CHALMERS
EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			
DIT185	SOFTWARE ANALYSIS AND DESIGN			
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
DIT185-0017 - HCM		23.08.2023	10	

* I confirm that I've no mobile or other similar electronic equipment available during the examination.
Jag intygar att jag inte har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under eximinationen.

Solved task Behandlade uppgifter No/nr		Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylles av lärare.
1	X	12/12	
2	X	12/12	
3	X	12/12	
4	X	10/12	
5	X	12/12	
6	X	8/10	
7	X	10/15	
8	X	12,5/15	
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			
Total examination points Summa poäng		88.5/100	

[redacted]

3

8/8

Clarification:

- "Upload papers" extends "Hand in assignment", since not all assignments require attachments
- "View courses and tasks" were grouped for readability, they should've been two seperate use cases

3

Task 1B

Use case name: Hand-in assignment
Goal in context: Successfully submit an assignment
Primary actor: Student
Success End Condition: Assignment is submitted for grading
Failed End Condition: Student is unable to submit the assignment
Trigger: Student goes into the assignment page
Main success scenario:
Step #1 Student enters the assignment page
Step #2 Student selects the option to submit the assignment
Step #3 User fills out the fields necessary for submission
Step #4 User confirms, that the assignment is their own work
Step #5 User selects the option to send the assignment for grading

keep using same role!

Extensions:
Step #3a User selects the option to add attachments
Step #3b User selects file from their device
Step #3c User confirms the upload of selected files

Sub-variations:
Step #2a The assignment is past deadline
Step #2b User is unable to make a regular submission
Step #2c If the assignment allows it, the user may make a late submission

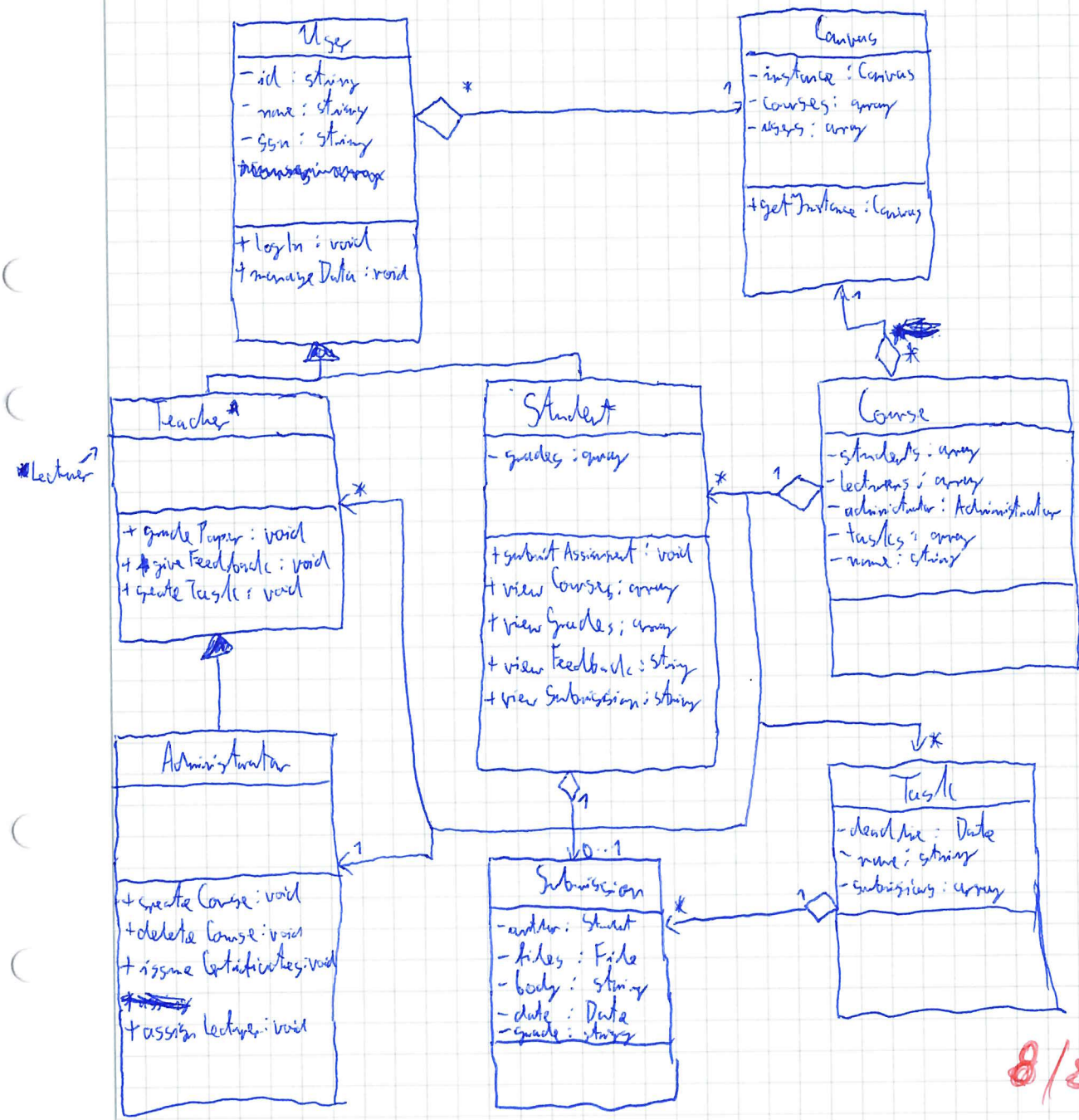
that's not a step but the name of the variation

