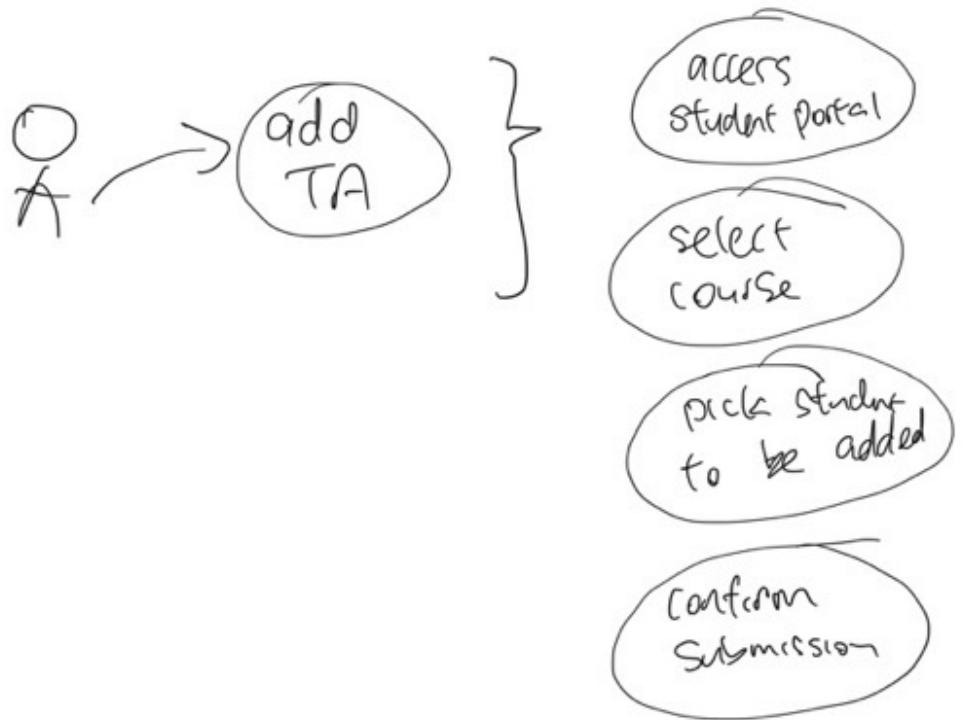
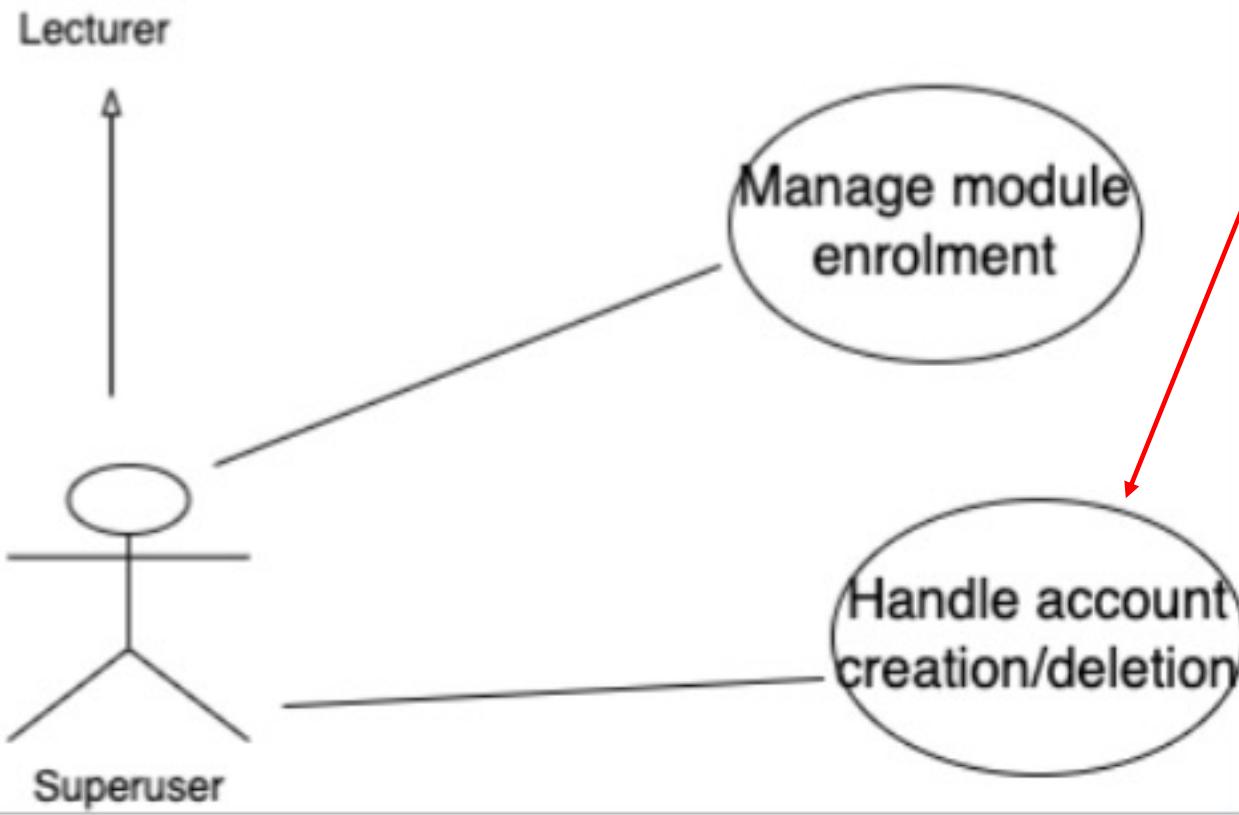


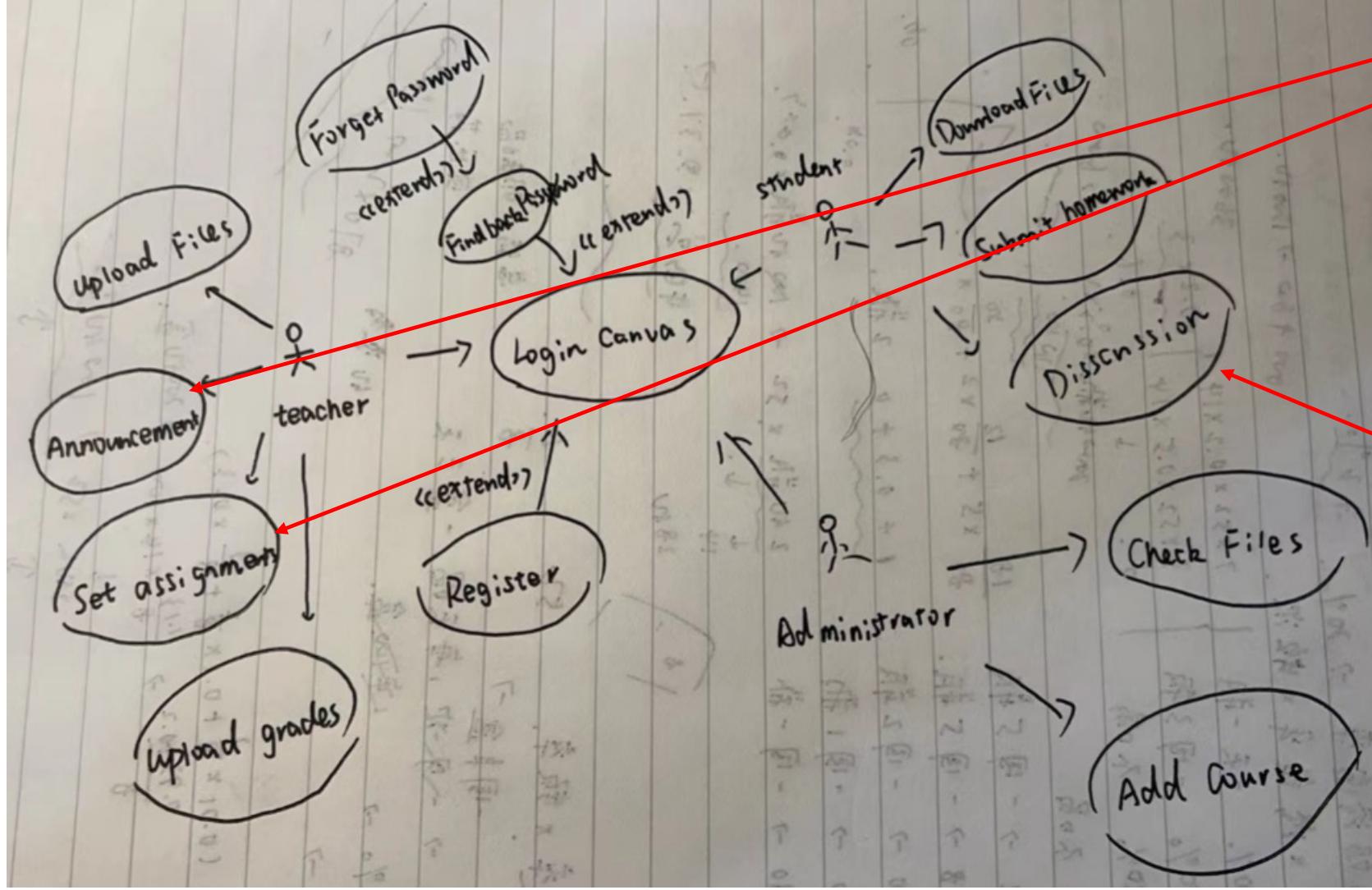


We don't provide the steps of a use case in the use case diagram

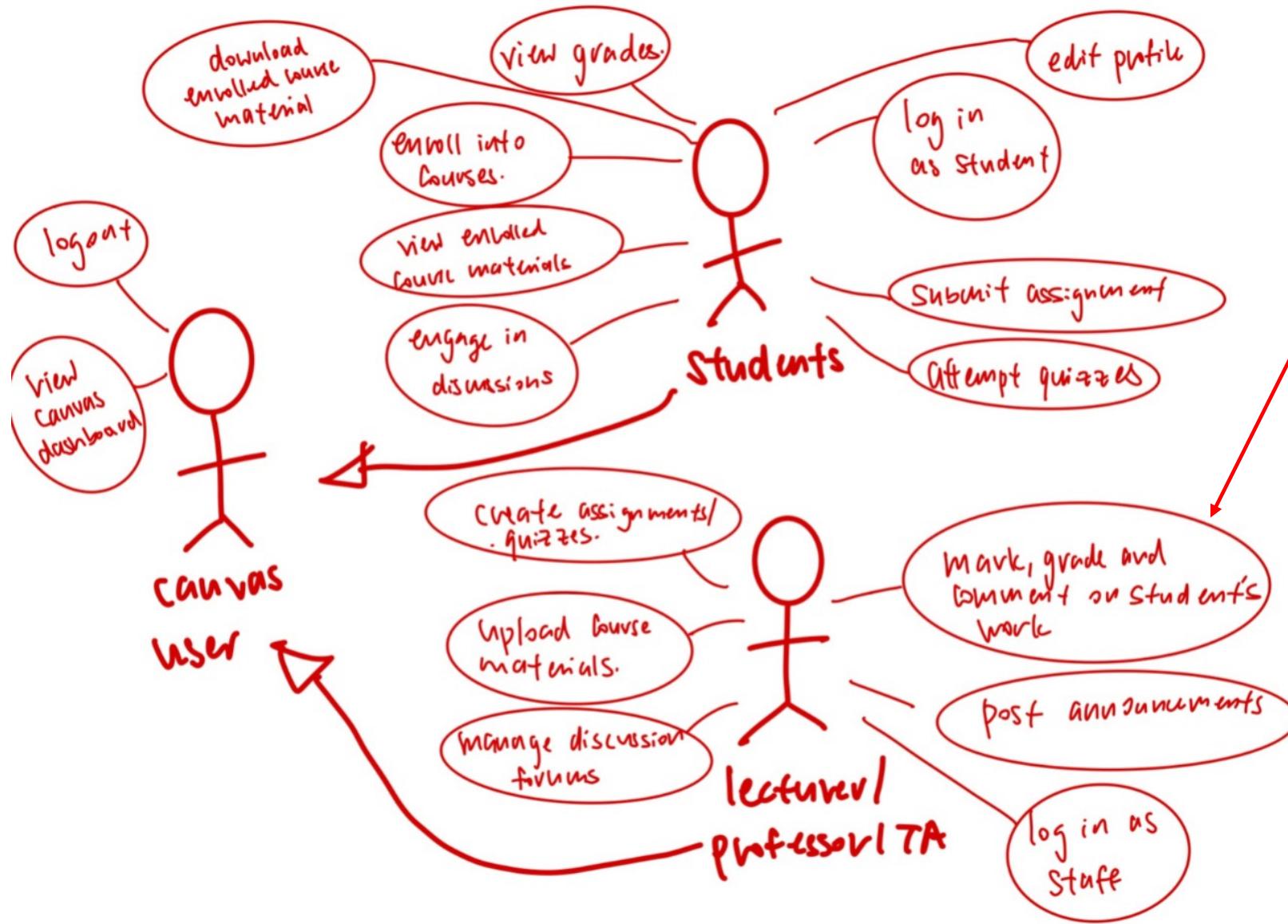


1 use case should be 1 action, don't combine use cases together unless they are the same system functionality



