

CHALMERS

EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn		
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad
	470	22-08-24	13
			Grade Betyg 4

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
 den här tiden.

Index o/nr	Marked task Markade uppgifter	Points per task Poäng på uppgiften	Observation: Areas with bold contour are to be completed by the teacher. Anmärkning: Rutor inom bred kontur ifylls av lärlare.
1	X	12	
2	X	7	
3	X	13	
4	X	12	
5	X	4	
6	X	10	
7	X	7,5	
8	X	15	
9			
10			
11			
12			
13			
14			
15			
16			
17			
Bonus poäng			

CHALMERS	Anonymous code	Points for question (to be filled in by teacher)	Consecutive page no. Löpande sid nr
	Anonym kod DIT185 470		
Task 1. a)	<pre> classDiagram class MicrobloggingService { CreateAccount ReceiveActivationCode ActivateAccount GetUserPage ConfigurePage WritePosts SharePosts SubscribePostings SendPrivateMessages FollowUser BanAccountsUsers temporaryBanAccountsUsers } class User class Admin CreateAccount "include"--> > ReceiveActivationCode GetUserPage "extend"--> > ConfigurePage BanAccountsUsers "extend"--> > temporaryBanAccountsUsers </pre>	Points for question (to be filled in by teacher)	Consecutive page no. Löpande sid nr

Notes:

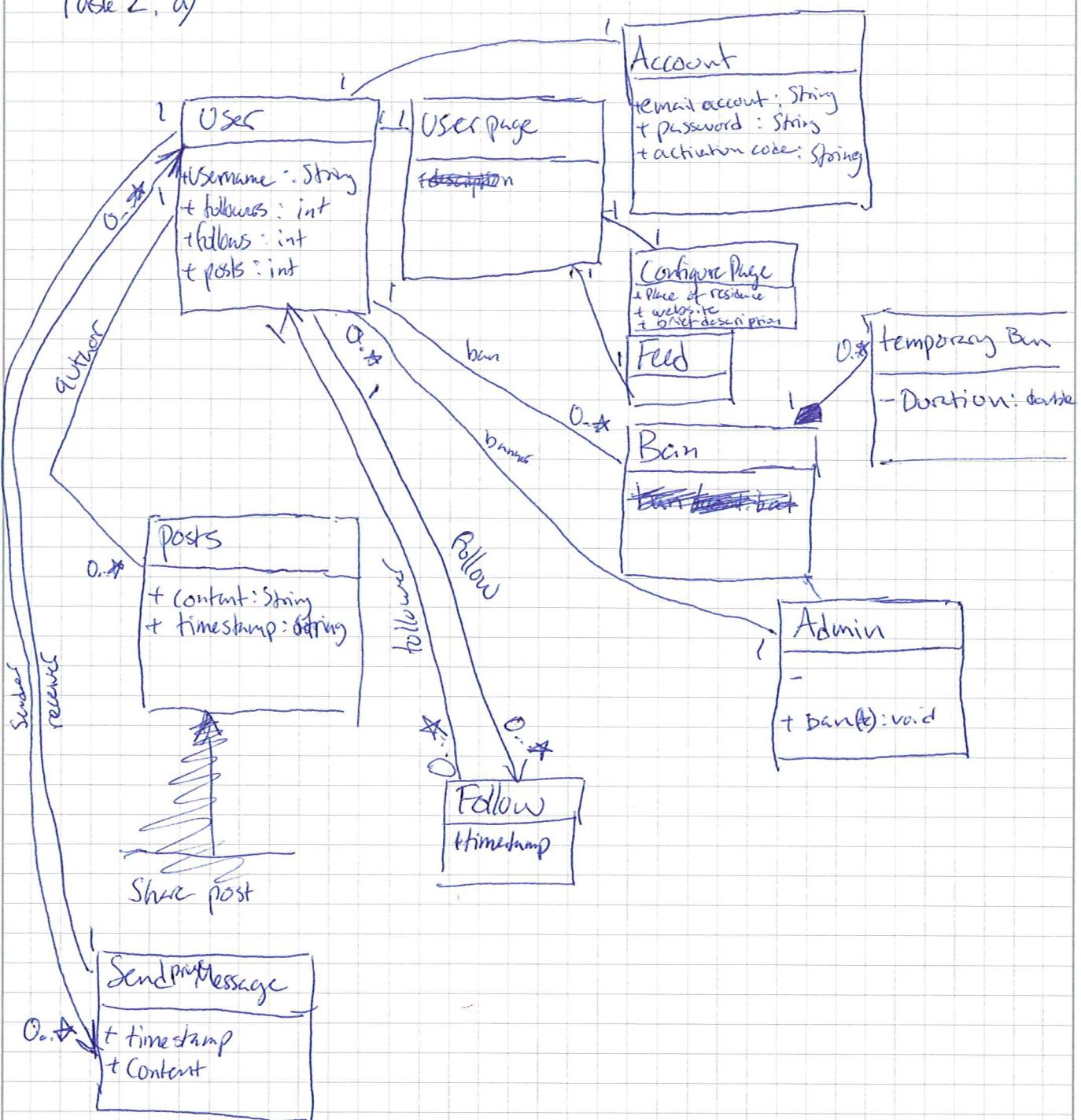
- All that the user can do can the admin also do. And more.

