To Do on Chair Project:

1. To do in db on server:
   1. Create new academic years (2012-13 and 2014-15)
   2. Create new semesters there
   3. Add in info from 2012-13 for copying purposes
2. On the Schedule drop-down, add an option for “Course” (or maybe “Course Offering”).
3. Delete unused stuff in urls.py, views.py and forms.py (?)
4. Put in safety feature to stop one chair from editing another dept’s stuff!!!
5. Add permissions – should do this by defining a custom template tag(!)
6. Add capacities to rooms.
7. 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?
8. Weekly Schedule page:
   1. Add function that checks for time block conflicts in faculty schedules
      1. If there are conflicts, needs to send a warning to the schedule
9. Room Schedule page:
   1. Add function that checks for time block conflicts
      1. If there are conflicts, needs to send a warning to the schedule
10. Course Summary page:
    1. Add credit hours for courses in a new column
    2. Add the ability to edit/delete courses
    3. When adding new sections (under Manage), show some info about the current sections, such as the professor or something…? If there are several, it could be difficult to see which one is which.
    4. Under Manage Offerings, indicate # credit hours, to make load hour decision easier
11. Other Features:
    1. Maybe make professors m2m instead of ForeignKey, so that profs can be part of more than one dept?
    2. What to do about courses like IAS…, which are not really dept courses? Work-around: IASm for IAS(math dept), etc.
    3. Add some sort of “copy” feature so that some amount of information can be copied from one year to another (what should all be copy-able in this way? Could be tricky if faculty members leave/join, etc.) Maybe just copy over the course offerings and course meeting times and rooms, but not the load information (i.e., professors). Or ask which parts the user wants to carry over….
    4. Set things up so that only the chair (or certain superusers) can edit information on the webpage – other faculty members have read-only access (they could each have their own profiles, though)
       1. Maybe each faculty member can only look at info for their own dept

**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.