Note: After Step #2 I've mixed up "Student" with "User", and while technically Student is a User, all steps should say "Student" ✓ ok!

4/4

3

Task 2a



8/8

Task 2b

Assumptions:

- The system is named „Canvas" for simplicity, and is a singleton
- Method manageData allows users to change their data
- It it's too messy, I'm sorry, I tried :c

3

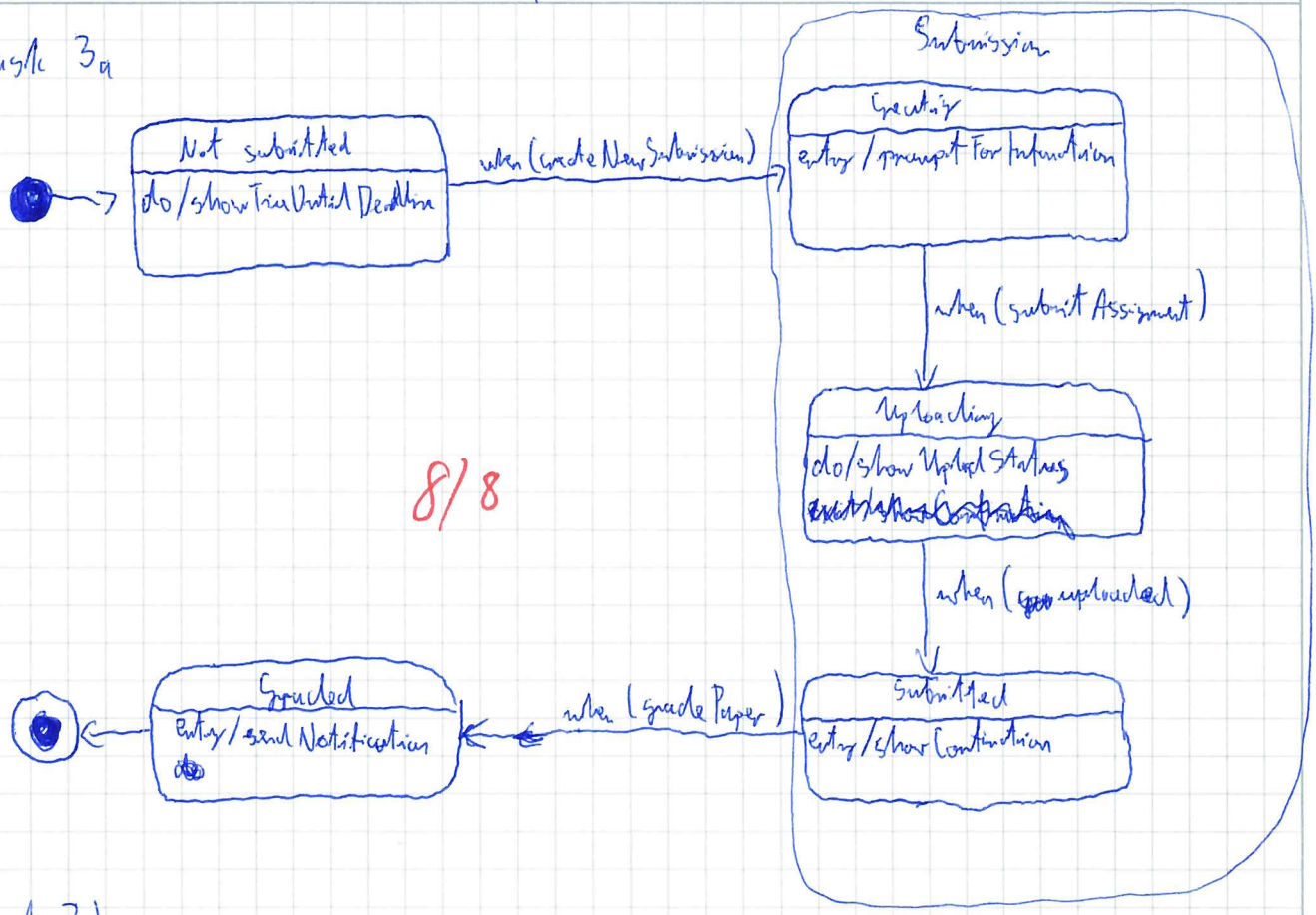
Task 2b cont.

