Final Report

While individual advisers have considerable flexibility to structure their project requirements to fit the students and the project, the final report and presentation are required elements for all projects.

The written report must include the following sections. There is no specified page count. This report is your opportunity to document the scope of the project and the work that was done. Fifteen weeks times 2-4 group members is a lot of available time and effort.

A report should always be written to an audience. I find it most useful to consider the audience to be people wanting to pursue a similar project, looking to learn from your experiences and build on what you've done.

Introduction

- Problem summary
- Motivation
- Comparison of existing solutions
- Proposed solution summary

Body of work

- Functional requirements: specify what the system does (mockups, screenshots, user stories)
- Architecture: libraries, servers, data organization and storage, communications and data flow. Emphasize the notable choices you made and justify them based on trade-offs involved.
- Algorithms and data structures: detail core algorithms in your project and the data structure choices that go with them. Emphasize the notable choices you made and justify them based on trade-offs involved. The trade-offs may be computational complexity, but can also be related to security, concurrency, data integrity and other computational concerns.
- Code roadmap: A detailed roadmap for anyone inspecting your code. Explain from the top down how everything is organized (projects, modules, classes, etc) and document the scope of the work.
- Testing and evaluation: describe your systematic approach to verifying that the system meets the requirements (evaluation) and works as intended (testing).

A common mistake in technical writing is to adopt a narrative structure. First we did this, then that, etc. In this section, we only care about the final product. The road to get there is discussed in lessons learned at the end.

Project management

Describe the tools and processes that your team used to:

- Decide what to work on each week
- Assign work to everyone
- Stay on the same page
    - Shared knowledge of everyone's progress
    - Familiarity with everyone's design decisions and code

    - Shared knowledge of obstacles and solutions
- Minimize the effort, redundancy, and inefficiency in combining individual work
- Make adjustments to design and schedule

Summary and self-evaluation

- Summary and evaluation of system (how it did or did not meet the requirements)
    - Changes along the way
    - Evaluation of external dependencies, tools
    - Lessons learned
- Summary and evaluation of the project
    - Project management obstacles, solutions
    - Teamwork
    - Lessons learned