**Team CACTUS Iteration 1: groupwork record**

**Meeting minutes**

[Note: I noticed that we should preserve our timestamp for meeting minutes. However, I record each meeting on my local computer in this word file and didn’t git push every time , so here’s a proof that I didn’t start doing it until deadline. I created the file a week ago and modified it each day. I also uploaded the teamwork summay on Teams days ago.]



From Team cactus 3.7

**Overall teamwork situation**

* Our team planned to have two synchronous online meeting through Microsoft Teams each week, and each member participated in the meeting to discuss the current progress of the project. Meanwhile, we used group chat and gitlab issue boards to hold asynchronous standups on a daily basis. Apart from that, during lab sessions, we also discuss the issues we met during implementing the project and set future goals.

In week 2, we’ve achieved satisfactory outcomes during both online meeting sessions and asynchronous standups. About half of Iteration 1 has been finished, and we tried to help each other with arising issues. If the issue couldn’t be fixed, we went to help session to figure that out.

We tried pair programming. Since there are 4 members in total, we have 2 pairs, reviewing each other’s codes and merge requests, as well as track each other’s progress. Pair programming works well for us.

In week 3, we found two meeting on Teams wasn’t enough. As we had 4 meetings on Team, we also use zoom to have meeting in pairs. We could track each other’s progress, review others tests and implementations through each meeting. Most importantly, meeting help us solve any merge conflicts, potential data structure difference, and misunderstanding of specification.

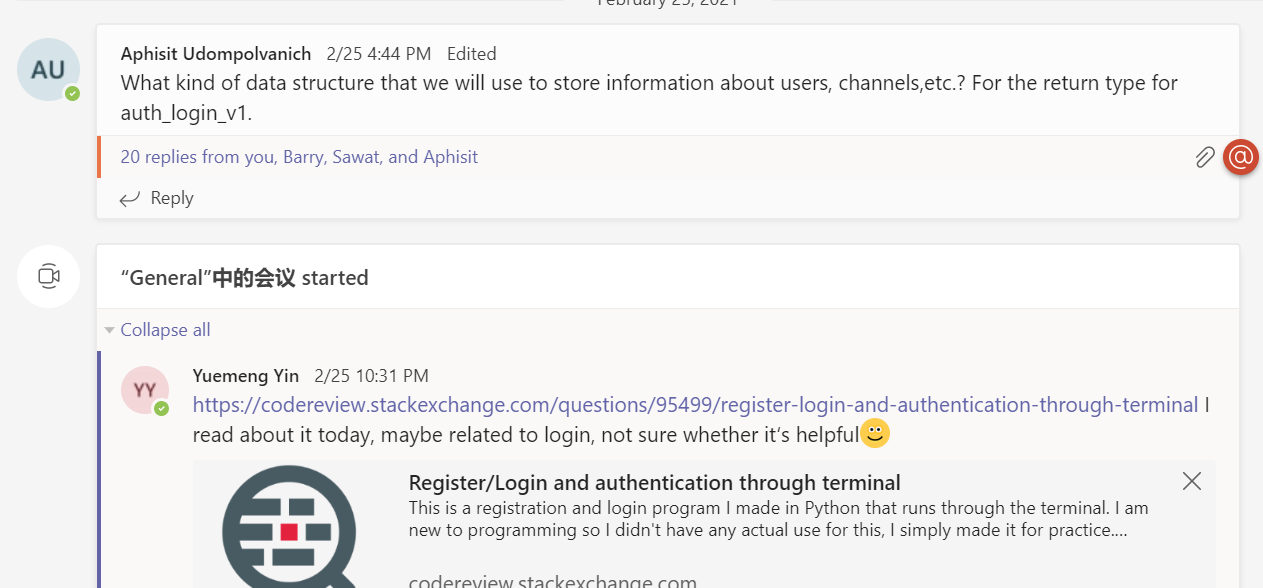
Everyone in the team has their own schedule, so it’s a challenge to find a suitable meeting time for everyone. We managed to have every member participated in Teams meeting twice a week, but if anyone couldn’t attend, we would summarize the meeting content and message him/her later. Another challenge is that different members have different coding preferences, some work on vlab and some work on their own machine, therefore, it’s essential to keep in-time communication and track the overall progress through chats and gitlab history.

