CIS 492 - Project Handout - Spring 2012
Version 1.1 - January 19, 2012

The core of CIS 492 is a semester-long team project modeling, designing, and implementing a working, demonstrable prototype database application for a scenario.

You have already been assigned to a team, and you already know your team's project scenario.

Note that peer evaluations will be required at several points during the semester. Your thoughtful participation in these is also part of your project grade.

Note, too, that the final version of your project will not be returned when the semester is over --- it will be retained by the department for departmental assessment purposes. You will be provided with a binder and dividers for your project notebook. You are welcome to return these and use better ones that you provide if you prefer -- I am quite willing to return such binders and/or dividers to you after the semester.

The project milestones are described on the following pages. These may be subject to change -- if any do change, either a new version of the project handout will be posted with an updated version number and date or dated project handout addendums will be posted.

Minimum Project Requirements

- It must involve a modeled, designed, and implemented database, implemented using the Oracle DBMS on the HSU Oracle `student` database.
- This database must have at least a somewhat-interesting structure, involving multiple distinct, significant entities and multiple distinct, significant relationships.
- On the database server side, some appropriate PL/SQL stored procedures and/or stored functions and/or triggers are expected to be implemented. A lack of such code will affect the project grade. (You need to show that you learned about these in CIS 318.)
- There is some latitude on what you use to build the applications atop your database -- you may certainly use some combination of software to provide these. However, the teams are expected to make these choices from among software currently available on campus, to keep the "playing field" fair for all teams, for a reasonable opportunity for oversight by the instructor, and as a slight example of the kinds of restrictions one sometimes encounters even in "real-world" projects.
- Web applications using your project database as a sink and source of data and with reasonably-sophisticated graphical user interfaces (GUI's) are expected.
  - These should be attractively-designed prototypes with at least a moderate degree of functionality, designed with ease of use in mind.
  - Use of valid XHTML and CSS is expected; the teams can decide whether they wish to include use of JavaScript or other client-side technologies as well. Your XHTML and CSS should be valid according to the W3C Markup Validation services.
  - External CSS style sheets should be used; use of internal style sheets and inline styles should be

minimized.

  - In the spirit of Enterprise Information Portals, appropriate use of sessions is expected. Once a user has logged in, he/she should not have to re-log in again to perform actions before having logged out.

- Security should be considered as well in your prototype; you should attempt to defend against cross-site scripting and SQL injection attacks, for example.
- An appropriate report generation capability (separate from the project forms!) is expected to be included as part of the project.

Grade breakdown:

The total project grade is the sum of the following:

Max Pts Possible	Project part	Description
10.0	Milestone 1	Database model, initial business rules, and preliminary schedule
7.5	Milestone 2	User stories and first-version acceptance tests
5.0	Milestone 3	First formal team presentation
7.5	Milestone 4	Database design (schema)
5.0	Milestone 5	Project coding progress report
5.0	Milestone 6	Project reports progress report
30.0	Milestone 7	Final version of project
5.0	Milestone 8	Formal presentation/demonstration to class and to other observers
5.0	Milestone 9	Individual reflection paper
5.0	Participation 1	Based on 1st peer evaluations, instructor's evaluation of performance
5.0	Participation 2	Based on 2nd peer evaluations, instructor's evaluation of performance
10.0	Participation 3	Based on final peer evaluations, instructor's evaluation of performance

The grades for Milestone 9, Participation 1, Participation 2, and Participation 3 are individual grades, and may differ for each team members. Except for unusual circumstances, the grades for the other milestones will be the same for each member of a team (although I reserve the option of giving different members of a team different grades for any project part, if I feel this is appropriate).

The sum of the grades for the above project components will then be multiplied by 50% in computing your CIS 492 final semester grade.

