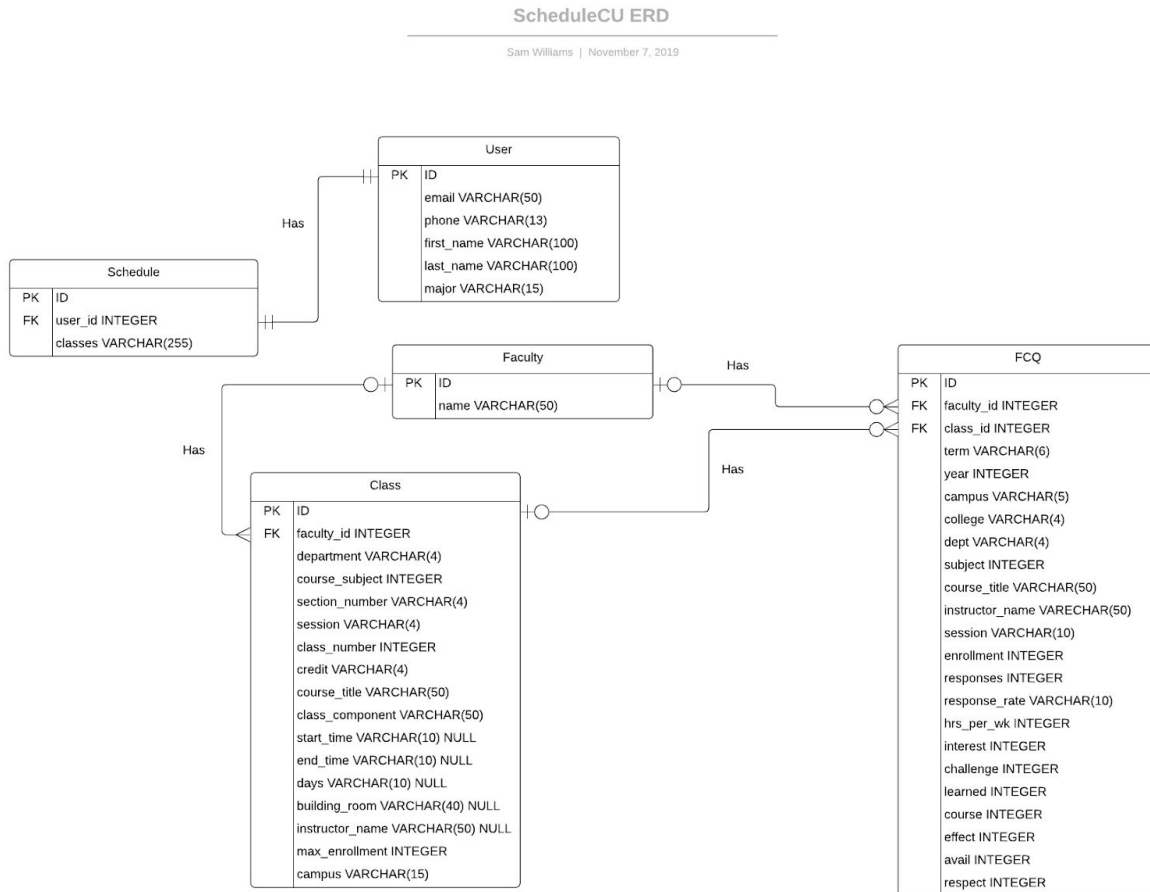


## Revised list of features:

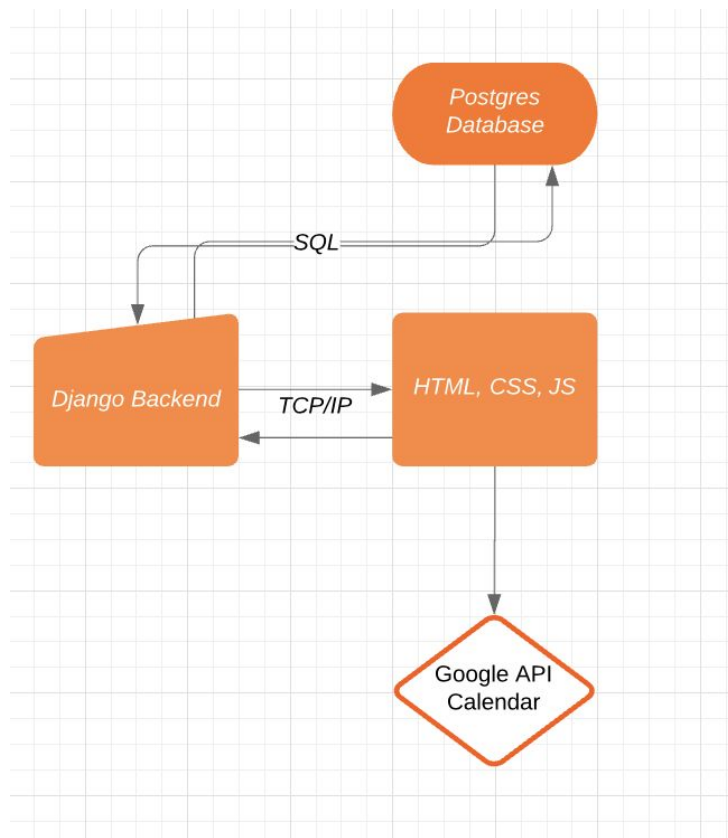
- Database:
  - Functional:
    - Populates data on loaded pages
    - User table
    - Class table
    - Professor table
    - Schedule Table
    - FCQ Table
  - Non-Functional:
    - Obtain ODA spreadsheets to use for data
    - Parse Class PDF to save to our database
    - Setup relational database inside django with postgres
    - Host the database on the cloud
- Website:
  - Functional:
    - User can access homepage
    - Contact-us form that sends us an email
    - Accessible from outside the local network
    - Other features of the app (fcq data, course scheduler, etc.) available to the user on the website
    - Logins and password authentication
  - Non-functional:
    - Framework with Django
    - Database populates information on each page
    - User login persists across multiple sessions with cookies
    - User changes save in database for future use
    - Some cloud hosting service to allow connections outside the local network
- Course Scheduler:
  - Functional:
    - Users can build a schedule through the class search
    - Users can view their schedule in a calendar view
    - Users will be suggested schedules
    - In progress schedules are saved so a user can return to their previous work
  - Non-functional:
    - Parse data from Course PDF to populate database
    - Compare times with current schedule, ensure classes do not overlap
    - Use degree requirements to determine what a user is allowed to take after
    - Save schedule with a foreign key to the user, so the user owns their schedule

- “Rate My Professor” with FCQ:
  - Functional:
    - Suggest professors based on FCQ ratings
    - Allow students to browse professors and view FCQs
    - Feature to search professors by name
    - Integrate with course scheduler to show Professor FCQ data with their class
  - Non-Functional:
    - Create teacher csv with statistics on every teacher from ODA data
    - FCQ table to hold data in our database
    - Professor table to hold professor info in our database
- Profile Pages:
  - Functional:
    - Display user image and information
    - Connect to “friends” or other users
    - Recommend people with similar courses to be friends
    - Users remain logged in with consecutive sessions
  - Non-functional:
    - User table saves profile data
    - Users can modify their data, gets saved in our database
    - Private user information is flagged so only the user can see sensitive information
    - Login authentication to prevent users from editing other profiles
- Degree Audits
  - Functional:
    - A student is able to run a degree audit for any degree
    - Accessible from the webpage
    - Updates recommended courses based on the audit
  - Non-functional:
    - Degree requirements need to be found in ODA spreadsheets
    - Python scripts to parse ODA into readable csv files to save to our database

