Chan Lee and Ryan Lansdon

Nomenclature:

- Project: A project created by a user within the application
- Owner: The user who created a user project
- User: Any CU Student on the web application
- Member: A user involved in a user project who is not the owner

1) MyCUProject is a web application that allows users to find projects where they can post information about projects they have in mind or are currently working on and to find users who want to join their project. It is essentially a platform where college students who are currently enrolled through CU credentials can create and join in on user-created projects that fit their academic and personal interests. This app will be primarily for posting project ideas and finding members, not for managing currently running projects.

2) Technologies we are planning on using: (We scaled familiarity from 0 to 10)

- Back end:
  - Python - Chan: 8/10, Ryan: 5/10
  - Django framework - Chan: 6/10, Ryan: 1/10
- Database:
  - Postgresql - Chan: 4/10
- Front end:
  - HTML - Chan: 4/10, Ryan: 6/10
  - CSS - Chan: 3/10
  - Javascript - Chan: 4/10
  - Adobe InDesign - Chan: 4/10 Ryan: 8/10
- Hosting:
  - Heroku - Chan: 2/10
- Version Control:
  - Git - Chan: 6/10, Ryan: 6/10
- Additional Libraries/frameworks:
  - Bootstrap 4 - Chan: 3/10, Ryan: 5/10
  - React library - Ryan: 3/10, Chan: 1/10

3) Essential Project Elements:

- Our project won't work if it doesn't have a login system.
  - MyCUProject is specifically geared towards students at CU. Users who verified their CU credentials can view the details of and join projects. It was designed this way so that students at the same school can benefit from joining projects.
- Our project won't work if it doesn't have dashboards.
  - MyCUProject requires dashboards so that people can post their project details. Thus this functionality is crucial in the success of MyCUProject. The user project details involve current progress of the project through a visual timeline, description of the project, users who are already involved in the project with

details of their job duties, a list of all job duties and what kind of users the owners are currently looking for.

- Our project won't work if it doesn't have a database.
    - MyCUProject needs a database to store the information about the specific users. So, by viewing those details the project owner can decide if they want to add them into their projects.
- Our project won't work if it doesn't have a user appeal page for each project
    - MyCUProject needs a page where each user can write an appeal to the project. Users who are interested in a project should be able to send a message to the project owner with some information about themselves and why they want to join the project.
- Our project won't work if it doesn't have a user profile page
    - Within the user profile page, the user can list their tech stacks and their experience, so that the project owner can easily view what their candidates have as their skills. (Another way of doing this is allowing the user to upload their resume on to the web).
- Our project won't work if it doesn't have a functionality where the user can sort the projects by their nature.
    - Project without sorting would mean a nightmare for the users since they have to go through multiple pages in order to find the project of their desire.
- Our project won't work if users cannot have customized sets of projects they are involved in.
    - Users should be able to see the projects they are involved in without needing to search for them every time.
- Our project won't work if it doesn't have search functionality
    - Users should be able to find the project through typing key words. So, that greatly saves their time in searching for the post they want to.

Back Up Plans

We acknowledge getting CU Credentials can be hard to do, so if this seems to be a too much work, we will use dummy email address that ends with 'colorado@edu' to test our purpose.

4) Outside resources required for this project will be minimal since the project owner will write the details about their user project themselves and users will view the user project and write a project appeal of some sort if they are interested. Some graphics will be required to show the current timeline of a user project, but other than that, not much is needed.

5) Since we are going to use the django framework, our frontend will all be positioned in 'views.py' file within the framework. Within the 'views.py', we will have all the html and css files. I have not figured out or felt necessary utilizing javascript at this point. Our front end and back end will mainly follow the flow of the django MVT pattern which is Model-View-Template which is

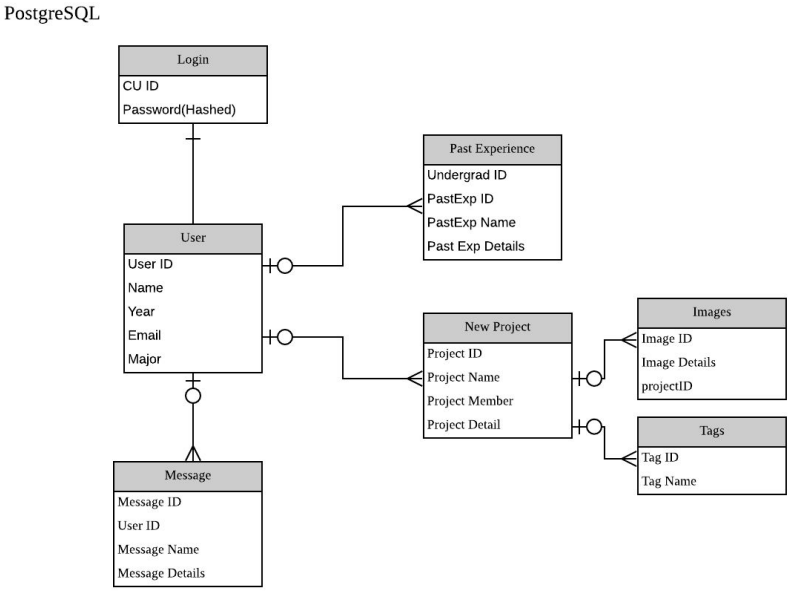
similar to MVC pattern where Controller corresponds to Template in this case. View corresponds to what will be displayed to the user. Model and Template essentially function as back-end where model saves the schemas of the database and Template controls how the Model and View will interact.