Milestone 1 - Database model, initial business rules, and preliminary schedule

Due:

Thursday, February 2, beginning of class

Turn in:

Your project binder with 10 dividers containing:

- labels on its spine **and** on its front including at least the team number and "Spring 2012"
- a cover sheet placed inside the binder at its very front, including:
  - (optional) the name of the team, if desired
  - the names of the team members
  - CIS 492, and Spring 2012
- after the first divider, labeled **Description**:
  - place the approved revised scenario description that I will supply for you to include here
- after the second divider, labeled **Schedule**:
  - place your initial Gantt chart depicting your initial estimated project schedule
  - include **at least** the following task categories and subcategories (you may also add others, if you would like):
    - Design - must include separate estimates for **each** of the following design subcategories:
      - design of the database
      - design of the application interfaces
      - design of reports
      - design of project code
    - Implementation - must include separate estimates for **each** of the following design subcategories:
      - implementation of the database
      - implementation of application interfaces
      - implementation of reports
      - implementation of project code
    - Testing - must include separate estimates for **each** of the following design subcategories:

    - testing of the database
    - testing of application interfaces
    - testing of reports
    - testing of project code
  - Documentation
- after the third divider, labeled **DB Model**:
  - place your initial database model for your project, expressed as an Entity-Relationship diagram, making sure that it includes a clearly-visible last-modified date.
  - (Note: required ERD format standards will be posted on the CIS 492 public web page.)
- after the fourth divider, labeled **Business Rules**:
  - place your initial set of business rules for your project, including a last-modified date.
  - While these will change over the course of later milestones, a reasonably thorough initial attempt at such rules is expected.
  - Optionally, you may also include a **separate** list of database/system assumptions within this section, for those system assumptions (database or programming details) that are not truly business rules (that are not operational, day-to-day rules within the project scenario). You may **not** include such database/system assumptions within your set of business rules.
- after the tenth divider, labeled **Partitioning**:
  - at the end of this handout (and eventually posted on the public course web page), you will find:
    - a team-meeting form that must be filled out for each team meeting, and
    - a milestone partitioning report form that must be filled out for **each** project milestone (except for Milestone 10)
  - place your milestone partitioning report for Milestone 1 followed by your team-meeting forms so far, with the team-meeting forms in reverse chronological order, most recent team-meeting form first.

Milestone 2 - User stories and first-version acceptance tests

Due:

Thursday, February 16, beginning of class

Turn in:

Your project binder with 10 dividers containing the previous milestone's contents, along with:

- after the **Schedule** divider:
  - first, place an **updated** Gantt chart, which is quite similar to the original, except:
    - it reflects any required changes from comments about the original,
    - it now **also** depicts "actual" work done since the date of the first Gantt chart **along with** the planned estimates, and
    - it has an appropriate Last modified/printed: date.
  - followed by the original version.
  - (Note: do not remove previous milestones' contents -- revised versions of project components should be placed in front of the previous versions within the appropriate project binder section.)
- after the **DB Model** divider:
  - IF you have made any revisions to your database model since the previous milestone, THEN place that revised model in the front of this section, clearly marked with a last-modified date,
  - followed by the comments sheet that your team received for your original database model,
  - followed by your original database model.
- after the **Business Rules** divider:
  - first, place the current version of your set of business rules, clearly marked with a last-modified date,
  - followed by your original set of business rules.
  - (It is expected that your business rules should change during the process of model correction and design, as you become more familiar with your project scenario.)
- after the sixth divider, labeled **User Stories/Acc Tests**:
  - I'm assuming that each user story consists of a short name, which category (or categories) of users it is a story for, and a short description of that user story.
  - So: the first thing in this section should be an index of user stories, listing just the user story name and category(ies) of user of that user story. Make sure that this index is clearly marked with a last-modified date (since you may be adding user stories later).

