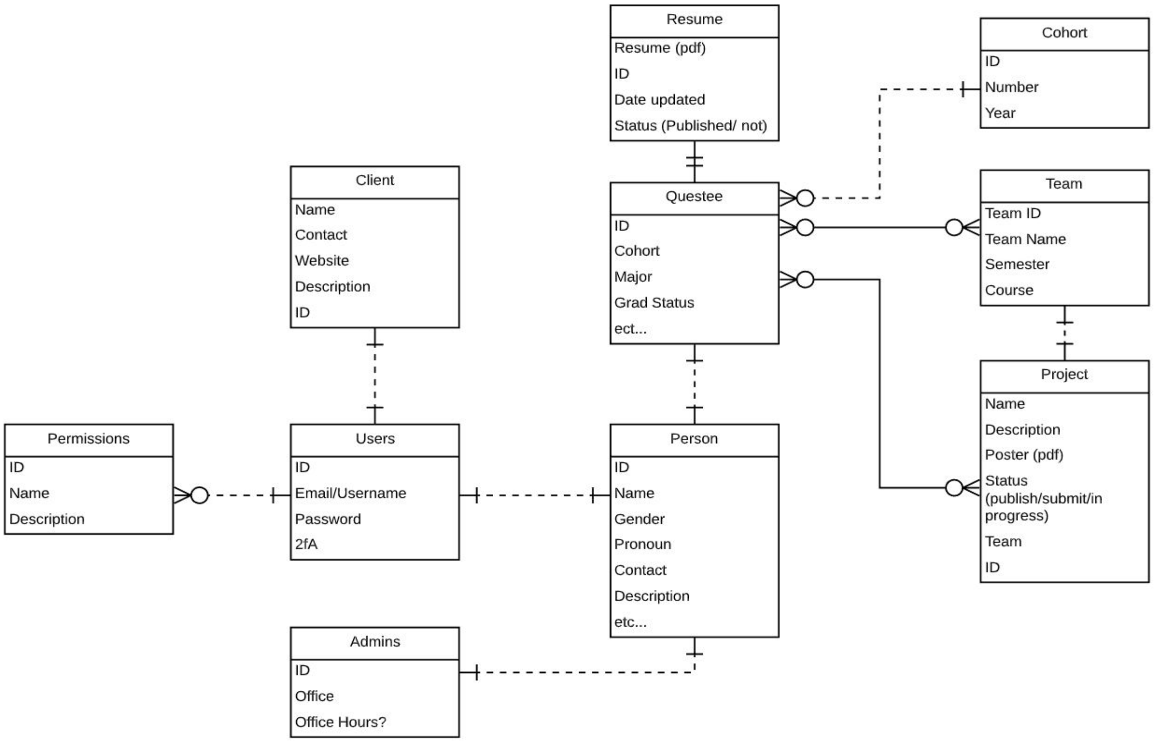
Rich & Creamy API

-- From meeting notes --



API

- Builds on the db
- Get all function, for each table
- Based off ID, return JSON results
- How are we access data from the DB
- Relational DB
- RESTful API Interface
- Swaggger.io
- JWT tokens - authentication

POST - add info to db
PUT - revise existing info
GET - request info from db
DELETE - delete existing info

Rich & Creamy API

Users

(POST) **createUser**(String email, String username, String password, String permissionName, String permissionDescription)

>Creates a user in our database
>Creates their default or selected permissions
>Returns result indicating success or failure

(PUT) **updateUser**(int id, String email, String username, String password)

>Updates selected user (based on id)
>Changes the values for any non-empty parameters
>Returns result indicating success or failure

(GET) **getUser**(int id)

>Gets selected user (based on id)
>Returns user information on success
>Returns indication of failure on fail

(DELETE) **deleteUser**(int id)

>Deletes selected user (based on id)
>Returns result indicating success or failure

(GET) **getAllUsers**()

>Returns all users information on success
>Returns indication of failure on fail

(GET) **searchUsers**(int id, String email, String username, String password)

>Returns all user information that matches all non-empty search criteria (parameters) on success
>Returns indication of failure on fail

Permissions

(PUT) **updatePermissions**(int id, String name, String description)

>Updates permissions (based on id)
>Changes the values for any non-empty parameters
>Returns result indicating success or failure

(GET) **getPermissions**(int id)

>Gets selected permissions (based on id linking to its user)

>Returns permissions information on success

>Returns indication of failure on fail

Person

(POST) **createPerson**(String name, String gender, String pronoun, String contact, String description)

>Creates a person in our database

>Returns result indicating success or failure

(PUT) **updatePerson**(int id, String name, String gender, String pronoun, String contact, String description)

>Updates selected person (based on id)

>Changes the values for any non-empty parameters

>Returns result indicating success or failure

(GET) **getPerson**(int id)

>Gets selected person (based on id)

>Returns person information on success

>Returns indication of failure on fail

(DELETE) **deletePerson**(int id)

>Deletes selected person (based on id)

>Returns result indicating success or failure

(GET) **getAllPersons**()

>Returns all persons information on success

>Returns indication of failure on fail

(GET) **searchPersons**(int id, String name, String gender, String pronoun, String contact, String description)

>Returns all person information that matches all non-empty search criteria (parameters) on success

>Returns indication of failure on fail

Admins

(POST) **createAdmin**(String office, String office_hrs)

