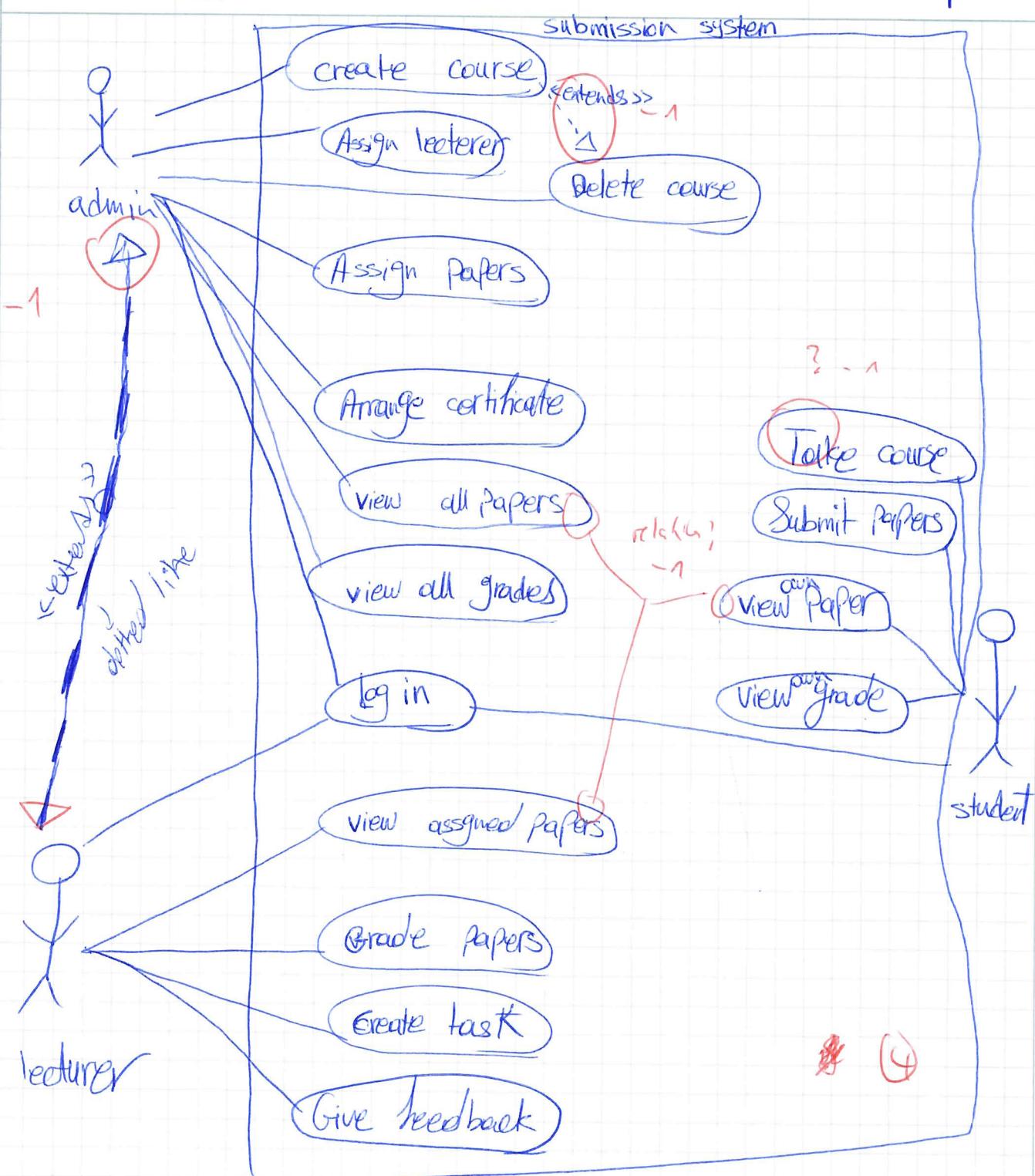


CHALMERS

EXAMINATION / TENTAMEN

Course name/kursnamn			
Software analysis & design			
Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg	
-UJZ	2024-01-03	14	4

mobile or other similar electronic equipment available during the examination.
eller mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under