  - (the idea here is to give me some reference point when I am looking at this section at the remaining project milestones -- hopefully it will be a useful reference for you as well.)
- follow that index with your user stories -- since index cards would not fit so well in a binder, put each user story on its own page (including its title, category(ies) of user(s), and short description). It should not fill the page, note!
- Finally, each user story should be followed by your current version of acceptance tests for that story (how you will test that that user story has been satisfied).
  - The acceptance tests for each user story are indeed initially written before the user story is implemented, to get a little Extreme Programming flavor in here (coming up with tests before writing code)
  - The number of acceptance tests per user story will vary, depending on the particular user story involved;
  - THE FORMAT OF THESE MAY CHANGE -- but if it doesn't, here's what we'll go with:
    - start with a statement of what is being tested (what is the goal of this acceptance test? what behavior, of what feature, in what situation, are you concerned about in this acceptance test within this user story?)
    - then include a description of the specific test data and situation for this acceptance test (note: you can use a sketch of a screen/form showing what should be typed/entered where, if that would make this specific description easier)
      - NOTE: do NOT just say "enter a fake name" or other such vague directions. The idea here is to create something that anyone, even a non-team-member, could use to run this acceptance test. This leads to the kind of acceptance tests that can be re-run as many times as needed/desired (because ideally one should rerun all acceptance tests whenever a change to the system is made...)
    - Then include a description of the **expected** behavior, **before** doing the test
    - Include a blank/area that can be filled in later with the dates that this acceptance test is actually attempted
    - Include a blank/area that can be filled in later with the **actual** behavior encountered in this test attempted on that date
    - Include a blank/area/checkbox/something that can be marked to indicate whether this acceptance test attempted on that date is considered to have **passed** or **failed**.

- after the **Partitioning** divider:
  - first, place your milestone partitioning report for Milestone 2,
  - followed by your team meeting forms for meetings that took place between the time that Milestone 1 was submitted and Milestone 2 was submitted, in reverse chronological order (most recent team-meeting form first),
  - followed by the milestone partitioning reports and team meeting forms from Milestone 1.

Milestone 3 - First formal team presentation

To be presented:

Tuesday, February 28, during class

Turn in (before presenting):

- (Note: you do *not* need to turn in your project binder for this milestone.)
- a typed outline of your group's presentation
- print-outs of any slides, illustrations, etc. used in your presentation to the class that are not directly from the project binder
  - black-and-white printouts with 4-6 slides per page will be fine for meeting this requirement
  - (if something being presented is directly from the project binder, indicate within your presentation outline what was shown at that point in the presentation, and from which project binder section it came, rather than printing out an additional copy of it)

Presentation purpose:

The purpose of this presentation is to:

- add to your experience in giving a formal team presentation
- present the **initial** user stories that you plan to implement first, (selected to implement first from those you included in your project binder's Section 6, **User Stories/Acc Tests**), along with displaying at least a few application interface "screen" prototypes/mock-ups related to these user stories
- solicit comments from the class about the user stories selected to be implemented first and those "screen" design prototypes related to those user stories

Presentation guidelines:

- everyone on the team should participate significantly and roughly equally in this presentation!
- each group should plan to take **20** minutes.
  - It is expected that you will have planned, organized, practiced, etc. enough so that your presentation will be neither shorter nor much longer than this. If the presentation takes **less than 20 minutes** or takes **more than 23 minutes**, there will be a **time penalty**.
  - A 5-minute question-and-answer period will follow the 20-minute presentation.
- a formal **overview** bulleted-list of the presentation's main parts should **start** the presentation, and it should be both displayed and verbally summarized at the presentation's beginning.
- you should briefly introduce/describe your project to the class before proceeding with discussion of the initial user stories that you will implement and display of the some related application interface

"screen" designs related to those user stories

- make sure that, however you choose to display your mockups, they are easily visible to the viewing audience!
- it is expected that you will carefully plan and organize your presentation -- for example:
  - each presenter should introduce the next presenter ("Now Sally will discuss the mock-ups for the widget-adjusting user story...")
  - each presenter should avoid reading directly from notes and instead make eye contact with the audience (each should have practiced his/her part so that he/she can just glance at notes while presenting rather than reading them word-for-word)
  - each presenter should know what he/she plans to do next at any given point, as should the overall team
    - (that is, avoid having to discuss, during the presentation, what should be done next -- everyone should already know that from having planned and practiced the presentation beforehand)
  - one team member should operate (or "drive") the computer while another actually speaks/presents, and this "driver" should also know what to do next at any given point (he/she should not have to ask, and the speaker/presenter should not have to tell him/her what to do); this also requires planning and coordination in advance.
- you will use the BSS 308 classroom projector; a laptop can be connected to this projector, if you would like. It is the team's responsibility to make sure in advance that they can successfully project their application interface "screen" designs and slides; if you need to make special arrangements, do so well in advance of the presentation!
  - I am happy to let you into BSS 308 to practice in advance, subject to that room's availability, of course.

