COMP 145 UNC-Chapel Hill
Design Specification

PROJECT TITLE
Submitted to
Prof. Greg Welch

Date

________________________
(signature)  AUTHOR NAME

________________________
(signature)  AUTHOR NAME

________________________
(signature)  AUTHOR NAME

________________________
(signature)  AUTHOR NAME

________________________
(signature)  AUTHOR NAME
PREFACE {LESS THAN 1/2 PAGE}
This should let the reader know that this document is intended to be a formal but primarily an internal (team and Boss) document that lays out the design.

GLOSSARY
Define anything you need to here. Omit if not appropriate/necessary.

1. Introduction {~1/2 page}
* What is the history of the design so far? For example...
* Any prototypes done?
* Are you re-doing an existing design?
* Any “significant” design constraints that you should introduce here?

2. High-Level Design Specification {~1-2 pages including diagrams}
* “Structural model” (s)
* “Control model” diagram(s)

3. Detailed Design Specification {~2-3 pages including diagrams}
* Object model diagrams or similar (include interface)
* Data flow diagrams
* Lower-level “Structural model” (s) ?
* Lower-level “Control model” diagram(s) ?

4. Requirements Traceability {~1/2 page or less}
* Include here an Excel spreadsheet (or some table) that has the Detailed Design module numbers/IDs from Section 3 listed down the rows on the left, the System-Level Requirements numbers/IDs from your Contract (II) listed across the columns, and x’s in the appropriate cells to indicate a connection between the two. This table will serve as a checklist for you so that as you are developing each module (listed down the rows) you can periodically check it against the requirements marked by the x’s to help make sure you are not corverting something.
* The matrix should be targeted at the module level, so will likely have many more entries than I have shown in the lame example below. As always, this is a tool for you, not something to please me (although I will be pleased to see you learning to use such tools!) so make it as detailed as you think will be useful to you as a checklist. If you have done this right, you will be able to code in comfort, knowing that you are not forgetting anything.
5. Schedule {~4 pages including diagrams}
Here is yet another opportunity for you to assess the realism of your goals. You should update your Contract II schedule as appropriate to reflect the work on specific systems (or major sub-systems) including the name of the person responsible. Here you can now schedule at the design level, including as tasks major system development, etc.

5.1 Development Milestones {~1 page}
Try and update your milestones to include the internal delivery or certification of some major modules. The milestone could be marked by a planned vote among team members to certify that a particular module is functioning “good enough” to go on. The approval resulting from that vote would be a good milestone.

5.2 Gantt Chart
An updated Gantt chart w / {~1 page}
Include new certification (vote “OK”) milestones

5.3 Risk Analysis and Management {~1/2 page}
5.3.1. Identified Schedule Risks
* What specific schedule tasks/items are you concerned about?

5.3.2. Plans
* How are you addressing that concern?
Avoidance or assignment
Minimization or acceptance (w/ relevant contingency plans)

APPENDIX A
* Include appendices if necessary. For example....
* Any relevant design/structure documents/pictures from a previous system?
* No need (in our case) to include previous deliverable documents. Please do not.