Our user interface is how our user will mainly interact with the program. Good user interface is essential to the site because users need to easily browse and create projects, and interact with other users and project owners. InDesign will be used for the website layout, so the user interface can look exactly how we imagine it, and Bootstrap 4 will be used to fine-tune the presentation of the site. Moreover, it is crucial for our program to have a functionality where each interested user can talk to the project owner and vice versa so groups can be made with the right members.

Inheritance relationships will be often used in the 'views.py' where we will base view and use other views to extend upon base views. Also, we will use inheritance in different project models classes where there are going to be many different projects with slightly different features. For example, a programming project model will have a field for languages, whereas an art model will have a link to the user's portfolio. So, for example, we will have a general project model and extend it to create a programming project model and an art project model.

For the use of design patterns, as we have mentioned above, we will extensively use MTV pattern(similar to MVC pattern) in building our web application. For other patterns, I think we can use factory patterns for the programming project model example as well. I believe factory patterns can be applied to account for different types of projects ex. art project model or computer project model.

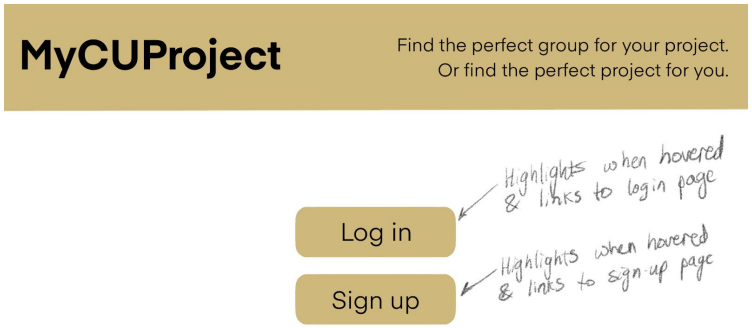
Our views are going to consist of User Profile Page, Project Edit Page, My Projects Page, Dashboards Page, My Applications page. Many of the pages will start from a base main page. Thus we need to inherit base pages to create those pages. Moreover, the Django framework will allow significant use of objective-oriented Programming Principles.



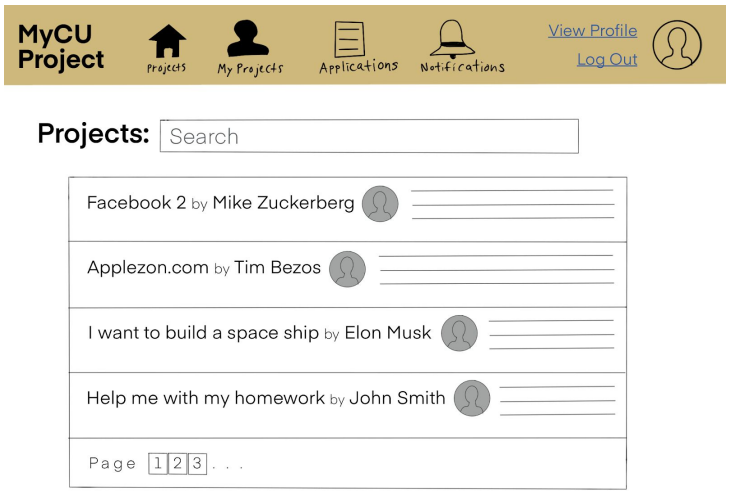
<Database Model>

6) Our user interface will use CU marketing colors, since it is meant for CU students. After the landing page, each subsequent page will have the same header, with MyCUProject on the left hand side, and the profile on the right side, with links to view the user profile, and a link to log out. There will additionally be four tabs on the header: *Projects*, which will link to the dashboard, *My Projects*, which will link to a page similar to the dashboard which displays only projects the user is involved in, *Applications,* which will link to all open applications the user has received for their created projects, and *Notifications,* which will tell the user when they have been accepted into a project, or when they have received an application. Assume all interfaces described are below the header.

