meeting 2020.02.19

requirements

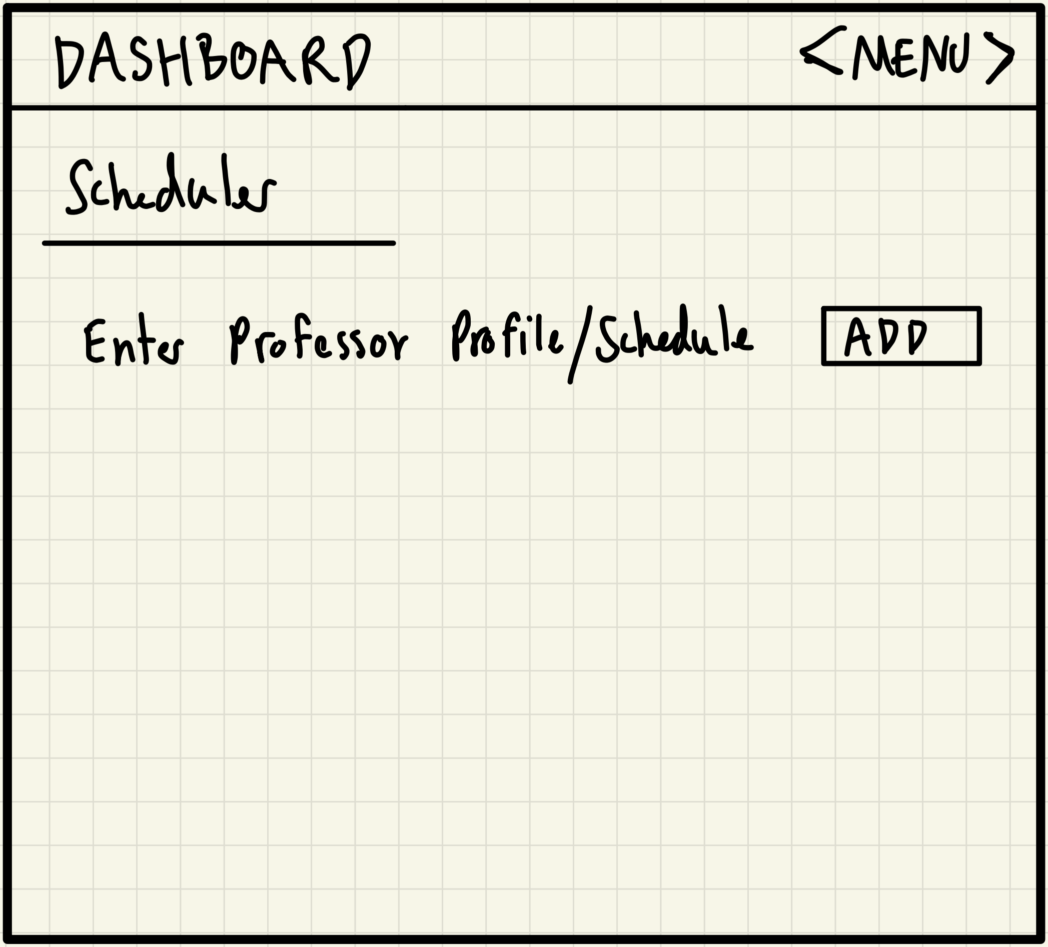
- take in professor schedules
  - exceptions
  - sequences (Prof X must teach after Prof Y)
  - incompat (Prof X cannot teach during Prof Y)
  - locations
  - times
- generate potential schedules

