

TESTING

- [high priority] Write unit tests for all the code!!! Done by May 16th
 - The models for course manage
 - The views for course manage
 - The models for note sharing
 - The forms for uploading
 - The file handling internals
 - File deletion
 - Filling in empty names
 - The views for file uploading and downloading

FUNCTIONAL

- [high priority] Write templates for account self management
 - include forms for:
 - email change
 - [Done] password change
 - [Done] allow user to see and delete all their files
- [low priority] make a filter by user file list Done by May 20th
- [low priority] join the password reset system into the site
- [?] Keep track of user's upload quota
- [low priority] Add a winter/spring/summer field to courses and files

REFACTORING

- [low priority] Refactor the code into three separate django apps [DONE]
 - User_manage
 - Course_manage
 - Note_sharing
 - So to have the site extensible in the future
- [low priority] Wrap the views of individual models into Subclasses of Class based generic views

FRONTEND DESIGN

- [High priority] Finishing the base template for the overall site
 - Fix up the existing template to conform with the site design
- [High priority] Load this onto a skule server for testing.
- [low priority] have a combined register and login page.

AJAX

- Implement JSON serialization of course list and sending via AJAX
 - api for list of all courses
- Implement sending dialogs via AJAX + adjustment of these forms to use AJAX client-side

- Look into what I want to do wrt AJAX (tastypie? Tehn I can get an API too)

Client UI

- JSON-based course list search
- All courses/filtering dropdown

Views

- Enable views to use AJAX (either using dajaxice, tastypie [?], or just creating something manually with <http://brack3t.com/ajax-and-django-views.html>)
- Sending of CSRFtoken through AJAX

Templates/HTML/CSS

- Coding of "Dialog" templates
- Refactor CSS to make more structured/consistent/self-documenting

Visual tweaks

- Change layout to use cssgrid.net 1140px responsive fluid grid
- Create colour scheme

Feature

- file hosting - done
- notes upload from users and admins - done
- Browse course resources
 - Filtering by course - done
 - Filtering by user
 - user pages - need work
- account management
 - let the users add themselves - done
 - Future
 - Password management
 - Content management
- Admin
 - Super user control of content - done
 - Add delete users - done
 - Password reset - need work
 - Future
 - Content review
- Future
 - search
 - batch download
 - embedded file viewer?
 - comment
 - Incentive system?
 - tie-in with Skule Kup

Support

- Eng-Sci Club
- Web master
- K.P.S.
- Influential upper year

HR

- Brian
- Oliver
 - Tony
 - Shen
 - Brandon?

Architecture - Done

1. OS
 - a.
2. Server
 - a. You probably want Gzip enabled stuff
 - i.
3. Back-end
 - a. Storage

- i.
- b. account management
 - i. mark-us
 - ii. Django?
- c. content management
Organize by user?
 - i. Roll our own?
 - ii. Mark-us?
 - iii. Django?
4. Admin interface
5. Front end

List of things to ask people about

- Mark-us
- Eng Sci club
- K.P.S.
- Web master
 - what os are we we running it on.

Design 1

Back end Django

- File management
 - use the build in file management system - storing file on the same server
 - <https://docs.djangoproject.com/en/1.3/ref/files/storage/#django.core.files.storage.Storage>
 - or write our own - if we want to store the files some where else
 - <https://docs.djangoproject.com/en/dev/howto/custom-file-storage/>
- User management
 - <https://docs.djangoproject.com/en/1.3/topics/auth/>
- Front end
 - Use the build-in template engine
 - <https://docs.djangoproject.com/en/1.3/topics/templates/>
 -