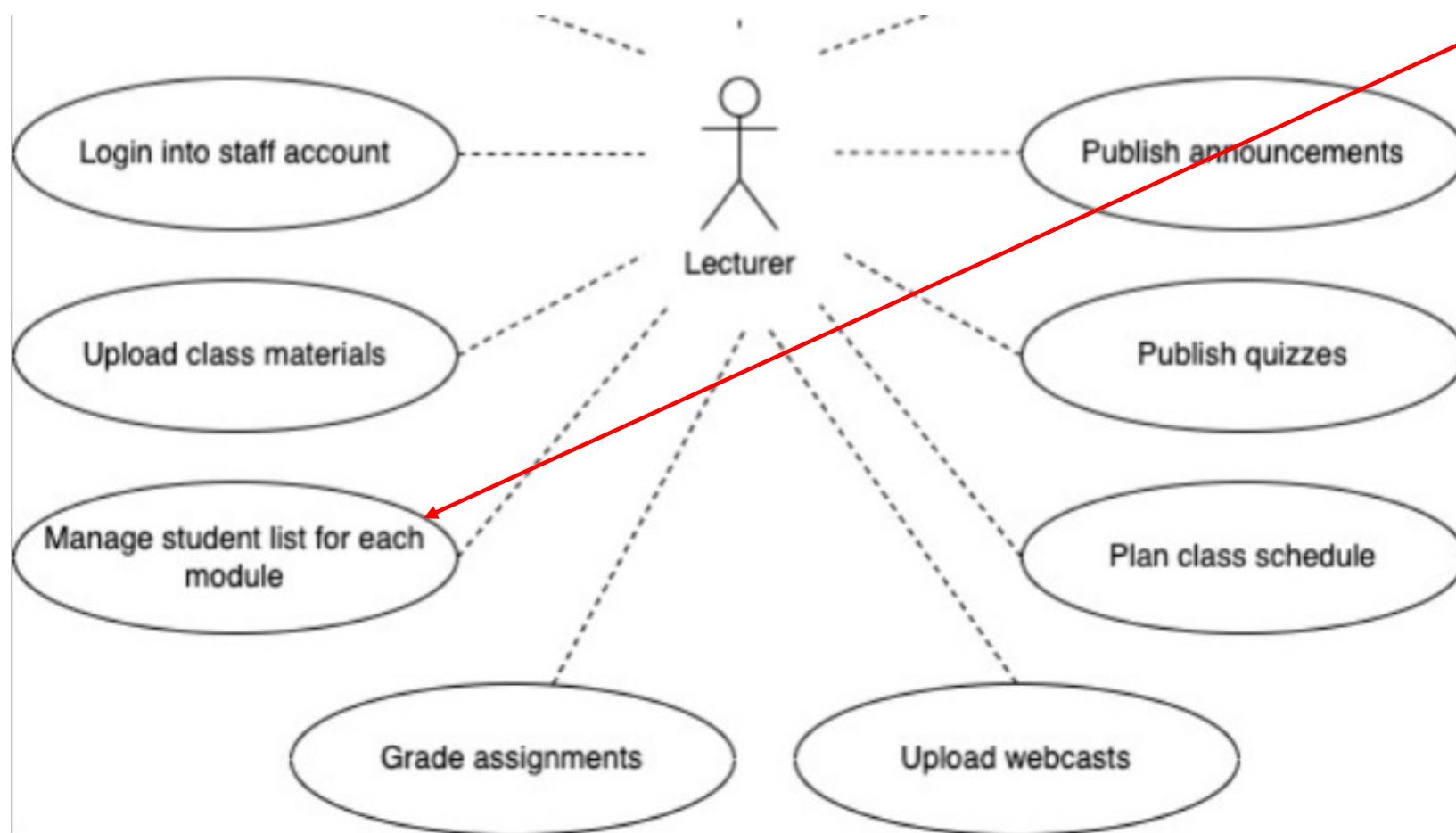


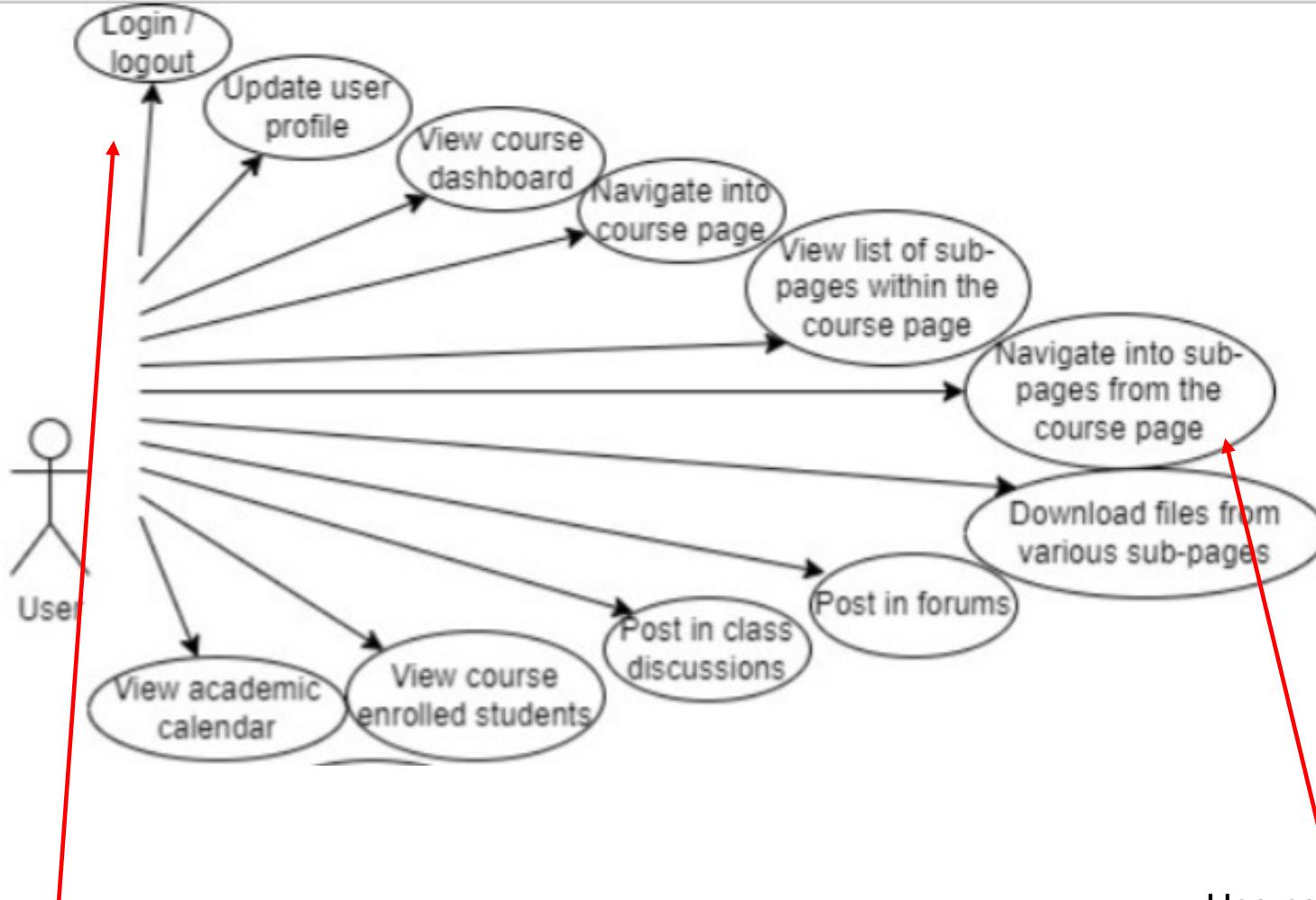
should think of each use case as an action that the user is doing rather than describing an idea of the system functionality
e.g. "announcement" is not clear
"create an announcement" is a use case
"view a list of announcement" is a separate use case, etc
comment: should think of each use case as an action that the user is doing rather than describing an idea of the system functionality e.g.
"announcement" is not clear
"create an announcement" is a use case "view a list of announcement" is a separate use case, etc



Use case should describe the system action rather than describe the objective or the rough idea

Don't combine use cases. Should break into the individual possible actions



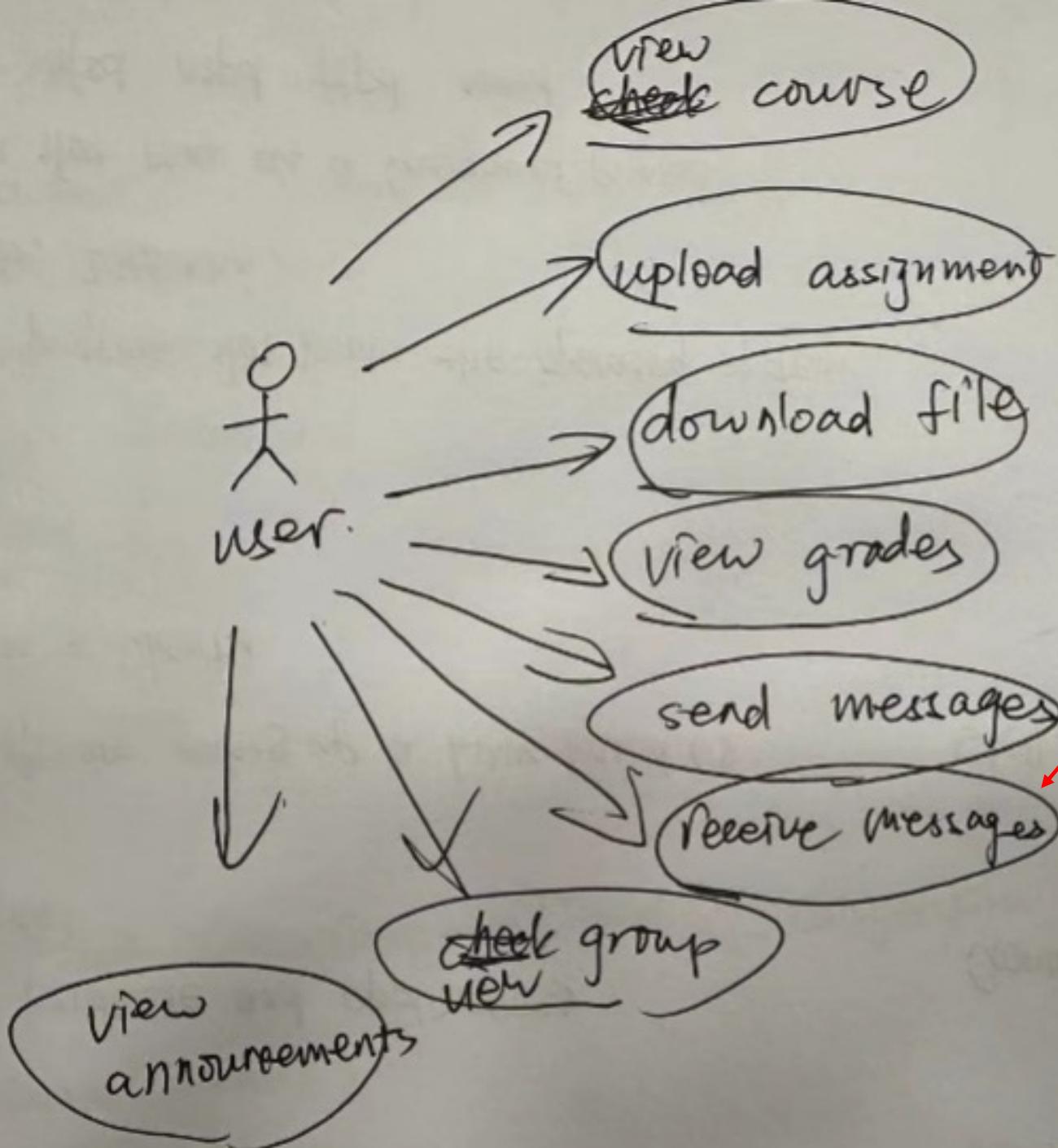


Do not combine use cases.
They should be 2 separate
action that a user can do in the
system

Use cases are not pages that
one can see

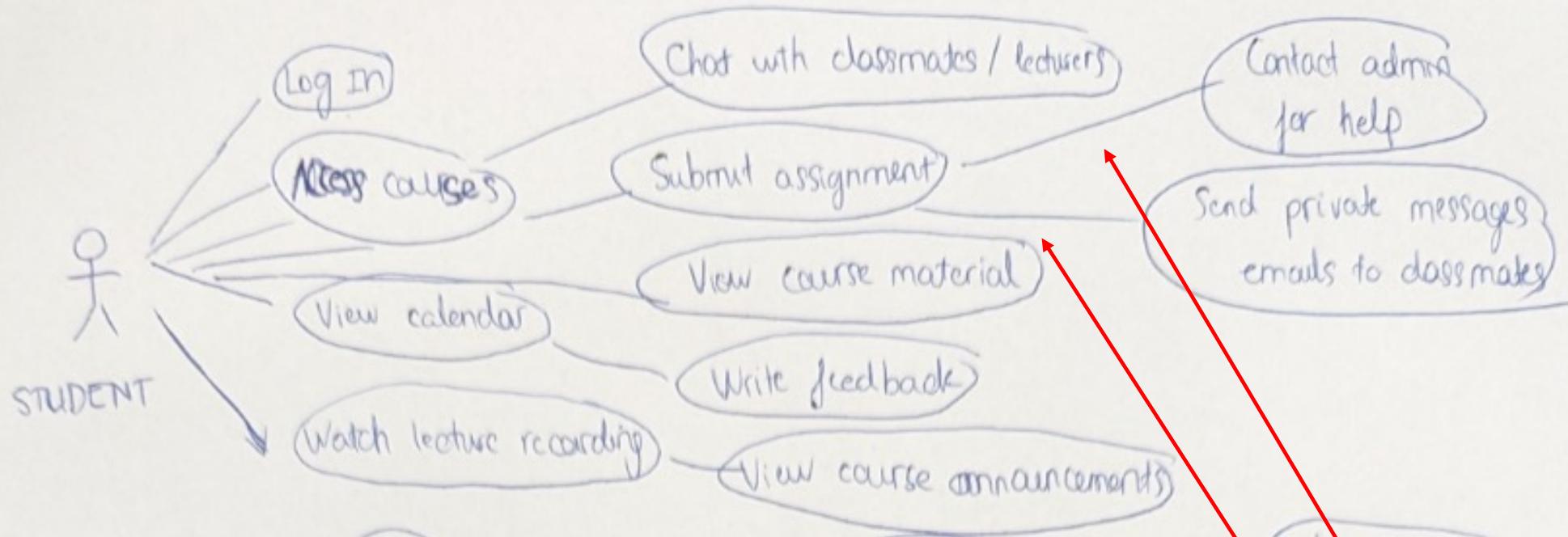


This is very vague what it means.
Whether it is:
View a list of files
Download a file
Download a list of files
... or what



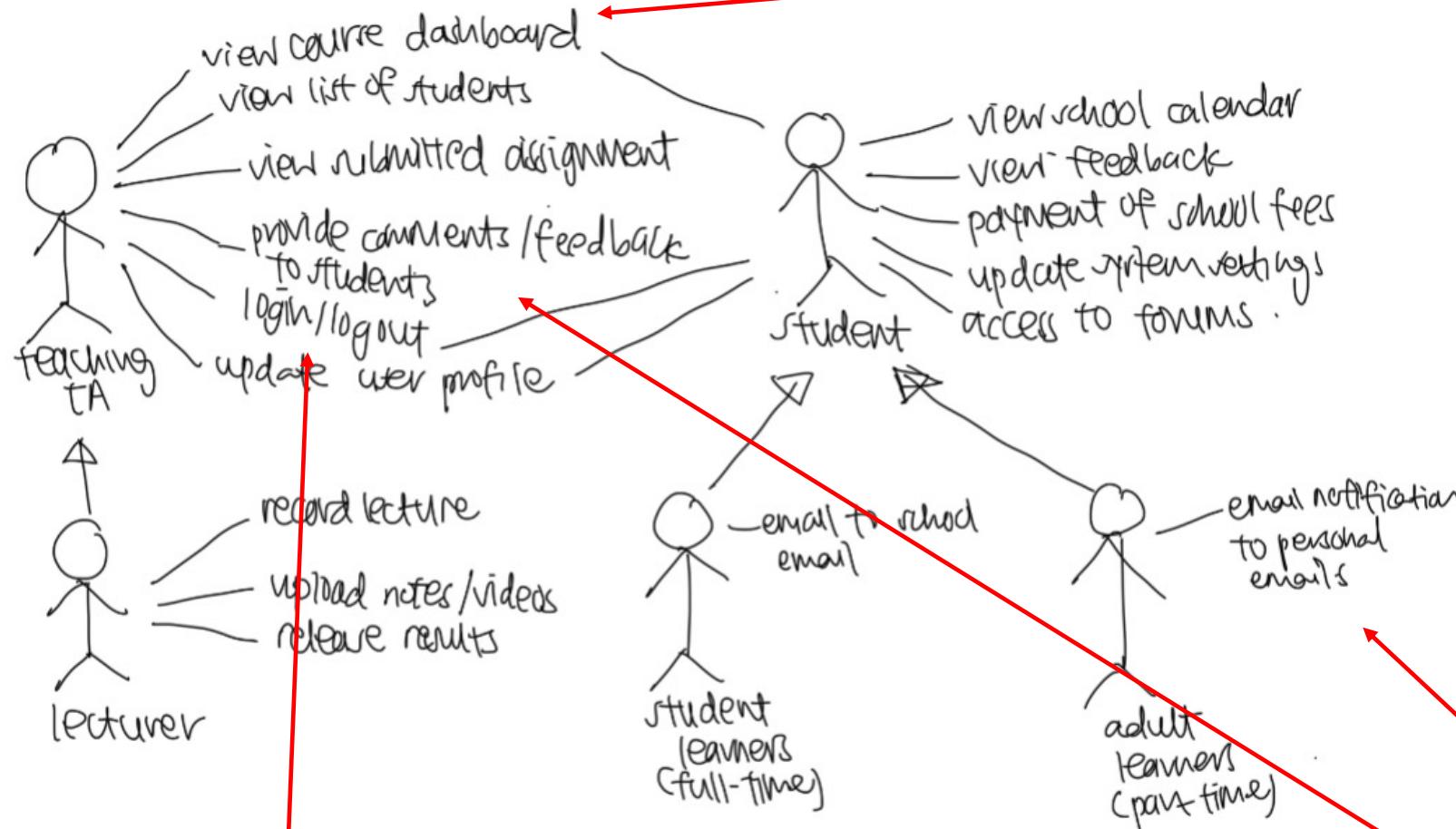
Use cases should be seen from the point of view of the user rather than describing what happens (i.e. if the user wants to see his/her message, the user will execute “view all messages” instead)