2.25 Group first meeting

1. a brief read together over the project Iteration 1 interface

2. division of between member and add issues on gitlab issue board

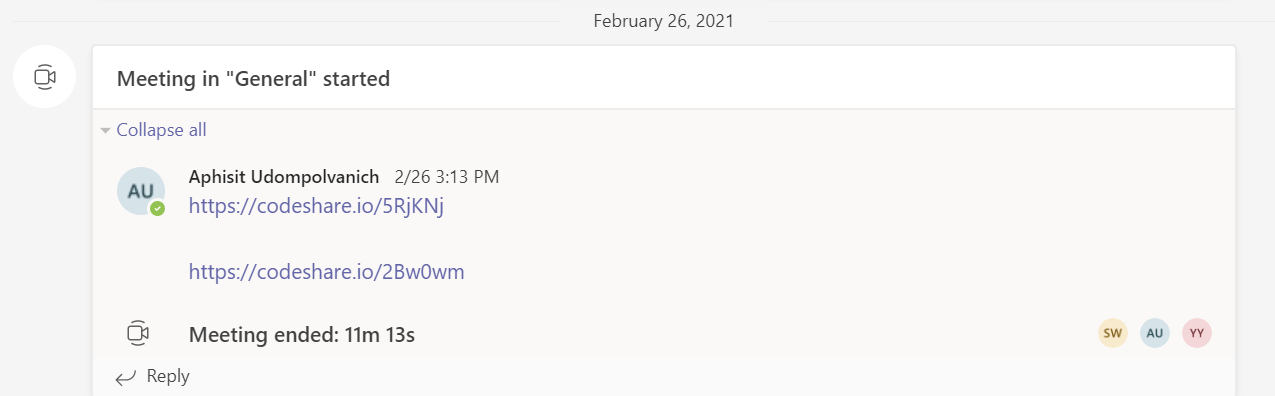
3. sharing of ideas on function return type and git merge



Meeting lasts for: 31m 40s

2,26 Group second meeting

1. Helping each other out: Aphisit share ideas on how to start with auth\_login
2. Discuss our current progress and keep everyone updated
3. Raise the suggestion of working in pairs : Yuemeng and Barry; Aphisit and Sawat since each pair of members are in the same time zone



Meeting lasts for: 11m 13s

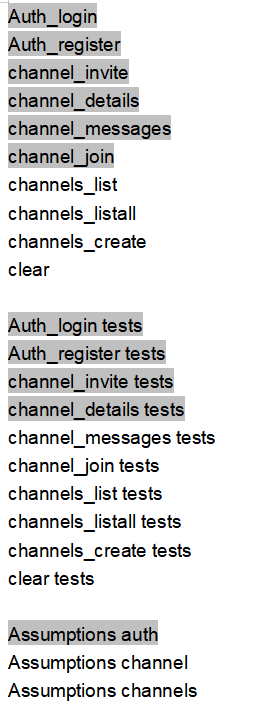
2.28 Project assessment 1

1. Summarization of the work each member have done
2. Barry read other members code and help with debugging
3. Rearrange project plan to quicken the pace



3.3 Project Meeting 4 (in pairs through zoom)

1. Discuss the current failing tests, redo some part of the functions implemented
2. Add more assert tests after learning from Hayden’s lecture
3. Record our progress through listing (the highlighted ones are done, but nothing merged)

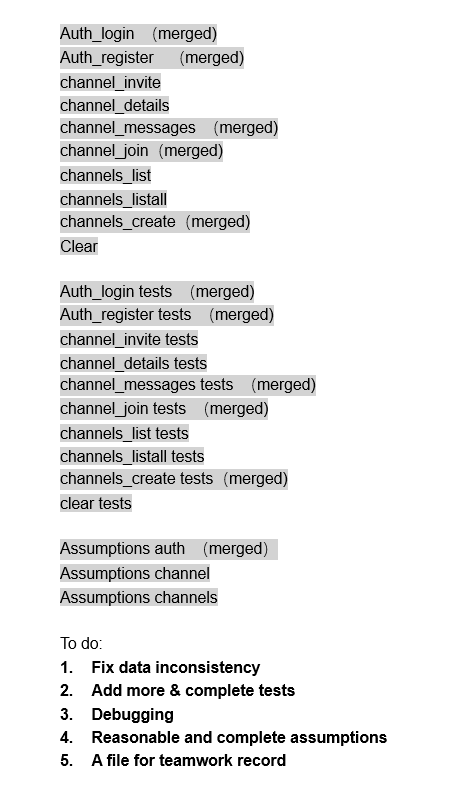


* 1. Project Meeting 5 (in pairs through zoom)

