

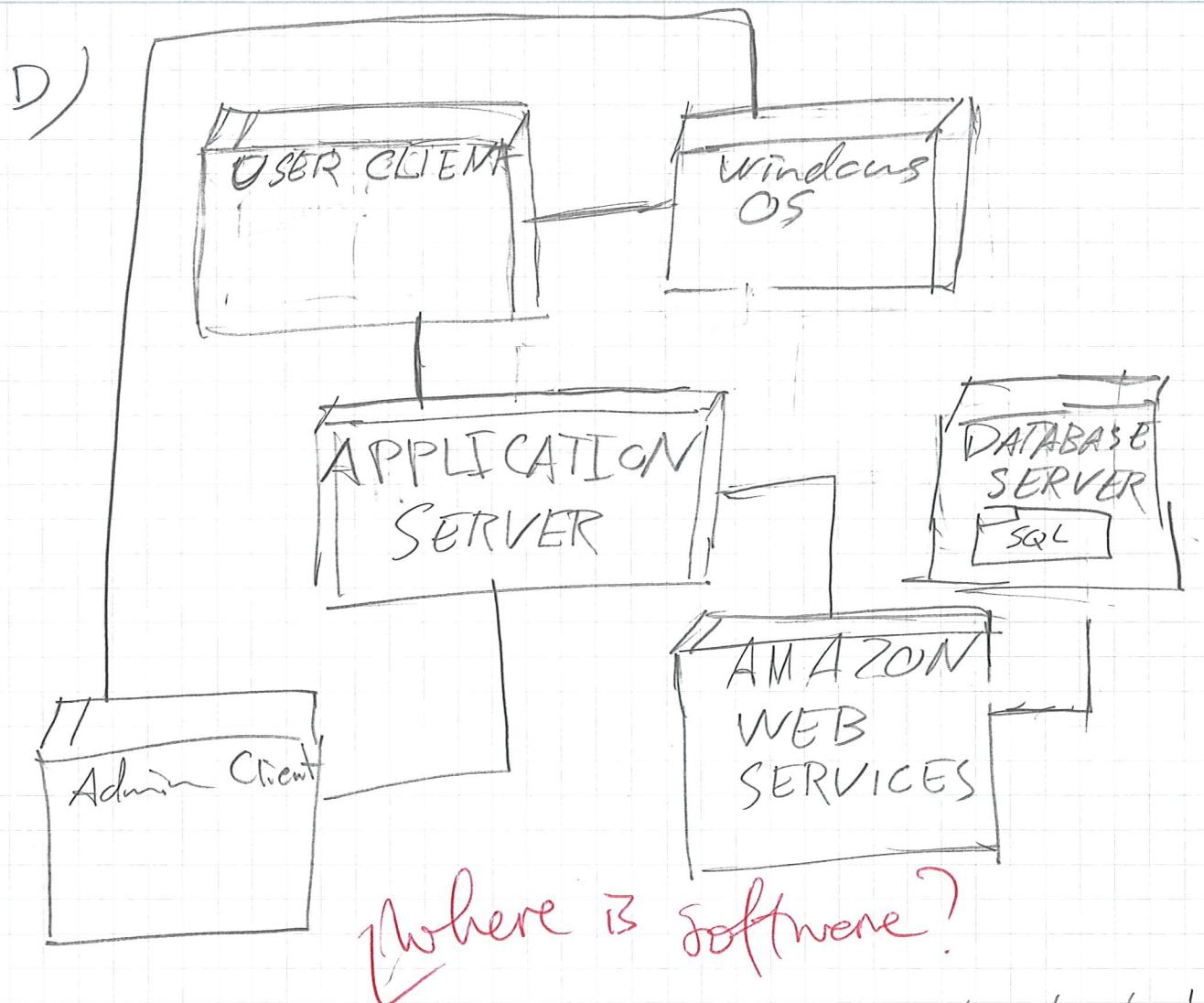
CHALMERS

EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn		
DIT541	Software Architecture		
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad
719		21/8-19	9