Milestone 4 - Database design (schema)

Due:

Thursday, March 8, beginning of class

Submit:

Using `~st10/492submit` on nrs-projects, submit your SQL script containing the SQL create-table-statements version of your design (as described below) using a homework number of `4`.

Turn in:

Your project binder with 10 dividers containing the previous milestone's contents, along with:

- after the **Schedule** divider:
  - first, an **updated** Gantt chart, which now also depicts the actual work done since Milestone **2**
  - followed by the previous versions
- after the **DB Model** divider:
  - IF you have made any revisions to your database model since the previous milestone, THEN place that revised model in the front of this section, clearly marked with a last-modified date,
  - followed by the previous contents of this section.
- after the **Business Rules** divider:
  - first, place the current version of your set of business rules, clearly marked with a last-modified date,
  - followed by your previous sets of business rules.
  - (It is expected that your business rules should change during the process of model correction and design, as well as from designing your application "screens", as you become more familiar with your project scenario.)
- after the fifth divider, labeled **DB Design**:
  - place your initial (logical) database design (database schema) for your project.
  - For our purposes, the database design will consist of two parts:
    - first, depict your design in **relation-structure** form, making sure the primary key attributes are noticeably depicted for each table, and using SQL foreign key notation to indicate foreign keys; (be sure to ASK me about this if you aren't sure what I mean by this)
    - then, to add in some domain information, depict your design in the form of nicely-formatted SQL create-table statements, meeting the following additional specifications:
      - include neat comments before each create-table statement describing that table's purpose,

and including information about any attribute whose meaning is not immediately clear from its name

  - primary keys must be explicitly defined for each table
  - foreign keys, as needed, must be explicitly defined for all relationships between tables; these should be expressed as table constraints rather than column constraints, and should be within the create-table statement unless there are unusual circumstances.

- Note that these create-table statements, then, will make clear the structure of each table, its attributes, at least rough partial domains for each attribute, and the relationships between the tables. This information, plus the business rules in their section, constitute a reasonably-complete database design for our purposes.

- after the **User Stories/Acc Tests** divider:
  - include its previous contents,
    - modifying the initial index if user stories have been added, removed, or significantly modified,
    - adding/removing/modifying the user story pages accordingly, and
    - modifying and filling in acceptance tests pages as you attempt acceptance tests for different user stories
  - (it is expected that you will continue work implementing and testing user stories until all are complete or until Project Milestone 7, whichever comes first...)
- after the **Partitioning** divider:
  - first, place your milestone partitioning report for Milestone 4,
  - followed by your team meeting forms for meetings that took place between the time that Milestone 2 was submitted and Milestone 4 was submitted, in reverse chronological order (most recent team-meeting form first),
  - followed by the milestone partitioning reports and team meeting forms from the previous milestones.

Milestone 5 - Project coding progress report

Due:

Thursday, March 29, beginning of class

Submit:

Using `~st10/492submit` on nrs-projects, submit all of the database-server-side, client-side, and application-server side code you have started and/or completed so far, using a homework number of `5`.

Turn in:

Your project binder with 10 dividers containing the previous milestone's contents, along with:

- after the **Schedule** divider:
  - first, an **updated** Gantt chart, which now also depicts the actual work done since Milestone 4
  - followed by the previous versions
- after the **DB Model** divider:
  - IF you have made any revisions to your database model since the previous milestone, THEN place that revised model in the front of this section, clearly marked with a last-modified date,
  - followed by the previous contents of this section.
- after the **Business Rules** divider:
  - IF your business rules have changed since the previous milestone, THEN place that revised set in front of this section, clearly marked with a last-modified date,
  - followed by your previous sets of business rules.
- after the **DB Design** divider:
  - IF you have made any revisions to your database design since Milestone 4, THEN place that revised design in the front of this section, clearly marked with a last-modified date,
  - followed by the previous contents of this section.
- after the **User Stories/Acc Tests** divider:
  - include its previous contents,
    - modifying the initial index if user stories have been added, removed, or significantly modified,
    - adding/removing/modifying the user story pages accordingly, and
    - modifying and filling in acceptance tests pages as you attempt acceptance tests for different user stories

  - (it is expected that you will continue work implementing and testing user stories until all are complete or until Project Milestone 7, whichever comes first...)