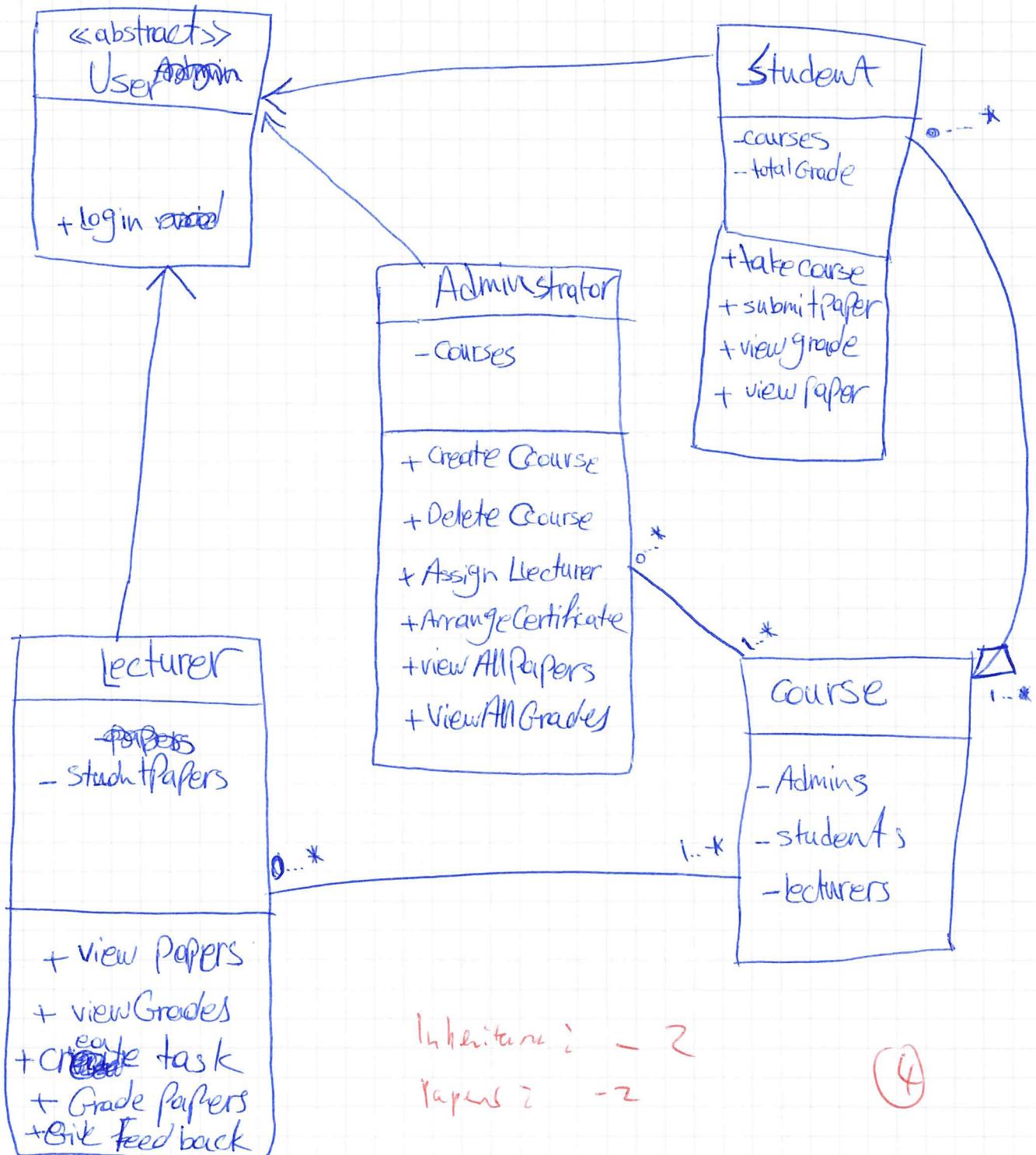


*Just to clarify, give feedback is comments on papers and assumed that

not grades, so created two different ones.

(20)

CHALMERS	Anonymous code Anonym kod DFT 185-0012-UJ2	Points for question (to be filled in by teacher) Poäng på uppgiften (fyller av lärare)	Consecutive page no. Löpande sid nr 2/4
			Question no. Uppgift nr 2 : 1



Description next page:

CHALMERS	Anonymous code Anonym kod DIT185-0012-UJ2	Points for question (to be filled in by teacher) Poäng på uppgiften (ifylls av lärare)	Consecutive page no. Löpande sid nr 3/4
			Question no. Uppgift nr 2.2

Since all users (student, lecturer, Admin) share some user auth requirements, I have created an abstract user class.

"every course has lecturers assigned to it" and admin creates courses, hint that the relation between them. So course must have at least one lecturer but it could be more (1-*), and admins and lecturers could be without any course (must not have at least one).

