
Project Information and Suggestions

Some of you may come into this class with a very clear idea of what you'd like to do for a project, and others may have not thought about this at all. Even if you have a general idea, flushing it out to a set of specifics can be challenging. The only real requirements are that your project be approved by the instructor, and that it involve a strong design component (so porting an existing application or fixing bugs in an existing project wouldn't qualify). As you consider proposing a project topic, keep in mind the following suggestions.

- *Make it relevant:* Don't treat this as just an assignment you have to do to fulfill a requirement — try to do something that someone else would care about: a useful project that you can contribute as open source, an application or tool for a non-profit or a research group, a research study that answers a question that others might find interesting, ...
- *Make it something you think is cool:* You're going to be spending a lot of time working on this project, and it's easier to get motivated for the amount of work required if it's something that really sparks your interest.
- *Use what you've learned:* One goal of this project is to bring together different things you've learned to create something interesting. For example, you might use software engineering principles, data structure and algorithm design, grammars from theory of computing, user interface design, and maybe even things like artificial intelligence, security, networking, or operating systems.
- *Try to learn something new:* Maybe this is a time to learn a new language, like PHP or Python, or a time to learn a new framework/API, like Facebook or Google gadgets. Don't go overboard though: taking on multiple new topics (like learning Objective-C and the Apple iPhone API) and trying to get a non-trivial application designed and written might be too much.
- *Be Realistic:* Remember that you've only got a single semester for all phases of development, from design through implementation and testing, and that you have only about 4 weeks for the main coding work. There are lots of great ideas for cool software — just make sure you can do what you propose. The instructor can help talk you through what's involved in your project to determine if you're biting off too much for a single semester.

On the back are some project ideas to get you thinking — including specific projects, general ideas, and ways to find ideas. This list reflects some of my biases about what I find interesting, and you shouldn't feel locked in to this list. Hopefully it will get you thinking though!

- Extend a previous project in interesting directions (e.g., last semester’s project wireless signal propagation).
- Create a user-friendly graphical interface for configuring and maintaining advanced filesystems. Some advanced filesystems have been introduced recently: zfs from Sun (for Solaris, ported to BSD, and available in a limited way in Linux) and btrfs for Linux (initiated by Oracle, and still in the very, very early stages). Administration of such a filesystem can be a challenge, so a well-designed GUI tool would be an excellent contribution.
- Create an easy-to use encrypted “container” as a filesystem that can be used in Linux (and in Windows for an extra bonus!). Current encrypted filesystems in Linux (and Windows!) make you choose between either encrypting everything, but putting it in a fixed-size allocation, or encrypting individual files, which can grow inside the regular file system but expose directory structures and usually file names as well. Using FUSE in Linux, you should be able to make a filesystem that is allocated as a single block (so file names and directories aren’t exposed) and yet grows and maybe even shrinks with demand.
- Develop a solid, easy-to-use, encrypted, incremental off-site backup system for Linux. Some solutions do exist (such as `rsync.net` and others), so finding something unique and distinctive to do here might be a challenge.
- Write an Eclipse plugin that does something interesting (for example, manages programming assignments and submissions for an intro programming class).
- Write a “data merge” web application — for example, manage keeping an online calendar in sync with multiple other calendars.
- Work with a UNCG faculty member developing software for their research.
- Write a social game based on a social networking framework like the Facebook API or OpenSocial.
- Design and perform a research project exploring and comparing different game AIs (e.g., for a simple card game).
- Make a cool iPhone or iPod Touch or gPhone (Google Android) application.
- Develop the next cool web application that will make you and your faculty mentor a billion dollars (OK, just checking to see if you’re paying attention).
- Look through Google “Summer of Code” projects for interesting ideas.