CANVAS USE CASE DIAGRAM



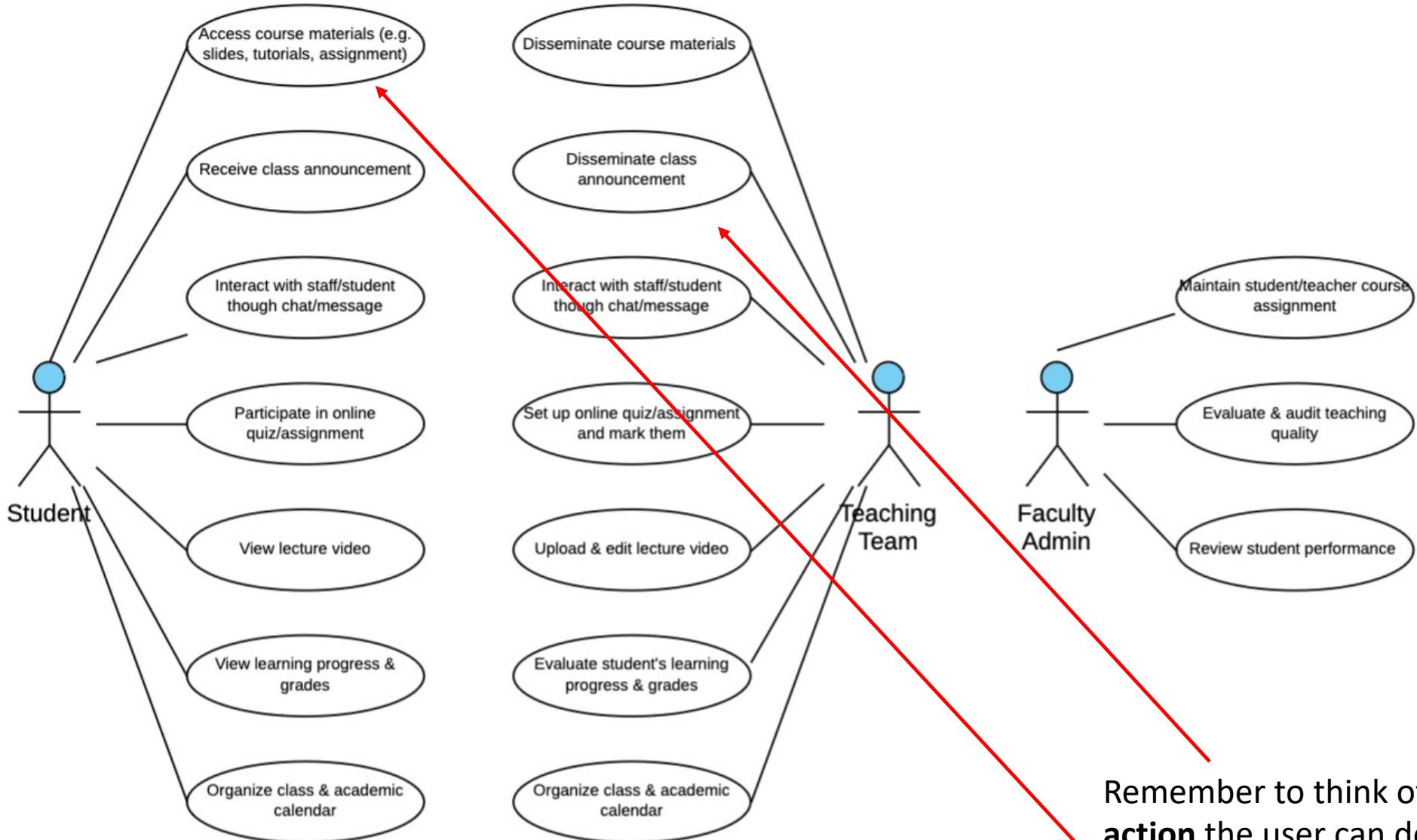
Don't link use cases together
and also don't draw it such
that it looks like they are linked

Use cases are not pages

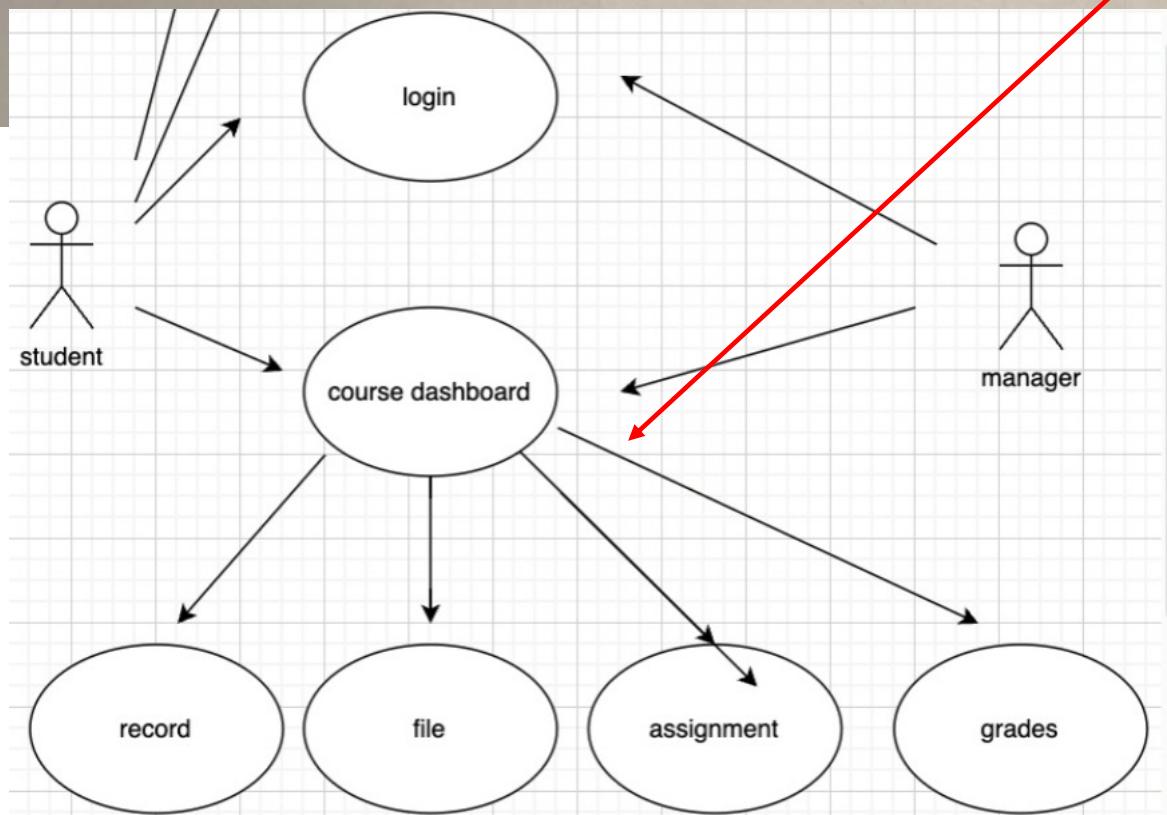
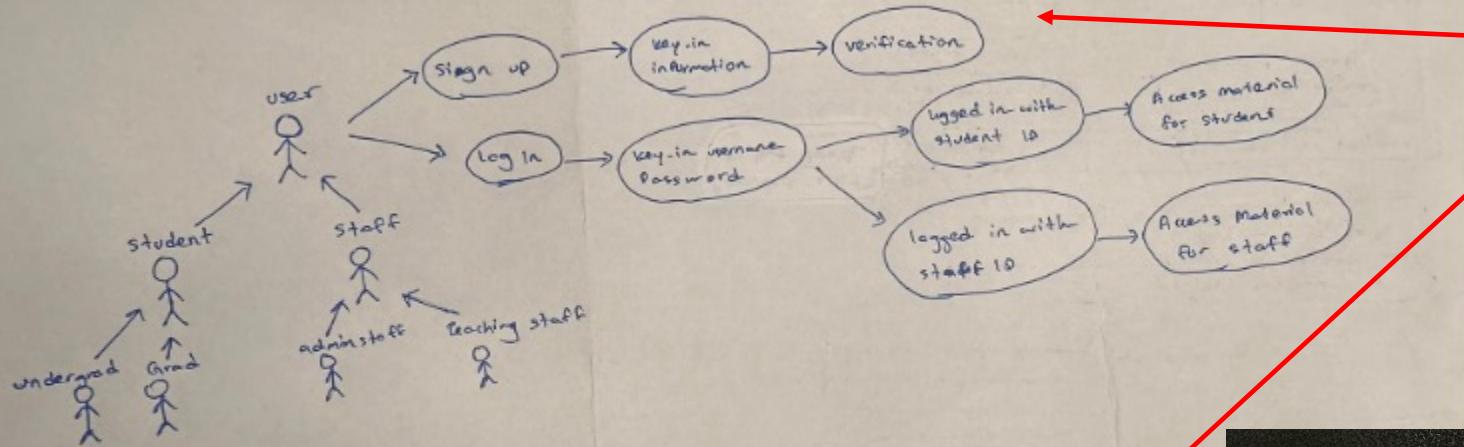


Don't combine use cases

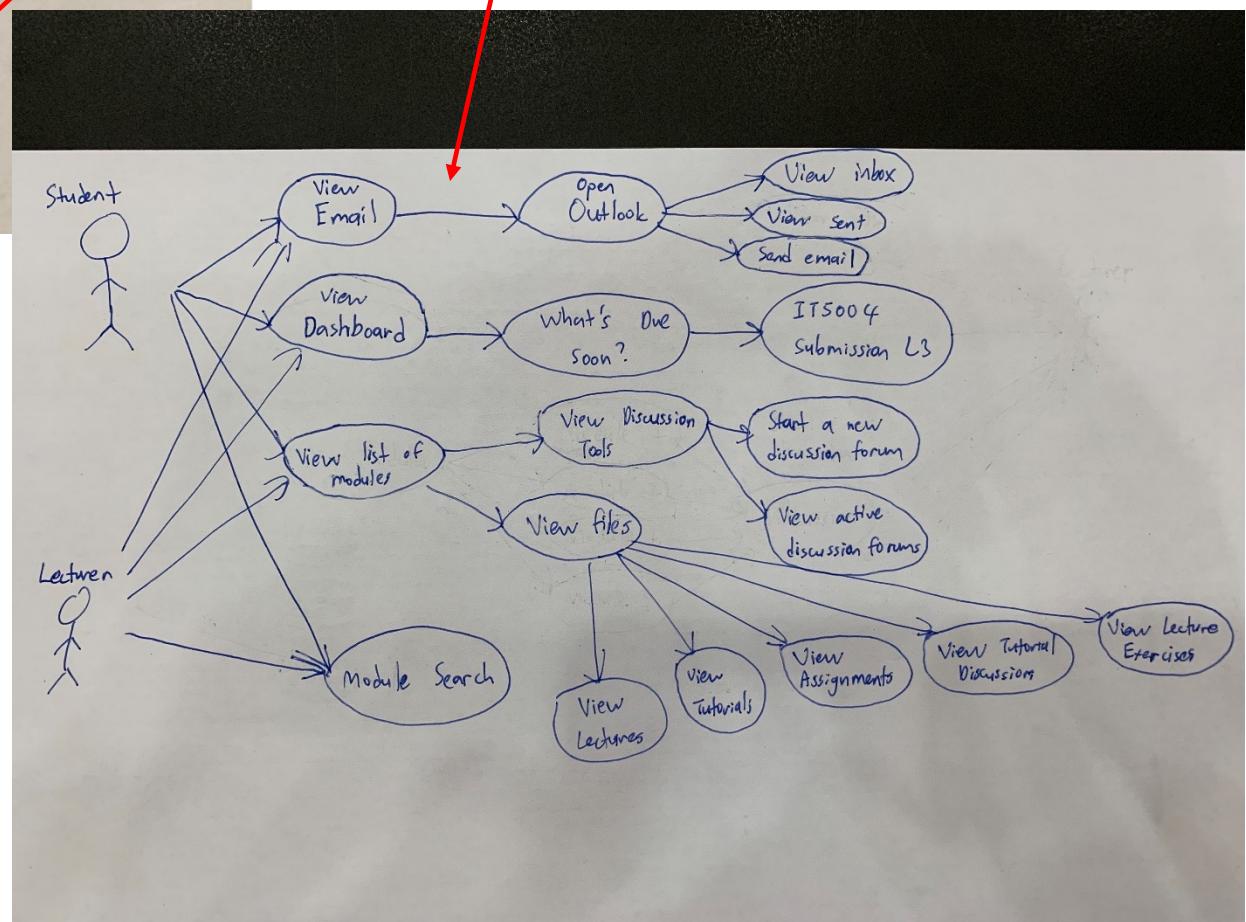
Use cases should be something that a user can do in the system rather than describing what happens



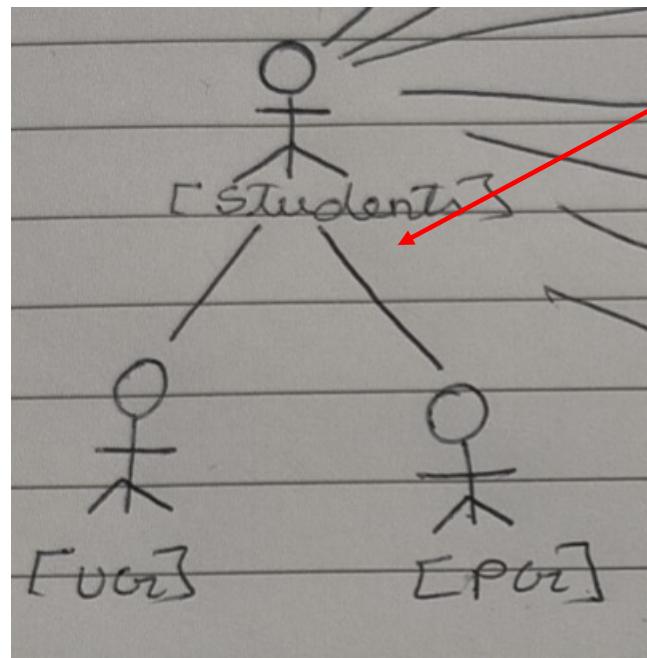
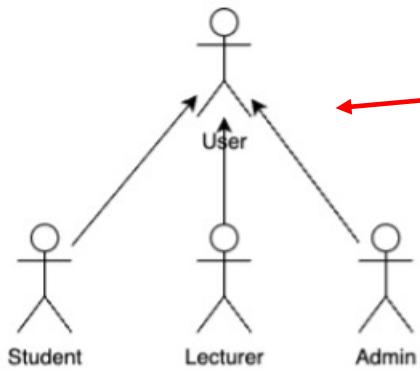
Remember to think of a use case as a **system action** the user can do in the system rather than trying to describing an idea of the user is doing.