so

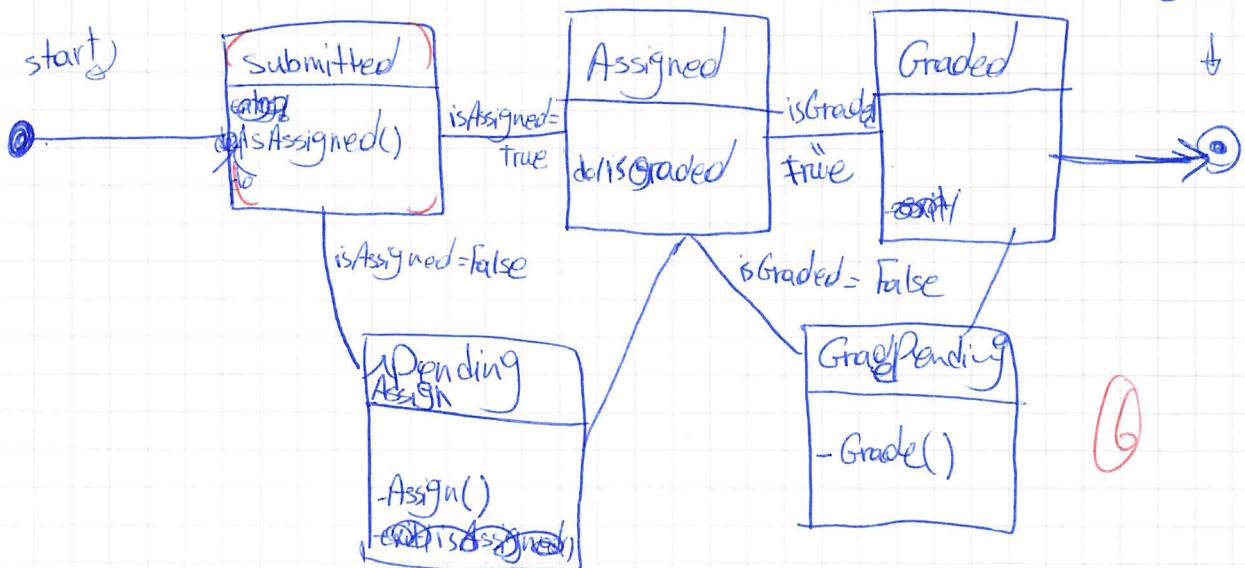
For relationship between student and course, courses must have students (it isn't a course without student), but students must not have at least one course (it could be zero or many)

V

Ø

Task 3a)

-2 notation

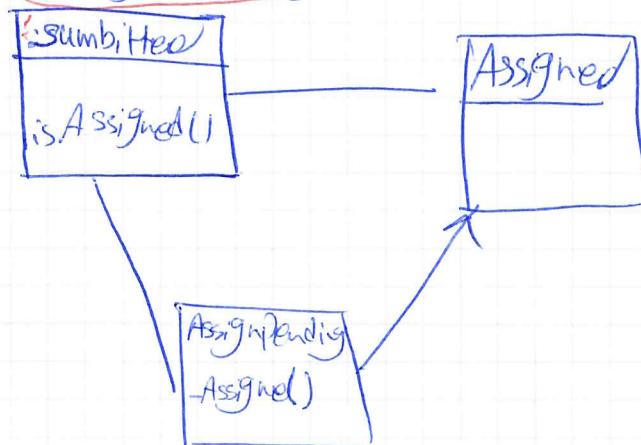


"Pending" state seen
superfluous

Task 3b)

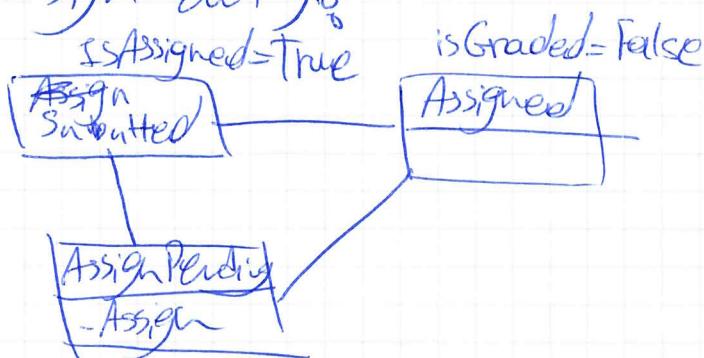
Before:
(isAssigned = False)