I've developed the domain model by analyzing the case description and identifying all actors and their responsibilities. Then I thought of potential solutions, as to how the system should look like. I decided to make the system (Canvas) a singleton, since there should only be one instance of it. It stores the data about all users and courses. The top-level users are User, which then divide into Lecturers and Students. Some Lecturers may be Administrators. Specific types of users are then assigned to Courses, which can also have Tasks (assignments).

4/4

3

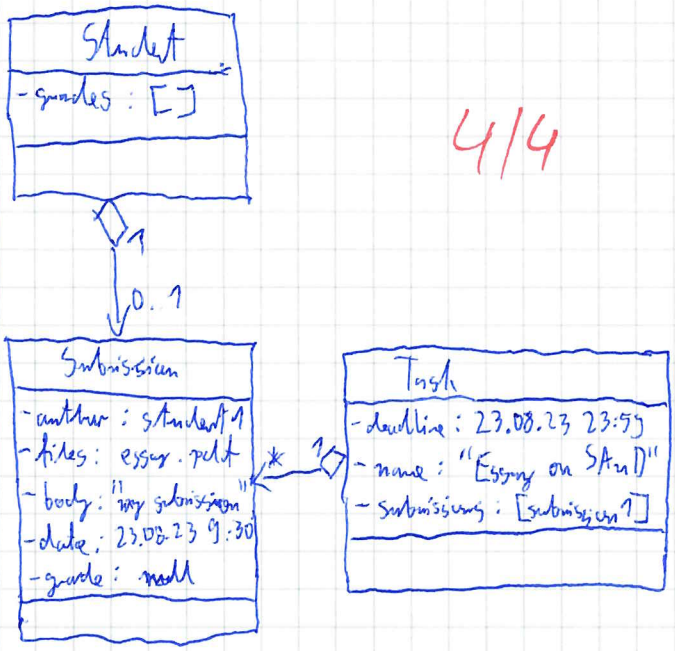
Task 3a



8/8

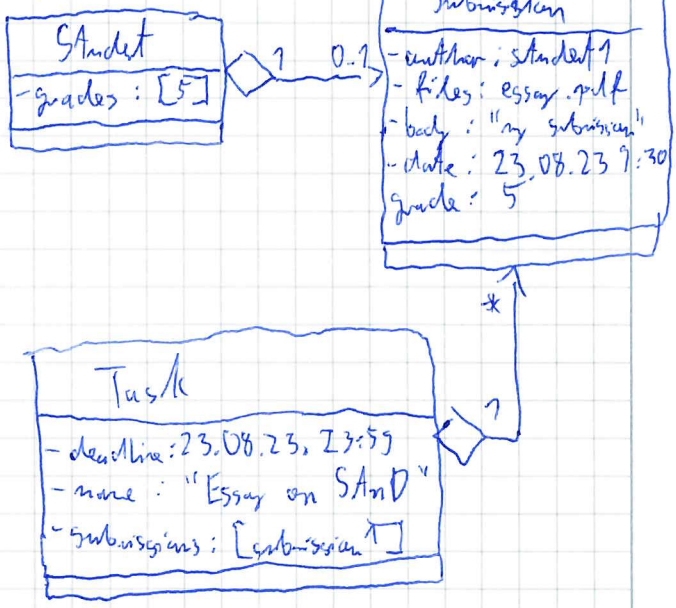
Task 3b

Before (Submitted)