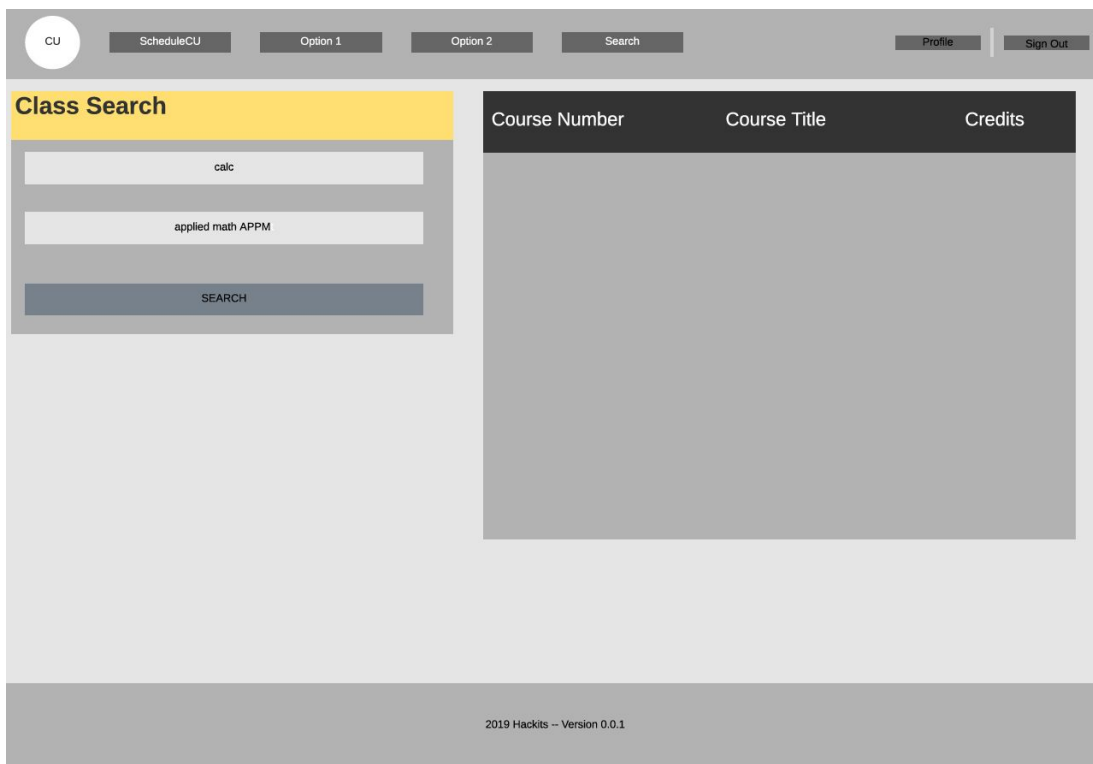
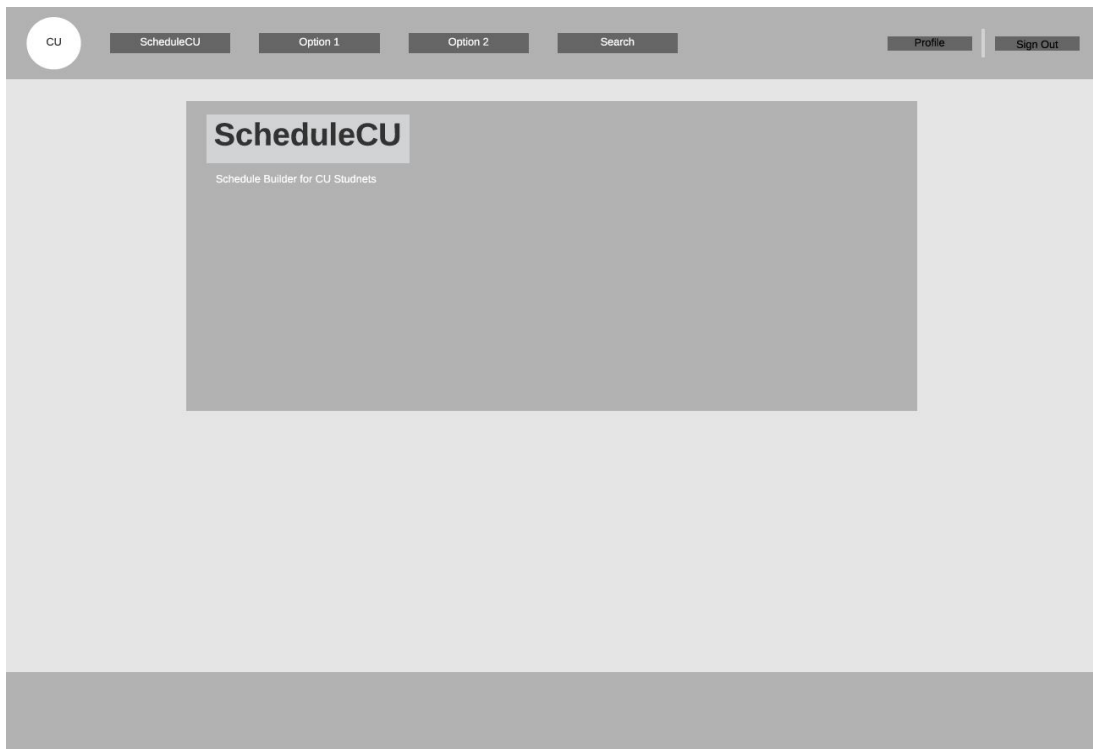
## Database Design (using postgres):