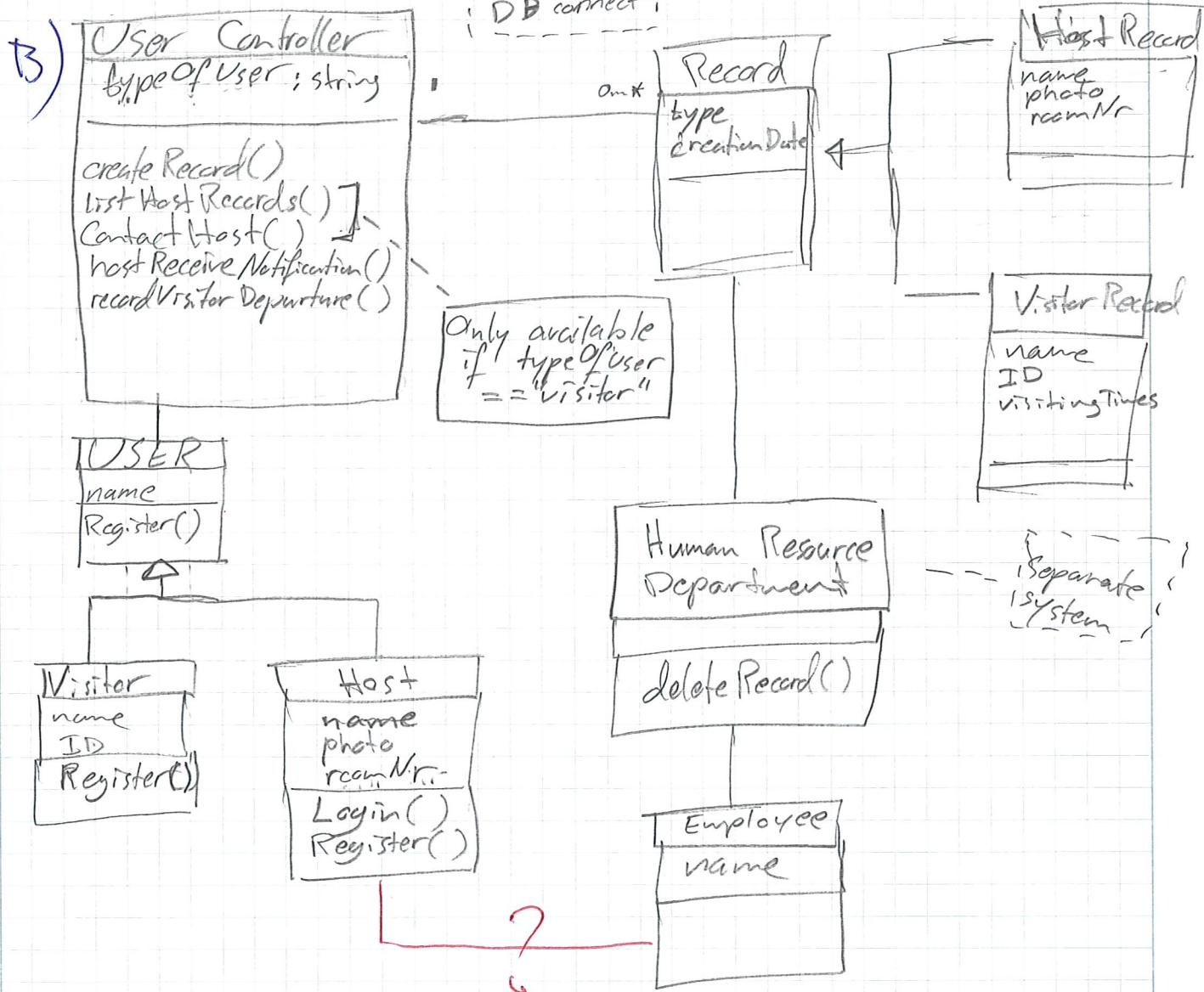
* I confirm that I've no mobile or other similar electronic equipment available during the examination.
 Jag intygar att jag inte har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under
 eximinationen.

Solved task Behandlade uppgifter	Points per task Poäng på uppgiften	Observe: Areas with bold contour are to be completed by the teacher. Anmärkning: Rutor inom bred kontur ifylls av lärare.
1 ✓	9	
2 ✓	9	
3 ✓	5½	
4 ✓	1½	
5 ✓	5	
6 ✓	10	
7		
8		no/ =6.7
9		60/60
10		
11		
12		
13		
14		
15		
16		
17		
Bonus poäng		
Total examination points		
Summa poäng på tentamen		



* Admin and user clients are ~~separate clients hosted on different nodes~~ but both talk to same application server.

The servers and user records database are hosted on AWS "Amazon web services" which is on the cloud.