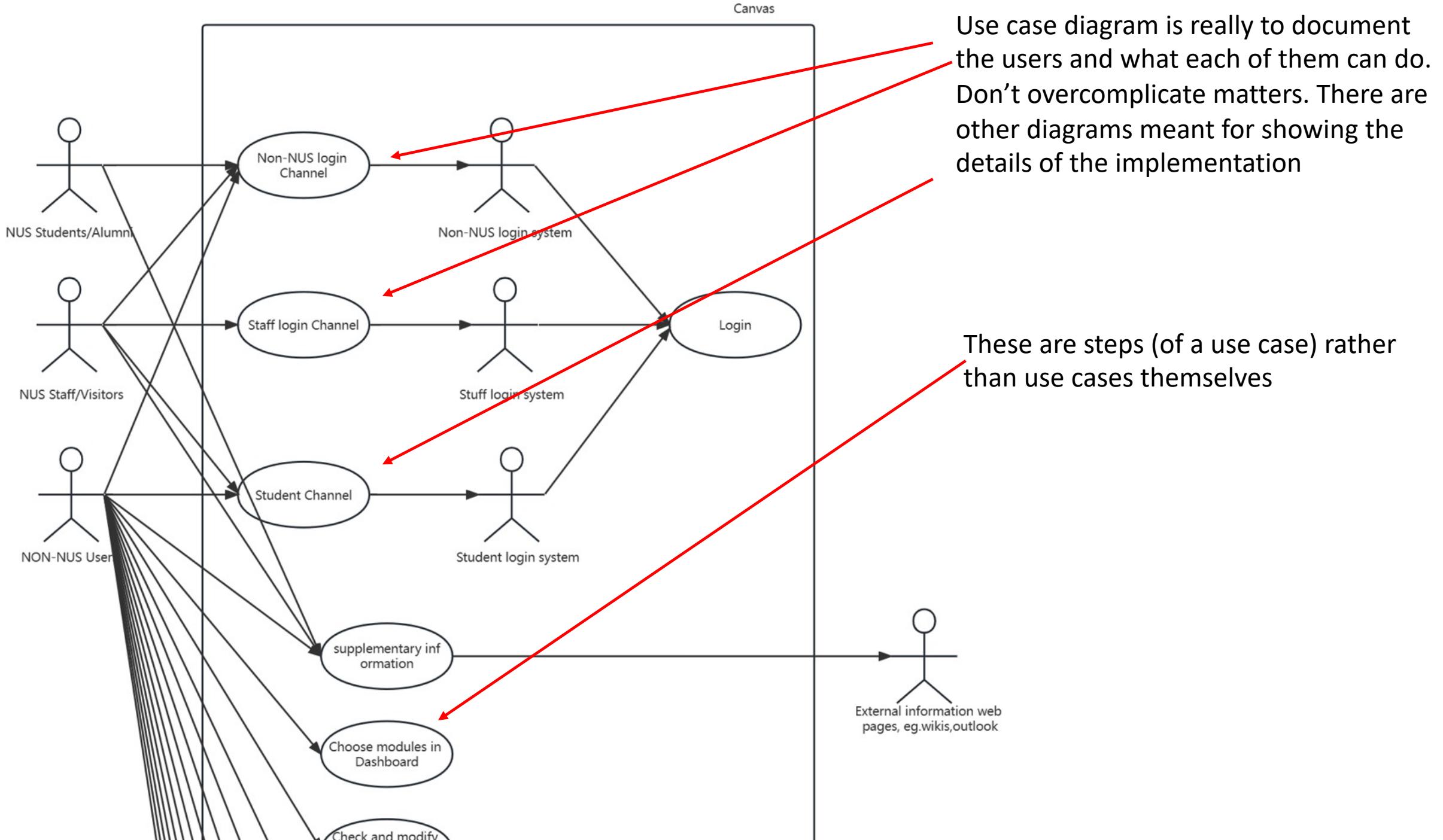


Don't link the use cases up.
They are meant to be showing the different functionality not the flow



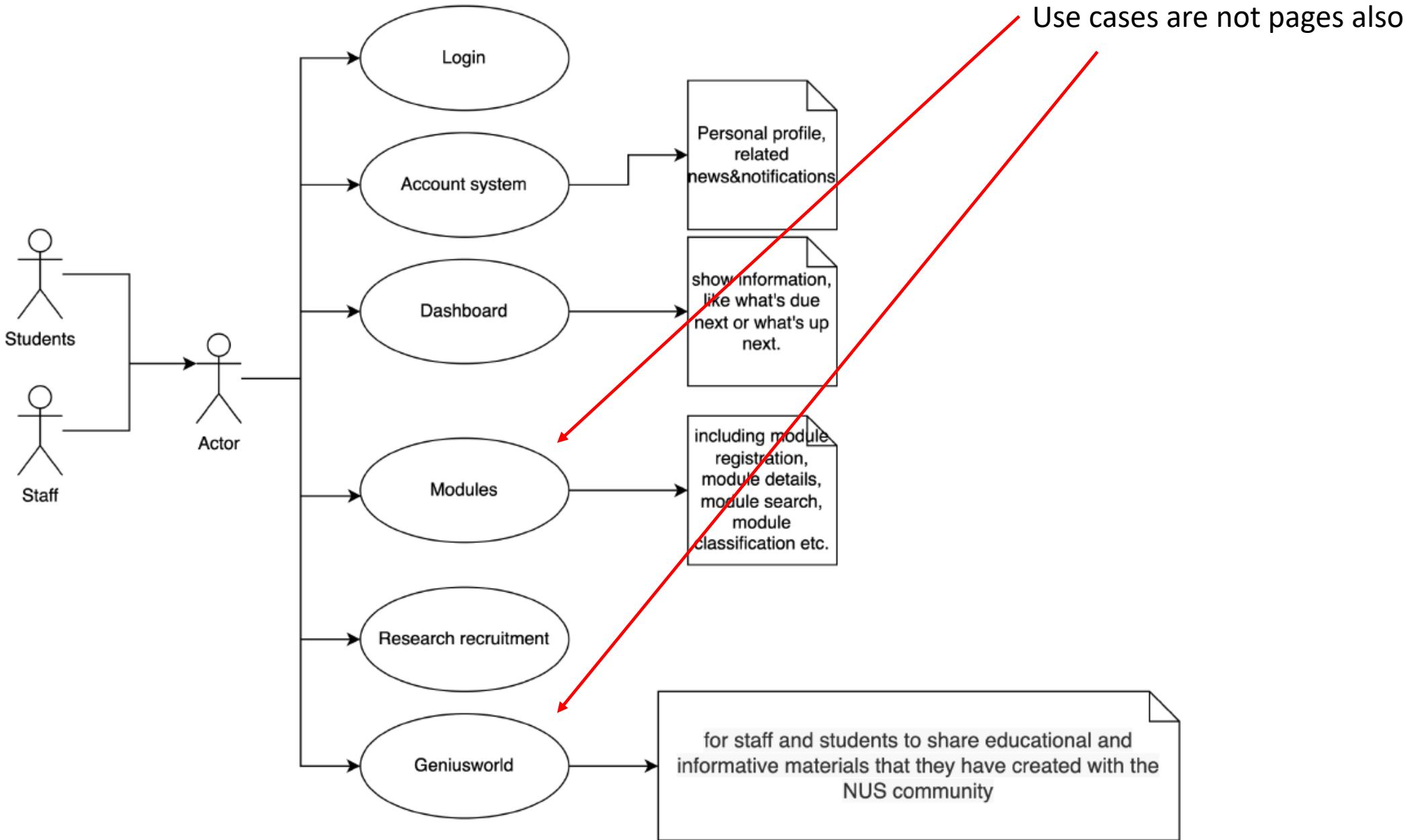
Make sure you use the correct arrow for
the generalization/specialization

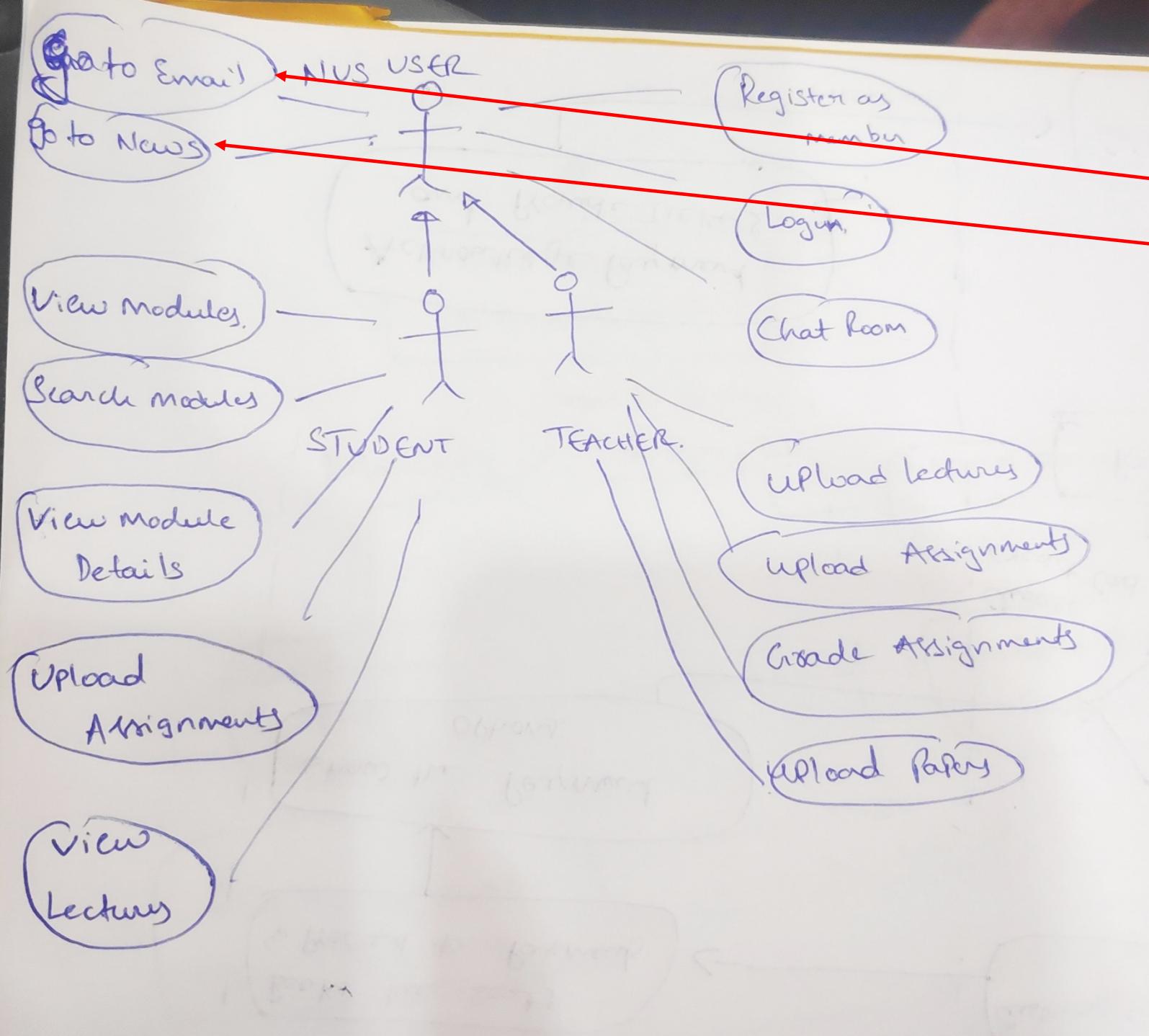




Use case diagram is really to document the users and what each of them can do. Don't overcomplicate matters. There are other diagrams meant for showing the details of the implementation

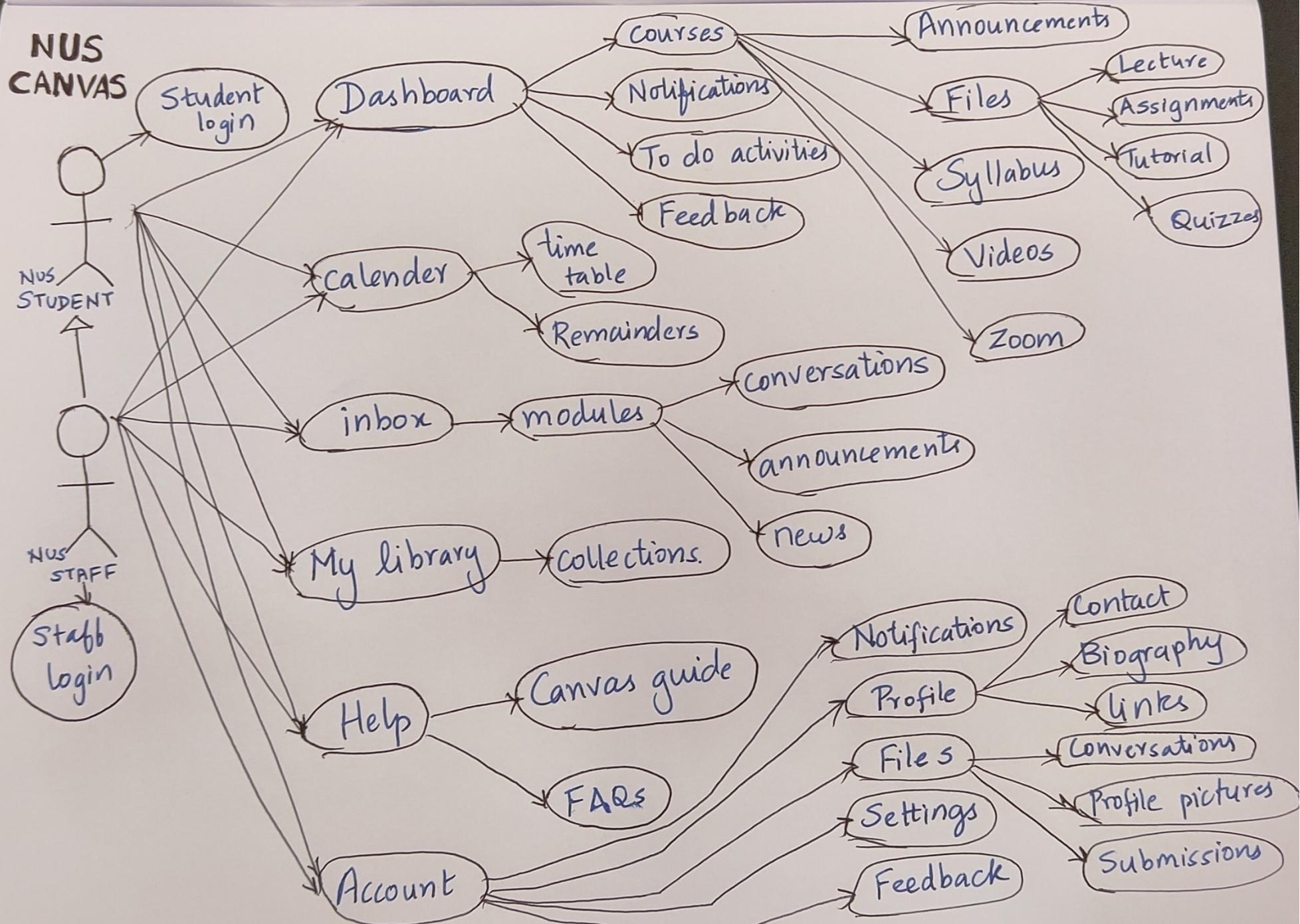
These are steps (of a use case) rather than use cases themselves



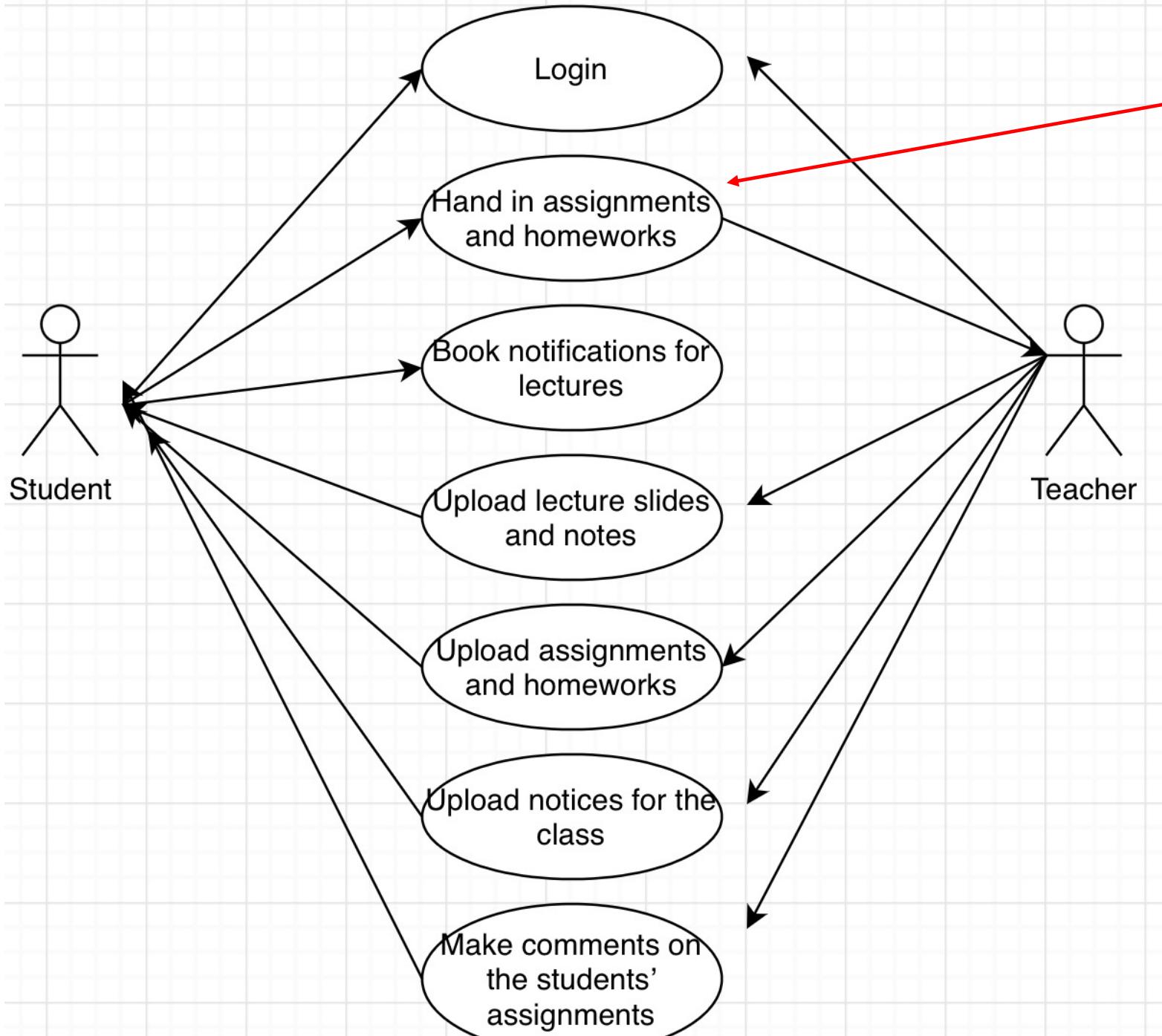


Use cases should not be thought of as pages (i.e. accessing a certain page is not a use case) but of course when you access a page, the system would need to retrieve certain records (those could be the use case)