4/4

After (Graded)



Well done!

CHALMERS

Anonymous code

Anonym kod

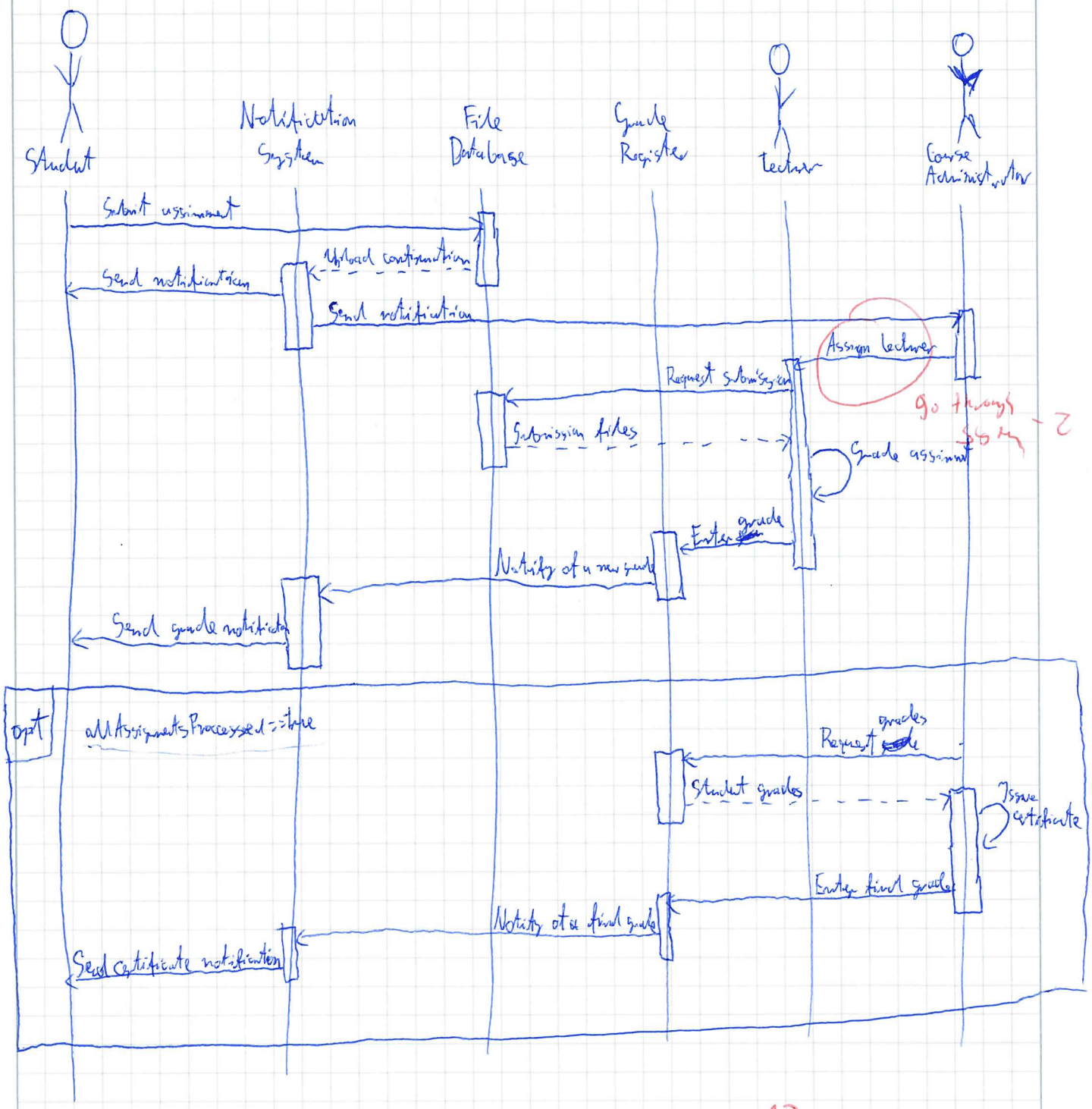
DIT185-0017-HCM

Points for question (to be filled in by teacher)