- after the seventh divider, labeled **Coding**:
  - this section begins with an index page, clearly marked with a last-modified date, listing **all** code that is planned, started, and/or completed so far. Separate the code into database-server-side, client-side, and application-server side, within each category organizing the code within that category in a coherent fashion for your particular project.
    - for each item in each category, indicate whether it is 1) not yet begun, 2) started but not yet completed/tested, or 3) completed and tested
    - remember that it is required that all projects include significant appropriate DATABASE-server-side code/triggers/stored procedures/stored functions, in addition to client-side and application-server side code -- these were important aspects within CIS 318, and I expect to see signs that you are making use of this knowledge from that course here
  - this index page is followed by printouts of **just** the five most significant/interesting "chunks" (programs, procedures, functions, etc.) in each of the database-server-side, client-side, and application-server-side levels; if necessary to meet this requirement, include stubs for proposed code yet to be written.
    - make sure to include examples from each language used within your project, to represent all of the code you have written in that particular language
    - (you will electronically submit all of your code so far, including stubs so far, as noted at the beginning of this milestone; for the notebook, however, you will only print out this representative sample of that code)
  - each code "chunk" should be prominently labelled at the top of the page on which it begins
- after the **Partitioning** divider:
  - first, place your milestone partitioning report for Milestone 5,
  - followed by your team meeting forms for meetings that took place between the time that Milestone 4 was submitted and Milestone 5 was submitted, in reverse chronological order (most recent team-meeting form first),
  - followed by the milestone partitioning reports and team meeting forms from the previous milestones.

Milestone 6 - Project reports progress report

Due:

Thursday, April 12, beginning of class

Turn in:

Your project binder with 10 dividers containing the previous milestone's contents, along with:

- after the **Schedule** divider:
  - first, an **updated** Gantt chart, which now also depicts the actual work done since Milestone 5
  - followed by the previous versions
- after the **DB Model** divider:
  - IF you have made any revisions to your database model since the previous milestone, THEN place that revised model in the front of this section, clearly marked with a last-modified date,
  - followed by the previous contents of this section.
- after the **Business Rules** divider:
  - IF your business rules have changed since the previous milestone, THEN place that revised set in front of this section, clearly marked with a last-modified date,
  - followed by your previous sets of business rules.
- after the **DB Design** divider:
  - IF you have made any revisions to your database design since the previous milestone, THEN place that revised design in the front of this section, clearly marked with a last-modified date,
  - followed by the previous contents of this section.
- after the **User Stories/Acc Tests** divider:
  - include its previous contents,
    - modifying the initial index if user stories have been added, removed, or significantly modified,
    - adding/removing/modifying the user story pages accordingly, and
    - modifying and filling in acceptance tests pages as you attempt acceptance tests for different user stories
  - (it is expected that you will continue work implementing and testing user stories until all are complete or until Project Milestone 7, whichever comes first...)
- after the eighth divider, labeled **Reports**:
  - begin with a brief description, clearly marked with a last-modified date, of **how** users will

generate reports in your project, followed by a list of the kinds of reports implemented/envisioned;

- follow this with printouts and/or sketches of your implemented and/or proposed report designs thus far.
- note that reports are distinct from application interface "screens" or forms --- for example:
  - there is a higher expectation for good formatting (of both numbers and text),
  - their rows ought to be logically and explicitly ordered,
  - they are not interactive,
  - they need to be readable on their own by even a non-user of your project (which implies explicit titles, well-labeled columns, etc.), and
  - they should be printable.
- even if not yet implemented, it is expected that, by this stage, you have a mostly-complete set of detailed sketches of reports that you plan to include, some of which will have been implemented (and those implemented should have example reports printed out).

- after the **Partitioning** divider:
  - first, place your milestone partitioning report for Milestone 6,
  - followed by your team meeting forms for meetings that took place between the time that Milestone 5 was submitted and Milestone 6 was submitted, in reverse chronological order (most recent team-meeting form first),
  - followed by the milestone partitioning reports and team meeting forms from the previous milestones.

