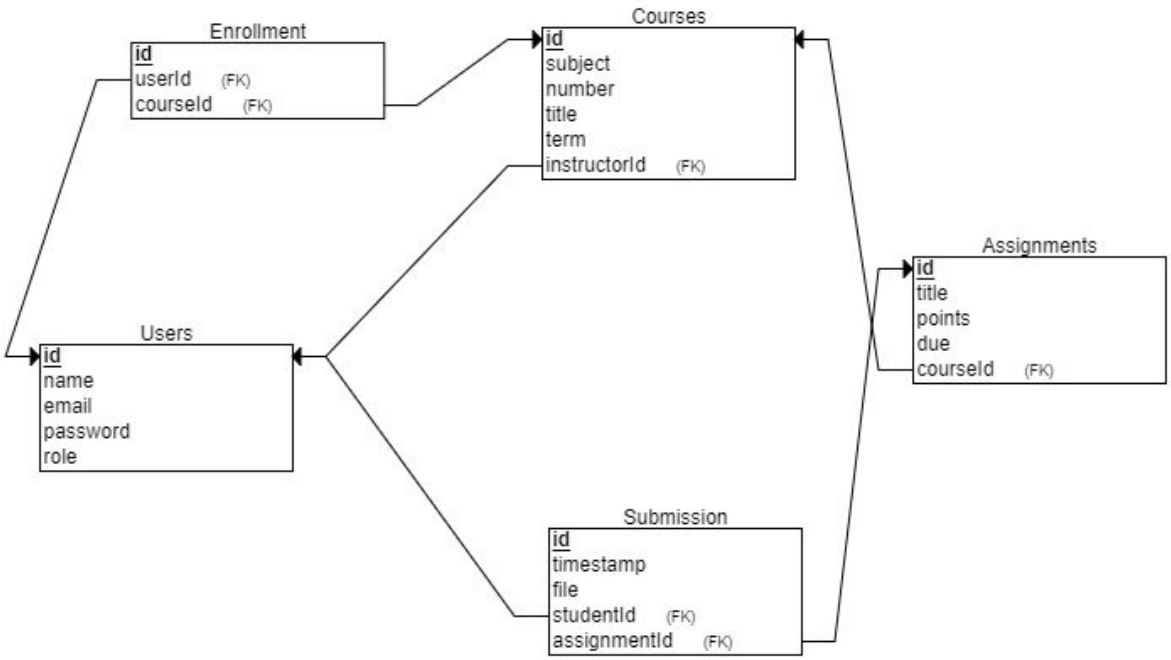
API Architecture Document - Group 8

Aaron Didner
Jared Tence
Brooke Weir

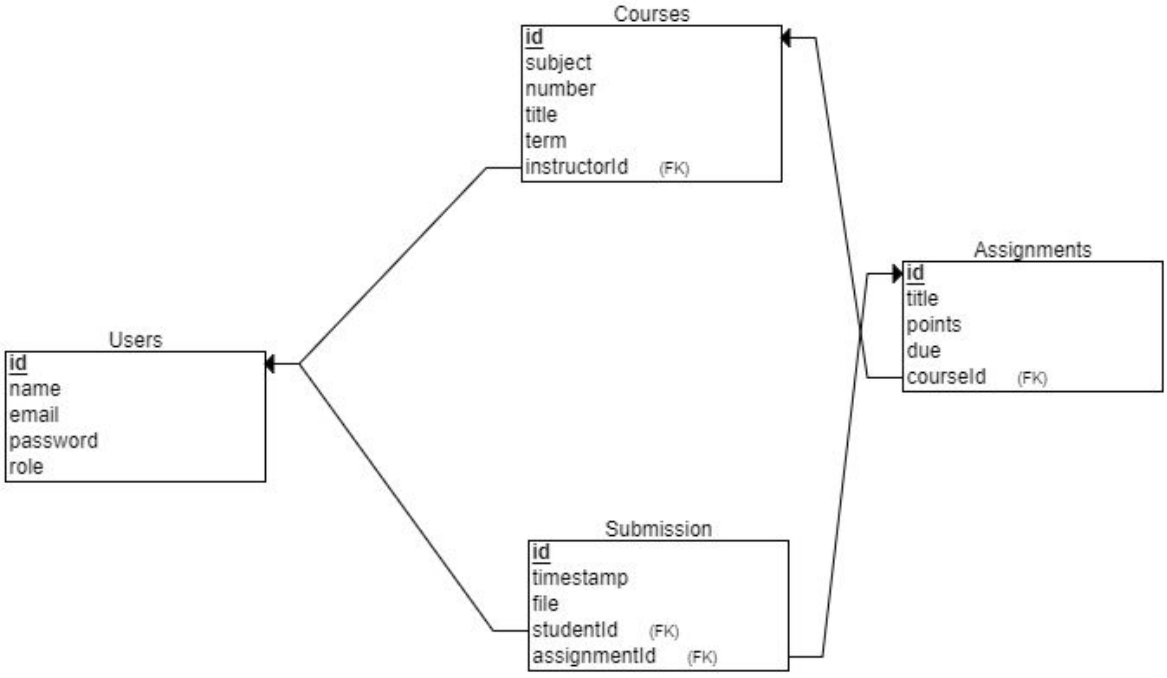
API architecture: – We are using Nodejs for our single api server, this Nodejs server will serve all GET, POST, PUT, and DELETE requests from users. Users will be limited to making at most 5 requests per a minute. Our Nodejs server will interact with a single MySQL database. The MySQL database will hold all data by storing data into four tables User, Course, Assignment, Submissions, and Enrollments. The MySQL will also store files associated with requests. Downloading a CSV is handled by a npm package called json2csv and sent as an attachment in the response.

API data layout –
Created with https://erdplus.com/#/standalone

Current ER Diagram:



Old ER Diagram:



API design reflection –

We ended up having to make changes to our ER diagram by adding an enrollment table which kept track of which students were enrolled in which courses for verification purposes when submitting assignments. This was an oversight on our part and in hindsight we should have looked through the endpoints in more detail before making our architecture. In addition we also think it would have been interesting to split up the work differently. Mainly by type of end point rather than by table in the schema diagram. For example having one user do all the post and get requests while a different user did all the delete, patch and put. Perhaps having someone else go in at the end to do all the authentication as well. We also should have discussed the order of completion in more detail so as to allow everyone to complete work at their ease without having to wait for others to finish their parts first.