CS 179J Project Report Information, Winter 2012

It cannot be overemphasized that this class is a design project, not simply an implementation project. In particular, even if you produce a beautiful piece of working software by the end of the quarter, you could still get a failing grade unless you have convinced me (via the contents of your project report, and to a lesser extent, your presentation) that you used a high quality design process to arrive at your solution.

At the end of the quarter each group will hand in a white 3-ring binder, which contains, on the last page, a double CD wallet containing two copies of your archive (more on this later). Please note that you will not get this binder back, so you may wish to make an extra copy for yourself.

The front clear face of the binder must have the following information, in exactly this format. The names are from a past CS 179 offering.

University of California - Riverside
Computer Science and Engineering Department

CS 179J Project: Winter 2012
“Project Title in Quotes”

Instructor: Dr. Philip Brisk

Team: Jones-Smith-Patel

Mike Smith  smith@cs.ucr.edu  (949) 123-4567
Susan Jones  jones@cs.ucr.edu  (949) 987-6543
Anita Patel  patel@cs.ucr.edu  (949) 135-2480
The spine of the binder must have the following information, in exactly this format.

<table>
<thead>
<tr>
<th>CS 179J Project: Winter 2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Project Title in Quotes”</td>
</tr>
<tr>
<td>Team: Smith-Jones-Patel</td>
</tr>
</tbody>
</table>

The first page of the binder should contain an index to the main body, which should contain all the various parts of your project, separated by annotated tab dividers:

<table>
<thead>
<tr>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Description</td>
</tr>
<tr>
<td>Requirements Analysis: First Draft</td>
</tr>
<tr>
<td>Requirements Analysis: Final Draft</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Slides Used for presentation</td>
</tr>
<tr>
<td>CD Wallet (see readme.txt)</td>
</tr>
</tbody>
</table>

All design projects reports must include the following steps:

1. **A Clear Technical Problem Statement**—before proceeding beyond this step, each group should be certain that it has addressed the following questions:
   - Is the problem solvable within the allotted time frame?
   - Does the group have (or can it obtain) the expertise to complete the design, prototype, and testing?
   - Does the group have access to the required software/hardware?
   - Is this a design problem (not just trivial implementation)?
   - Is the project significant enough to be worthy of 4 credits?

It is mandatory that all groups visit the professor in his office in the first two weeks, fully prepared to defend the worthiness of you project with reference to the above. Meetings via skype are also possible.
2. **A Requirements Specification**—this describes quantitatively what problem your group is trying to solve. You will create this by doing a requirements specification, in which the professor will play the role of a customer. Note that while playing this role, the professor will act as a businessman who needs a solution to a problem, not as a computer scientist. It is therefore your group’s responsibility to pull the proper requirements from the professor, and not for the professor to push the proper requirements onto you.

- This exercise is required for all groups, including those who have selected a project suggested by the professor. This is an important learning experience.

- Several iterations may be required. Include all versions in the report. It is important to document the evolution of the project over time.

3. **System Design**—this level of design includes the entire system, including the people and processes involved, and not just the software architecture.

4. **Program Design**—each component of the selected solution approach should be carefully designed.

5. **Construction of a Prototype**—Given the relatively short duration of the class, completion of the project may not be possible; however, you must do your best to produce a working prototype that implements the most important core functionality of the system you have envisions. If graphical output is not possible, create some screen mock-ups on your own.

- **Intermediate Project Reports**—Document your progress. Both intermediate project reports should describe your progress toward the construction of the overall prototype. This section should include two brief summaries that document your project at the time of the intermediate demonstrations.

6. **Implementation**—Here, provide an overview of the high-level code structure of your application. What are the key data structures, modules, etc. Do not provide source code here; it should be an Appendix instead.

7. **Testing**—You are expected to have a detailed test plan, including

- Unit testing
- Integration testing
- Acceptance testing

If you are unfamiliar with these terms, look them up. Search engines are marvelous inventions.
8. **Maintenance Plan**—You are expected to produce a maintenance plan, which states how you plan to keep the software solution current with future environment changes. Here, we assume that you will maintain your project following the 10 week course period, even though we know that this is not really the case.

9. **Engineering Effort and Societal Impacts**—Each project must consider realistic constraints on time and money, and should consider safety, reliability, aesthetics, ethics, and other possible social impacts. Produce a short (1-2 page) statement that describes how your group has addressed these key issue.

10. **Conclusions**—Assume that you write this just before handing in your document. Summarize briefly what you did for you project. Address the following; what most surprised you about the process of creating the project. What would you do differently if you had to do it again?

11. **References**—Every, book, paper, webpage you used must be cited in a standard format. You could use the American Psychological format [1, 2], or the IEEE standard [3], or any other format so long as you are consistent and complete. You should also reference any standard template libraries used in your code.


12. **Appendices**—Include the following:

- Source code printed in the 2 “pages” per page format.
- A printed copy of the slides used for your final presentation. If you use animation in your slides, then you need to “sanitize” the slides so that they are readable when printed..
- A professional quality one-page resume for each member of the group. Use the same template for each resume.

13. **Acknowledgements**—Thank anyone who helped your group with the project.

14. **Two (identical) CD ROMs in a high quality plastic wallet**—Containing source code, test data, all the weekly archives, and a readme.txt file that contains a brief (one line) description of all the files. In recent years, USB sticks have come down to less than $6 for 128mb, and at the same time many machines do not ship with CD burners. So you can replace the CD-roms with USB sticks if you want.