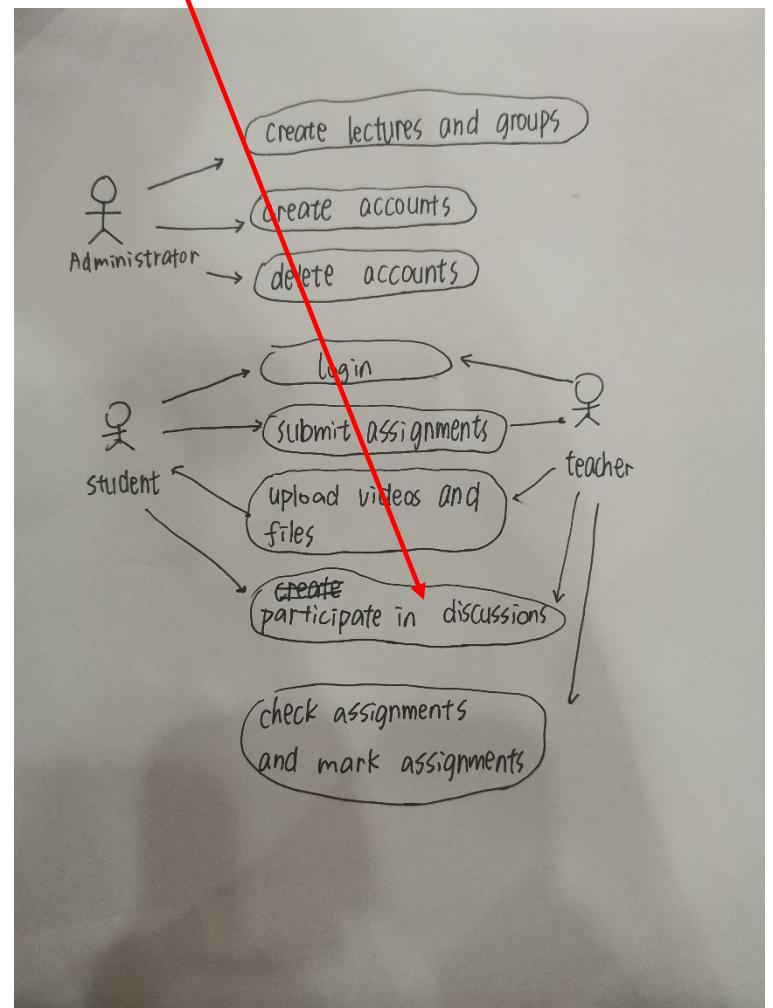
NUS
CANVAS

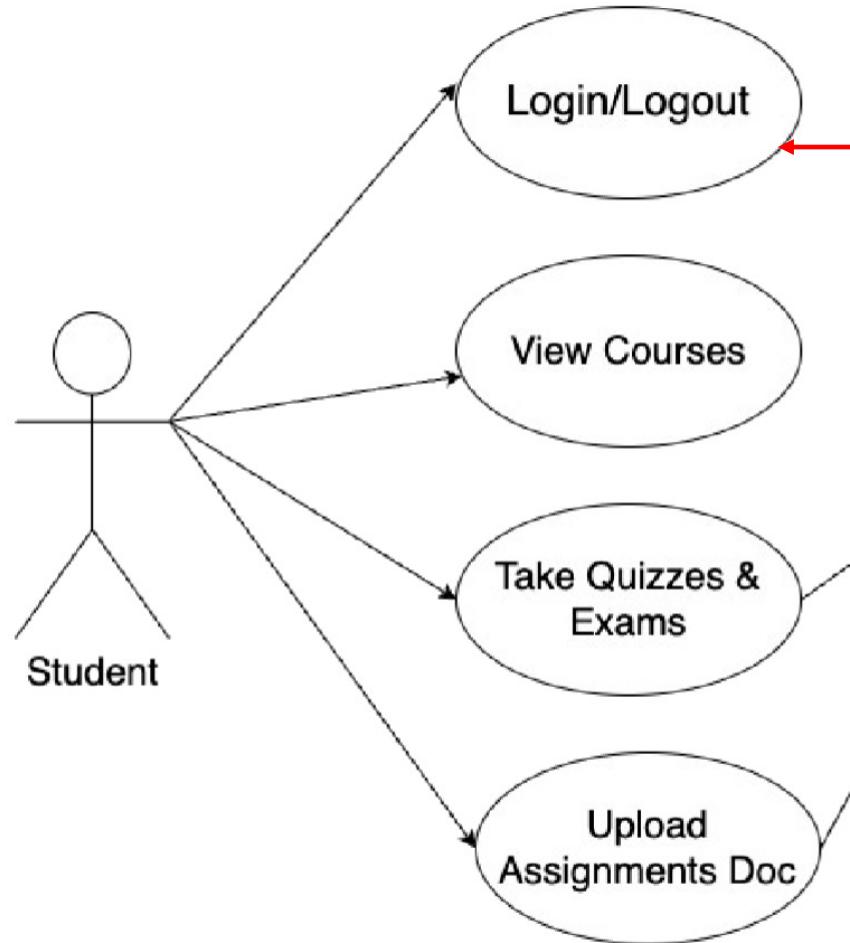


Use case diagram is
not meant to be used
to model the
“navigation paths”

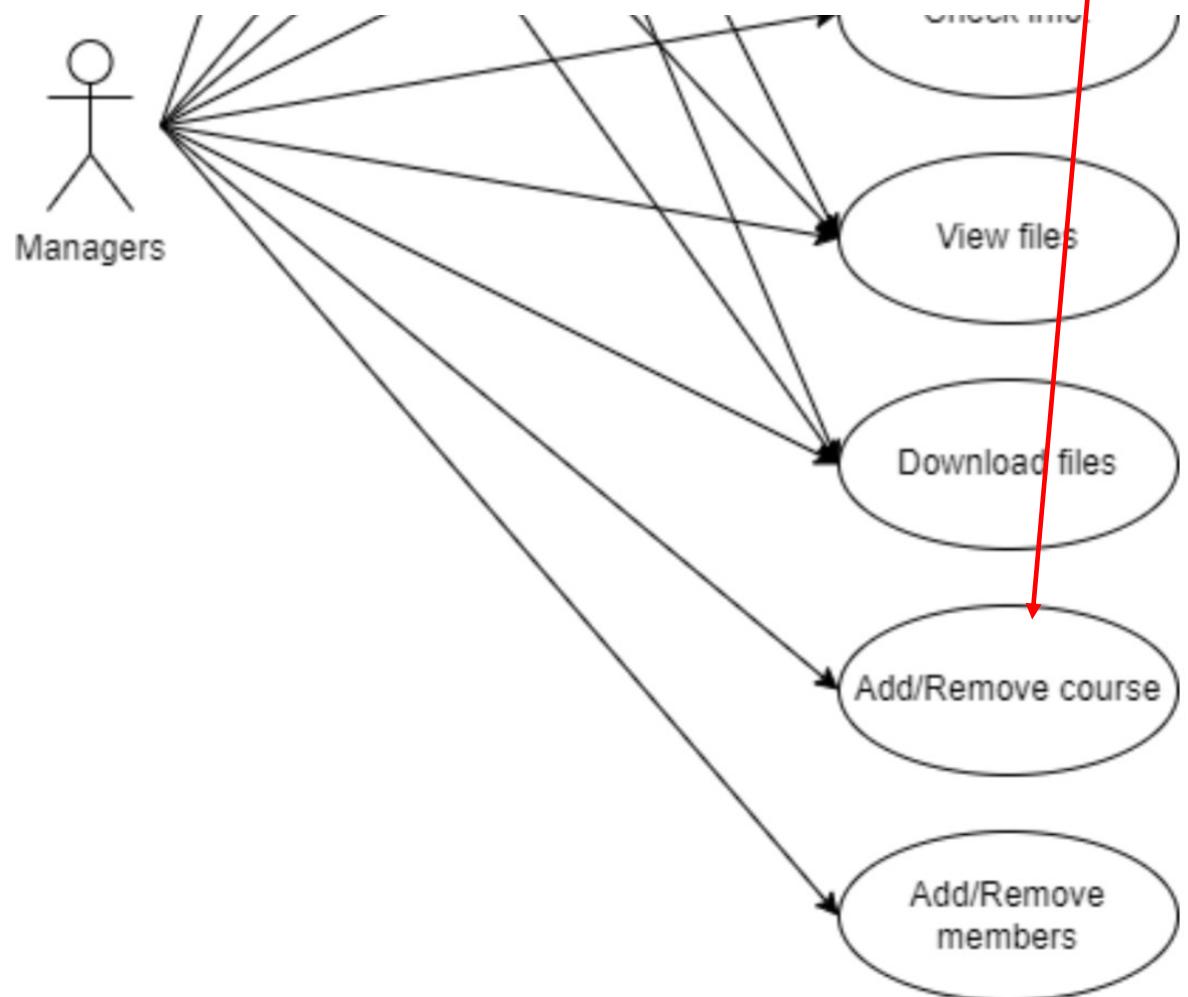


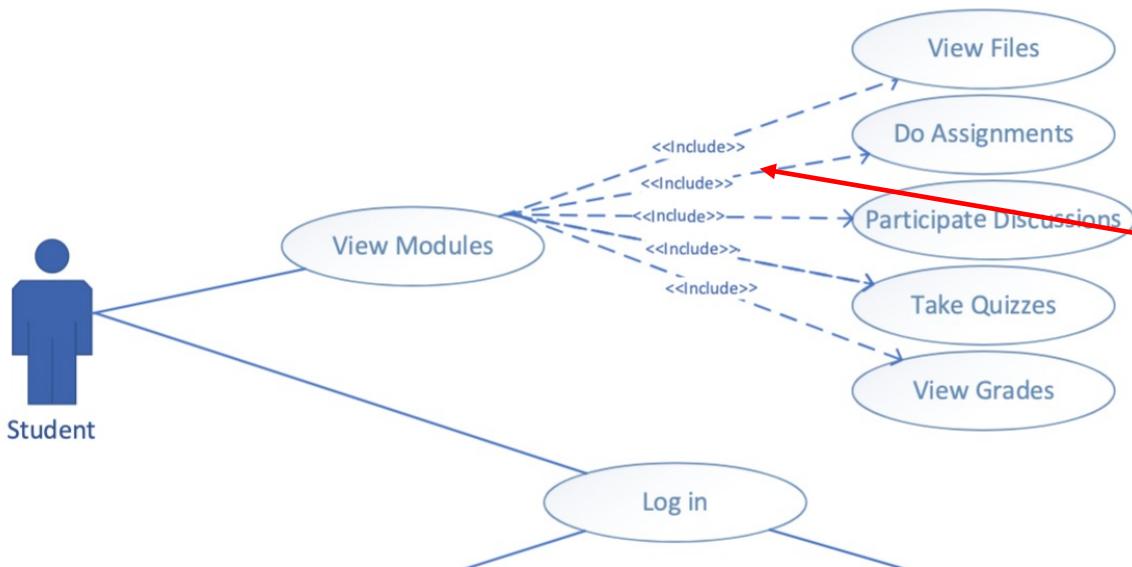
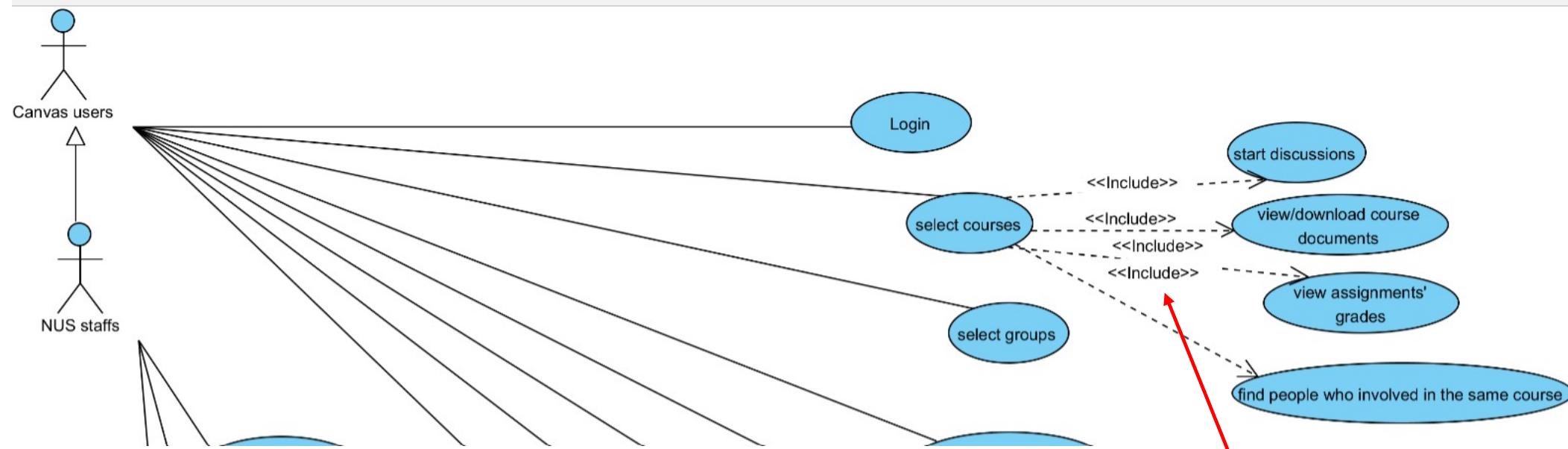
While it's important to think about the business objectives, should translate each of the business objectives to system functionality