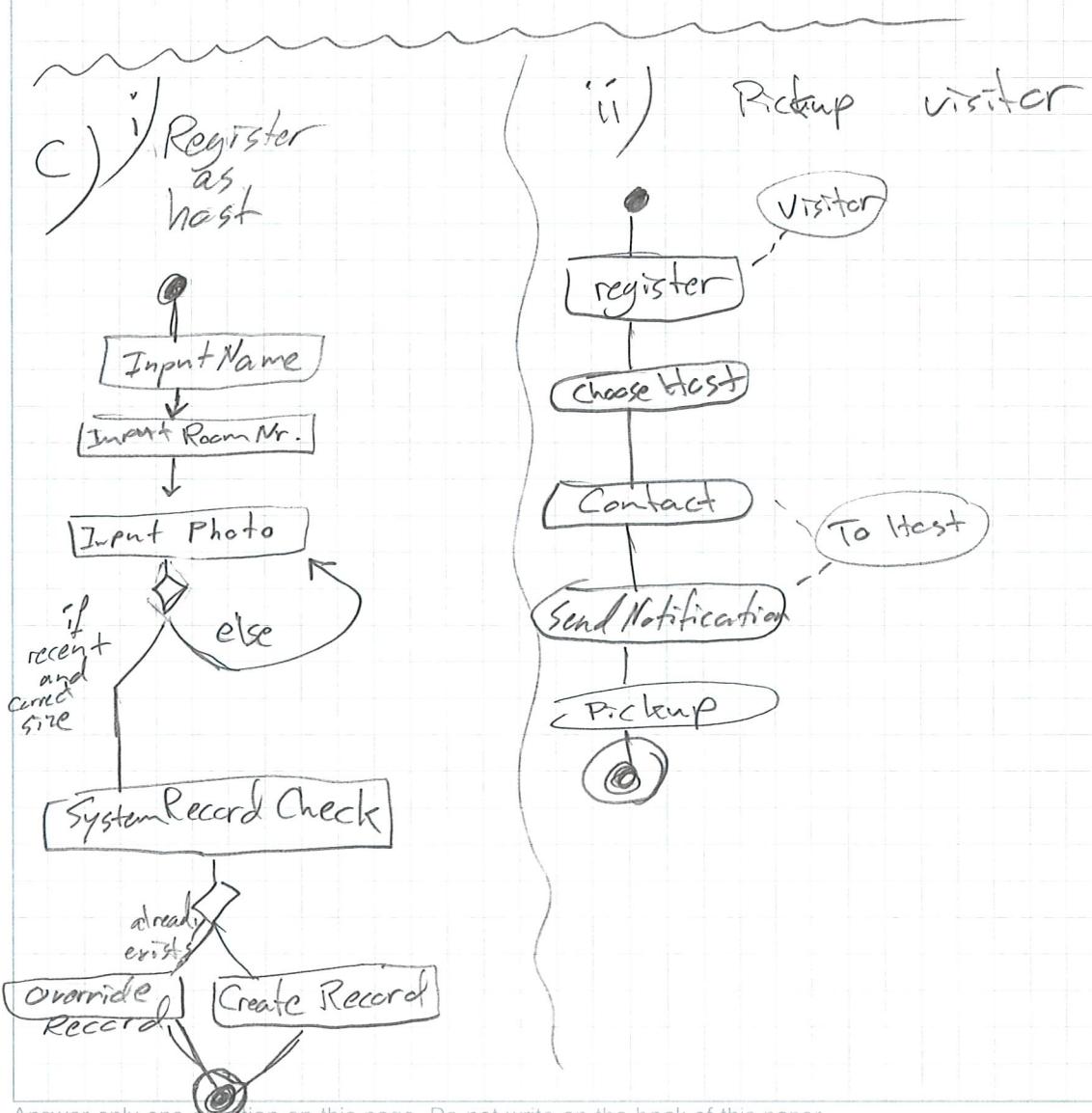
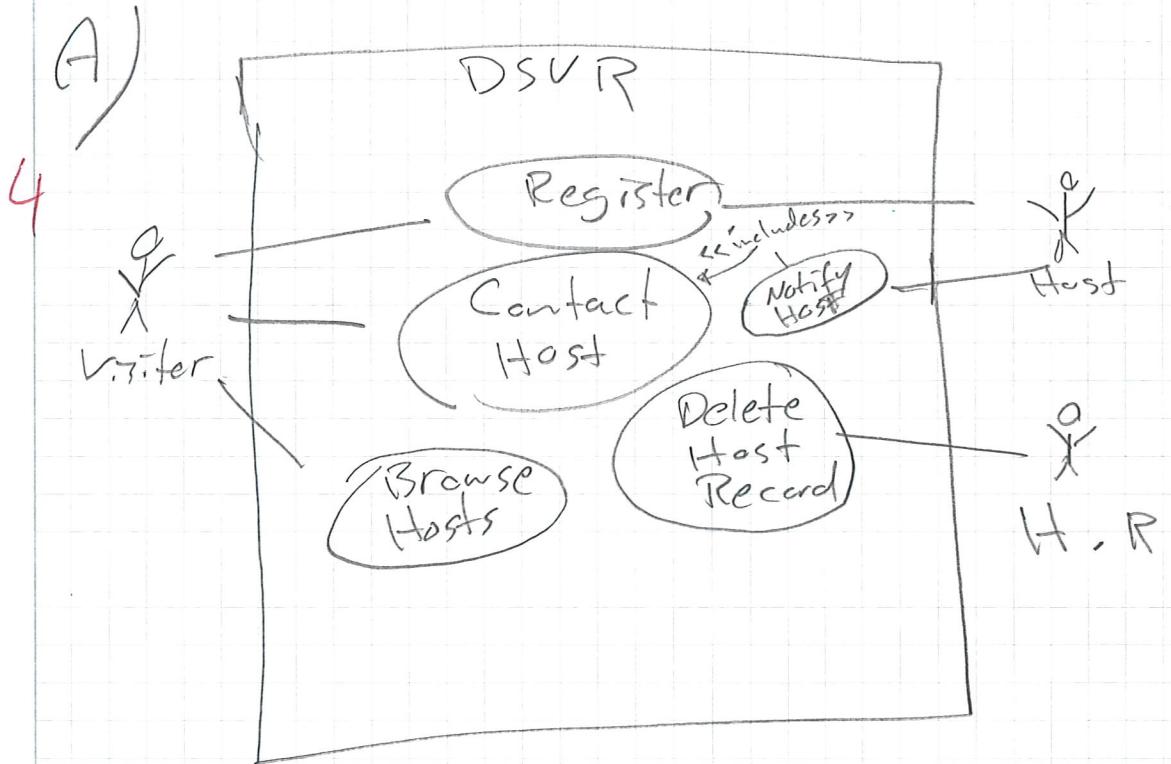


