did you make any trade-offs that involve these metrics? Etc.

Because every system design is different, you may need to discuss additional metrics specific to your system in the evaluation. Your evaluation should also address the use-cases presented in the design project write-up. You may pull those out into their own subsection, or mention them at different points within your evaluation. Whatever structure works best for your report is fine.

In almost all cases, we are looking for you to evaluate your system with respect to the expected and worst cases, not the best case.

You should also consider how your system could evolve. How would it fare if every class at MIT were to use your system? What if storage beyond video or audio was required?

Finally, some design decisions may not correspond to calculable metrics; the most common case is preferring a simple, modular design over a complex one. Your design report should note when you made choices in the name of simplicity or other design principles. Depending on your report organization, it may be more appropriate to include this information in the design section than in the evaluation.