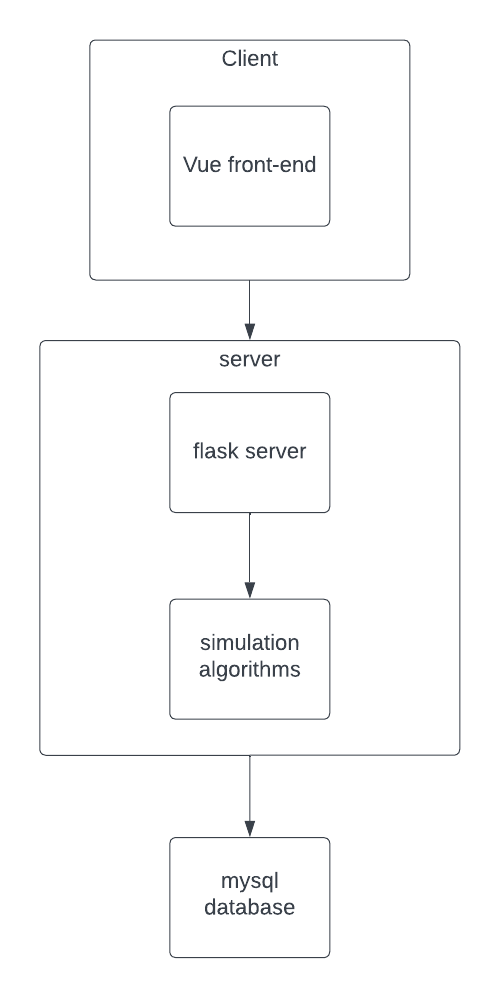
### Requirement document

##### Application structure



### Features (front-end):

**Registration:**

* Students will have to activate their own accounts in the system using their directory ids.
  + Process:

1. input directory id as account name and type a password.
2. (Usernames displayed in the system will be their true names.)
3. Server checks if directory id exists and has not yet been registered.
4. Prompt registration results according to server response.

**Grouping:**

- Students will have to create or join a group to participate in simulation.

(Students who want to solo will have to create a group that contains only themselves.)

* Info needed to create a group: group name.
* Students can only belong to one group.
* Group should be deleted if all members are gone.
* Only when students are in a group can they access the workbench module.
* To enhance group work, students should only join groups created by students who are of the same track.

**Simulation:**

* For each case, there should be some limit on the number of trials (5?) could be used before the group’s final submission.
* Students should be prompted if they are trying to submit final before using up all their trials?
* All members of the group can use trial and see the outcomes. There should be a confirm dialog box popped to the user before the trial is executed to avoid accidental clicks.

**External:**

* There should be a process for integrating student info from the university’s database.

**Records and Leader Board:**

* Records module should display simulation records made by the student’s group.
* Leader board should display the simulation records in the order of their score.

**Admin:**

* Admin should be able to view the records of all the users.
* Admin should be able to change the availability of cases to a track of users?
* Admin should be able to change the hyper parameters of the simulation?
* Admin should be able to import student data?

### Features (back-end)

* List of api’s:

root format 🡪 host/bmgtelp/

* + sign-in [post]
    - params: user\_did, password
    - response: status, message, set-cookie:KV’s
  + sign-up [post]
    - params:directory-id, password
    - response:status, message
  + forget-password [post]
    - params: directory-id
    - response: status, message: “email already sent to corresponding email address”
  + get-user-info?user-id=1 [get]
    - params:user-id
  + get-user-list?page=1&asc=0&by=name [get]
  + get-case-info?case-id=? [get]
  + get-case-list? [get]
  + get-case-record-list?page=1&asc=0&by=name [get]
  + get-case-record?case-id=1&group-id=1 [get]
  + get-group-info
  + get-group-list
* Cookies:
* There should be a process to import students’ info, including their names and directory ids.
* User registration:
  + Method: post
  + Params: {directoryId, password}
  + Returns {status, message}
* Forget password:
  + Method: post
  + Params: {directoryId}
  + Returns {status message}
  + Note: email should be sent to user. Front end should not prompt if the email exists.
* User sign in:
  + Method: post
  + Params: {directoryId, password}
  + Returns {status, message, cookie}

##### User requirements

Students:

* Sign up an account
* Sign in to the system
* Retrieve password
* Simulate a case
* Create or join a group
* See past records
* See simulation leader board

Administrators:

* Create account
* See simulation records
* See leader board
* Change availability of cases to students

Base user:

* Register
* Login
* Access available simulation cases
* View accessible leader board
* Create/join simulation group

Admin user:

* Import user account?
* Change availability of a simulation case to users
* Tag users