Poäng på uppgiften (ifylles av lärare)

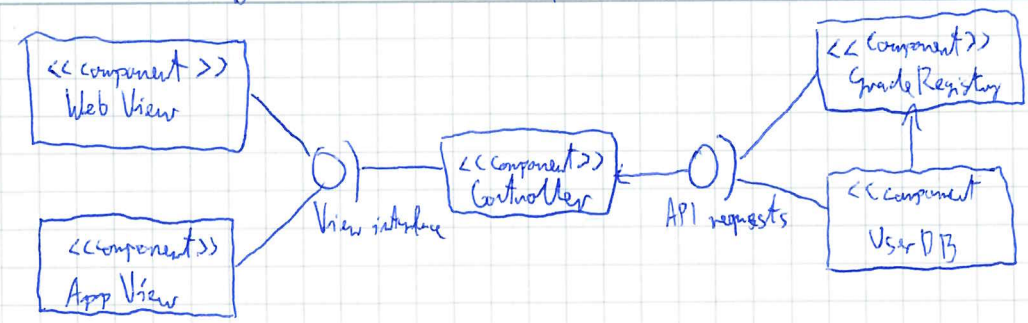
Consecutive page no. Löpande sid nr 6

Question no. Uppgift nr 4



10

3



8/8

I've chosen an MVC architecture style, because ~~the~~ the system will require multiple views (i.e. web and apps), as well as multiple data sources. This structure allows for an easy expansion, if one is needed, and ~~s~~ seperates the end-user frontend from the backend via the controller

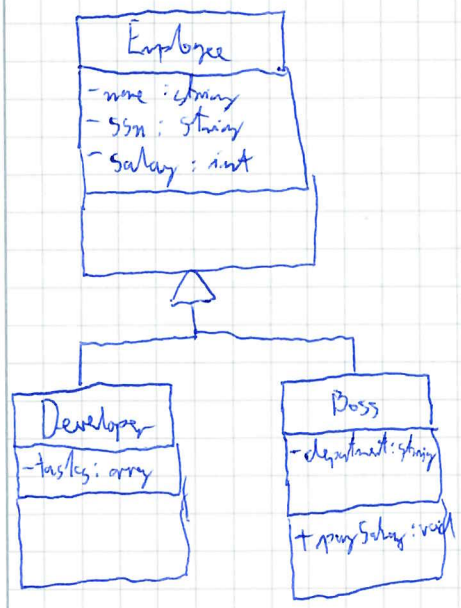
4/4

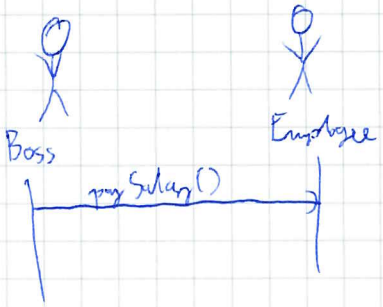
impressively minimal complete solution :)

3

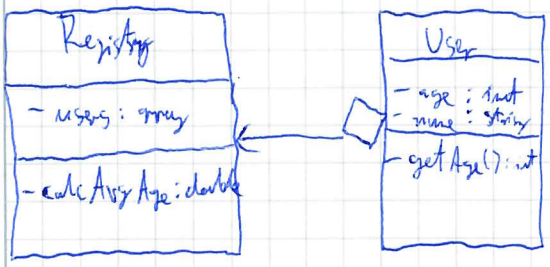
1.