>Creates an admin in our database

>Returns result indicating success or failure

(PUT) **updateAdmin**(int id, String office, String office_hrs)

>Updates selected admin (based on id)
>Changes the values for any non-empty parameters
>Returns result indicating success or failure

(GET) **getAdmin**(int id)

>Gets selected admin (based on id)
>Returns admin information on success
>Returns indication of failure on fail

(DELETE) **deleteAdmin**(int id)

>Deletes selected admin (based on id)
>Returns result indicating success or failure

(GET) **getAllAdmins**()

>Returns all admins information on success
>Returns indication of failure on fail

(GET) **searchAdmins**(int id, String office, String office_hrs)

>Returns all admin information that matches all non-empty search criteria (parameters) on success
>Returns indication of failure on fail

Questee

(POST) **createQuestee**(String cohort, String major, String grad_status, int cohortNumber, int cohortYear)

>Creates a questee in our database
>Creates their cohort designation in our database
>Returns result indicating success or failure

(PUT) **updateQuestee**(int id, String cohort, String major, String grad_status)

>Updates selected questee (based on id)
>Changes the values for any non-empty parameters
>Returns result indicating success or failure

(GET) **getQuestee**(int id)

>Gets selected questee (based on id)
>Returns questee information on success
>Returns indication of failure on fail

(DELETE) **deleteQuestee**(int id)

>Deletes selected questee (based on id)

>Returns result indicating success or failure

(GET) **getAllQuestees**()

>Returns all Questee's on success

>Returns indication of failure on fail

(GET) **searchQuestees**(int id, String cohort, String major, String gradStatus, int cohortNumber, int CohortYear)

>Returns all Questee information that matches all non-empty search criteria (parameters) on success

>Returns indication of failure on fail

Resume

(POST) **createResume**(File pdf, boolean status)

>Creates a resume in our database that is tied to the calling Questee

>Returns result indicating success or failure

(PUT) **updateResume**(int id, File pdf, boolean status)

>Updates selected resume(based on id linked to Questee)

>Changes the values for any non-empty parameters

>Returns result indicating success or failure

(GET) **getResume**(int id)

>Gets selected resume (based on id linked to Questee)

>Returns resume information on success

>Returns indication of failure on fail

(DELETE) **deleteResume**(int id)

>Deletes selected resume (based on id linked to Questee)

>Returns result indicating success or failure

(GET) **getAllResumes**()

>Returns all resume information on success

>Returns indication of failure on fail

Cohort

(PUT) **updateCohort**(int id, int number, String year)

>Updates selected cohort (based on id linked to questee)

>Changes the values for any non-empty parameters

>Returns result indicating success or failure

(GET) **getCohort**(int id)

>Gets selected cohort (based on id linked to questee)

>Returns cohort information on success

>Returns indication of failure on fail

Client

(POST) **createClient**(String name, String contact, String website, String description)

>Creates a client in our database with the given parameters

>Returns result indicating success or failure

(PUT) **updateClient**(int id, String name, String contact, String website, String description)

>Updates selected client (based on id)

>Changes the values for any non-empty parameters

>Returns result indicating success or failure

(GET) **getClient**(int id)

>Gets selected client (based on id)

>Returns user information on success

>Returns indication of failure on fail

(DELETE) **deleteClient**(int id)

>Deletes the selected client (based on id)

>Returns result indicating success or failure

(GET) **getAllClients**()

>Returns all client information on success

>Returns indication of failure on fail

(GET) **searchClients**(int id, String name, String contact, String website, String description)

>Returns all client information that matches all non-empty search criteria (parameters) on success

>Returns indication of failure on fail

Team

(POST) **createTeam**(String name, String semester, String course)

>Creates a team in our database with the given parameters

>Returns result indicating success or failure

(PUT) **updateTeam**(int id, String name, String semester, String course)

>Updates selected team(based on id)

>Changes the values for any non-empty parameters

>Returns result indicating success or failure

(GET) **getTeam**(int id)

>Gets selected team(based on id)

>Returns user information on success

>Returns indication of failure on fail

(DELETE) **deleteTeam**(int id)

>Deletes the selected team(based on id)

>Returns result indicating success or failure

(GET) **getAllTeams**(int id)

>Returns all teams associated with Questee of the given id on success

>Returns indication of failure on fail

(GET) **searchTeams**(int id, String name, String semester, String course)

>Returns all team information that matches all non-empty search criteria (parameters) on success

>Returns indication of failure on fail

Project

(POST) **createProject**(String name, String description, File poster, int status, int team_id)

>Creates a project in our database with the given parameters
>Returns result indicating success or failure

(PUT) **updateProject**(int id, String name, String description, File poster, int status, int team_id)

>Updates project (based on id)
>Changes non-empty parameters
>Returns result indicating success or failure

(GET) **getProject**(int id)

>Gets selected project (based on id)
>Returns project information on success
>Returns indication of failure on fail

(DELETE) **deleteProject**(int id)

>Deletes selected project (based on id)
>Returns result indicating success or failure

(GET) **getAllProjects**()

>Returns all projects on success
>Returns indication of failure on fail

(GET) **searchProjects**(int id, String name, String description, int team_id)

>Returns all team information that matches all non-empty search criteria (parameters) on success
>Returns indication of failure on fail