- When creating an account, every time, there will be an activation code sent out. Therefore **<include>**
- A user has userpage where it is optional to configure a page. Therefore **<extends>**
- When an admin ban an account it is optional to ban temporary. Therefore **<extends>**

-1 admin is a secondary user, right side

CHALMERS	Anonymous code	Points for question (to be filled in by teacher)	Consecutive page no. Löpande sid nr
	Anonym kod DIT185 470	Poäng på uppgiften (fyller av lärare)	Question no. Uppgift nr
Task 1	Codeburn template		1
b)	Description		
	<p>Use Case: Create account</p> <p>Goal in context: As a user, I want to be able to successfully create an account on the microblogging service.</p> <p>Primary Actor: User</p> <p>Pre-conditions: That the user does not have an account to begin with.</p> <p>Success End Condition: Successfully create an account.</p> <p>Failed End condition: Not be able to create an account, find it confusing and fail</p> <p>Triggers: Creating an account by providing their Username, email address and password.</p> <p>Main Success Scenario:</p> <ol style="list-style-type: none"> 1. User registers for an account by providing username, email address and password 2. The user receives an activation code on their email 3. Use this activation code to activate their account. As it is inactive until then. 4. Sign in to account <p>Extensions:</p> <ol style="list-style-type: none"> 1.1 User registers for an account either on Phone app or web browser. 2.1 Provided that they have an email, or they need to get one. <p>Sub-Variations:</p> <ol style="list-style-type: none"> 1.1 If the user already has an account with that email then he/she can not create a new one. 2.1 The activation code could end up in "Junk mail" 		

Task 2, a)



- 1 User and Account is the same class
- 1 Admin is a user (inheritance)
- 1 Ban / Temp ban are missing attributes

6

CHALMERS	Anonymous code	Points for question (to be filled in by teacher)	Consecutive page no. Löpande sid nr
	Anonym kod DIT185 470	Poäng på uppgiften (fylltes av lärare)	Question no. Uppgift nr 2

Task 2

b) I analysed the text and decided to create a class for all nouns. This way I can break all classes, make it more organized and separate where needed.

