# CS160 - Final Project
# Fall 2009

- Overview

  Throughout the first half of this course we have learned about the foundational concepts required to build a modern information retrieval system. Through our assignments, we have constructed the framework for a basic IR system. In this project, you'll be adding the frills and whistles. It is a chance for you to learn more about a sub-field of IR and to explore the experimentation process in CS. Not only will you be implementing an addition to our system, you will be evaluating your addition and doing a final write-up.

  This project may differ from many of your other class projects that you've had before because your project will be not be viewed in isolation. My hope is that with good planning, we will be able to integrate all of these projects together into one (yet to be named) IR system. Besides having something interesting you can show other people about your hard work, this will also give you some exposure into the challenges of working on a modestly sized software project.

  The major implication of this will be that planning for this project will be **very** important since your project will interact with other people's projects. We will all be building from the same starting code base, and so you will need to specify any interactions that you will have with our current code.

- Schedule

| date | description |
|---|---|
| 11\4 | Project proposal draft and in-class discussion/planning |
| 11\9 | Finalized project proposal |
| 11\18 | Status report 1 |
| 11\25 | Status report 2 |
| 12\6 | Paper draft |
| | initial code submission |
| 12\9 | Reviews |
| 12\14 | Final paper |
| | Presentation |
| | Final code submssion |

- Project requirements
  - must be related to something we have discussed (or will discuss) in class
  - must integrate with our current system, though, it is likely we'll need to make changes to interface properly
  - should work out to around 20 hours of coding work per person
  - evaluatable
- Project ideas
  - GUI and backend
  - Snippet/summary generation
  - Index compression
  - Distributed indexing
  - Faster/approximate ranked retrieval
  - Spelling correction/Query suggestion
  - Relevance feedback
  - Audio retrieval
  - Image retrieval
  - Clustered search results
  - Crawling (though come talk to me beforehand if you're thinking about doing this)

- Project proposal

  Your first task will be to come up with your project group and figure out what you'd like to work on. A first draft of the project proposal will be due in class, where we will discuss the different projects and attempt to work out how all the projects will fit together. Based on our discussion in class, you will then go back and revise any issues and submit the final version.

  Your project proposal should include the following information:

  - Members of the team. I'm *strongly* encouraging pairs. If you want to work solo or in a group of 3, please come talk to me.
  - A one paragrah description of your project
  - What you hope to accomplish and how you will evaluate your approach. You should think about what would be an appropriate way for evaluating your success. For example, for index compression you could measure the size reduction, for distributed indexing you could measure the speed increase and for spelling correction you could measure retrieval performance improvement for mispelled queries.
  - What resources you will use/need including code, data, etc. All of these resources MUST be obtained by the time the final draft of the proposal is due. Come talk to me (early) if you're having trouble finding appropriate data.
  - Two papers in the literature that tackle the same problem.
  - Logistics:
    - How you will integrate with the current code base including any changes that need to be made to existing code.
    - Things to be accomplished by status report 1
    - Things to be accomplished by status report 2

- Status reports

  In your project proposal you will specify a list of intermediary goals/accomplishements. A status report should be a one paragraph description of the current status of the project, including to what extent you accomplished your proposed tasks and any problems or issues that have arisen. This will also be an opportunity to communicate to me if there are any changes to your specification, etc.

- Paper

  Your paper must be complete enough for me (or anyone else) to fully understand what you did. I would like you to think of this as a real (potential) submission to a conference or workshop. It is unlikely that you can complete enough work to have a submittable paper in this short time (most workshop projects take several months to develop and write up, if not several years!). But if you get excited about your project, I would encourage you to continue working on it after the end of the semester and plan to submit it to a SIGIR workshop or symposium in the future.

  We will use the ACM paper format which is used by SIGIR (`http://www.acm.org/sigs/publications/proceedings-templates`). The website has templates for both latex and word, either of which are fine. You paper will be short (on the order of **3 pages**). Even though it is short, I expect it to be well written, well organized and present what you've done (including your results) clearly and concisely.

  You may organize the paper however you like, but a common approach would include the following:

  - Abstract: Gives a very high-level view of the problem, approach and results. An abstract almost never more than a paragraph.
  - Introduction: Describe the problem and motivate why the problem is interesting/useful.
  - Algorithm description: Clearly describe your algorithm including any challenges you encountered.
  - Results: Describe your data, experimental setup, evaluation critrion and how well your system performed. You should spend some time discussing the results, including if anything was surprising or interesting.
  - Conclusion: A brief summary of the paper including any challenges, where to next and any high-level comments you have at the end of the project.

- Reviews

  Each of you will be asked to review 2-3 papers written by other groups in your class. Your review will be similar to previous one's we've done in class, but should be easier since the papers should be shorter and should be on topics you're familiar with. For these reviews it will be important that you give very specific feedback about things that were good/bad/unclear in the paper since the authors will use your feedback to improve their final paper.

- Presentation

  Each group will give a short (15-20 mins) presentation of your work during our final exam period. At a high-level, your presentation will have a similar flow to your paper. Your presentation must include the following information:

  - Problem
  - Motivation: Why is what you did useful?
  - Approach: How did you solve the problem?
  - Results: How well does it work?

- Code submission

  Since we are developing a system as a class, you will need to submit your final working code. I will find a home for our project and will setup a location where you can make sure your code integrates well. When you submit your draft of your paper, you will also submit a working/compiling version of your code so that we can work out any integration issues. When you submit the final draft, you will submit a final version of your code that should address any other issues you have as well as tack any integration problems that have been identified.

- Grading

  The project will represent a majority of you work for the last half of the class including your final exam, so don't get too concerned that there are a lot of things to do. You'll have six weeks to accomplish the tasks, which should be plenty of time if you stay on top of things.

  - Project proposal draft (5 points) - Did you think about the project beforehand and have something reasonable written?
  - Project proposal (20 points) - Meets specifications above.
  - Status reports (10 points each)

- Paper draft (10 points) - Do you have a reasonable draft in place for other students to review?
- Initial code submission (10 points) - Compiles, follows integration specifications
- Paper reviews (20 points)
- Final paper (100 points) - Meets specifications above.
- Presentation (20 points)
    - Covered content
    - Organized and well-prepared
    - Presentation style
- Final code submission (10 points) - Compiles and all integration issues have been worked out.