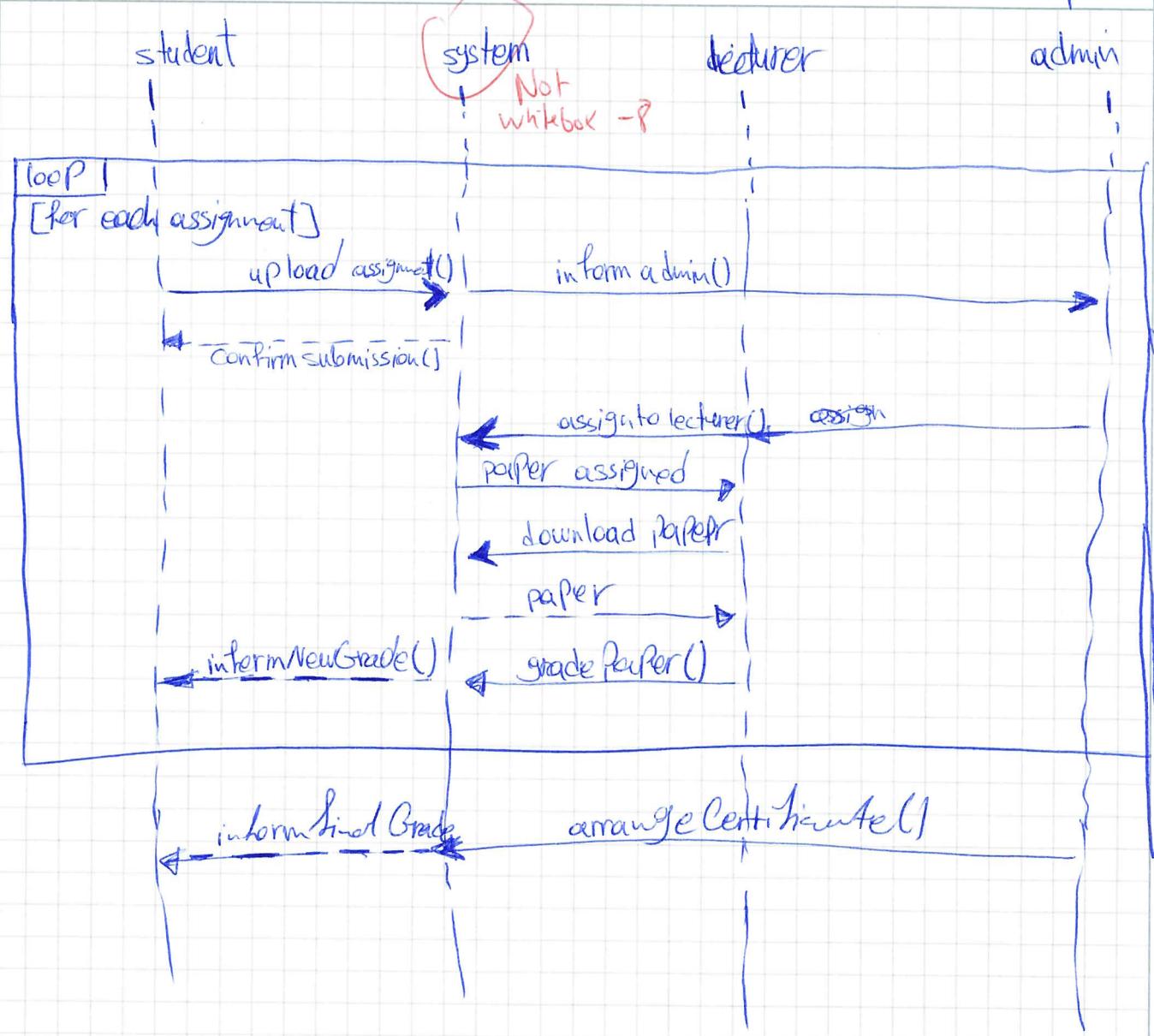
Notation -1



④
③

AFTER (Assignment):