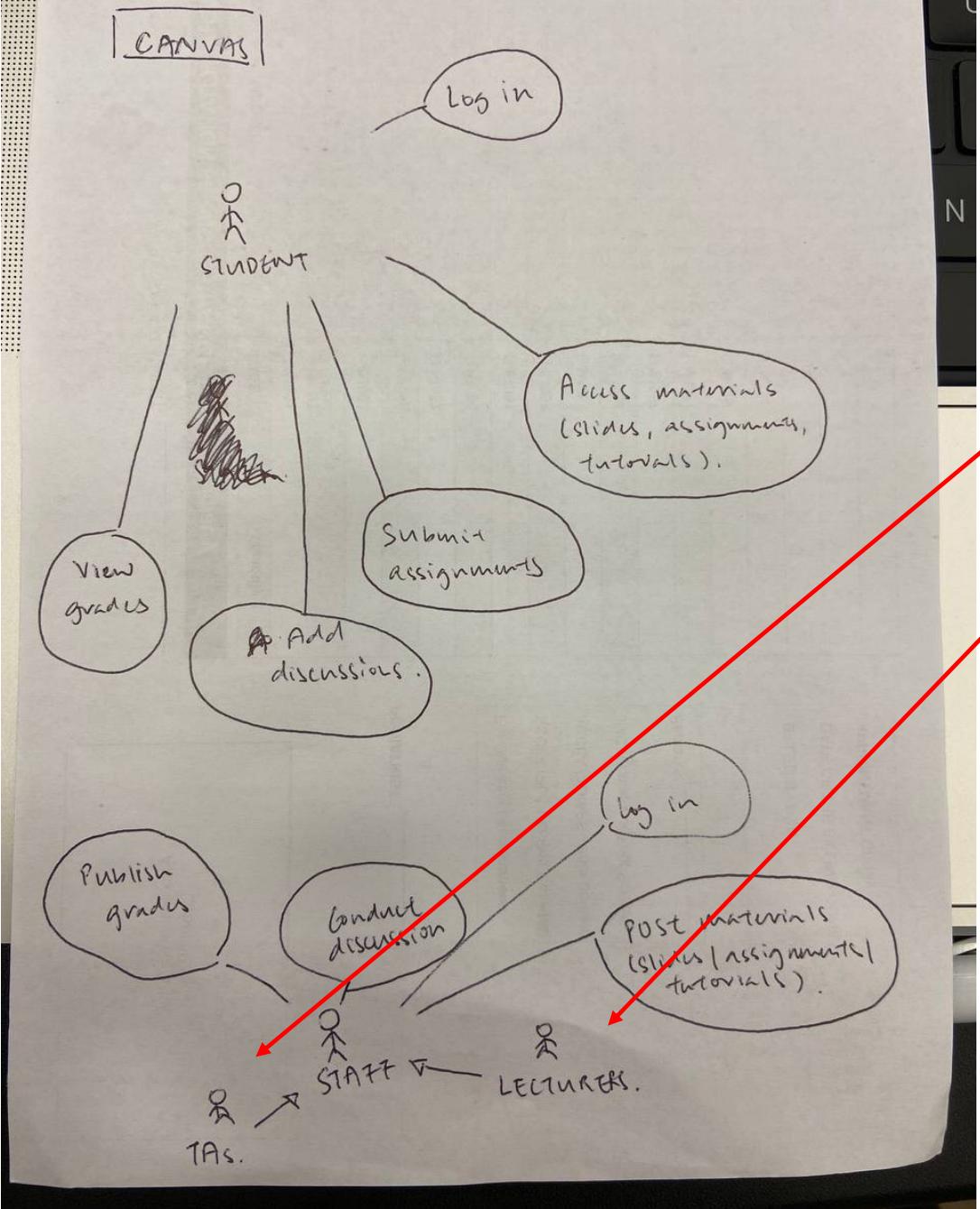


Should break it out to the individual system functions (i.e. don't group use cases up together)



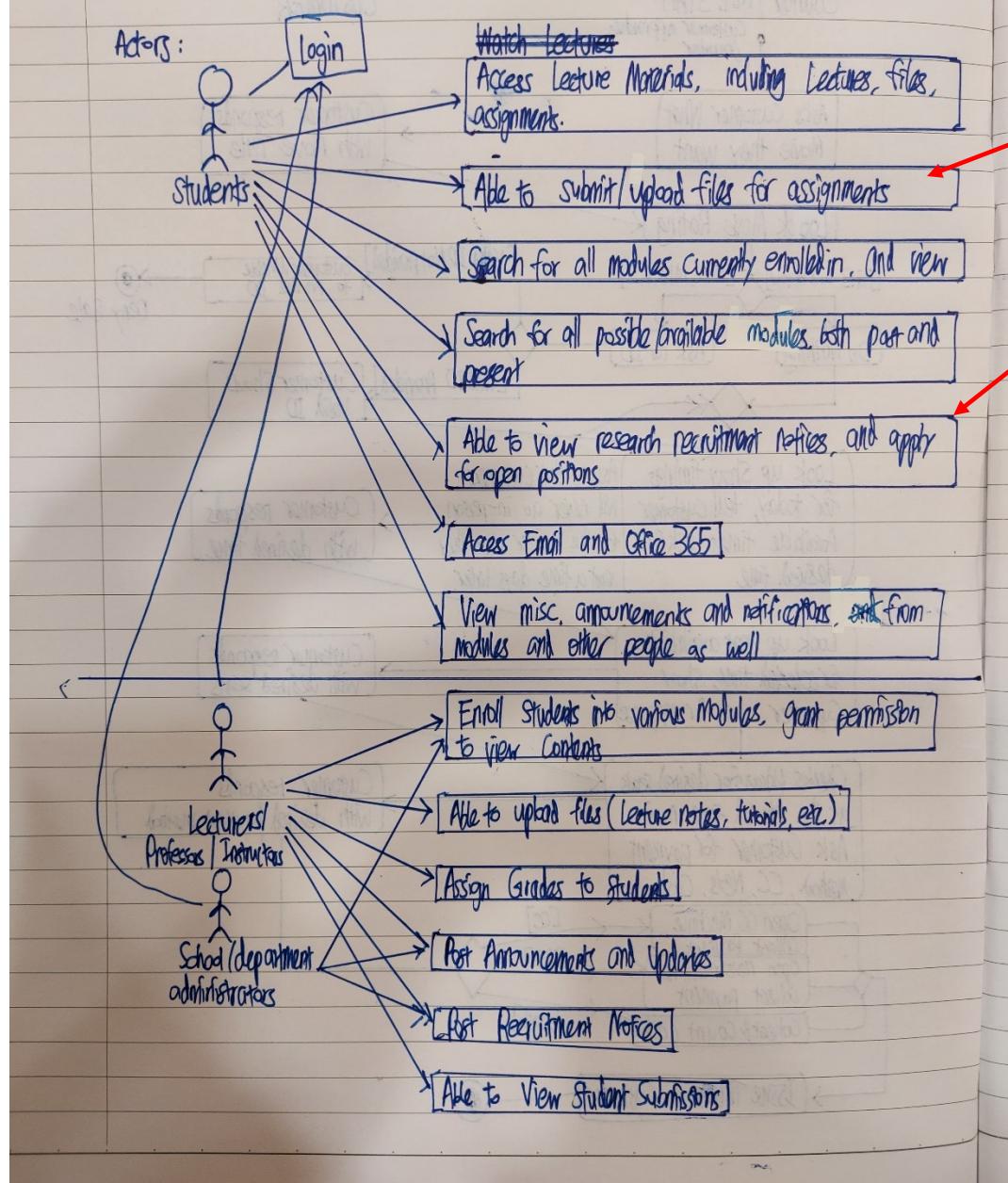


Wrong use of includes (we will cover includes later on)



Usually if we were to draw specialized actor, there should be some additional use cases they can do, if not, not really so important to include this

EX2: Use case Diagram for Luminus system



Write use cases as a system function rather than objective or "access rights"

management
sales

handle return tickets.

Inventory inquiries

collection

Scores registered

反思：

上源：

→ personal information

$\log h$

Student → Select course
(-units)

✓ test scores

Teache

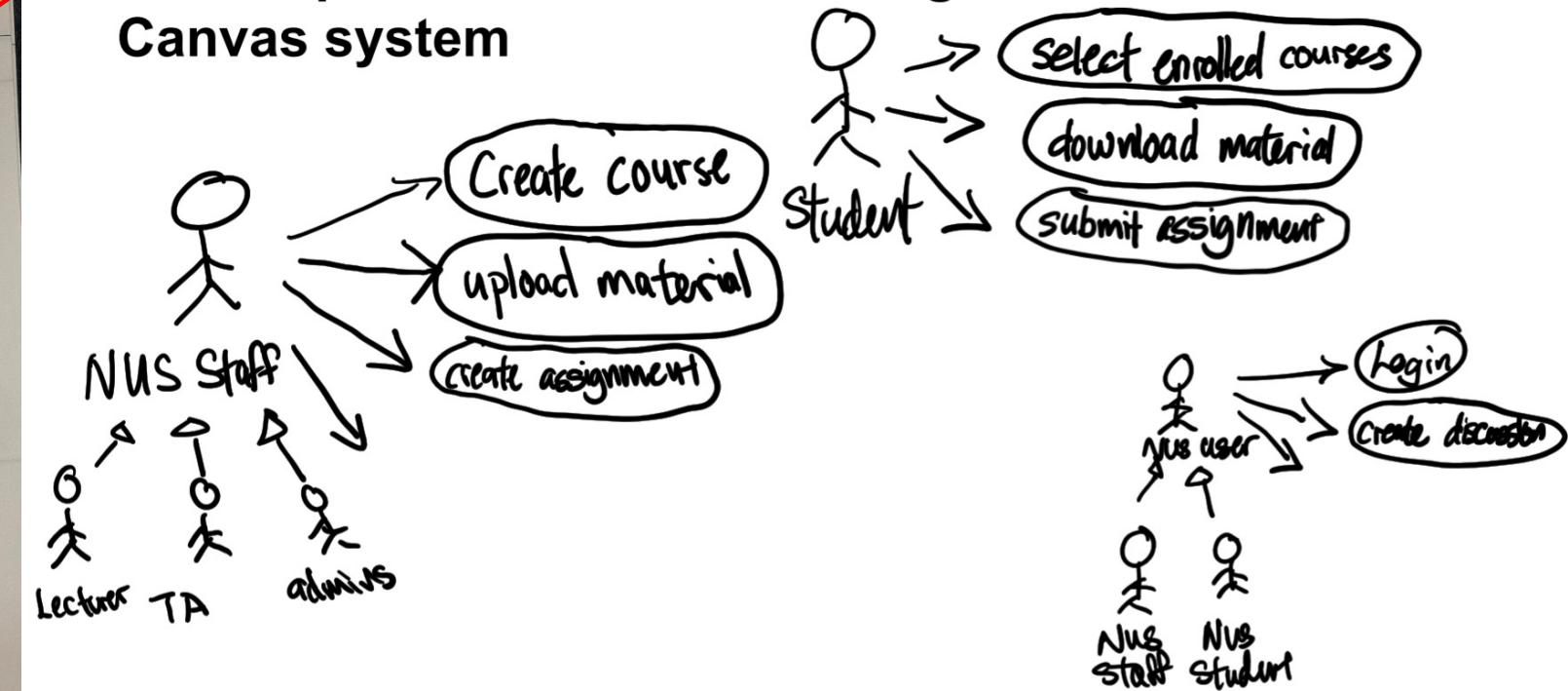
↓

Teaching

This is an interaction with the system but
not a use case
(unless you meant to “view details of a
module”)