User : Represents someone using the GUI of DVPS

User Controller : Module that contains possible operations on record DB, containing conditional methods.

Record : Represents either "Host" or "Visitor" record in database.

Human Resource : External system with admin rights on DVPS database, responsibilities?



A) The process of reverse architecting is to reconstruct an architecture from an existing system ~~information about the system~~ by gathering information about it, i.e objects, then process it with specific tools to create a design from it.

B)

- * Lack of information, insufficient data
 - 3 Sometimes some documentation is missing or important artifacts are out of reach due to original developers working elsewhere.
 - * Bad use of design principles in the code could end up with incorrect architecture or tools might not interpret them correctly.

5

A) ✓ By exposing only! what other modules need from the module or interface.

B) In high cohesion we want each module to have code that relates to each other strongly, so that module serves one main purpose.

For example:

The function "printName()", should not ~~do~~ perform 3D game animations.

↳ This is similar to "one responsibility per component"

A) It's a documented proven way to successfully structure an architecture. Every style has its own characteristics that provides a template for the actual architecture. ✓

B) Its components are publisher and subscribers.

The subscribers receive periodical updates/notifications from the publisher via "events".

✓

C) Pipes and filter is appropriate here

because each image enhancement step is an input, becomes processed, then output to the next enhancement until final output to the user. ~~regular~~

29 D) Client-Server because customer data is hidden and protected on server side, also the system can be scaled up by adding server hardware.
or upgrade

- 2½
- C) * ~~(By using different views)~~ the architecture increases communication and understanding amongst stakeholders.
- * By providing an abstraction of the system through easy to read diagrams.

A) i) The business drivers often (not all) answers why the system exists (will exist) which serves as higher purpose for the architecture to help achieve ~~the~~ business goals.

requirements

ii) The most architecturally significant ~~drivers~~ helps the architect in making early design decisions, i.e "what architecture style to be chosen?". By understanding the requirements, the architect can identify such that are most architecturally significant.

B)* When changes are introduced during development

then the architecture has to be updated to stay in sync with the codebase or requirements, which can be a hinder because then it's best to wait for the architecture changes made before new/more changes to the system, this costs time.

* If the architecture takes into account "modifiability" as a quality attribute requirements then adding changes to the system can become easier implemented.

A) Quality attribute is a measurable or testable property of a system that ~~satisfies~~ stakeholders needs.

Its relevance in Software architecture is that some of them may act as architecture drivers if they are strongly influential in shaping the architecture.

B) i) By abstracting the software into diagrams it becomes easier to understand.

ii) It can help reduce cost of maintenance by foreseeing future changes, (~~through~~^{i.e.} early quality attribute requirements focusing on maintainability.)

iii) In software architecture evaluation, one method to analyse system performance can be through simulation or mathematical modelling