When finding attributes it is ideal to identify in the text when it refers to a class and its qualities like "they specify", "needs an",

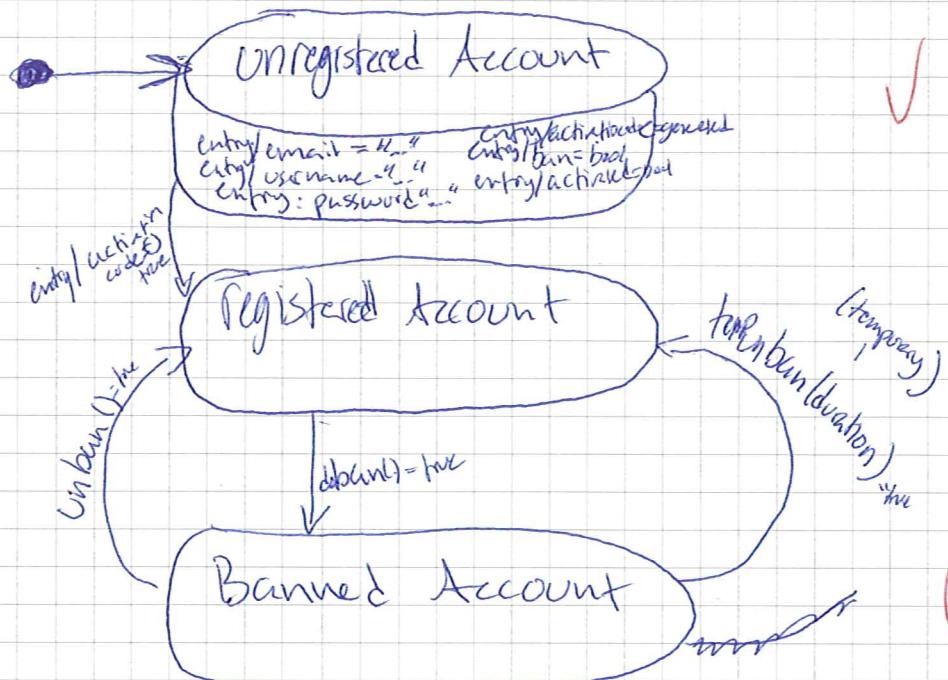
Regarding the relationships for all classes you simply look for other key words, like "can be" which means its not 1 to 1. More like 1 to 0...★
Every registered user owns a publicly accessible user page! Every! This explains that all users will have a user page. This is how you find relationships

-1 too short

Task 3.

a)

State Machine Diagram



b)

Before: Unregistered Account

```

email = "albin_carlsson@hotmail.com"
username = "Albin Dr Karlsson"
password = "A567B!!ef"
activationcode = "123456"
ban = false
activated = false
  
```

After: registered account

```

email = "albin_carlsson@hotmail.com"
username = "Albin Dr Karlsson"
password = "A567B!!ef"
activationcode = "123456"
ban = false
activated = true
  
```

In the "before" state the account is not activated.
In the "After" state the account is activated.

✓

(4)

