

CMSC B399 Senior Conference - Spring 2023
First and Second Drafts

Overview and Schedule

Your Final Report for this class represents the culmination of your undergraduate Computer Science education and presents a thorough investigation of the design, implementation, and evaluation of a computing-based solution to a real-world problem.

Prior to submitting your Final Report, you will need to submit two drafts to help you flesh out ideas through putting them into words and providing an opportunity for feedback from myself and your classmates.

The due dates for the drafts are as follows:

- **March 17:** First Draft due
- **April 10:** Second Draft due

There are no specific guidelines or requirements regarding the percentage of the Final Report that should be completed in each of the drafts, but the First Draft and Second Draft should contain at least some content for each section described below, and should be considered standalone, “complete” documents with full sentences, paragraphs, etc. as opposed to merely a collection of notes, unformed thoughts, bullet points, etc.

Structure

Your First and Second Drafts should be PDF documents of around 15-20 pages (single-spaced, 12-point font, 1.5 inch margins all around) consisting of the sections described below. You may reuse parts of your Proposal and Extended Abstract where appropriate (indeed, I encourage you to do so).

A separate template will be provided for your Final Report, which will consist of the same sections described here but also some boilerplate content that you do not need for the first two drafts.

Please be sure to use the following section names and numbers in your document! This makes it easier for the reader to follow, and also for your instructor and classmates to discuss.

1. Introduction

Start the document by introducing the problem area or domain that your project will address, or the overall goal of your project.

You can base this section of the Problem Statement/Goals section from your Extended Abstract, and should answer such questions as:

- What is the problem you're trying to solve, or the overall goal of the project?
- Why is this important?
- Why is this difficult?
- Who will benefit from solving this problem or achieving this goal?

CMSC B399 Senior Conference - Spring 2023

First and Second Drafts

Then, briefly (1-2 sentences each) summarize your solution/approach, and the design and implementation of your solution. Don't describe these in too much detail just yet, but rather give the reader a sense of how you plan on solving the problem that you introduced here.

2. Solution (Approach, Features, Requirements)

Describe the solution that you set out to implement in order to solve the problem stated in the Introduction. Note that you should not be focusing on any implementation details just yet, but rather only discussing the features of your solution, i.e. *what* it does, not *how* it does it.

In describing your solution, be sure to mention:

- What are its features/functionality? That is, what does it *do*?
- What is your intuition for why this will be a good solution? Show that it is clearly relevant and appropriate for solving the problem you identified.
- How will it address the things you identified as difficult in your problem statement?

Also, discuss any legal requirements regarding the use of your system and the data and functionality it provides, for instance:

- security and access control
- data privacy and integrity, e.g. adherence to FERPA, HIPAA, etc.
- record keeping and logging for auditing purposes
- user interface accessibility, e.g. adherence to the Americans with Disabilities Act (<https://www.justice.gov/sites/default/files/crt/legacy/2009/02/18/oldsoftware.pdf>)

Likewise, consider any ethical implications of your solution: are there ways in which individuals, or society in general, be put at risk through the use (or misuse) of your solution? If so, how are you going to address or mitigate those issues?

3. Design

Describe the major components/modules/pieces of your solution. Although it may be hard to exactly define what “components” means, think of them as “things that can stand alone but are then combined together to form our solution.” These may be things like:

- user-facing applications, e.g. mobile apps or webapps
- server software that uses APIs and protocols such as HTTP, WebSockets, REST, SOAP, etc.
- data sources such as relational or noSQL databases, file systems
- hardware components

For each of these components, describe their responsibilities are and what technologies it will use. For instance, “... uses a NoSQL database for persistent storage to as it provides easy storage and retrieval while allowing for a flexible record structure that — at least the development of — this project requires.” Also include an explanation or diagram showing how the components will relate to each other and how they will communicate and interact.

Since design is about tradeoffs, you should also answer these questions:

CMSC B399 Senior Conference - Spring 2023

First and Second Drafts

- What other design options did you consider?
- Why did you pick this particular one?
- What makes you think this design is “good” on its own, not just “better than the alternatives”?