Milestone 7 - Final version of project

Due:

Thursday, May 3, beginning of class

Submit:

Using `~st10/492submit` on nrs-projects, submit the final versions of all of the code and documents for your project using a homework number of `7`.

Turn in:

Your project binder with 10 dividers containing the previous milestone's contents, along with:

- after the **Schedule** divider:
  - first, an **updated** Gantt chart, which now also depicts the actual work done since Milestone 6
  - followed by the previous versions
- after the **DB Model** divider:
  - IF you have made any revisions to your database model since the previous milestone, THEN place that revised model in the front of this section, clearly marked with a last-modified date,
  - followed by the previous contents of this section.
- after the **Business Rules** divider:
  - IF your business rules have changed since the previous milestone, THEN place that revised set in front of this section, clearly marked with a last-modified date,
  - followed by your previous sets of business rules.
- after the **DB Design** divider:
  - IF you have made any revisions to your database design since the previous milestone, THEN place that revised design in the front of this section, clearly marked with a last-modified date,
  - followed by the previous contents of this section.
- after the **User Stories/Acc Tests** divider:
  - include its previous contents,
    - modifying the initial index if user stories have been added, removed, or significantly modified,
    - adding/removing/modifying the user story pages accordingly, and
    - modifying and filling in acceptance tests pages as you attempt acceptance tests for different user stories

- after the **Coding** divider:
  - include the **final**, **dated** index listing **all** the code included in this final/latest milestone, separated into database-server-side, client--side, and application-side, within each section organizing the code within that section in a coherent fashion for your particular project.
    - for each item in each category, indicate whether it is 1) not yet begun, 2) started but not yet completed/tested, or 3) completed and tested
  - this index page is followed by printouts of **just** the final versions of the five most significant/interesting "chunks" (programs, procedures, functions, etc.) in each of the database-server-side, client-side, and application-server-side levels at this final milestone;
    - make sure to include examples from each language used within your project, to represent all of the code you have written in that particular language
    - (you will electronically submit all of your code, as noted at the beginning of this milestone; for the notebook, however, you will only print out this representative sample of that code)
    - each code "chunk" should be prominently labelled at the top of the page on which it begins
  - followed by the previous contents of this section.
- after the **Reports** divider:
  - include a **final**, **dated** brief description of how users generate reports in your final prototype, followed by a list of the kinds of reports implemented
  - the above is then followed by printouts of the final report layouts/sample reports
  - followed by the previous contents of this section.
- after the ninth divider, labeled **User Doc**:
  - the purpose of this section is to include specific and general information for would-be users of your final, completed prototype.
  - the **first** item in this section should be detailed instructions of where to find the final prototype and all related code and files, how to build/create it, and how to run it
    - this part should include a README file that lists all of the files included in the prototype and the purpose for each
  - the **second** item (which must be clearly **separate** from the first!) should be the more general user documentation for your final prototype:
    - you may assume a computer-savvy audience for this portion --- but you must also assume that they are using your project for the first time. What would they need to know?
    - note that careful use of screen dumps within this documentation is **expected**, and greatly strengthens it, in general.
- after the **Partitioning** divider:
  - **first**, place a **final report** discussing **how** you partitioned the work involved in the project, and including discussion of:

  - **how well** the group's techniques of partitioning **did** and **did not** work
  - if the partitioning was easily accomplished or not,
  - any interesting means your group used to attack this problem, or to solve problems with regard to partitioning
- **then**, place your milestone partitioning report for Milestone 7,
- followed by your team meeting forms for meetings that took place between the time that Milestone 6 was submitted and Milestone 7 was submitted, in reverse chronological order (most recent team-meeting form first),
- followed by the milestone partitioning reports and team meeting forms from the previous milestones.

