

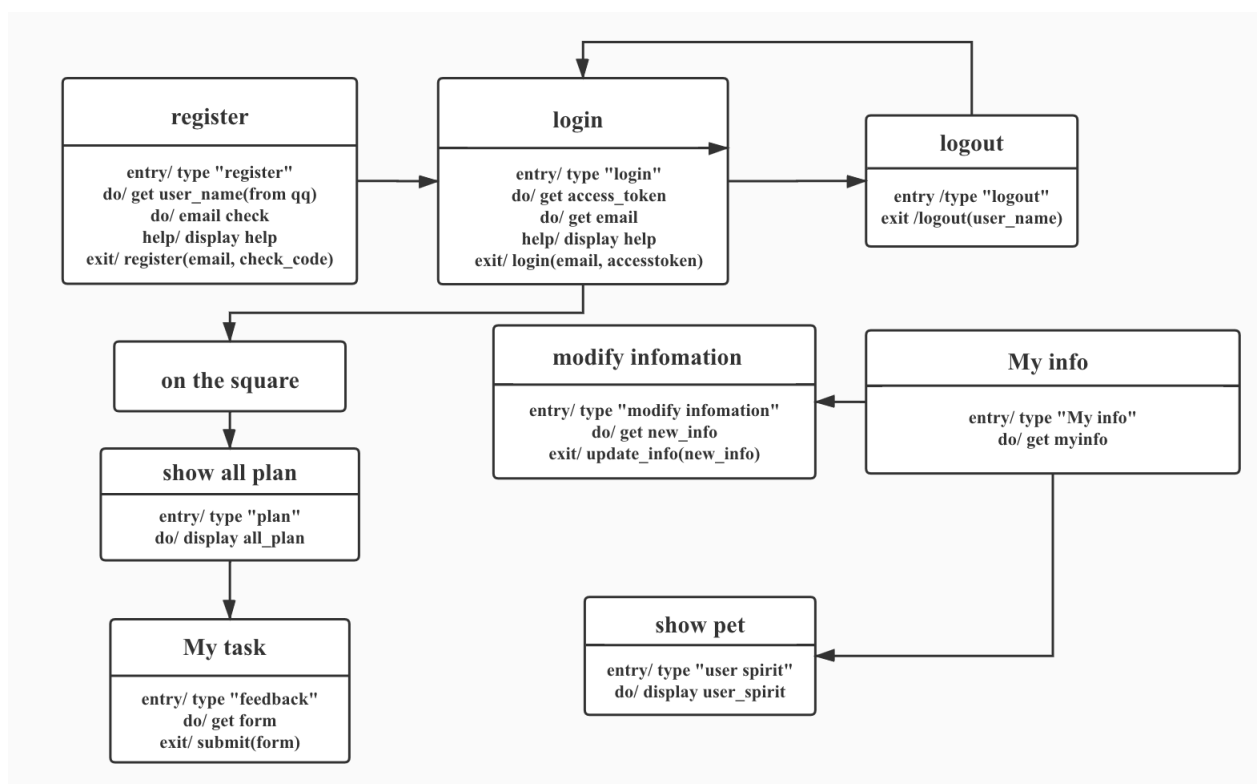
Cornerstone——基于双人打卡的竞争学习系统

复旦大学计算机科学与技术 16307130335 方煊杰

1 项目简介

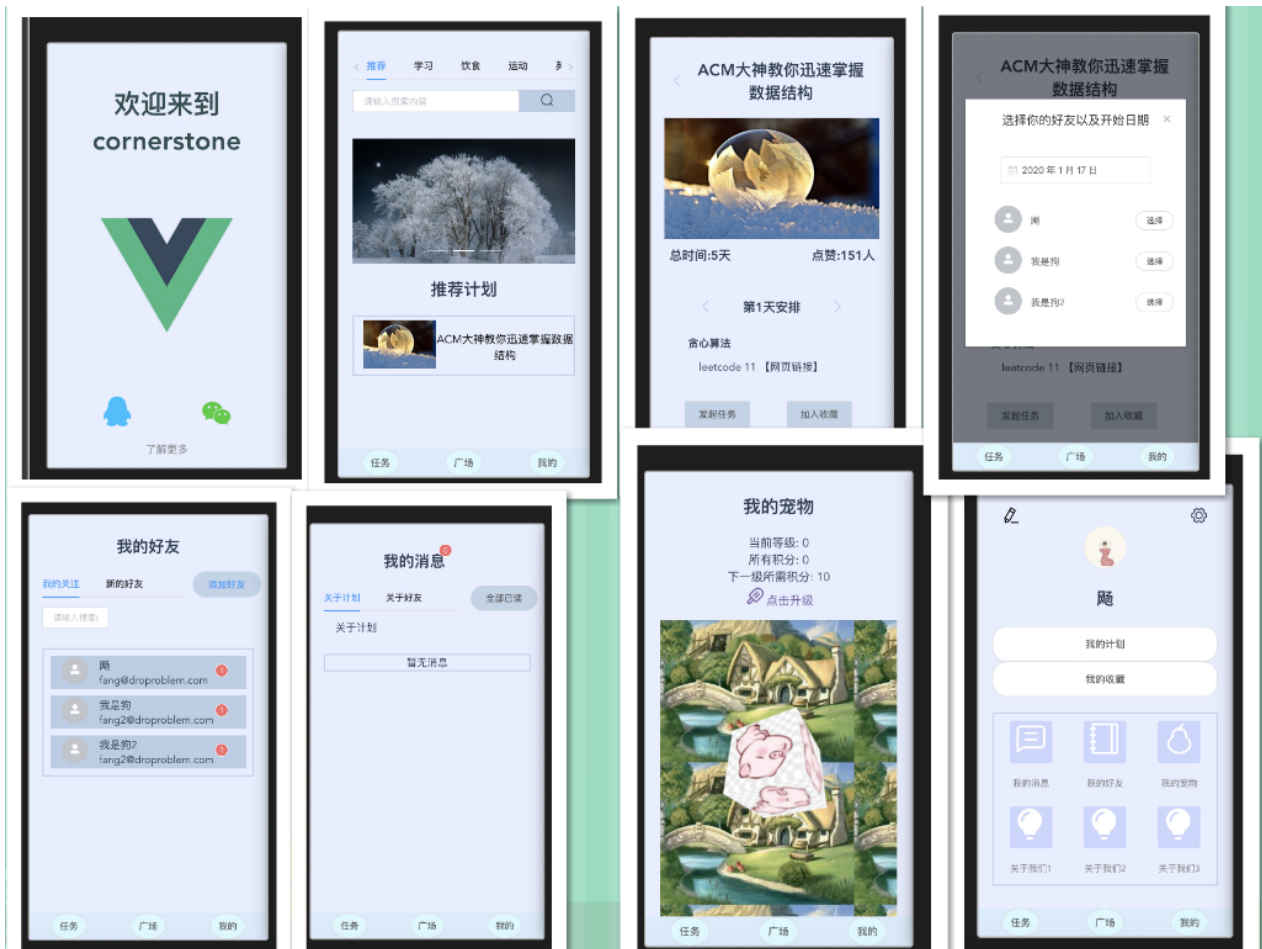
- 名称: **cornerstone**
- 功能: 基于双人互相打卡的学习系统
- 设计思想: 由于现如今我们有很多学习类的 app, 有各种单词打卡, 健身计划, 舞蹈教学等等, 但是能够坚持不断学习的人很少。我认为监督是促进坚持的有效方法, 希望基于此能够实现快乐组队, 共同进步

2 项目设计



设计模型可访问: <http://116.62.46.96/>

3 项目效果



因为空间有限，就不全部截图了，大概效果如上图。

4 运行

前端名： cooperation_front3

后端名： cooperation

因为当前后端已经部署在服务器上，所以直接运行前端即可。

```
$ cd cooperation_front3
$ cd vuefront
$ npm install
$ npm run build
$ cd ..
$ cordova run android
```

若要部署后端：

```
$ cd cooperation
$ docker build -t cooperation .
$ docker run -d --restart always
    -v /srv/cooperation:/everything -p 3000:8000
    --name cooperation cooperation:latest
```