1.Discuss the current failing tests, redo some part of the functions implemented as same as yesterday

2.minor changes to code due to database difference, finalise the data structure in database.py

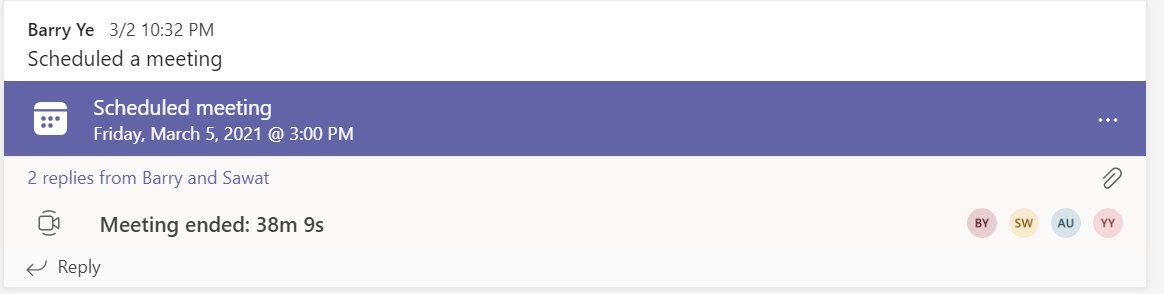
3.Record our progress through listing (the highlighted ones are done, and part of them are merged into master)



4. add the new issues to gitlab issue board

* 1. project assessment 2

We are still working on making sure database use consistency between each member, on branch fix\_datafile\_inconsistency, this really takes like forever to fix, and consume so much of our time, we learnt that **we better make sure data structure is clear to everyone before starting instead of fixing it later,** this is really a lesson for our group, and we will do better on that next time

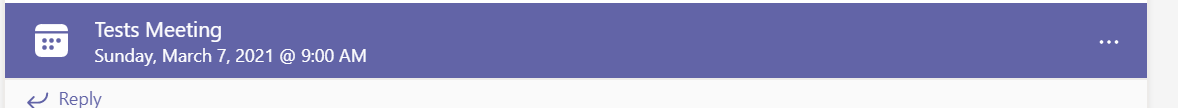


Meeting lasts for : 38m 9s (basically discussing about database the whole time)



(fixing database)

* 1. Tests meeting



Meeting lasts for: 31m 44s

1. Each member add more tests to ensure complete test coverage
2. Fix issue like **assertion error, key error**
3. Discuss the unclear idea of **handle generation** together, and modify auth\_register codes
4. Check Yuemeng’s assumptions for auth\_login and auth\_register, delete two that’s already in specification
5. Fix some bug with channel\_details

Auth\_login （merged)

Auth\_register （merged)

channel\_invite （merged)

channel\_details （merged)

channel\_messages （merged)

channel\_join（merged)

channels\_list （merged)

channels\_listall （merged)

channels\_create（merged)

Clear（merged)

Auth\_login tests （merged)

Auth\_register tests （merged)

channel\_invite tests （merged)

channel\_details tests （merged)

channel\_messages tests （merged)

channel\_join tests （merged)

channels\_list tests （merged)

channels\_listall tests （merged)

channels\_create tests（merged)

clear tests（merged)

Assumptions auth （merged）

Assumptions channel （partially merged)

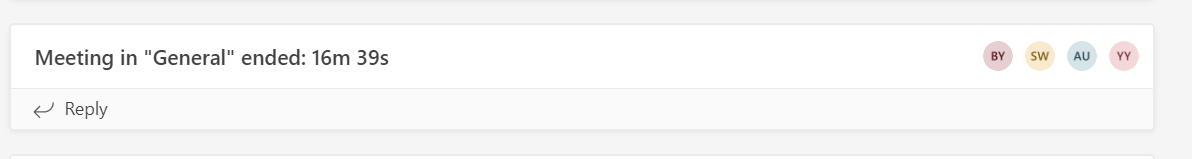
Assumptions channels

**Merge into debug\_ver1 branch**

To do:

1. **Add more & complete tests**
2. **Debugging**

* 1. last meeting: finalization



1. docstrings for each function
2. tests coverage
3. finish some issues on gitlab issue board

In this meeting, we summed up our work, and see if there’s still more adjustments to make. It is generally a very good but tiresome project work, we learnt a lot in iteration1: python, pytest, git, SDLC, groupwork, leadership, compromise, responsibility ….

Prepared to start on iteration 2 next week.