~~TASK~~

Come up with the Use Case Diagram for the Canvas system

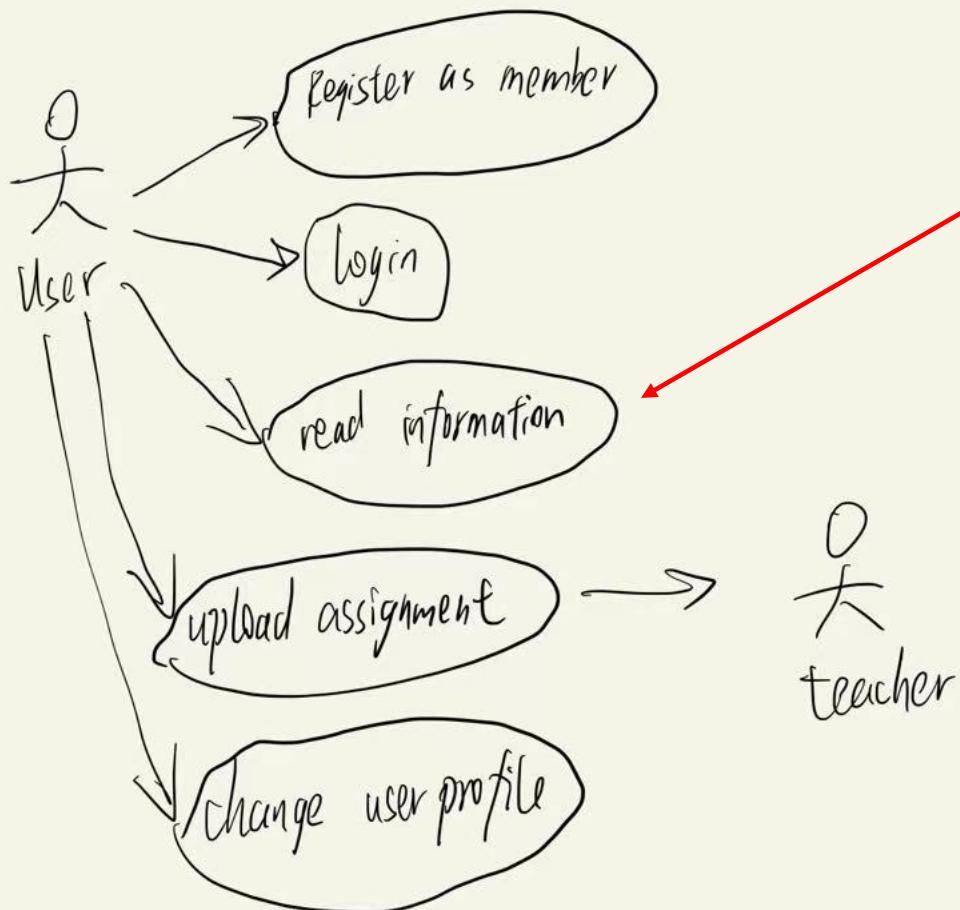


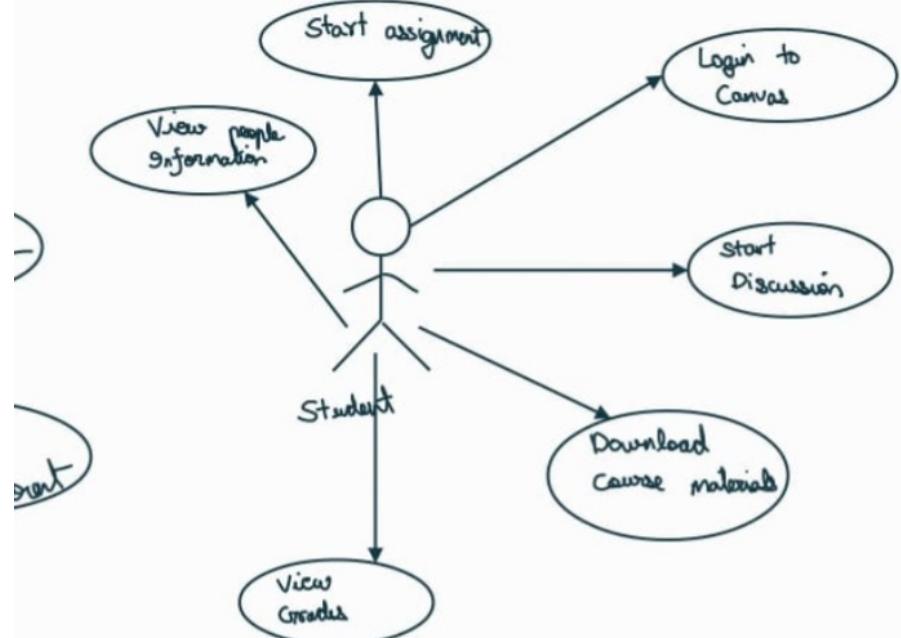
L3 Requirements Gathering

无标题的笔记本

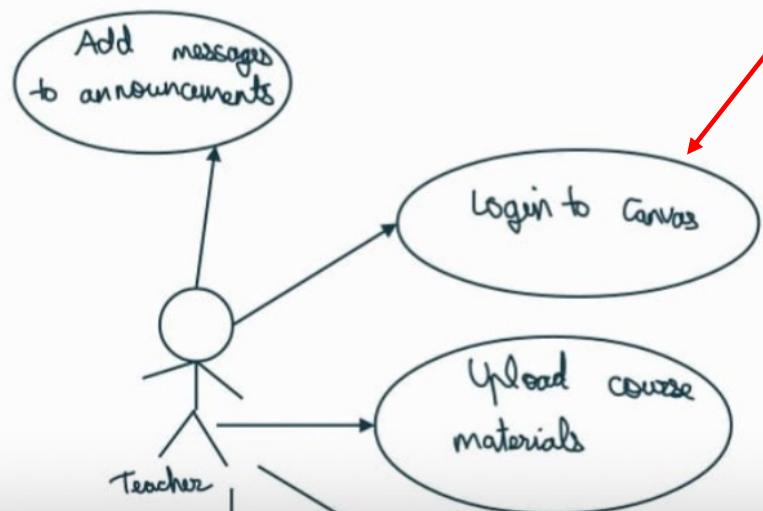


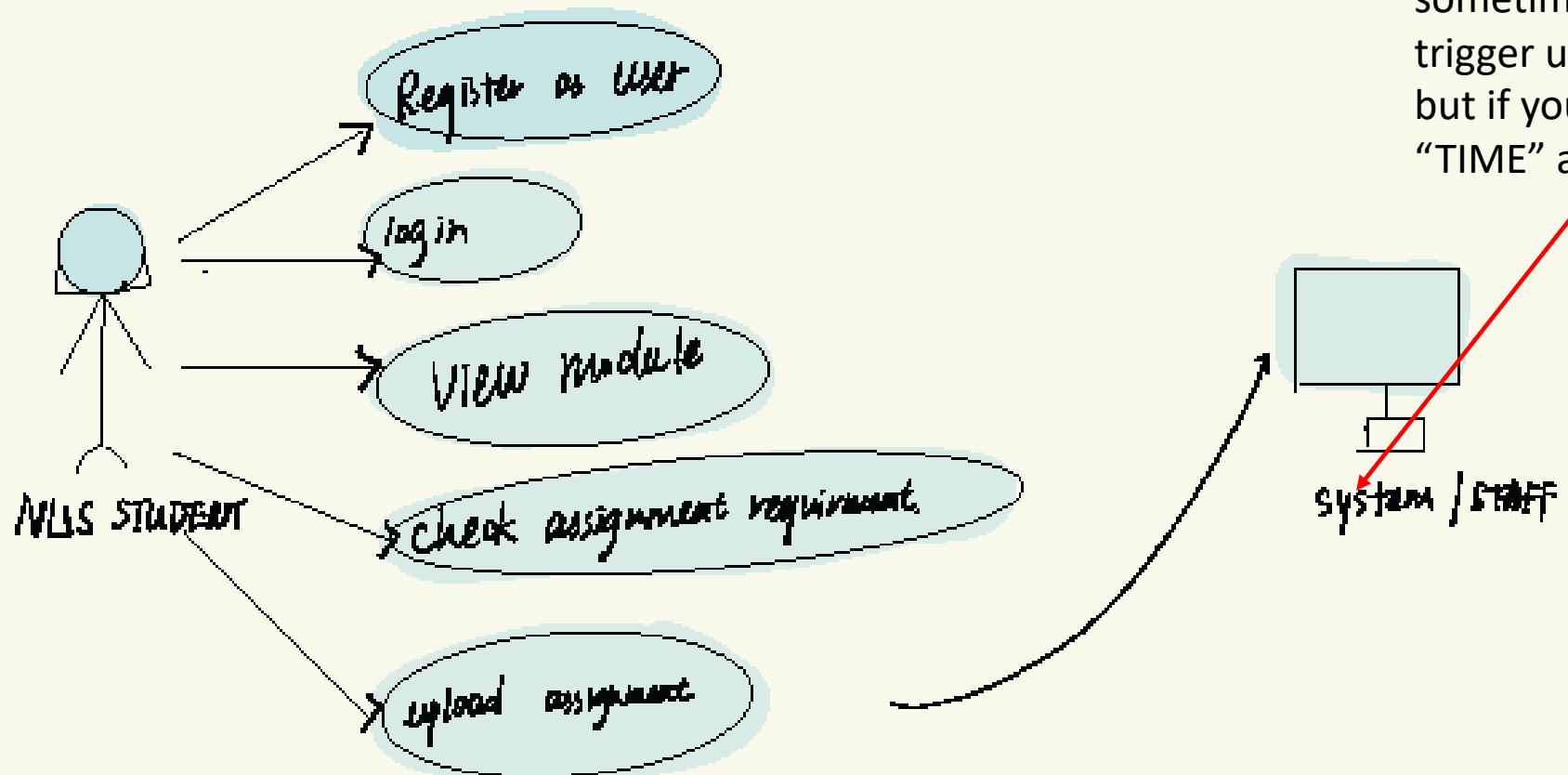
This is too vague, should be a system function that's clear and we know what to implement





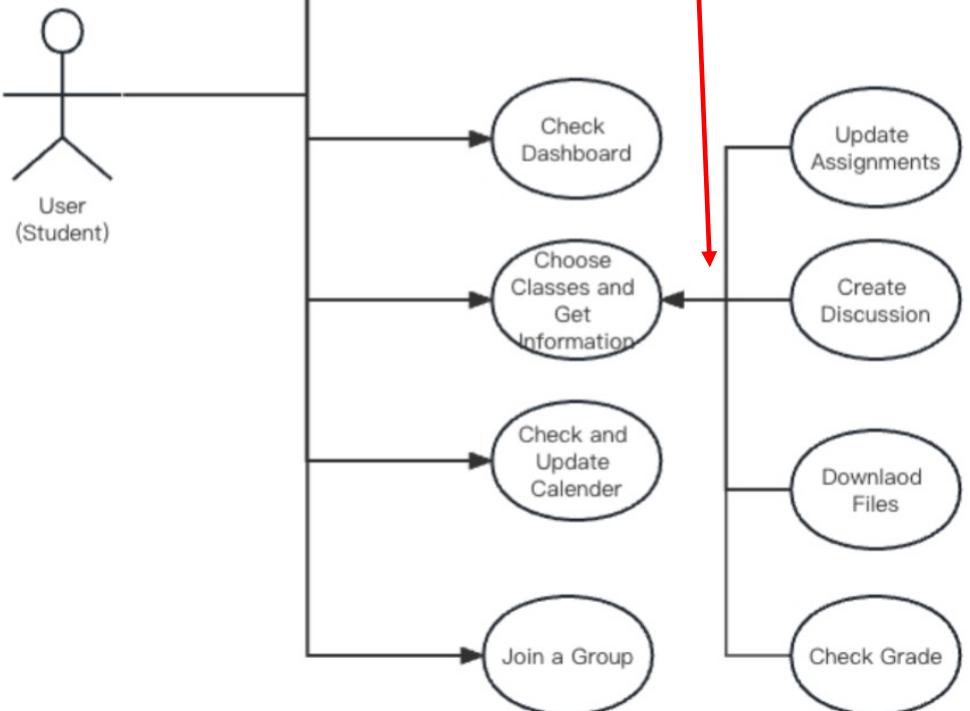
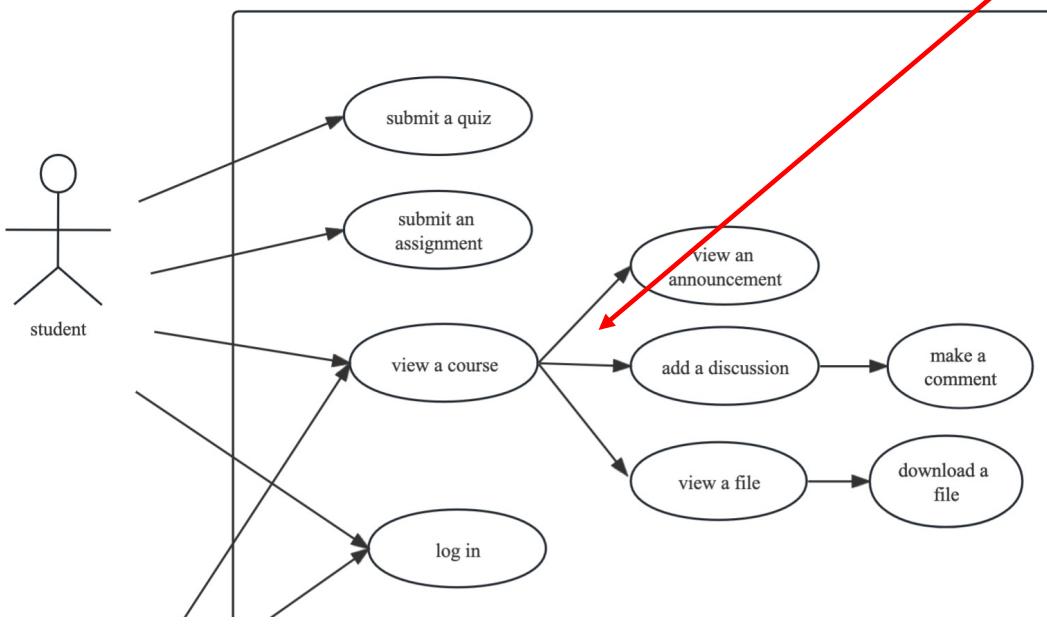
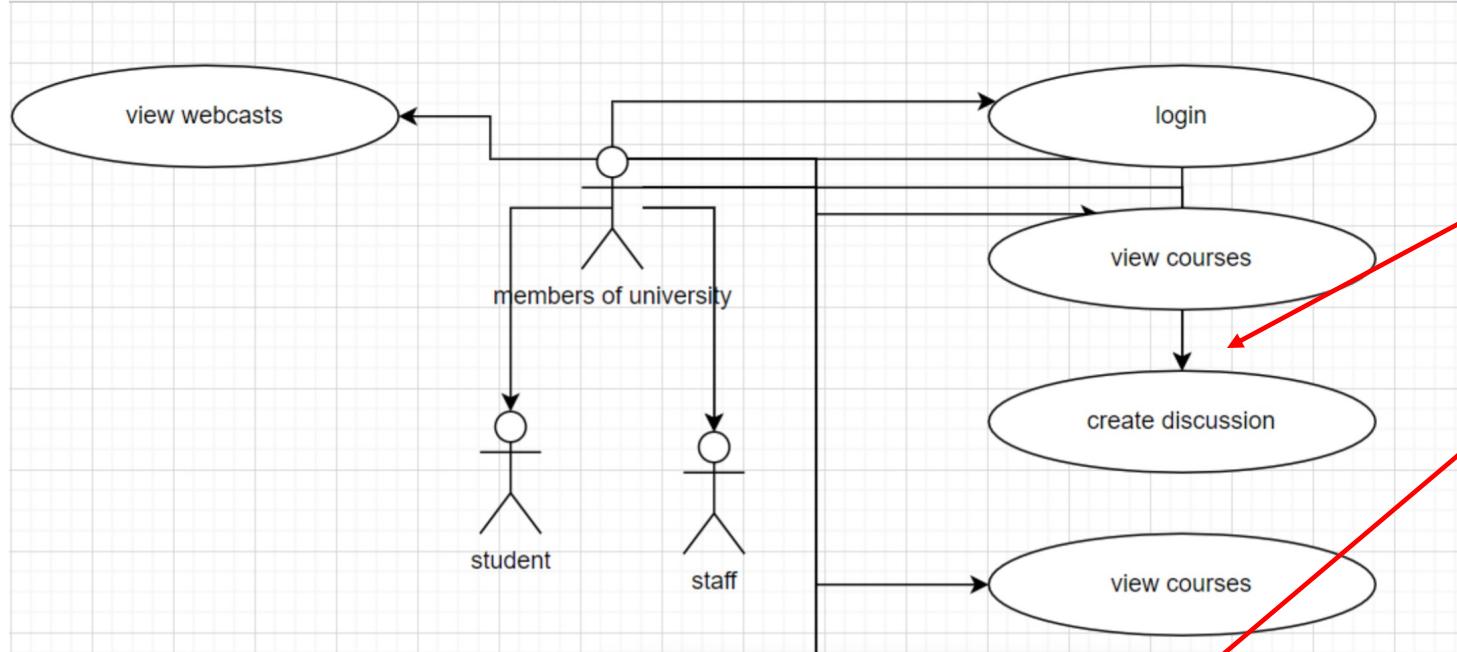
Don't duplicate use cases. Link the actors to the use case if it is the same use case