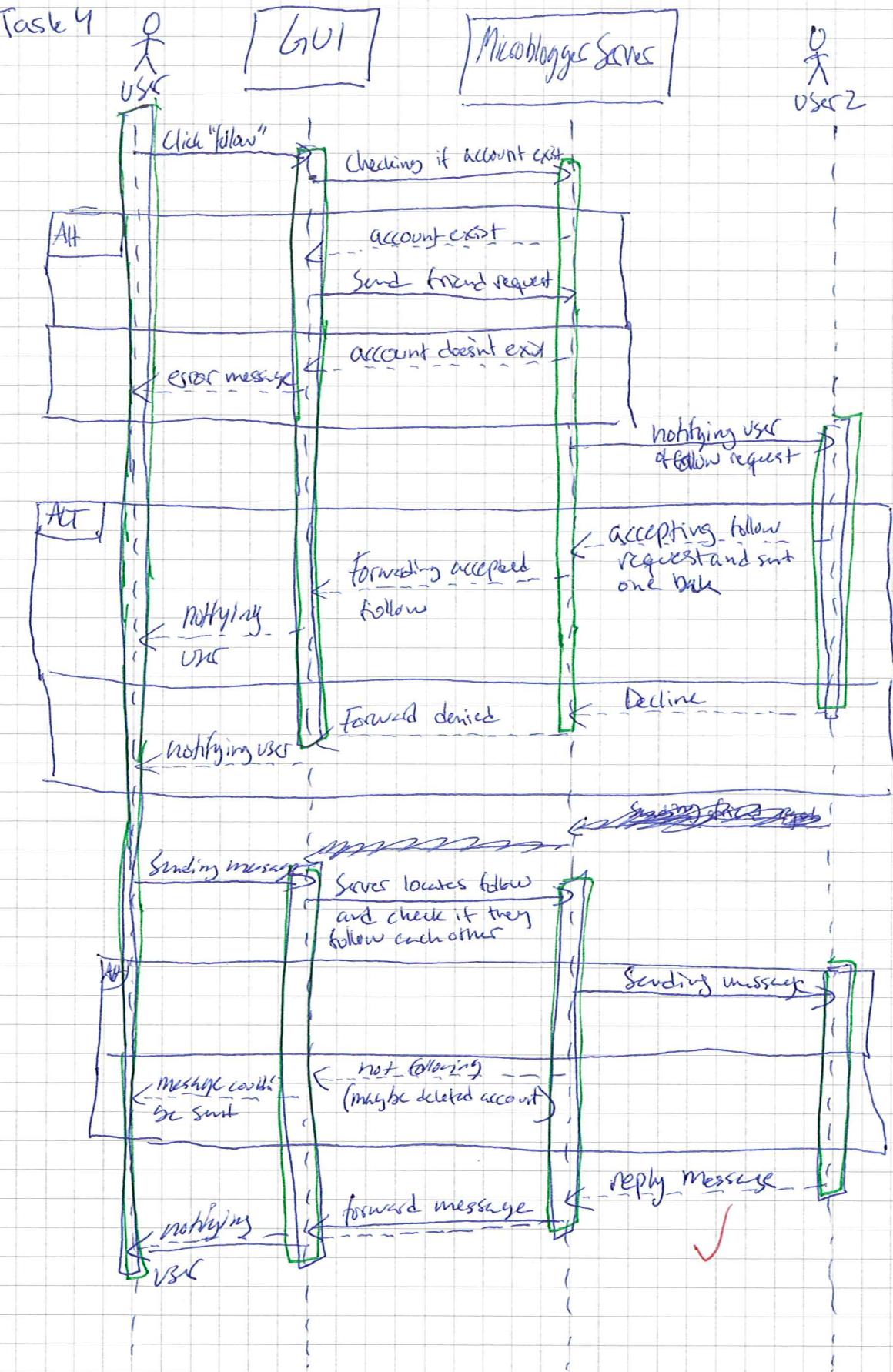
Task 4

User

GUI

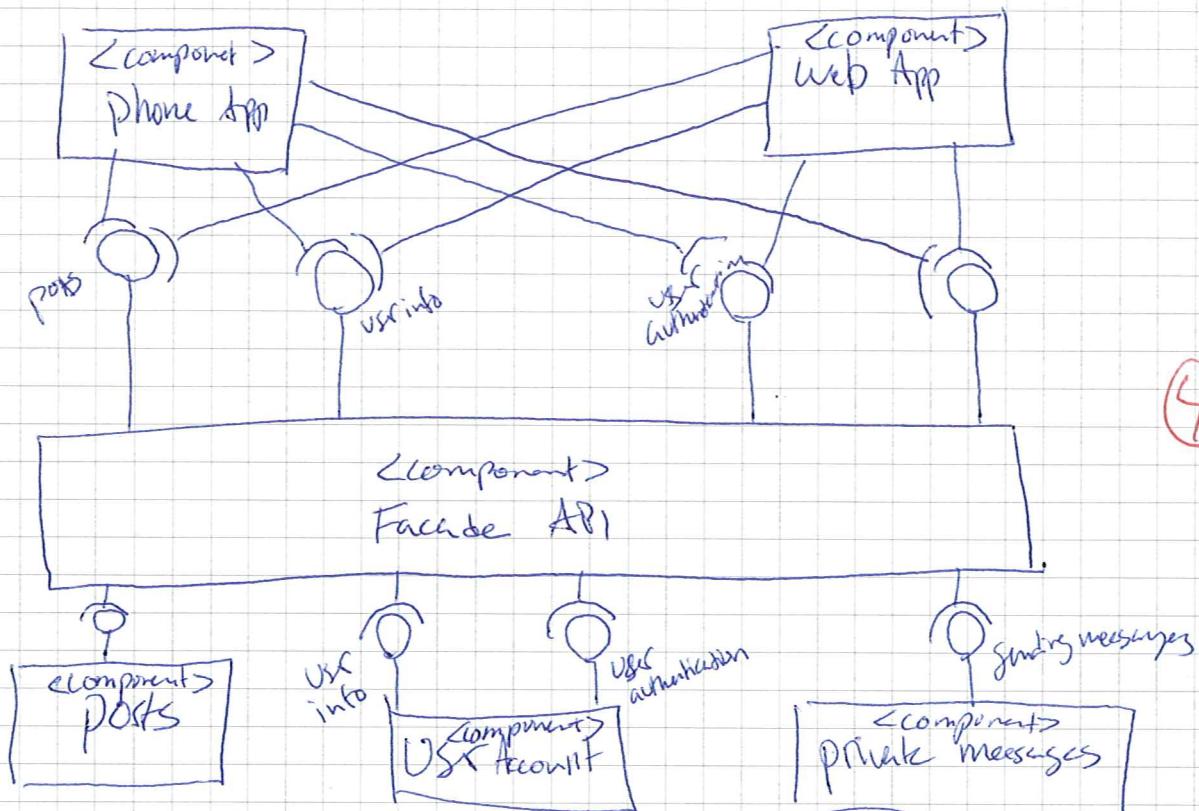
Micoblogge Server

User 2



Task 5.

a)



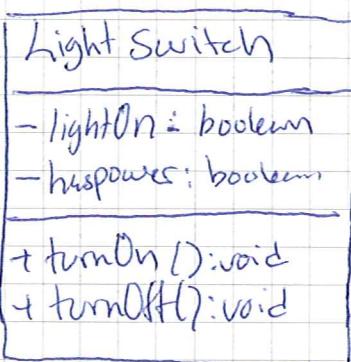
