

Report #3 - Specification & Design – Iteration 2

Report Format

This report collates Reports #1 & #2 into a single document. This report should be self-contained and contain all the information that is relevant to your project. It should be possible to discard all previous reports and read this one alone to obtain all the relevant information about the project. The report should reflect the revisions and additions since the previous reports were submitted.

The report format should follow the formats of the previous two reports.

- Cover Page
- Table of Contents - Ensure that the page numbers listed here are correct
- Summary of Changes - Provide an itemized list of key revisions since the previous two reports. This includes key changes in project objectives, use case descriptions, and system design (e.g., interaction diagrams, class and package diagrams).

The report should contain the following sections:

1. Customer Statement of Requirements (as in Report #1, revised as needed)
2. Glossary of Terms (as in Report #1, revised as needed)
3. System Requirements (as in Report #1, revised as needed)
4. Functional Requirements Specification (as in Report #1, revised as needed) - Elaborate the use cases that will be finished by the time of the final demo. For the use cases that will not be implemented for the final demo, provide a casual description for each and indicate that these could be considered for future work. Update your System Sequence Diagrams to incorporate the use cases that will be completed for the final demo. In this section, you should include the Traceability Matrix that shows how your use cases are related to your system requirements.
5. Effort Estimation using Use Case Points
6. Domain Analysis (as in Report#1, revised to incorporate the use cases that will be completed for the final demo). Include the Traceability Matrix that shows how your use cases map to your domain concepts. Include text description, not only a table with checkmarks.
7. Interaction Diagrams (as in Report #2, revised as needed)
Explain and justify the patterns that you use in your new design. State explicitly in what sense the use of the specific design pattern in the particular interaction diagrams improves the design.
8. Class Diagram and Interface Specification (as in Report#2, revised as needed)
This section must include the Traceability Matrix that shows how your classes are related your domain concepts and a text description of the concepts-to-classes evolution.
9. System Architecture and System Design (as in Report#2, revised as needed)
10. Algorithms and Data Structures (as in Report#2, revised as needed)

11. User Interface Design and Implementation (as in Report#1 and Report#2, revised to incorporate the use cases that will be completed for the final demo)
12. Design of Tests (as in Report#2, revised as needed)
13. History of Work, Current Status, and Future Work - Instead of the section Plan of Work have the section History of Work which documents how the actual milestones and deadlines evolved. Compare these against the milestones as planned in Reports #1 and #2. Also, summarize (as a bulleted list) your key accomplishments in this project. Discuss the possible directions for future work on this project.
14. References (books, papers, URL's of the sources of information and tools used in the project)

Reflective Essay

A minimum 1-page must be included. (see 1.2 for detailed description)

Note: Do not submit separately "revised/corrected" versions of reports #1 and #2. All corrections suggested for those reports should be incorporated in report #3.

Note: If in doubt of whether to include something from reports #1 and #2 since it may be overlapping, then include everything that is not repetitious.

Discuss your diagrams! Describe all design decisions and other things that are not obvious from the diagrams. Any useful information is welcome. There is no limit on the number of pages for the report. Having good comments and explanations greatly helps in reading and evaluating the project and will certainly contribute to your grade.

Also, give exact references and URLs of any material that is used in the project and doesn't originate from the textbook.

Since this report is a complication and revision of the previous two, it is a good idea to address all the issues that were not adequately addressed in the first two reports. Think about what could have been done better in the first two reports and also take into account the comments that were provided to you earlier. **Do not repeat the same mistakes.**

1.1 Report #3 Should Show Project Evolution, Not Only the Final Product

We are interested in the evolution of your project as well as in the final outcome. If you began with certain ideas but now have modified them, you should not simply discard for the third report. First explain how you started, then explain why you moved on to a new design. Why the original design was inadequate? How is the new design better? If you scaled down your initial ambitions, indicate that this can be part of future work for your project. Polish only the ideas that you have implemented for the final submission.

Include both the domain model and the class diagrams in the report. You should know by now that your domain model does not need to exactly correspond to the class diagrams. In fact, it is very likely that they will be different, with class diagrams having many more classes. Moreover, if they do match exactly, that is likely a sign of a bad design.

On the other hand, there should be significant correlation between the two, since otherwise it would imply that the domain model was not used in the design.

The key difference between the domain model and class diagrams is that the domain model shows conceptual organization of your software, while the class diagram contains many implementation details. Therefore, the domain model being more abstract and summarized should be easier to understand. The class diagram contains many details and requires longer examination to understand. It may contain many specific details of your particular implementation of the conceptual domain model. Both have value and both should be included.

Ideally, you should show the first Domain Model that was part of Report#1, then you show the most recent Domain Model that you derived at the end of the semester. Same for the Class Diagram, first the one from Report#2, then the final one.

Provide description how new domain model evolved from the old one, and how the class diagram evolved from the domain model.

1.2 Reflective Essay

Write a reflective essay focusing on the course topics and the design project. This should be a reflection from your personal, individual perspective on how you felt the course met your needs or fell short. You should demonstrate what you have actually done in the course and what you have learned. What was the most difficult aspect of this course in general, and what was the most difficult aspect of the project?

- - Describe the technical challenges you encountered in the development of your software project. What topics do you feel should be covered in a pre-requisite course for this course?
 - Describe how the software engineering techniques you learned in this course helped you to address those challenges
 - Which techniques were the most useful to you and why?
 - Which techniques were the least useful to you and why?
 - What challenges did you experience by working on your project?
 - What benefits you by working on the project?
 - Describe what knowledge you feel might have helped you with the project development
 - Overall, how did the course meet your needs or fall short? Discuss topics covered that helped or did not help in advancing your knowledge of system implementation

Grading

Report Section Points

- - Summary of Changes (5 points)

1.

1. Customer Statement of Requirements (6 points)
2. Glossary of Terms (4 points)
3. System Requirements - See detailed breakdown in Report#1 (6 points)
4. Functional Requirements Specification - See detailed breakdown in Report#1 (30 points)
5. Effort Estimation (4 points)
6. Domain Analysis - See detailed breakdown in Report#1 (25 points)
7. Interaction Diagrams - See detailed breakdown in Report#2 (30 points)
8. Class Diagram and Interface Specification - See detailed breakdown in Report#2 (10 points)
9. System Architecture and System Design - See detailed breakdown in Report#2 (15 points)
10. Algorithms and Data Structures (4 points)
11. User Interface Design and Implementation (11 points)
12. Design of Tests (12 points)
13. History of Work, Current Status, and Future Work (5 points)
14. References (negative points if missing or inadequate) (-5 points)

•

- Reflective Essay (33 points)

Total: 200 points

The grading focus will be on traceability along with the domain, class, interaction, etc., diagrams as well as the program code. To avoid confusion, it is extremely important that the names of the concepts, classes, methods, and attributes are consistent across all diagrams.