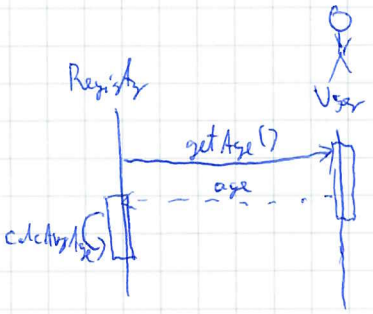
Rule #8





Rule #9





✓ (5)

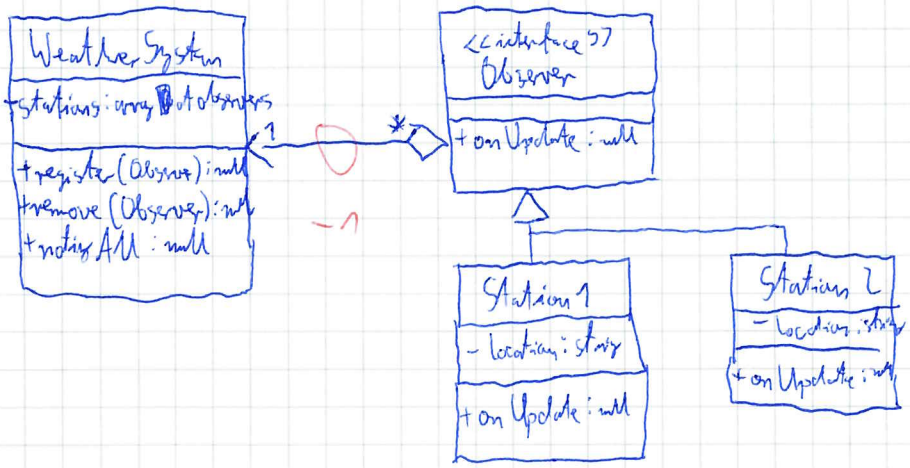
2.

Vague

- Make the class diagram more white-box, creating for example notification service, databases, and grade register.
- There are no getters/setters, so for example Course Administrator wouldn't be able to request student grades ✓
- There should be a more concrete way of handling final grades ✓ (3)

