

11/30

Progress Report

- Finished MVP, features included:
 - Email authentication
 - Create postings (skills, student, course preference)
 - Enter skill name and their proficiency (no autocomplete)
 - Match students with urops (based on skill level only, no department or time commitment)
 - See matched urops with descriptions (student)
 - see posted urops with descriptions (staff)
- Revised design
- Got access to people API
- Missed milestones
 - None
- Identifies difficulties encountered
 - It's easy to miss things in the checklist
- Identifies changes to problem analysis or design
 - No longer using certificates
 - Skill is no longer a schema, but a collection

Agenda

- Demo MVP to mentor
- Feedback from mentor
- Logistic question
 - What are the remaining deadlines?
 - Identify what features we need for the final
- Backend
 - routes/user - should I refactor some code in controller.js?
 - what exactly does it mean to exclude details in schema? (actually how do we read "schema representation details excluded"? Does it mean we need to include schema or not?)
 - advice/critique about design choices?
 - Include skills/tags back-end design choice thing?
 - Deleting posting or not - design choice
- Frontend
 - "Appropriate and skillful use of advanced language features", do we need to use map, filter, reduce?
 - "Data types immutable when appropriate", example for this?
 - Do we include autocomplete feature in our revised design doc

Meeting Minutes

- Logistics

- If we need extensions, let Graeme know beforehand
 - Should have all planned features by feature freeze, which is next Wednesday
 - Next meeting is next Monday 2:30
- Backend
 - Use postman for email, avoid having email credentials in repo
 - Include matching on similar terms in skills/tags e.g. python = python3
 - Good idea to refactor code in routes/users
 - Do not delete things immediately, it's good to keep things around. Storage is cheap.
- Frontend
 - Fix cramping on staff posting page at skill bar
 - In student page, suggested UROPS, highlight first UROP or having a preview on the right side
 - Advanced language features - closures, functionals, similar to assignment standards, avoid bad js, clean, concise and elegant code
 - "Data types immutable when appropriate" - for example, put things in closures, reducing scope of variables
 - Yes, include autocomplete feature in design doc