Try to demonstrate that you considered other reasonable alternatives and made a conscious decision to move forward with the one you described in the previous section. Of course, in some cases there may not have been a decision to make, e.g. if a particular data source is only available from one web service using one protocol, but for other components for which there are reasonable alternatives, you need to justify your decisions.

4. Implementation

For each component in your solution, describe what its requirements/responsibilities are, what technologies it uses, and an overview of how it is implemented.

In describing the implementation, the main idea is that there should be enough detail so that someone familiar with that area of CS will understand what you did and how you did it, or that if someone wanted to modify it, they would know where to start.

In particular, if you devised algorithms or approaches to implementing particular aspects of your solution, you should describe those here. It is also okay to describe algorithms that you considered and then abandoned, and to explain why those were deemed unsuitable.

Do not list code. Pseudocode for describing algorithms here is fine if it would help explain the details of your solution. If you would like to include actual code, put it in an Appendix, described below.

Be sure to also describe the communication between components, i.e. what data is transferred and what protocols/technologies are used.

5. Evaluation

Describe how you determined that your solution solved the problem that you have identified. Note that this is more than just performing software testing and checking that your implementation conforms to the specification, but rather that it achieves the goals that you set out in the Introduction.

If you performed a user study, be sure to include: Who are the users? What did you ask them to do? How did you collect information? Once you obtained that information, how did you relate it back to your problem statement?

If you did some sort of experiment, e.g. to evaluate an algorithm, be sure to describe: What data did you use? What was your hypothesis? How did you try to validate it? How successful were you?

CMSC B399 Senior Conference - Spring 2023

First and Second Drafts

Use diagrams and other charts as needed in order to demonstrate that you didn't just build a solution: you built a "good" solution to the problem you're trying to address.

Additionally, you should also identify **threats to validity** -- assumptions you've made and other factors that may indicate that your results do *not* actually demonstrate what you've set out to prove -- and describe how you have mitigated them or taken them into consideration.

6. Related Work

Give an overview of related work in this area. This can include other attempts to solve the same problem (or similar problems), or other solutions that are similar to yours but perhaps were targeted at different problems.

This section should convey the idea that you have looked into what other people have tried to do and understand where your solution stands in relation to theirs. In a related work section such as this, however, it is not enough to say, "these people did this; these other people did that" because you need to convince the reader that what you are proposing is in some way *better* than what has been done before.

For each work that you describe, provide a sentence or two explaining why your solution will be *better* than what already exists. It may simply be that other solutions weren't necessarily addressing the exact same problem, or that they tried to tackle some other aspect of the problem, but demonstrate here that you are aware of other work in this space and where your work will be placed within it.

See the "References" section below for information about citing other work.

7. Contributions

Your paper should end with a section that:

- Summarizes your solution and the findings from your evaluation
- Addresses potential future work in solving this problem
- Refers back to the original problem statement and explains how your solution has addressed it in some way
- What are your contributions? Winston in "Make it Clear" argues that the three previous bullet points should be phrased in terms of what your project uniquely contributes.

In describing potential future work, you can explain what you would have done if you had more time, e.g. additional features you would implement or additional evaluation you would have performed, but you can also identify other approaches that someone else could take, or that you perhaps would have taken if you had to do this over again. If you discuss future works, do not make that the last thing you discuss; that ends things on a down note.

If you have significant thoughts on potential future work, you may separate that out into its own section, but be sure that the last section of the paper concludes it by tying everything back to the problem statement from the Introduction.

CMSC B399 Senior Conference - Spring 2023

First and Second Drafts

8. References

You must cite all related work that you refer to in your paper, even for things that are not specifically academic papers. Please use the APA citation format. In particular, these are the types of works you are likely to cite, along with links to the proper way to cite them:

- [Conference proceedings](#)
- [Journal article](#)
- [Website](#)
- [Book \(entire\)](#)
- [Book \(chapter\)](#)

For anything not listed above, find the appropriate format described [here](#).

9. Appendix (optional)

If there is any other information that you would like to include in your paper that does not seem to fit into one of the sections above, you may put it here.

Likewise, you can include information such as code listings, diagrams, lists of data, etc. that may take up too much space within the text in one of the above sections, and/or could be considered supplemental material.

Submission

Submit each draft by email to me (gtowell@brynmawr.edu). In addition, bring a hard copy to class on the due date.