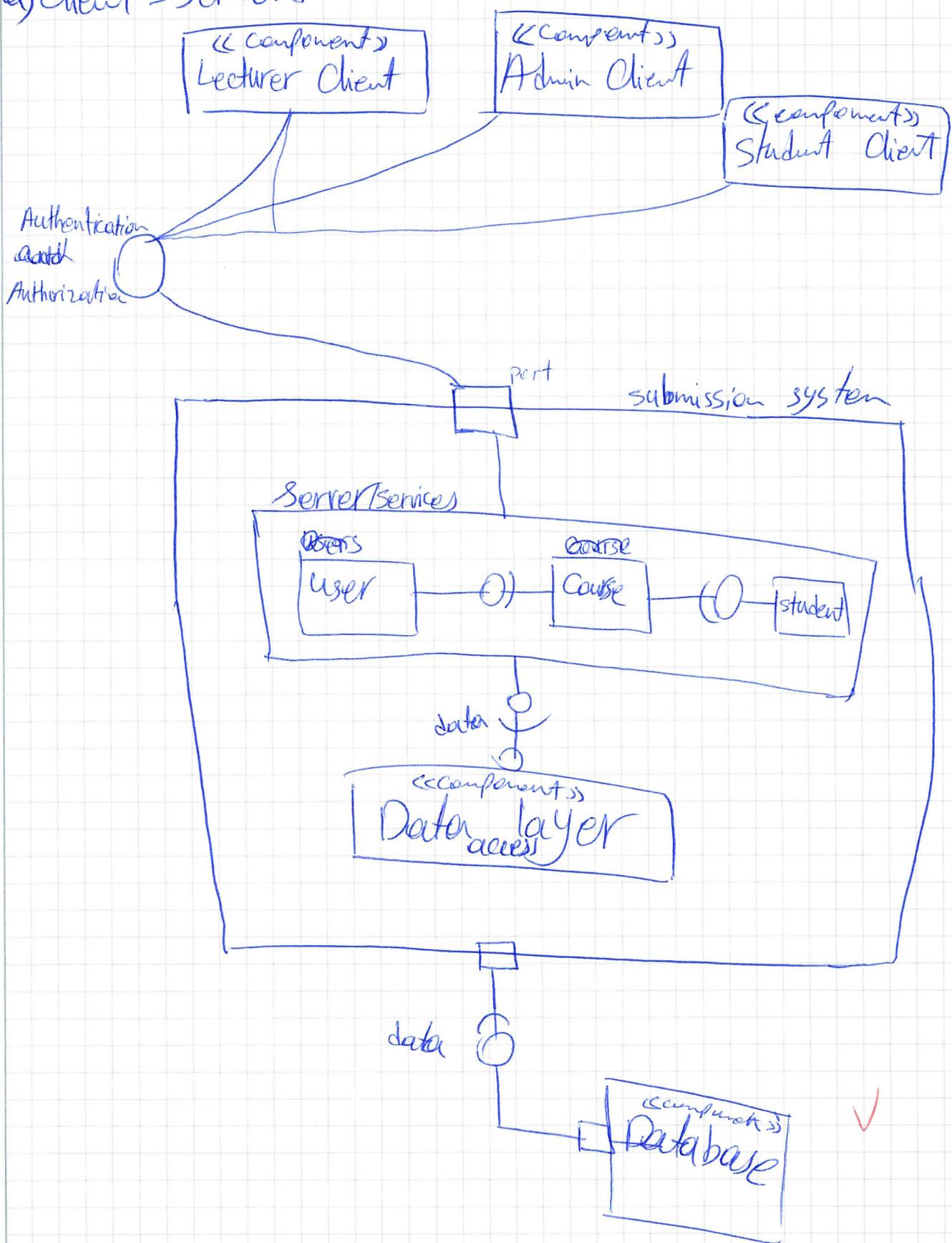




V

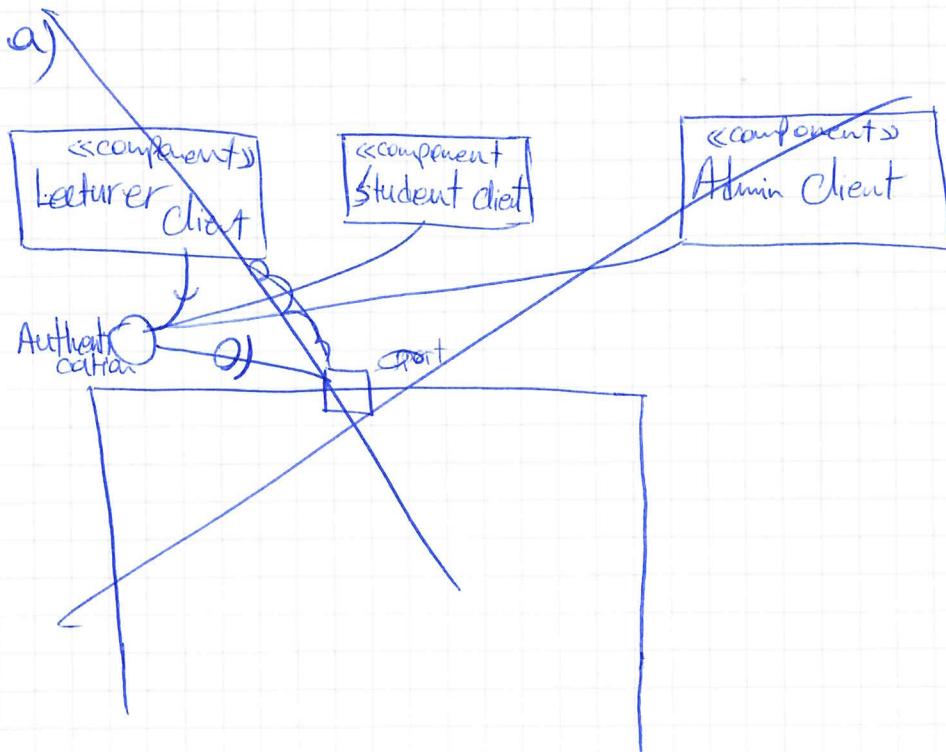
⊕

a) Client - Server



next page:

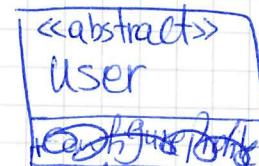
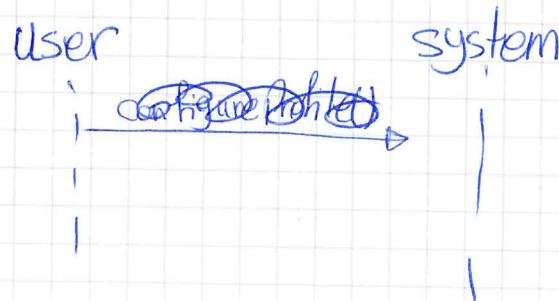
CHALMERS	Anonymous code Anonym kod DST185-0012-WJ2	Points for question (to be filled in by teacher) Poäng på uppgiften (fylltes av lärare)	Consecutive page no. Löpande sid nr 714
		Question no. Uppgift nr 5.02	



b) chose a client-server to separate the client and server. Since all users need to log in there is authentication and authorization, then the requests will go through the services where we have the domain related logic. There are services for users (as mentioned in question's description). "the info about user and admins comes from another system". Where needed, services can modify, add, delete data as that is stored in database through a data access layer. This will separate concerns of changing data and business logic.