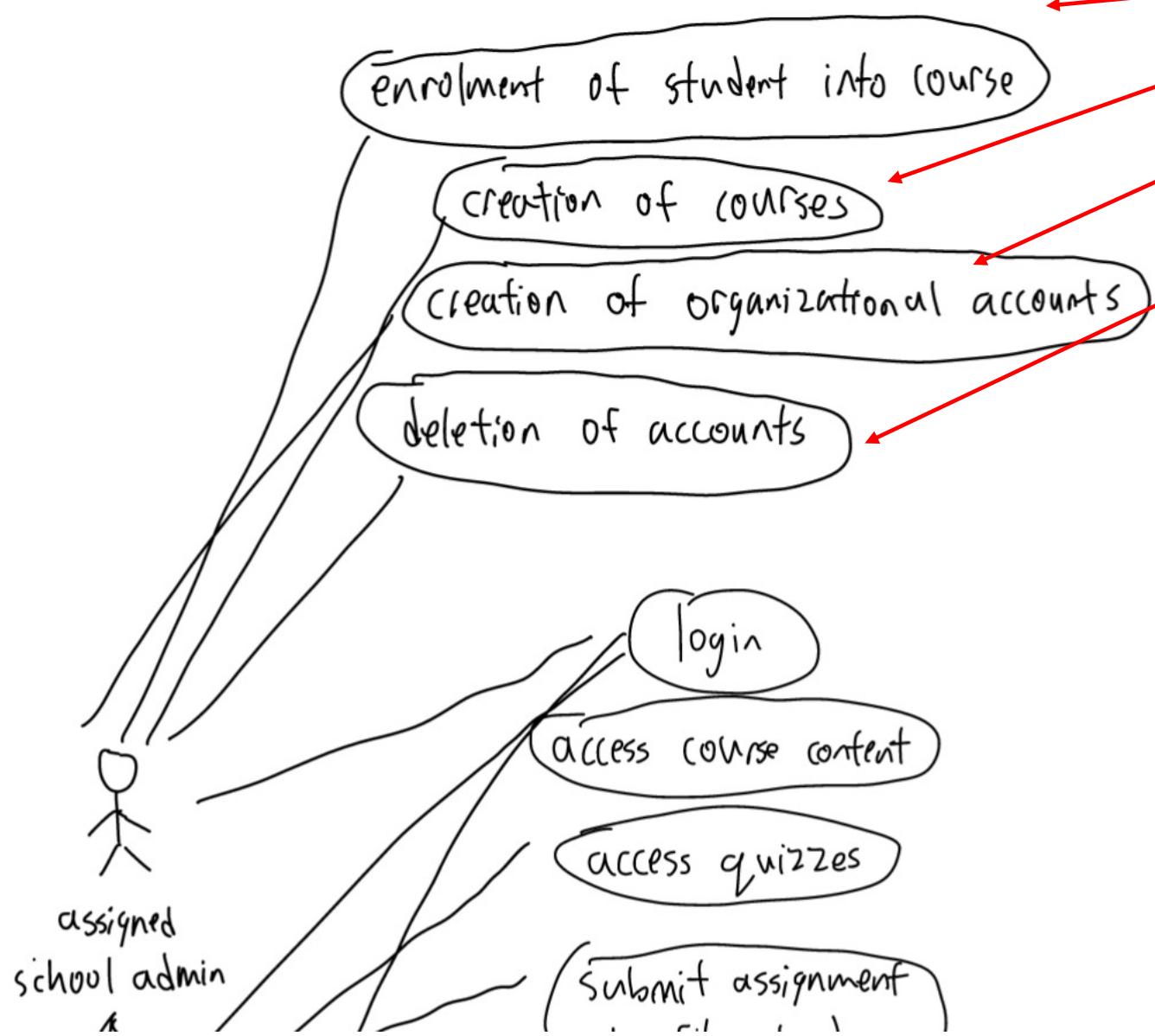




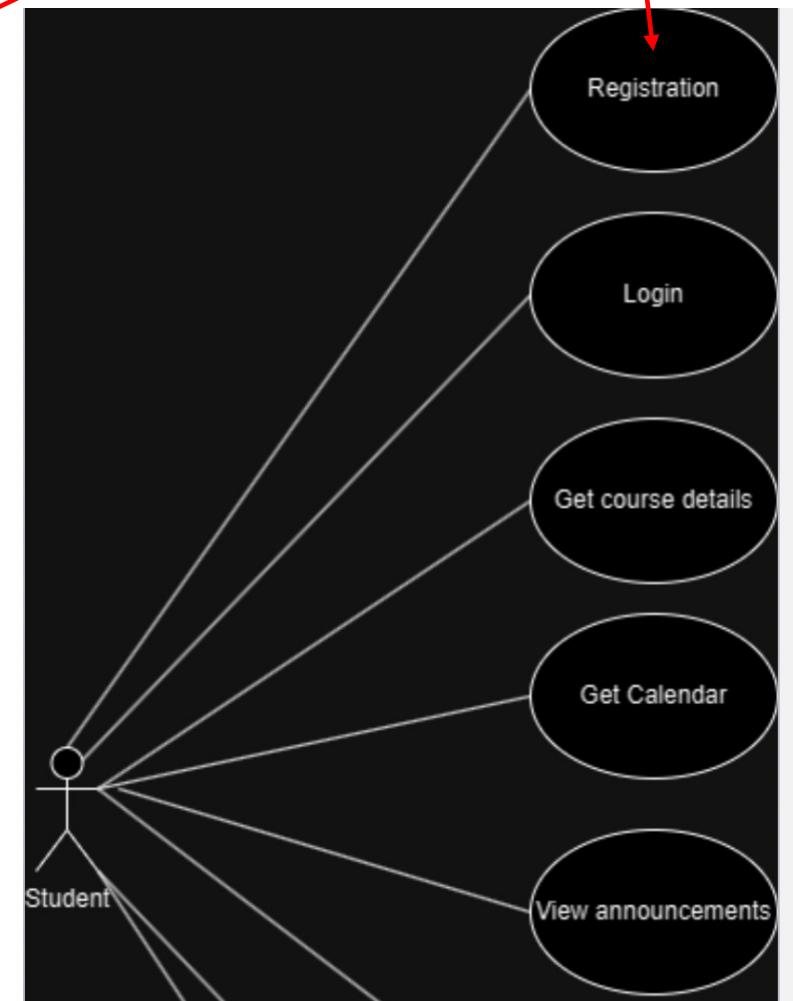
The entire collection of the use cases is the system so there should not be a system actor (unless you are talking about another system). On that note, sometimes we do not show the time-trigger use cases in the use case diagram but if you want to, you can create a “TIME” actor

Don't link use cases together. At least not this way





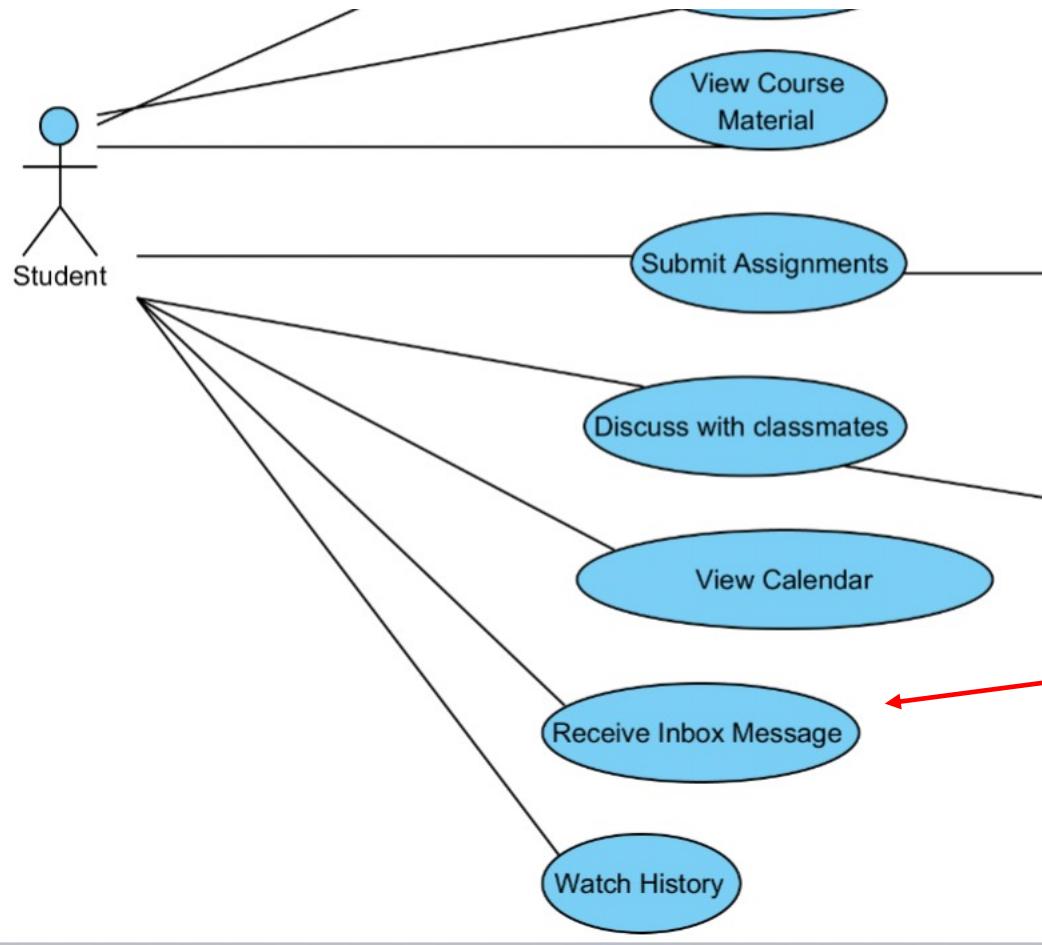
Name use cases using an action. E.g.
Delete an account, etc



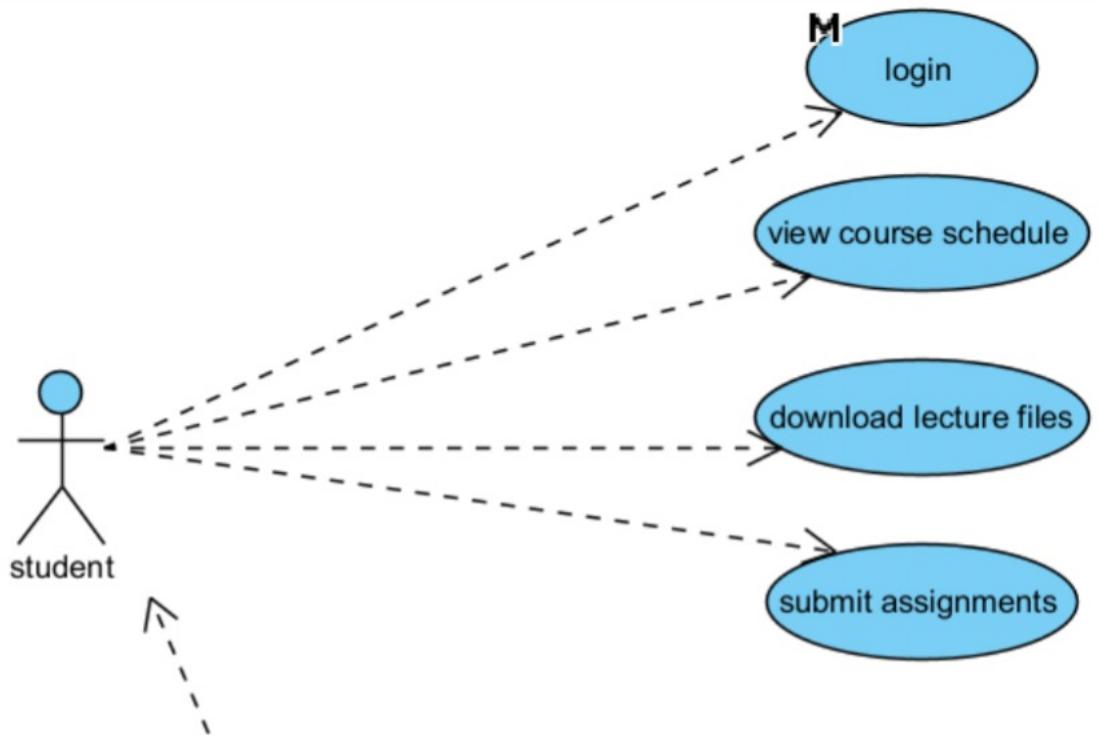
Actors:

- Students
 - Access learning resources
 - View courses
 - Submit assignment
 - Create discussions
 - Participate in discussions
 -
- Professors, Teaching assistants
 - Manage course content
 - Make announcements
 - Create assignments
 - Grade assignments
 - Create & manage quizzes

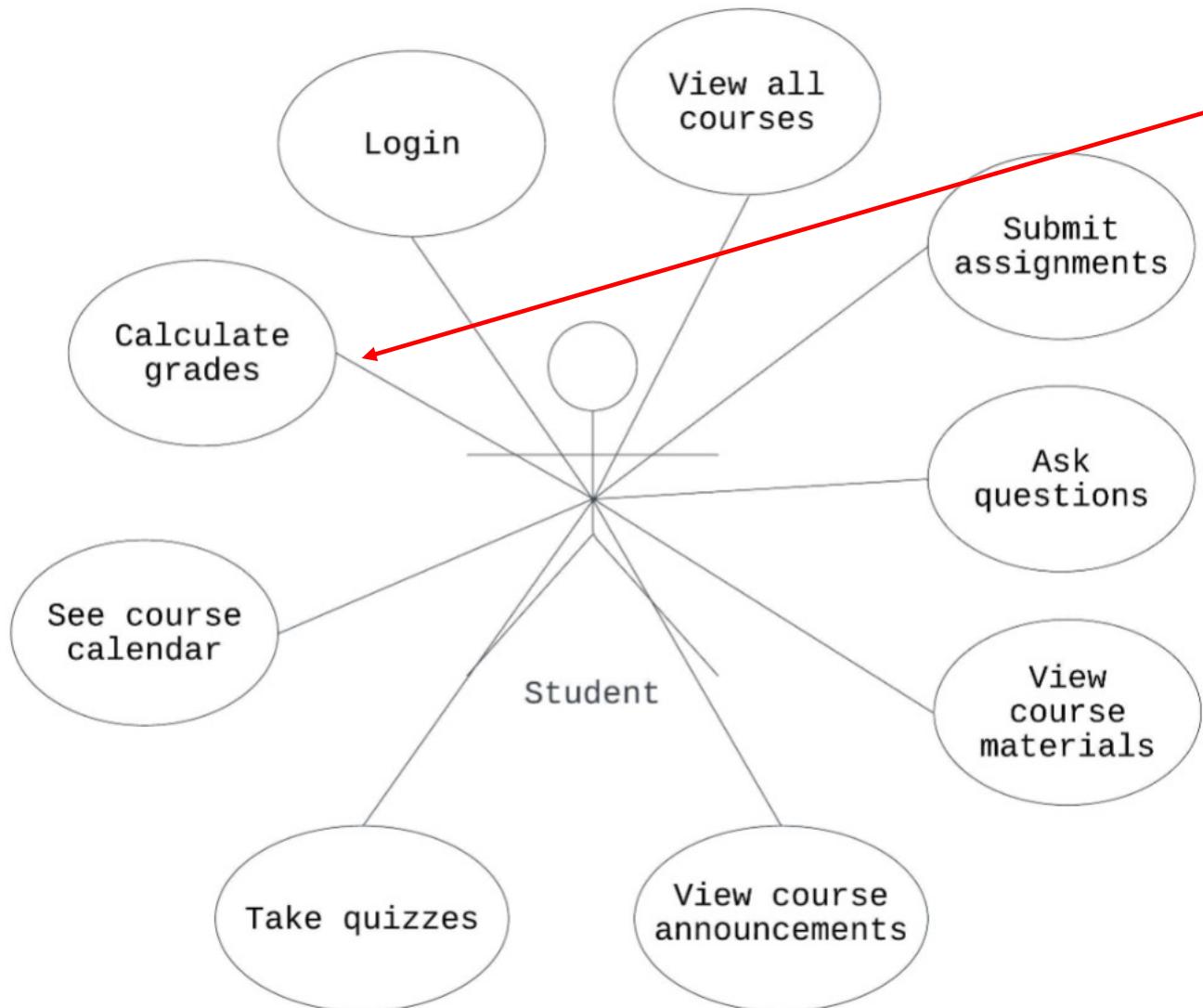
Should split this into the individual actions (i.e. don't use manage XXX to group up all the actions for managing the course content)



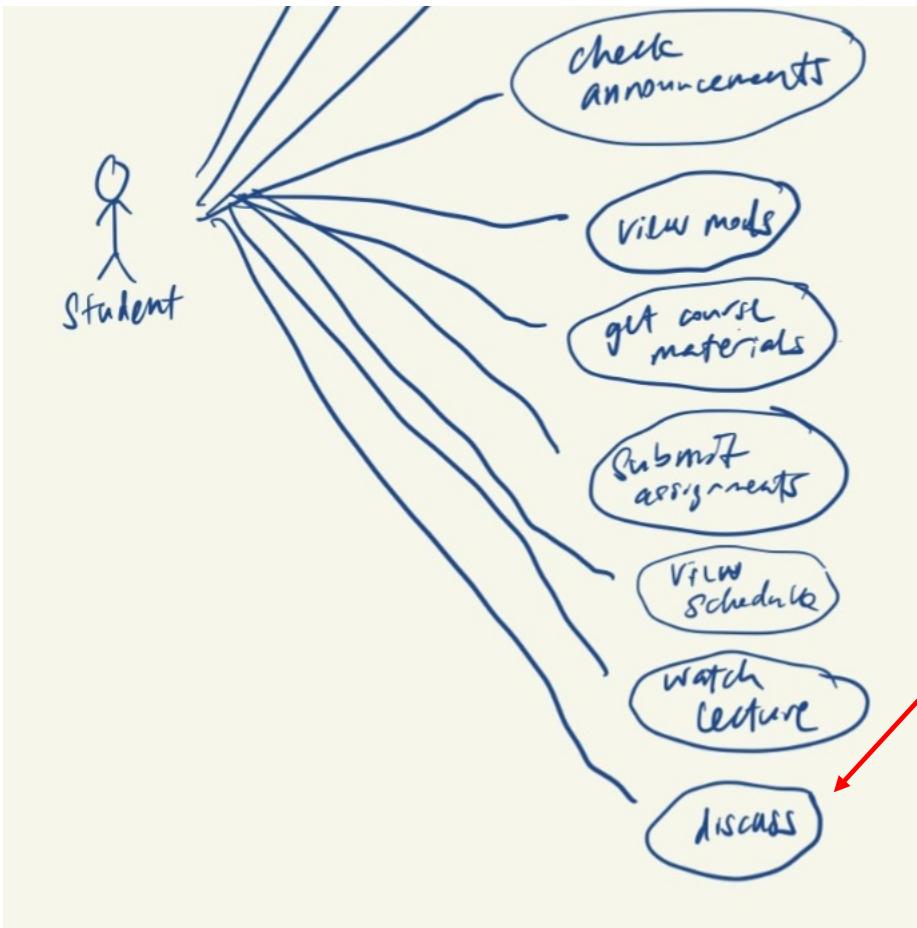
Should think of use cases as a list of actions that the actor does rather than things that is done to the actor.
For example in this case, it should be more of “view a list of inbox conversations”, and “view details of a inbox conversations”



Use solid association lines (unless you are drawing “includes” or “extends” which will be covered later)



This is not a use case. Use cases are not just what the system needs to do. We look at use cases as from the point of view of the user, a list of system functions that each user can do



Use case is too vague