Point of using facade unclear, making it impossible to judge. The instantiations makes score ~4

b)

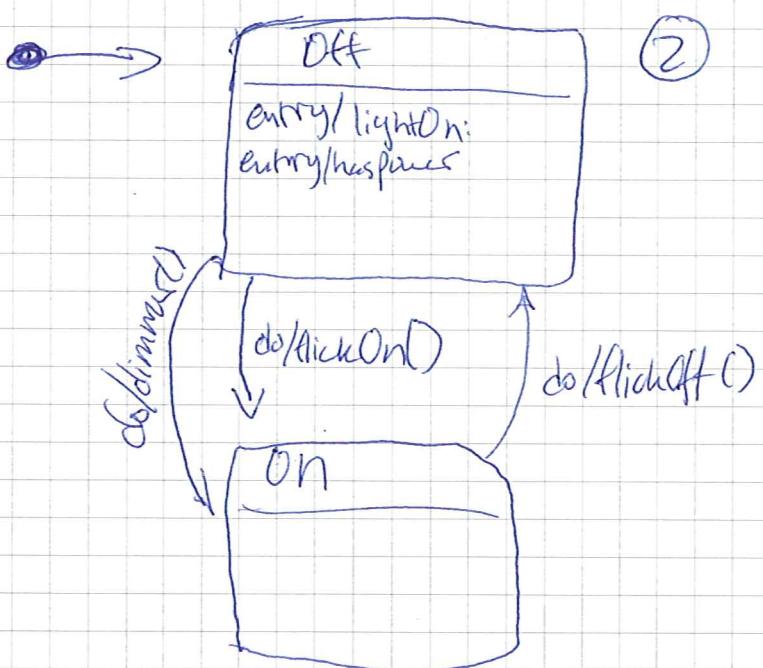
Task 6.