technical stack

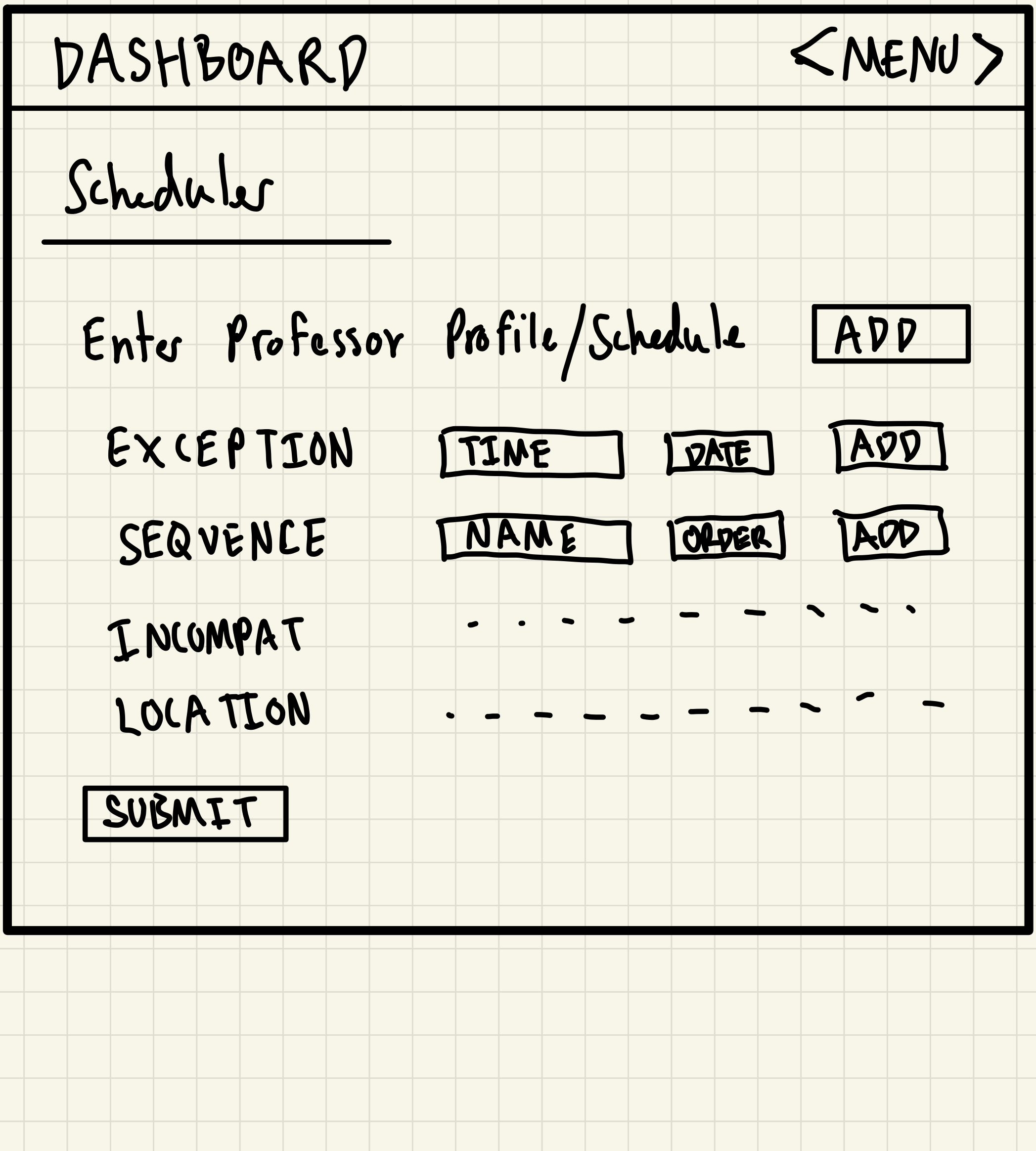
- Python (Back-end)
  - Flask
  - SQLALCHEMY
  - WTForms
  - Bootstrap / Pure.css
  - Jinja
- Digital Ocean (hosting)
  - Linux
  - Nginx
  - gunicorn
  - supervisor