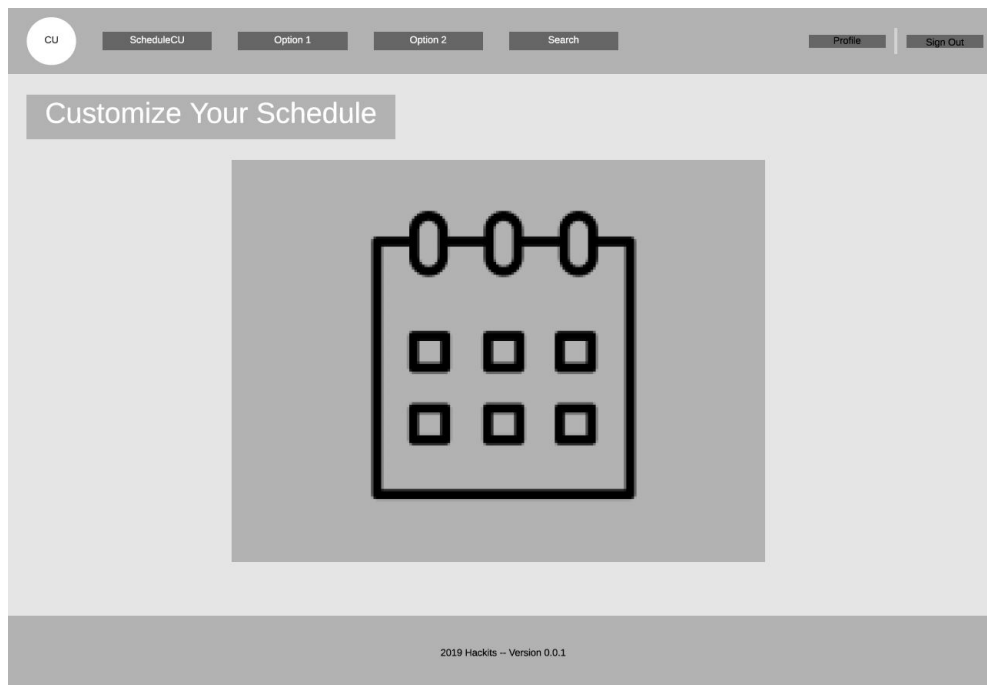
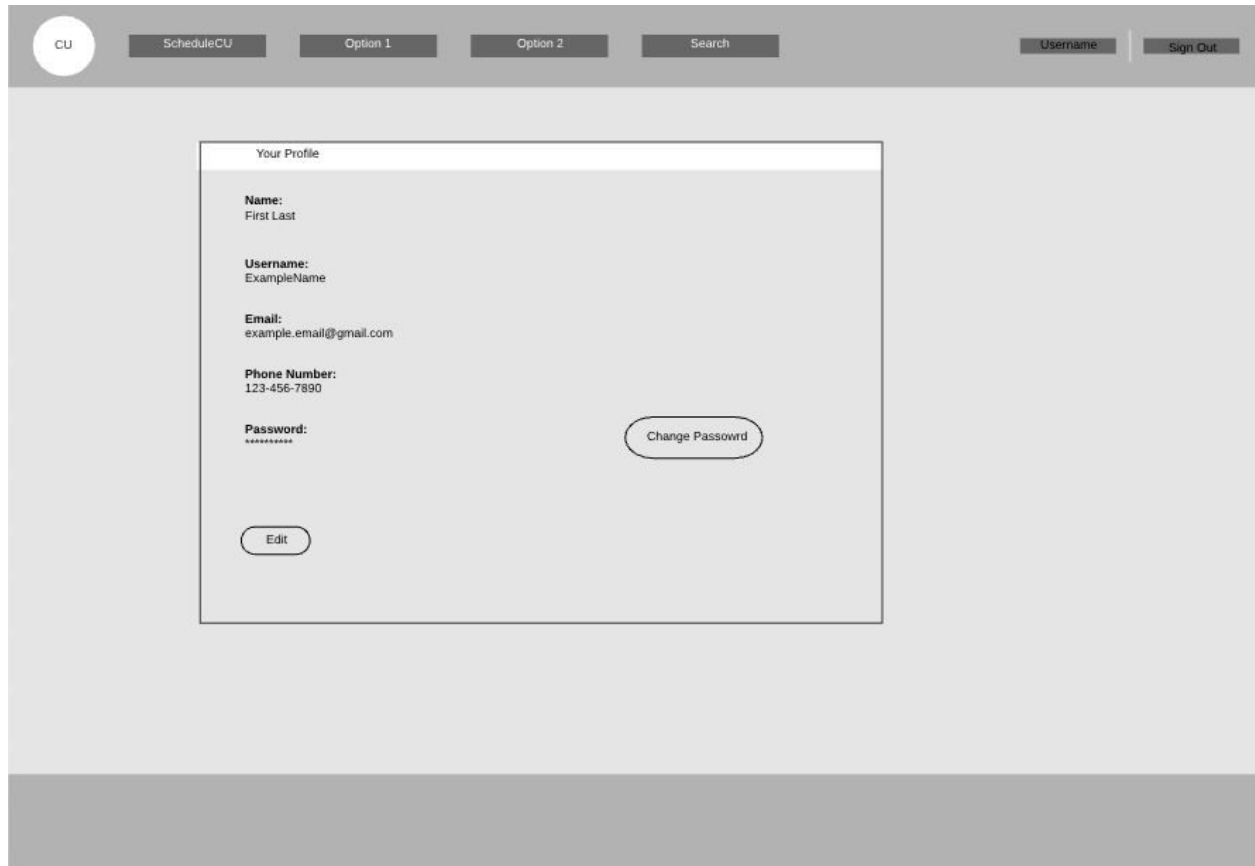


## Architecture Diagram:



Front End Design:





### Web Service Design:

- We are using a Google API calendar in order to display the user's schedule

- User will select classes
- In order to view their schedule, they will be prompted to sign into a google calendar
- We will fetch the user's information from the database
- This data will then be filled in on the user's unique calendar
- The user will then be able to view the class information displayed on the calendar