1.

(1)



SwitchL



Explanation:

1. Naming conventions: Both classes, some methods have different names which is not correct. Naming conventions should be the same. Not using other words or shortenings names. ✓
2. In the state machine diagram there is an extra method called dimmer() which the class diagram does not have. From the looks of it this is not done correctly! ✓

① (LightSwitch/SwitchL) ② turnOff() / turnOn()

6.

2. I'm in able to complete this task.

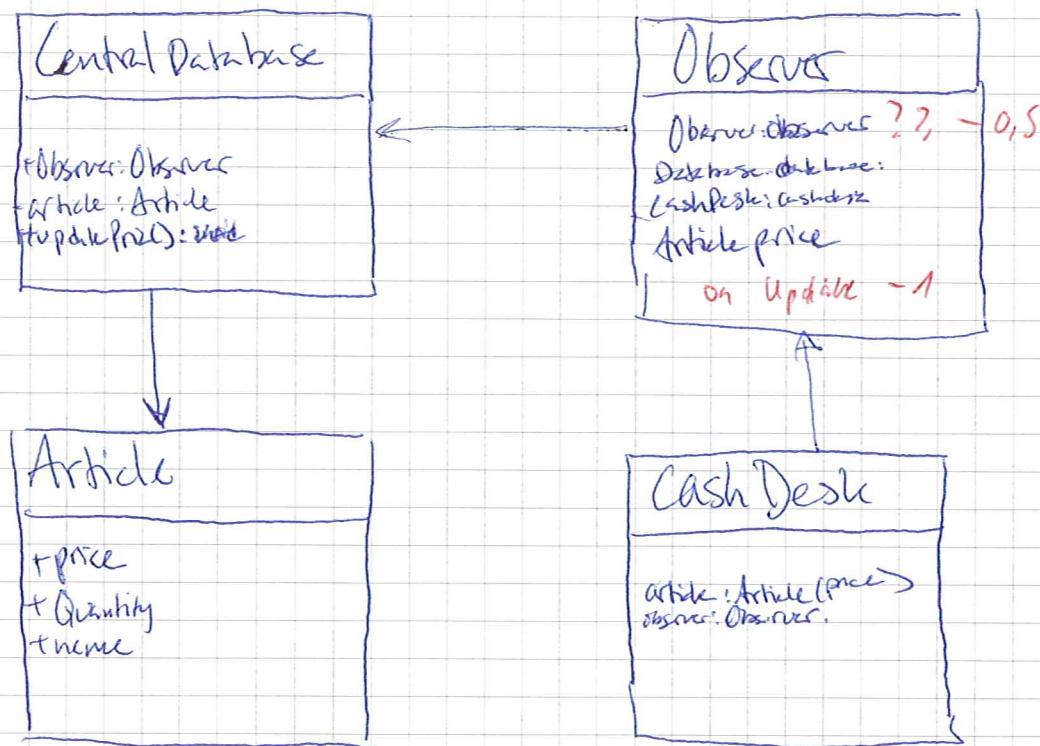
It asks me to consider my sequence diagram from task 3. There is no sequence diagram on task 3.

I'm not sure if it is sequence diagram from task 4 or the state machine diagram from task 3.

However, if it is the sequence diagram on task 4. Then the class diagram has a few methods like bahn() ~~etc~~ which is not presented on the sequence diagram. ✓

Task 7.

Case 1: For this case it is most appropriate to use the "Observer" design pattern.