Milestone 8 - Formal presentation/demonstration to class and to other observers

To be presented:

Thursday, May 3, during class

Turn in (before presenting):

- (Note: you will just have turned in Milestone 7, including your project binder, before this)
- a typed outline of your group's presentation
- print-outs of any slides, illustrations, etc. used in your presentation to the class that are not directly from the project binder
  - black-and-white printouts with 4-6 slides per page will be fine for meeting this requirement
  - (if something being presented is directly from the project binder, indicate within your presentation outline what was shown at that point in the presentation, and from which project binder section it came, rather than printing out an additional copy of it)

Presentation purpose:

The purpose of this presentation is to:

- add to your experience in giving a formal team presentation, especially one demonstrating a software prototype
- give a final report to the class (and to other audience members) about your final project prototype
- demonstrate your prototype in action -- to demonstrate its strengths, to show off its capabilities!

Presentation guidelines:

- everyone on the team should participate significantly and roughly equally in this presentation!
- each group should plan to take **30** minutes.
  - It is expected that you will have planned, organized, practiced, etc. enough so that your presentation will be neither shorter nor much longer than this. If the presentation takes **less than 30 minutes** or takes **more than 34 minutes**, there will be a **time penalty**.
  - A 5-minute question-and-answer period will follow the 30-minute presentation.
- a formal **overview** bulleted-list of the presentation's main parts should **start** the presentation, and it should be both displayed and verbally summarized at the presentation's beginning.
- there may be guests from outside of the class at these final presentations; for them, as well as for formality, then, you should briefly introduce/describe your project, and then briefly discuss the

"history" of your project;

  - you should briefly discuss interesting problems encountered, how you dealt with them, etc.
  - discussions of general group-project issues your group encountered are also encouraged --- in fact, it would be remiss to omit them at this stage.
  - and, ties you have seen/experienced to topics discussed in the in-class discussions are encouraged.
  - BUT be careful not to take too long for this part, because:

- ...don't forget that the **most important purpose** of this presentation is to **demonstrate** your prototype, "**live**", actually showing it in operation; **this should be the heart and the main focus of your presentation.**
  - demonstrate its strengths --- show off its capabilities!
  - you are expected to demonstrate features "live" rather than to simply talk about them
  - it is expected that you will carefully **plan** how you will demonstrate your project's capabilities (you know the order of demonstration of the different features, for example, and you have sample data already planned and ready to demonstrate;
    - ...you are **not** discussing within the team what to present next during the presentation, or what sample data to use!)
  - if at all possible, you should strengthen your demonstration by organizing it so that information added in earlier examples shows up in later examples (proving to at least some degree that the prototype actually functions), being sure to explicitly **point out** such information along the way
    - ("...and here you see the value of the just-adjusted widget in this report...")
- the presentation should be closed with a brief discussion of "future work" -- what would you work on next, in this prototype, if you had more time?
- as implied above, it is expected that you will carefully plan and organize your presentation -- for example:
  - as before, each presenter should introduce the next presenter ("Now Sally will demonstrate the widget-adjusting portion of our application...")
  - as before, each presenter should avoid reading directly from notes and instead make eye contact with the audience (each should have practiced his/her part so that he/she can just glance at notes while presenting rather than reading them word-for-word)
  - as before, one team member should operate (or "drive") the computer while another actually speaks/presents, and this "driver" should also know what to do next at any given point
  - as before, each presenter should know what he/she plans to do next at any given point, as should the overall team
    - (avoiding having to discuss, during the presentation, what should be shown next -- everyone should already know that from having planned and practiced the presentation beforehand)

- you will use the **BSS 313** lab projector; a laptop CAN be connected to this projector, if you do not want to use the lab computer. It is the team's responsibility to make sure in advance that they can successfully project what they need to for this presentation; if you need to make special arrangements, do so well in advance of the presentation!

Milestone 9 - Individual reflection paper

Due:

Thursday, May 10, beginning of final exam period

Turn in:

Individually-written papers describing:

- what you learned from this project
- how you might have learned more
- how you would do it differently if you could start over
- ways in which your project team was successful
- ways in which the project team was not successful
- an example of something you now know about working in a team that you did not know before this semester
- any other points, pertinent to course discussion, that you would like to make

These papers do not need to be long, but they do need to be taken seriously, and they need to discuss all of the areas listed above. Note that grammar and spelling will be taken into account --- as seniors presumably about to graduate, you should be able to write a technical reflection paper with proper mechanics!

The primary purpose of this milestone is to provide you with an opportunity for some final reflection on your project as the semester comes to an end.