

Agenda:

- Demo MVP
- Update about Questions in an Application
- How do we attribute overlapping authorship
- Implementation Questions
 - What is appropriate to put in req.session.user
 - How do we correctly use “next” in our controllers for functions of form function(req, res, next)
 - Using Boolean flags for handlebars: isSubmitted, isSaved, isRejected, isWithdrawn; is there a better way?
 - Any pre-existing API's for filtering and sorting? If not, opinions about the filter function in Listing
 - Can we expect the fields of the mongoose schema in the return of calls to the DB, or do we need to reformat them?
- Go over next stage work

Progress Report:

- Progress
 - We have the MVP finished with the following features
 - Sign Up Student, Sign Up Employer, Login
 - Student:
 - Fill in Common App
 - View all Listings
 - Add Listing to Current Applications
 - View all Applications of Student
 - Submit Application
 - Employer:
 - View all Listings of Employer
 - Create Listing with a Custom Application template
 - View all Applicants of a Listing
 - View an individual Applicant's Common and Custom Application
 - Tests for our models
- Milestones Achieved: Completed all of the MVP
- Milestones Missed: None
- Difficulties Encountered: No significant difficulties, we accomplished most of the heavy lifting last week!
- Changes
 - Did not go with separate Question model

Meeting Minutes:

- Demo MVP
 - Back button to listings on the applicant's page

- Logging out and logging back in may change from employer to student/vice versa
 - Logging out then pressing physical back button logs you back in
- Useful for instead of “Title” and “How many applied”
- Interpreting status and errors/error handling
- Updating Question
 - Kept our current model instead of adding a new Question model
- Attributing overlapping authorship
 - It's ok to have more than one person as a file's author
- Implementation Questions
 - What is appropriate for the req.session.user?
 - Current one is fine
 - How do we use next?
 - Refers to the next middleware function that we need to call
 - Instead of calling it, look into async functions
 - Return err/return
 - Using Boolean with Handlebars
 - OK
 - Handlebars helper functions
 - OK to register all at once
 - Preexisting APIs for filtering and sorting
 - Tablesorter (sorts things into tables easily)
 - Filtering (input/regex and then change on jQuery)
 - Expecting mongoose schema
 - Normal practice
- Next stages
 - Planned features for the next stage:
 - Logic for deadlines
 - Searching/sorting/starring/rejecting/etc. (sending email)
 - Hosting PDFs (maybe try integrating into dropbox/link)
 - Security
 - Cleaning up the UI
 - Looking at modal.js javascript document
 - Can generate as you go but generating it all at once is OK too