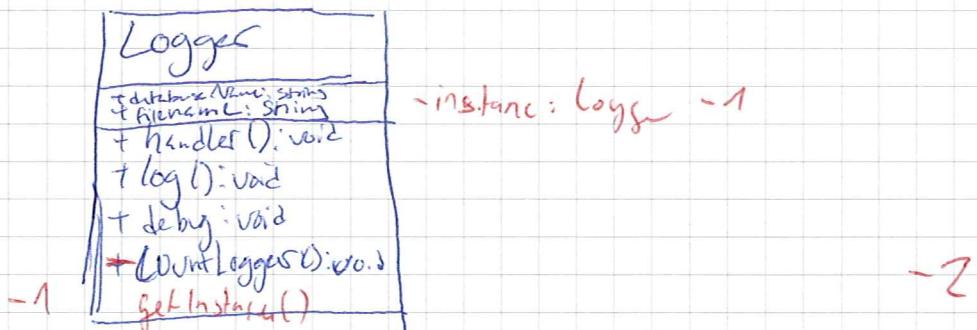


The database collects all data there is and the observer will observe the article price and the cashDesk will update the information every time the article price changes.

(2,8)

Task 7.

Case 2: For this case it is most appropriate to use the "Singleton" design pattern.

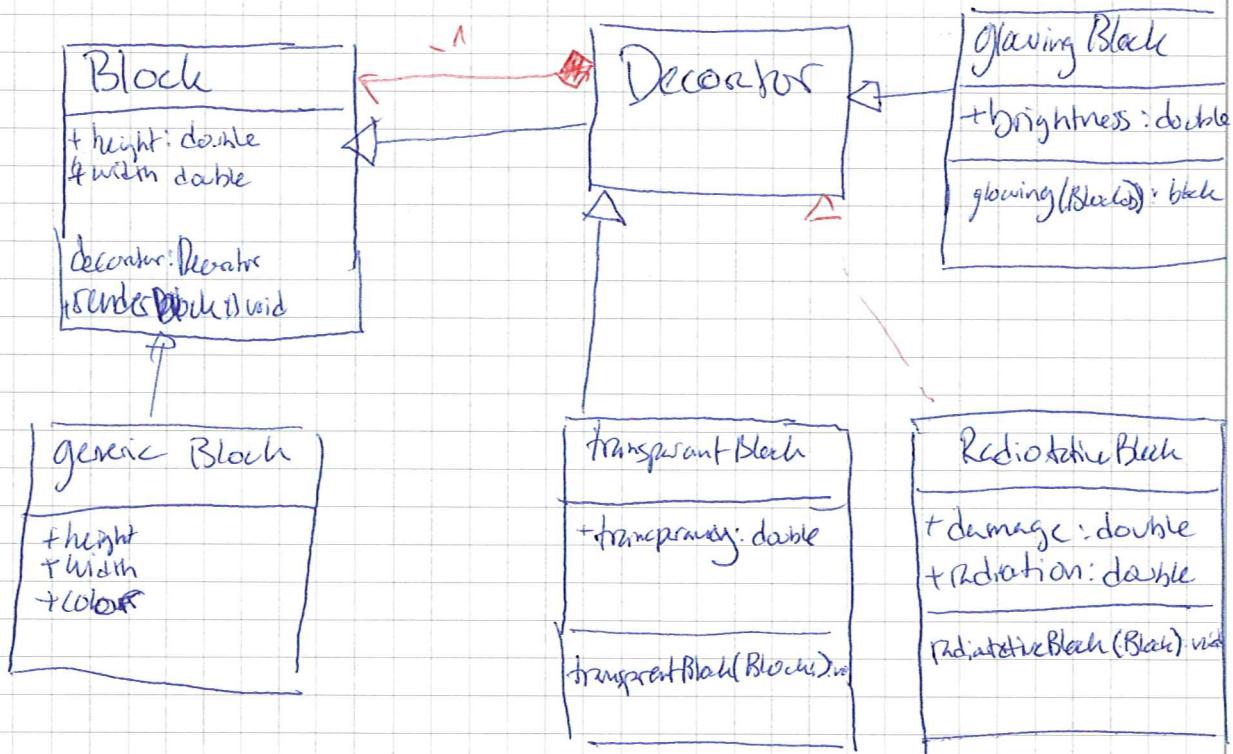


Single object handling all logging

(7)

Task 7.