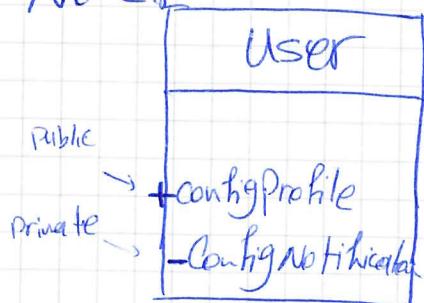
No 1:

Part 1a

6.1
6.2

The lifeline in a sequence diagram must be an interface or abstract class. (User is abstract class) ✓

No 2:



User

Public

Private

+configProfile

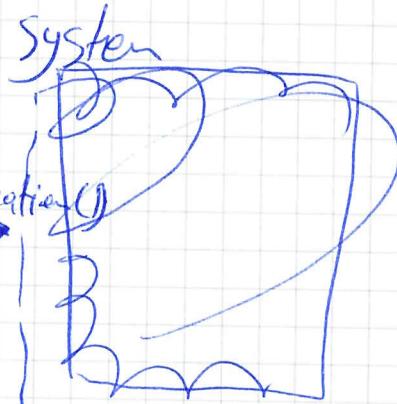
-ConfigNotification

async

>>System

configNotification()

<<



No operation can be used if it breaks the visibility rules of the class diagram (public, private). ✓

Part 2:

Next page

✓

3

① Downloading Paper does not exist as a method in class (for lecturer).

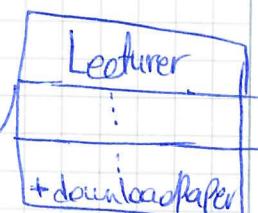
② Student should have a list of assignments ~~and not see the total grade.~~

③ Idea

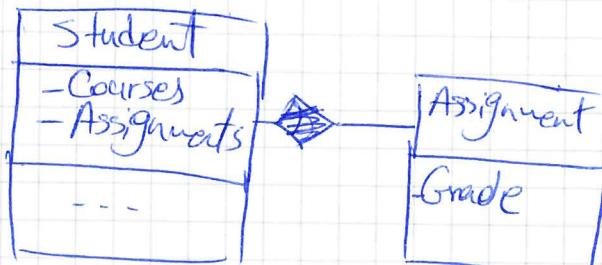
③ we don't have anything to "inform" or send notification. So a class to notify users should be added.

As a result:

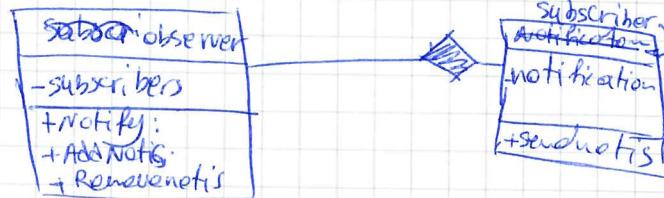
① ➔ lecturer will have a "downloadPaper" method



② ➔ student will have a list of assignments (new class called Assignment) to store extra info about it.



③ ➔ Adding a notifier



✓
③

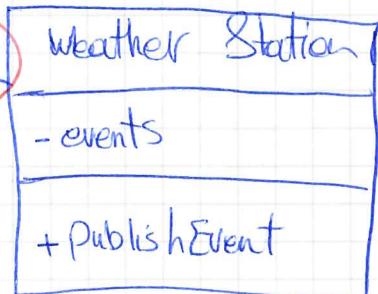
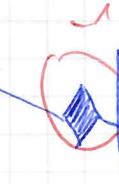
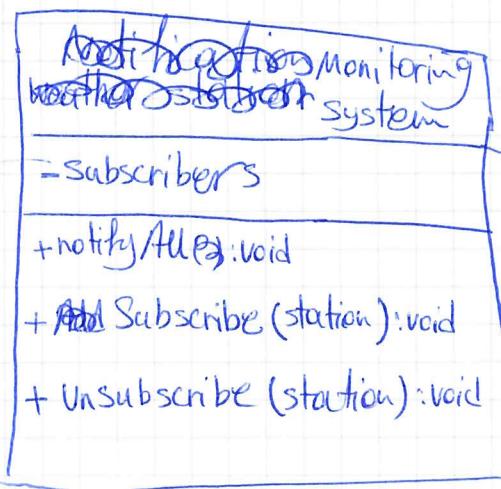
CHALMERS	Anonymous code Anonym kod DIT185-0012-UJZ	Points for question (to be filled in by teacher) Poäng på uppgiften (ifylls av lärare)	Consecutive page no. Löpande sid nr 10/14
----------	---	---	---

Design Patterns 8

Case 1:

Observer pattern

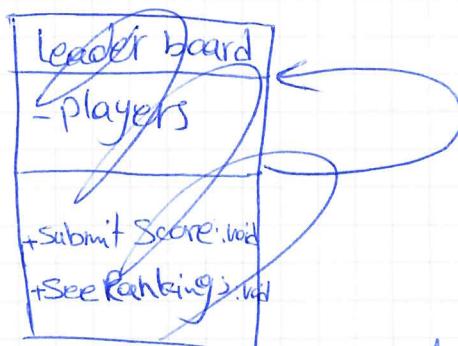
stations



(4)

Case 2:

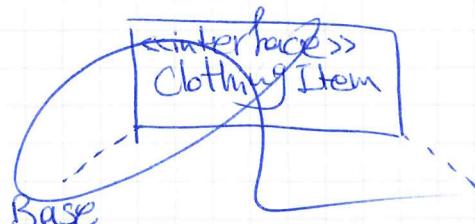
~~Single~~ Singleton pattern



& construct func -1

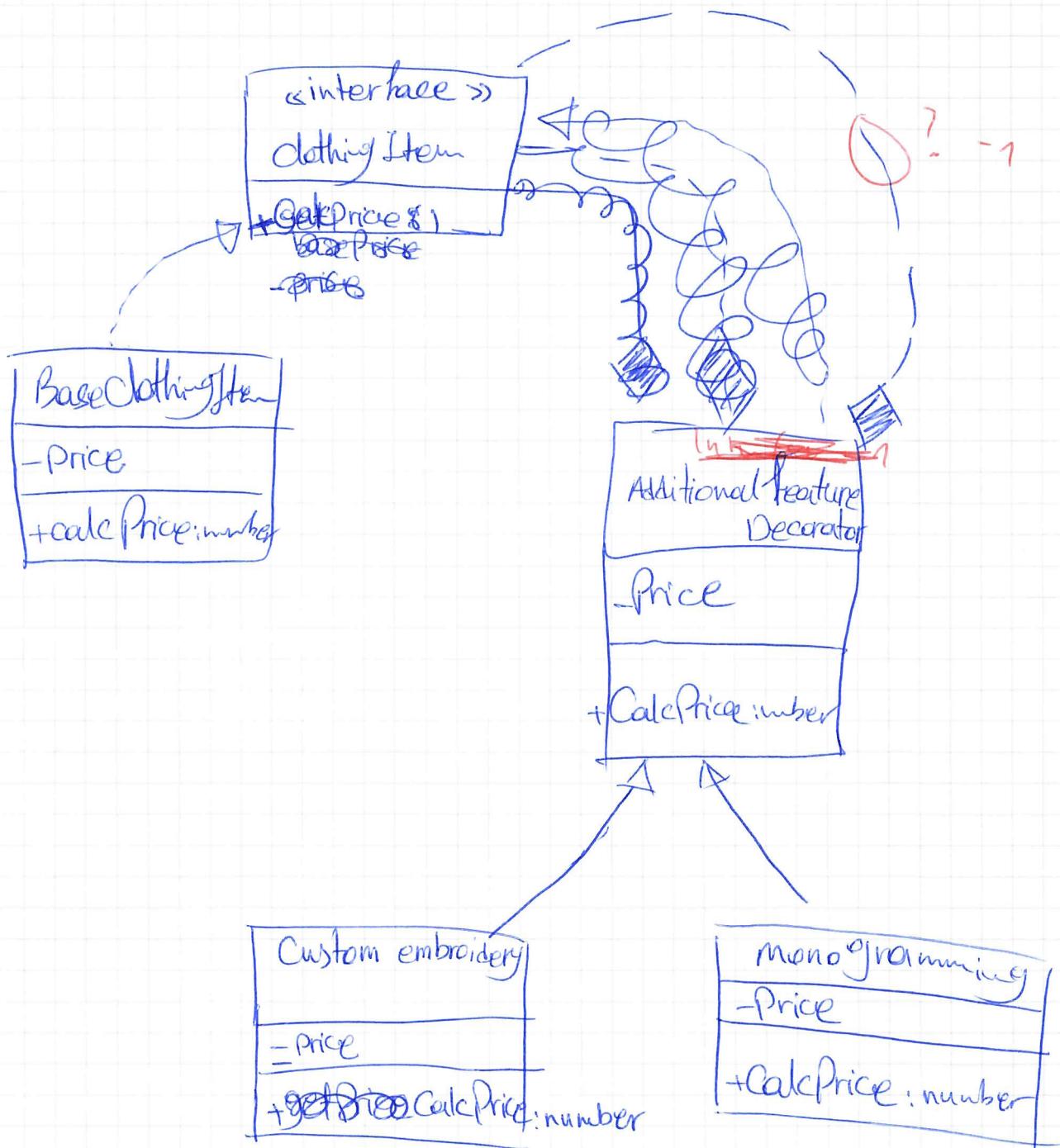
It will create a new board first time and return the same board all other calls.

Case 3:



Case 3:

Decorator Pattern



(4)

