# CIS440 Group Project Assignments

The group capstone project is a software development project to be developed iteratively and delivered in five "releases" as well as a final report. Acceptable projects include web applications, mobile applications, data analytics, digital marketing campaigns, and more. Each project must have a responsible client outside of the class who will serve as a contact person and ultimately help me to grade your work. These may include established companies, start-up businesses and individual entrepreneurs, nonprofit organizations, or academic researchers. Requirements and grading criteria for each release are documented below.

## Release 0.1 (4pts, September 3)

By August 27, you will ideally have formed teams and identified projects.
On September 3, you must present an initial overview of the project to your classmates. This will include crafting an initial README.md file, identifying the high level tasks to be completed this semester, and preparing paper mockups or digital prototypes to communicate the design goals. The README file and prototypes should be committed to a GitHub repository, tagged “v0.1”, and the tasks should be entered into a waffle.io task board. You will give a 5-10 minute presentation to your classmates.

## Release 0.2 (4pts, October 1)

By October 1, you are to have set up the "technology stack" for your project and integrated all the moving parts, so that it is possible to begin adding features. For example, if you are creating a PHP/MySQL based website, you need to have established a web hosting account, set up the database and some of its tables, decided on a PHP version, and created at least one basic "hello world" page with links to a basic CSS file. Determining a folder structure, and perhaps creating dummy files for pages yet to be developed, are also good ideas. Commit the current version of your code to the GitHub repository and tag it “v0.2”.

This step is vital because it makes it possible to test and demonstrate new features. With the technology stack in place, someone could make a change to the CSS (or something) and test it with all the other components working. For full credit, you also need to provide **a set of instructions** for others to use your software. I, or anyone else, should be able to follow these instructions to get your project up and running. Therefore, in addition to your code, you need to provide information about how your server is set up, and you'll probably need to provide the SQL scripts to set up your database. A sign of good design is that the list of instructions doesn't need to be too long!

In your presentation to the class, take up to 10 minutes to tell us about the tech stack and demonstrate that it works. This is also a good time to tell your classmates where you’d like help.

## Release 0.3 (4pts, October 29)

At this time, your project should be a "navigable" prototype that can be shown to potential users for their feedback. Not all of the features need to be working, but you should have prototypes of the core features, and dummy pages or “stubs” in place of other features yet to be developed, as well as site navigation so that you can pretend to demonstrate most of the intended features. You should also have a basic visual style (i.e. fonts and colors) although it doesn't need to be perfect -- just enough to have something people can give feedback on. Commit the files to GitHub and tag the commit as “v0.3”.

Carry out some "user testing" with at least 3 people (ideally, with intended customers or end users) to get some feedback on the usability of your software. You might want to video their experiences with your product. In your presentation to the class, tell us some of the feedback they gave you, and show us at least one specific change you have made as a result of this testing.

## Release 0.4 (4pts, November 12)

By now your software should be substantially complete. This is the time to get as many features done as possible, and let your client know which features probably won't get done in this semester. The client should be involved in determining these priorities and helping you decide what to include and what to defer. You may want to start removing links to pages that aren't going to be finished, and generally polishing the details. There are three weeks left before final presentations, and I'd like you to try to "go live" in two weeks (if you haven't already). In your presentation to the class, show us the new features you have built, and convince the professor that you will be ready to "go live" in two weeks.

The next three weeks should be mostly devoted to testing the software, fixing bugs, and completing the documentation, so that you can go live (at least to beta testers) by the day of the demonstration. Implement some form of testing – either "verification" testing to identify bugs, or "validation" testing to gather feedback from live users on the design and usability of your software. The testing plan should be documented in the GitHub repository (perhaps in a folder called "tests"). Tag the latest commit on GitHub "v0.4".

## Release 1.0 (7pts, December 3, 10:30-12:00 in McCord Hall, Avnet Lounge)

Give a public demonstration and presentation about your completed project. This will be conducted as an "expo" for other students, faculty, staff, and friends of the Information Systems department to see your projects. Each team will be provided with a table to set up their demo. There will be easels available for displaying posters. I can't guarantee you'll be near a power outlet, so make sure your batteries are charged. You should be prepared to give a good brief demonstration, and answer questions from me about:

- What feedback you received from the testing you conducted, and how you responded to it.
- How the project evolved and changed during the course of development.
- What's the path going forward for this project, and what you might have done if you had had extra time.

I'll invite your clients/sponsors to the showcase, and after this event I'll give them a questionnaire to provide feedback that impacts your grade. So give it a good effort!

## Project Report (7pts, due December 3)

Prepare a report of at least 20 pages that tells the story of your successful project. At a minimum, it should include the following:

- Title page, table of contents, and 1-page summary.
- Identify the client, the problem, and the proposed solution.
- Identify the team members.
- Give an overview of functionality, actual and planned.
- Identify the most significant technical or business challenges posed by this project, and how you approached them.
- Describe the technology stack and any info needed to understand the code.
- Describe your testing or evaluation activities and findings.
- Tell us what's next for the software. Did it go live? Will it be developed further?

Feel free to embellish the report with graphics, code documentation, usage instructions, and other valuable information. A major objective here is to have a "success story" that you can add to your portfolio and that I can share with other potential capstone project clients.

## Summary of Milestones:

- By Sept 3, show us design documents so we know what you intend to accomplish.
- By Oct 1, have the complete “tech stack” working and replicable.
- By Oct 29, have a navigable prototype and carry out usability testing with 3 people.
- By Nov 12, have the project mostly complete and ready for testing and bug fixing.
- On December 3, demonstrate your completed project and be able to talk about it.

## Extraordinary Projects

If your project is not a "typical" software development project, and some of the above milestones don't make sense for it, contact me directly for guidance on what to deliver and when.

## Additional Notes on Grading

Your clients will complete a feedback questionnaire after the final delivery (ideally around December 3). Their feedback determines 10% of your course grade.

Furthermore, at the end of the semester, all students will have the opportunity to rate their teammates' performance, and the average ratings can reduce your project score by 0-100%. (For example: if your team earns 40 points out of 50, but your teammates rate you at an average of 75%, you would get 40*0.75=30 points for the group project.) Therefore it is important to form a good working relationship with your team, and resolve any potential problems before they become critical.