Case 3. For this case it is most appropriate to use the "Decorator" design pattern.



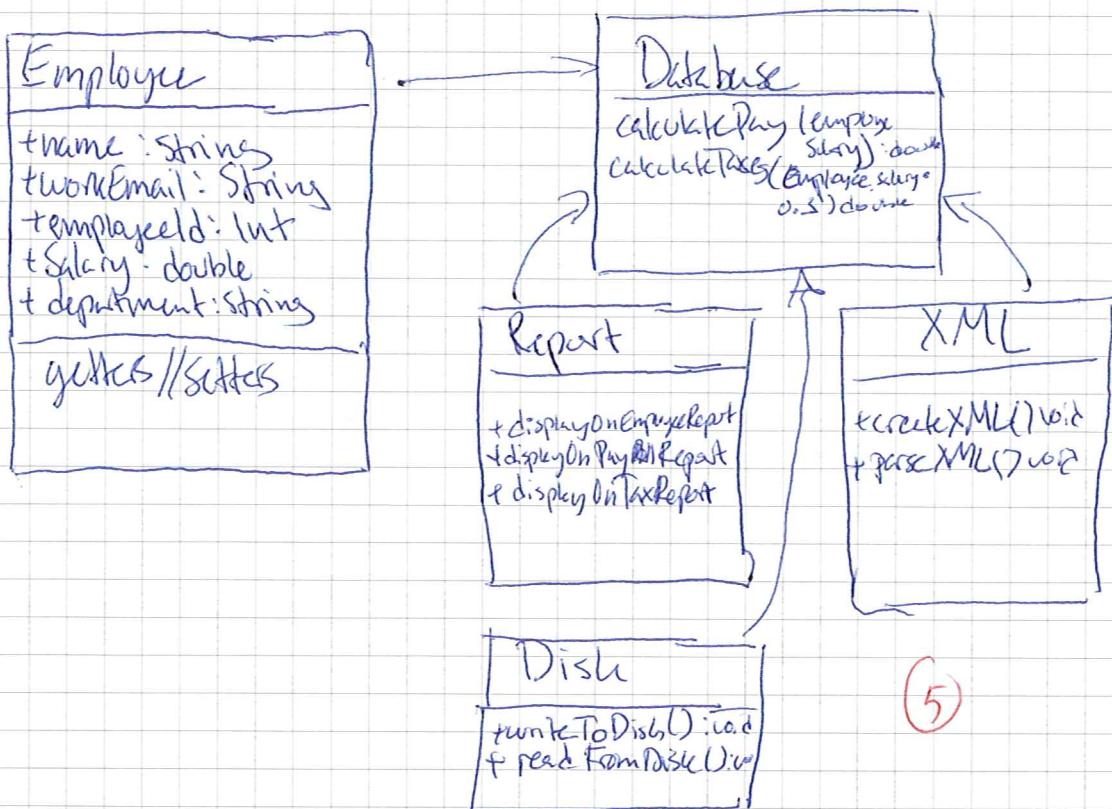
still ③

Task 8.

a)

- ① It is not good design principle that the Employee class does everything like calculating its own salary and displayOnPayRollReport. This is bad practice. It is better to divide up the class into multiple classes.
- ② There are no attributes at all. This is also not a good practice. What is an employee? No name? EmployeeId? Maybe department?

b)



- ③ Now we have separated the classes and added some attributes to the employee. ✓ (5)