此时访问localhost:3000即为api服务

5 项目工作及难点

- qq第三方登录：

- 申请过程十分复杂，在网页申请久久无法成功后转为申请应用宝软件，伪造了纯前端app上传才得以通过。
- qq提供的接入第三方sdk只有android studio (java) ，以及js sdk，但由于我的项目是cordova+vue，所以不考虑java sdk，而js sdk只支持网站接入（即申请的应用为网站，需要域名等条件），所以在我的项目中不可行。
- 解决：最终采用cordova的qq插件，在vue中集成后调用。但是此时开发环境下是无法调用插件的，只有真机安装后方可调用，对调试来说是个蛮麻烦的问题。

- 无底洞的界面细节

只能不停改了

- 跨域访问

- 采取的后端框架为sanic，这个框架较为小众（为了异步），如果说在vue的proxy_table中路由重定向，部署后并不能访问，因此考虑在后端处理跨域。
- sanic自带中间件关于跨域的库为sanic_cors，但是在使用过程中发现，如果我返回的是exception，并不可以成功跨域。

我中间有这样的返回：(重载了exception class，抛出指定异常)

```
▶ {type: "CornerstoneError", uuid: "CornerstoneError:1", code: 1, error: "宁没有登录"}
```

```
def assure_user_login(handler):
    @_wraps(handler)
    async def ensure_user_r_handler(request, *args, **kwargs):
        user = request['session']
        if user == {}:
            raise UNBOUND_USER(reason="宁没有登录")
        return await handler(request, *args, **kwargs)
    return ensure_user_r_handler
```

- **解决**：看了源码后发现，是因为sanic_cors中关于exception的返回函数忘记加async修饰，变成了同步函数，所以会被handler阻止，之后修改源码后成功返回。

- **three.js 的引入**

vue 中引入3d是坑中之坑

- 用于vue有个this指针的问题，这个劳什子 three.js 引入时，this内部找不到three.js, 也就是说根本调不到???
- **解决**：后来参考了小颜的做法，将three.js以及它的相关一堆插件js文件，在最外层增加 export function () {}，将它们作为函数export出，然后再在对应vue文件中引入变量绑定，这才能够成功调用。

- **docker 部署**

这也是为了放在服务器上我更新方便，不然每次要手动找一波python进程再kill掉

- 由于墙的原因，如果直接按照普通步骤（比如docker pull unbuntu , pull python3）既慢又大，还有可能中间挂。
- **解决**：采用国内别人写好的部分镜像，**FROM taoliu/gunicorn3**其中已经包含了python等必须，就可以少写dockerfile文件，能够快速生成对应镜像。

6 代码量：

后端：

```
xuanjiedeMacBook-Pro:Desktop fang$ cloc cooperation
  44 text files.
  44 unique files.
   5 files ignored.

github.com/AlDanial/cloc v 1.82  T=0.11 s (381.9 files/s, 46614.6 lines/s)
-----
Language                  files      blank     comment      code
-----
Python                     37         712         874         2906
XML                        4           0           0          613
Dockerfile                 1           6           0           16
-----
SUM:                       42        718         874         3535
xuanjiedeMacBook-Pro:Desktop fang$
```

前端

```
xuanjiedeMacBook-Pro:vuefront fang$ cloc src
34 text files.
34 unique files.
1 file ignored.

github.com/AlDanial/cloc v 1.82 T=0.10 s (345.2 files/s, 86575.9 lines/s)
-----
Language               files      blank      comment      code
-----
JavaScript              7          1694          235          2654
Vuejs Component        21           415           31          2410
CSS                     2            89           19           459
HTML                   1             18            2           196
JSON                   1              0            0            23
SVG                    1              8            3            21
-----
SUM:                   33          2224          290          5763
-----
xuanjiedeMacBook-Pro:vuefront fang$
```

应该要去掉一些，因为魔改了three.js等文件，可能这个也算在其中了

再次感慨，前端真是劳动密集产业啊（当然和我对js的复用能力不强也有关）