Project Report Template: Title

Student Name

Date (varies per deliverable)

CS 225, Fall 2016

Embry-Riddle Aeronautical University

Daytona Beach campus

600 South Clyde Morris Boulevard

Daytona Beach, FL 32114

**Introduction:**

This section should provide a top-level view of what your project is about and any results that you obtained. A good introduction should tell the reader two things as quickly as possible: What did you do and why does anyone care.

For example, this document provides guidance on how to write a final report for this course, and you should care because following these guidelines should help your grade.

The grading rubrics presented in the project assignment describe what your project grade will be based on. The rubric serves as a form of requirement document. This document gives you some suggestions as to how to meet those requirements. Use the format presented in this document and provide the content recommended for each section. Minor variations from this format are allowed to suit your individual project needs.

*[Should be completed by deliverable P2, and edited as needed for future deliveries.]*

**Problem Description:**

This section contains a more detailed description of your project, with more detailed descriptions of *what* your project is trying to achieve. It should not discuss *how* you achieved it; that tends to go in the problem solution sections where you present algorithms and UML diagrams.

In addition to discussing your specific project, you can place your project in context of some larger work. For example, if you are doing a project modelling server response times, or elevator wait times, then your model is a specific example of the more general class of queueing models. There is a lot of literature out there on queueing models. You could mention that the techniques you are using are important because they can also predict some other specific situations. You can also mention that your project has validity because the technique was successfully used by other people (true validity is more complicated than that – you could still have made an error!). If you are doing a game, describe what class of game your game is in and compare to other games in that class. The point is to give the reader some larger context in which your project makes sense.

Lastly, discuss some of the specific technical problems that made your project difficult. What decisions did you have to make? What were the software algorithmic or graphic problems that you had to deal with? The discussion of what problem you had to solve in this section sets up the discussion of the solution in following sections. Try not to mention a problem here without presenting its solution later, and try not to present a solution later to a problem you never told your reader about previously.

When you are turning in your early project deliveries you will tend to write in the future tense. Naturally, in your final delivery everything is past tense. It is acceptable to write in past tense for all deliveries in order to cut down on the editing you have to do at the end of the semester.

*[Should be completed by deliverable P2, and edited as needed for future deliveries.]*

**User Stories and Requirements:**

This section lists your user stories and provides requirements for your software. You should include a few sentences of text describing any terms, if needed, and simply telling the reader what they are reading. This section mirrors the test cases and results section placed further in your report in that your test cases must test the requirements.

If possible, make your requirements non-trivial. Try to make them relate to the actual operation of your important methods as opposed to the more trivial data input type errors. Since this section follows the problem discussion section, it is natural to make the two sections align with each other. Three non-trivial requirements are sufficient.

Helpful URL for writing requirements: <http://www.processimpact.com/articles/qualreqs.html>

*[Should be completed by deliverable P3, and edited as needed for future deliveries.]*

**Problem Solution:**

This section has two main parts. The first part discusses how you produced your final project in terms of algorithmic solutions. The second part presents the UML diagram for your software.

You should present the algorithms for any “interesting” methods or computations that your code performs. This is hard to define with such a broad array of projects being worked on, so you should be thinking about where in your software the most difficult computations are performed. Remember that algorithm discussions have text discussions followed by pseudo-code algorithms.

The UML diagram is a simple pictorial representation. Don’t just plop a figure into a document without discussing it. Give the reader a sentence or two describing what they are looking at. If there is any interesting relationship between your classes, point it out in your text.

*[Should be completed by deliverable P4, and edited as needed for future deliveries.]*

**Test Cases and Results:**

The main part of this section is the table listing the test cases and results. You need more than one test per requirement, though more than three may be redundant for a project of this size. As always, provide some introductory text telling the reader what they are looking at. You may also choose to describe any issues you had meeting the requirements. This could be done as a list of notes referenced in the table. If you do not meet a requirement – say so! Explain what you might do in the future to correct the problem in the conclusions section.

Table 1: Test Cases and Results: Suggested Format

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requirement | Test Case  Identifier | Input | Expected Behavior | Actual  Behavior | Pass? |
| Which requirement  is being  tested? | Numeric identifier.  Decimals can be used to tie test cases to requirements.  For example, test cases 1.1, 1.2, and 1.3 might be three test cases all relating to requirement 1. | Specific values for input. | Specific output value or program actions. | Fill this in after the test is completed. | Y or N  with notes  if needed. |

*[Should be completed by deliverable P4, and edited as needed for future deliveries.]*

**Conclusions:**

This is an optional section that sums everything up. The content is not optional, simply where you choose to put this information. It may make more sense to put your summary of results in the introduction or in the solutions discussion section.

The benefit of having a separate section for conclusions is that it makes them highly visible, so the reader has a sense of what you want them to remember from the report. It also serves as a place for you to put in your “future work” statements about how you would correct or improve on your project in the future.

*[Should be completed by deliverable P7.]*

**References:**

This is an optional section, though you should cite all sources used. Sources might be papers and texts in the general problem domain of your project, code snippets, libraries you incorporated in your project, or even algorithmic solutions to specific parts of your project.

*[Should be completed by deliverable P7.]*

**Appendices:**

This is optional, but you might choose to include external sources or source code.

*[Should be completed by deliverable P7.]*