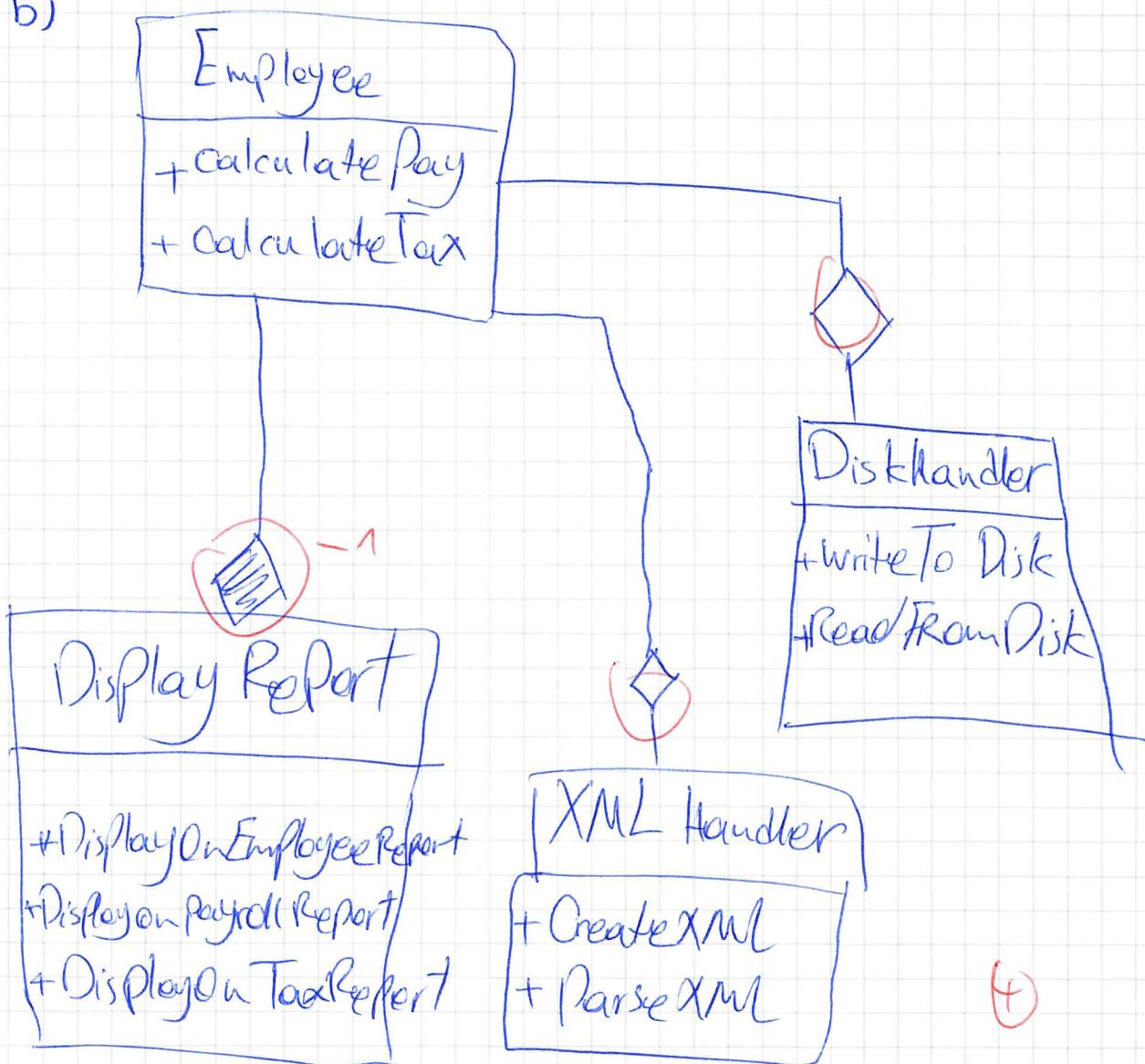
CHALMERS	Anonymous code Anonym kod DIT185-0012-UJZ	Points for question (to be filled in by teacher) Poäng på uppgiften (fylltes av lärare)	Consecutive page no. Löpande sid nr 12/14
			Question no. Uppgift nr 8.1

a) The given class, ~~has~~ ^{has} many different stuff (methods), doing different tasks. ~~Calculating pay and salary~~
~~and other~~ do what an Employee is
~~can have or might have different cases in future~~
~~for example,~~
 It ~~does~~ has methods for employee tax, pay, disk
 read/write ~~etc~~, XML related actions, and displaying
 reports, Each of these have their own implementation
 that can grow over time and the class will be less
Scalable.

The
Modularity and separation of concerns: As mentioned
 the Employee class does many multiple different
 tasks, that could had been group other classes
 to separate concerns. (For example read/write to
 disk ~~is not implemented~~ could be ⁱⁿ its own module.)

V
S

b)



Next Page

CHALMERS	Anonymous code Anonym kod DST185-0012-WJZ	Points for question (to be filled in by teacher) Poäng på uppgiften (fylltes av lärare)	Consecutive page no. Löpande sid nr 14/14 Question no. Uppgift nr 8.3
----------	---	--	--

c) Separating the XML and disk related methods

into their own classes will provide more modularity and makes it more testable, scalable for future.

~~because~~ It will separate the ~~possibility~~ implementation with all deps needed for XML and disk operations ~~and give~~ which makes it more scalable (for adding new feature). ~~testing~~ helps testing cause we can test each operation separately. ~~and without~~

DisplayReport related methods ~~had also~~ were also close enough to be in a separate class.

~~because~~ So that the specifics of working with stream is separated from the Employee class.

Although it would be more coupled with Employee than Disk and XML handlers.

To sum, the changes will ~~would~~ break down the class make it more modular, and separate the different concerns ~~and~~ of XML, disk, stream handling, making each module more scalable and fastable.

V5