- Landing Page: This page will have a brief description of the website, with two buttons in the center of the page, one to login, and another to register for the site. Our reach goal for this project would be to link the login for MyCUProject with identikey accounts. However, for the scope of this project, we will have a registration page where the user is required to submit an “@colorado.edu” email address in the email field.

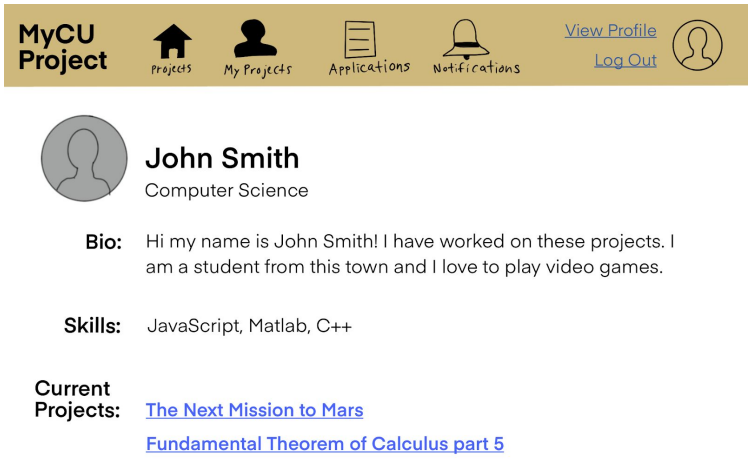


- Dashboard: At the top of this page will be a list of projects, with a search bar and a button to search by tags. Each project in the list will have a project name, and the creators name. Project results will be paginated, with a limited number of projects displayed per page. Each project will have a button on the right side of it that will cause that project to drop down
    - Project drop-down: This will be within the dashboard, and will simply be an expansion of the project the user clicked on. This will still display the project name and owner name, but will also have a longer description of the project, and the current members of the project. At the bottom corner of the project page will be a button to contact the owner.



- User-owner communication pop-up: Once the user clicks the contact button, a pop-up will appear on the center of the dashboard. There will be a text field where the user can write an appeal to the project. The bottom right corner of this project will have a button to submit to the owner, which will be greyed out until the user types something.

- Profile page: This page will show the user's own profile information. Top left will be name, and on the right side will be all other information the user chooses to display. Below name will be an edit profile button, where the fields that display for a user can be changed (only by that user).



7) Detailed Project Plan

For Sunday(2/23): Create Django app, setting up virtualenv environment

- Week 1 (2/24):
  - Ryan: React Tutorial, Setup login/signup system, user profile page
  - Chan: Connect Django with postgresql, create db models, user profile page
  - User testing: At this point, the user will be able to see the landing page, and either sign up or login to the site. After logging in, the user will be taken to their profile page, where they will see their information (pending completion of user profile page).
- Week 2 (3/2):
  - Ryan, Chan: Dashboard layout and functionality (both front-end, back-end)
  - Ryan : React Tutorial
  - Chan: User profile page
  - User testing: At this point, the user can now see the dashboard with some dummy projects, as well as their profile page after logging in.
- Week 3 (3/9): Homework 3 due
  - Ryan: React Tutorial, project edit page
  - Chan: Project Edit page, My project page (related to project formation)
  - User testing: The user will now be able to create a project, and upon creation of that project, see the project page with information for the project.
- Week 4 (3/16):
  - Ryan: React Tutorial

    - Ryan, Chan: application functionality
    - User testing: At this point, the user will be able to see projects on the dashboard, and will be able to use a button on the project page to apply to join a project by typing an appeal or attaching a resume.
- Week 5 (3/23): Spring Break
- Week 6 (3/30): Homework 4 due
    - Ryan, Chan: approval functionality + update on dashboard
    - User testing: Project owners will be able to approve or deny a project application. If they approve a new member, that member will now appear under members on the dashboard and project pages.
- Week 7 (4/6):
    - Ryan, Chan: Notification functionality
    - User testing: Users will be notified if they were approved for a project.
- Week 8 (4/13): Homework 5 due
    - Ryan, Chan: Exclusive User Testing, Possible Bug Fixes
    - User testing: At this point, the website should have all necessary functionality. Final user error testing will occur so bugs can be fixed.
- Week 9 (4/20):
    - Both: Making the presentation, Final debugging and testing
- Week 10: Final Presentation

If we have time, we will work on Tagging functionality (Search by tags)
If we have time, we will work on deployment

Each user will have their project list and based on the fact that if that person is the owner or not, we show slightly different views.

8) All work for this project will be done outside of class. Therefore, it will not interfere with in-class activities. We will be working an average of 10 hours a week on this project, and allocating separate time for the programming exercises.

Link:
https://docs.google.com/document/d/1qIeXJ1IQ-0_WMje87hsyXmoTdClYFceFnJvblHONlxY/edit?usp=sharing