Final Documentation

The final code and documentation for your project must be completed and submitted before 11:59 pm on Thursday Dec. 19th, 2019.

The Goals of the Documentation

- You should write your documentation as though you are explaining your project to a new developer who is about to join your team. You should include enough details so that he or she can get started quickly. On the other hand, the documentation should not overwhelm them with too much information that can be easily discovered by reading the code.

- Your documentation should include pointers to important facts that you wished you had known when you started the project, e.g., limitations of a framework you used or a workaround for a nasty bug you encountered, etc. These are things that people cannot figure out quickly unless it has been documented.

- Read sections 5 and 6 in A Rational Design Process: How and Why to Fake It for more details about code documentation.

Deliverables

Submit the documentation along with your code under the **final-release** tag of your code repository. The documentation should be placed in the root directory of your tagged commit. The documentation must be in PDF format (not Word, not OpenOffice, etc.) and should be named **project_documentation.pdf**. You can include the source (doc, tex, etc.) if you like, but I will only grade the pdf.

Final Documentation

The documentation is worth 20 points. It needs to include the following items (you can use this as a guide for the table of contents):

1. A brief description of your project (make use of HW1) - (2 points)

2. The process followed by your team (make use of your final presentation) - (5 points)
    - Make sure to address the issues of iterative development, refactoring, testing and collaborative development (even if you are not using XP, you have to address these issues in your documentation).
3. Requirements & Specifications of your project (make use of HW2 and HW3) - (5 points)
    - This should include the user stories and use cases which you implemented in your project. You do not need to mention the requirements which were dropped.
4. Architecture & Design (make use of HW2 and HW3) - (5 points)
    - Include multiple UML diagrams to show the important parts of your system (you must have UML diagrams).
    - Describe in a top-down manner the architecture of your system.
    - Include enough details about the design of your system such that anyone who refers to your documentation can understand the major components of your system and how they are related.
    - Describe how the choice of the framework influence the design of your system.
5. Reflections and Lessons Learned - (3 points)
    - Include personal reflections on the project and process by each team member. Describe what you learned.

You should be able to re-use material from the homeworks and your presentations while preparing the documentation.

Formatting Guidelines for Final Documentation

1. Your document must have a table of contents.
2. All class, method names, code must be in monospace font. Use a font like Courier, Monaco, etc.
3. All figures must be labeled.
4. Do not draw the diagrams by hand - use a diagraming tool.
5. All pages must be numbered.
6. Your documentation should follow consistent formatting for sections, headers, etc., for the *entire* document.
7. Your documentation should be **at least** 5 pages but not more than 10 pages.

Failure to follow *any* of the formatting issues will result in severe deductions (possibly resulting in 0).

Please proof read your documentation to avoid spelling mistakes and grammatical errors. Do not write in a colloquial manner. In short, make a *professional* document for submission. Note that it *can* include humor and yet be professional.

Keep your documentation short and to the point. If you are using a framework, you don't need to describe the framework in detail. Instead focus on the big picture. For instance, if you are using a web framework, then you should just explain the *metaphor* behind it (eg., "Web Framework X is based on the Model-View-Controller pattern. Our models follow the YYY pattern to interface with the database. ...").