This document is intended to help with planning the DB/API implementation

Possible API calls:

- remember searching — however we want to implement this…
- with all calls, user must be authenticated

Create account post call - for both buddy and student

Resources:

*Searching

create post —
sent: creator, title, content
stored: above + id, date

delete post —
sent: person deleting, post id

view posts —
get titles, ids

view post —
sent by client: post id
reply from server: posts and all of their info

Buddy connection system:
* Searching

Student:
search by industry
list of possible buddies (their company?, name) returned

another call to request additional information on potential buddy, such as a description
-- remember to handle if student already has request with buddy pending. May need to expand on this?

call to send request for connection to buddy(make sure that buddies which already have a friend request is grayed out)

Buddy:

- view all connection requests (client sends for which buddy account)
- get more info per student, e.g. client sends get request with student ID, server returns more info on student such as bio
- buddy can accept request e.g. ---accept request on my account from this student, new connection established
- buddy can deny request

Existing connections:

- for both student and buddy a list of all connections, and their info will be returned for now?
- call to delete connection

Profile:

For this section there will be a get request for each field, and also a post request to edit each
Field. I'm assuming google authentication handles password changes.

Let's decide on what fields to use

- student:
  - name
  - email
  - bio
  - interests/ desired industry?
- buddy:
  - name
  - email
  - bio
  - # students/connections
  - current industry
  - company (if applicable)

A similar approach of editing, deleting, getting applies to when we expand to support companies/contracts.