**To Do on Chair Project:**

1. Put in safety feature to stop one chair from editing another dept’s stuff!!!
2. Create a sandbox year that allows you to play around with different scenarios?
   1. Should it have the ability to copy things over from that year into the actual current year?
   2. In Profile, make it so that you can either show or not show the sandbox year. Uh-oh…that’s going to require a change to the model!!!
3. For courses that are offered spring and fall, the faculty load summary seems to list the SPRING ones first, and then the FALL ones. WHY?!? (See EDU150L for an example.)
4. Export feature:
   1. Fix redirect after downloading the file. Not sure how to do this at the moment…. From Tom:
      1. See the "checked" answer here: <http://stackoverflow.com/questions/20827665/django-redirect-after-file-download>, which references an informative page that features a solution: <http://johnculviner.com/jquery-file-download-plugin-for-ajax-like-feature-rich-file-downloads/>
5. Add capacities to rooms.
6. Superuser:
   1. Oops….crn was attached to Courses instead of to CourseOfferings….ruh-roh….
7. Add new field for actual course enrollments?
8. Add things for profs’ schedules (office hours, committee assignments, etc.)
9. Possibilities for data export:
   1. Weekly schedule (for Barb…for hanging on our doors) – I guess this is actually a Word document. What if this could be dumped out in pdf?
      1. Would need to add a new field for office hours.
10. Faculty Load Summary page:
    1. Under Meeting Times > Update Class Schedule, make it possible to delete all entries and immediately go to New Class Schedule.
    2. Add # students as a column?

**Nice AJAX/django reference:**

http://www.tangowithdjango.com/book/chapters/ajax.html

**Migrating instructions:**

Dr. Kiers,

I just saw this and if you have already solved the problem disregard this email.

Here are the steps I would use:

1. If you are not using south, use south. I think it will work with your version of python. You can install it with pip install south.

2. If you just installed south, make sure you have done an initial migration.

3. Add the field to the model with a default value that you can remove later in a later migration.

4. python manage.py schemamigration --auto <appname>

5. It should reply with something saying that it created the migration and you can apply it with migrate.

6. python manage.py migrate <appname>

7. The migration should have taken place and it will try to load the initial data which will fail because the schema has changed.

8. run the backup.sh script again to dump it.

9. Try to migrate again. (when using south, always used migrate instead of syncdb)

The above steps should also work for changing a foreign key to a many to many field. If you are not or can not use south then I have no idea what you would do.

I hope this helps.