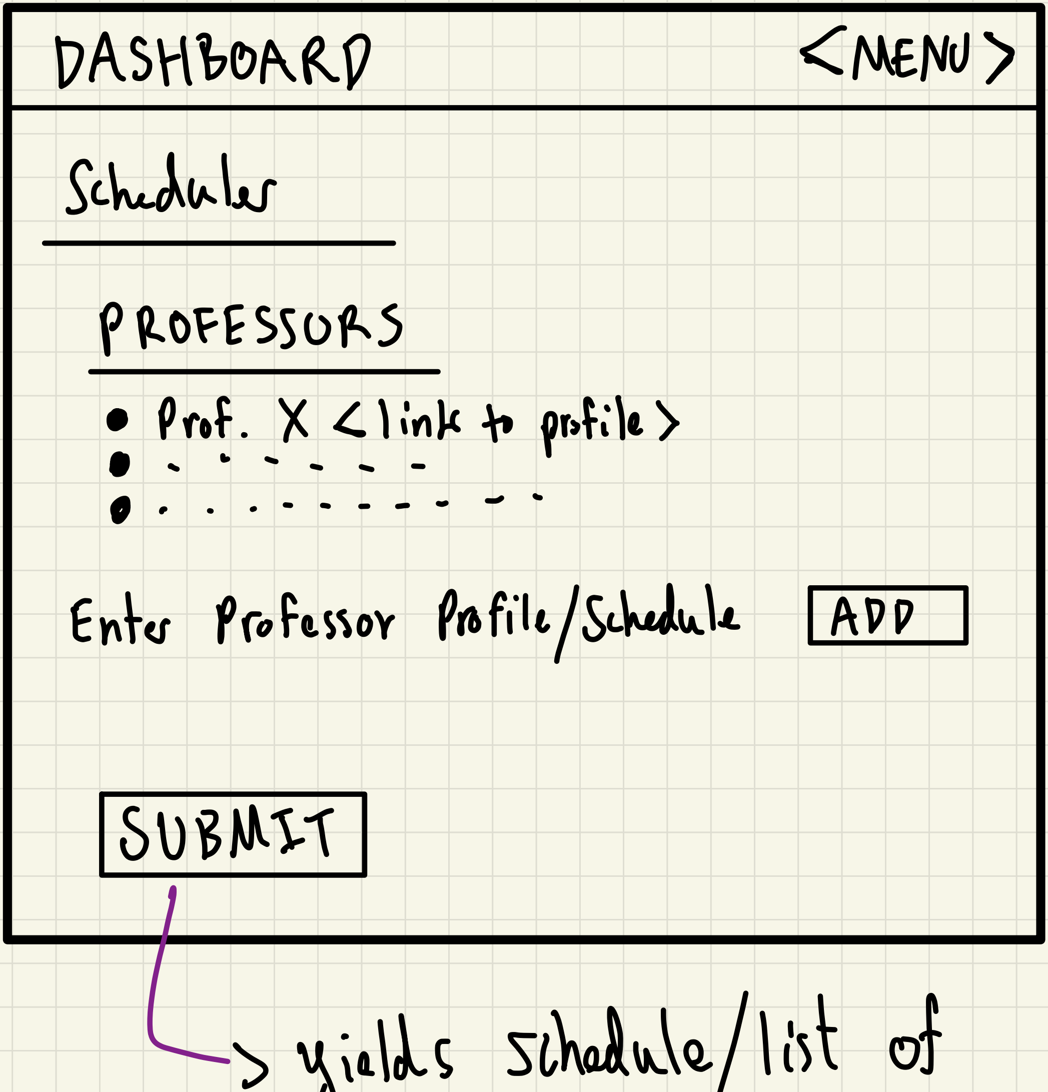
UI Design

browser



Click ADD


DASHBOARD
<MENU>
Scheduler
Enter Professor Profile/Schedule
ADD
EXCEPTION
TIME
DATE
ADD
SEQVENCE
NAME
ORDER
ADD
INCOMPAT
LOCATION
SUBMIT



yields schedule/list of conflicts

Back-end

- Establish dev environment
  - python, ~3.6
    - python version manager
    - python virtual env manger
    - pip
  - docker
    - replication/testing
  - Linux dev env (Ubuntu)
    - Nginx
    - PostgresSQL

- Basic Flask Env/App
  - Hello World
  - Basic API Routes
    - POST/GET Requests on index, '/'
  - integrating with Jinja templates
    - basic hello world example

- Schedule Representation
  - types of "exceptions"
    - time
    - sequence
    - interval
    - day
    - preferences

    Some type of Class/ Data representation

  - Concept of professors
  - Concept of classrooms/Courses

- Schedule Resolver
  - how to reconcile exceptions?

- Data permanence / Storage
  - data serialization (Pickle lib)
  - plain text
  - SQLite (Postgres later)

Contract

- Figure out real constraints
- Core deliverables
  - Scheduler
  - web app
- Risks
  - group project
  - knowledge gap
  - time limits
  - nature of problem