3

Case 1: Observer



-1

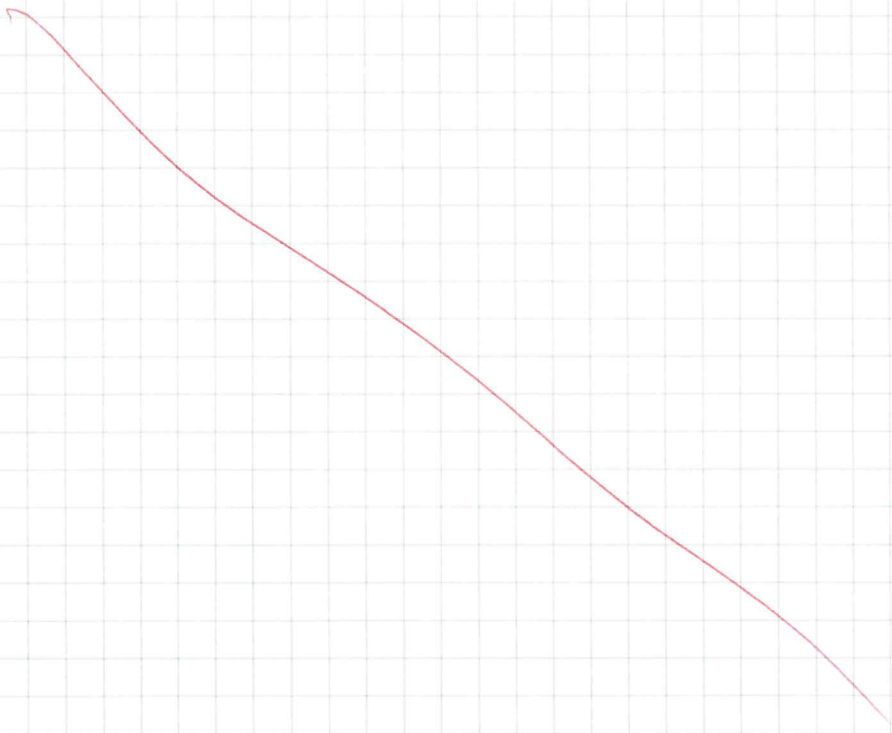
4

Case 2: Singleton

V

5

Case 3: Decorator V 1

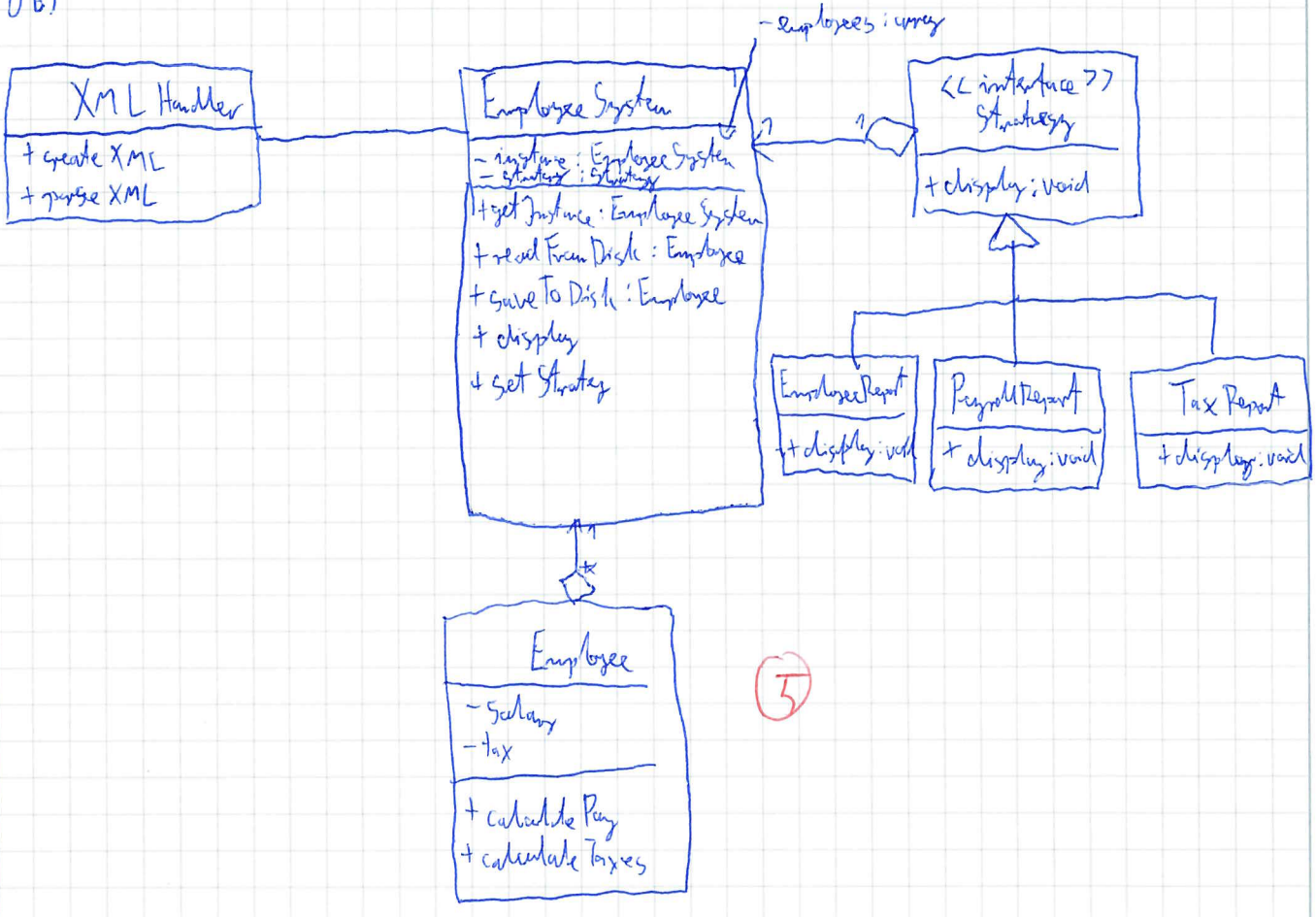


3

8a) The given class diagram is of bad quality, as it violates the Single Responsibility Principle and most likely is monolithic code, with similar repeating functions. ✓

Too short (no details) (2.5)

8b)



(5)

8c)

With the new diagram, each class is responsible for only things related to it (for example XML files are now handled by XML Handler, which is called when reading/saving to file). Different reports are now displayed by using a Strategy pattern, which allows for less repetition in code. (5)