

CHALMERS

EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn			*
DIT 344	Fundamentals of Software Architecture			X
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
963	2019-11-01		9	VG

* 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
 examinationen.

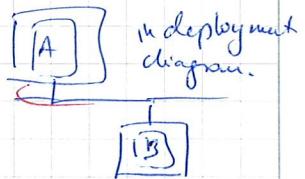
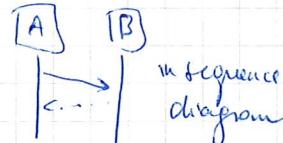
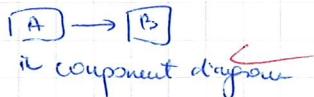
Solved task Behandlade uppgifter	Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylls av lärlare.
No/nr		
1	✓ 9	
2	✓ 6	
3	✓ 6	
4	✓ 20½	
5		86%
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
Bonus credits/ poäng		
Total examination points Summa poäng på tentamen		

- 1) a) Software Architecture is a discipline dealing with high level modelling, describing and communication about a software system. Architectural models capture a system from different views, thus facilitate communication between stakeholders and make sure that the software's high level design ^{addresses} captures the most important business needs and requirements, including risks.
- 2) b) ① In the beginning of the life cycle, designing an architecture helps ~~reaching~~ ^{achieve} ~~formulating~~ clear goals reducing risk, because by discussing the system's and capturing the system's goals and ~~other~~ likely scenarios, the system can be developed in order to mitigate these risks. Such as, development of a safety critical system should start with the choice of an appropriate architecture that resists towards failures.
- ② Throughout the development, weekly meetings are recommended so that ~~software~~ developers can make sure they follow the architecture that has been designed, even when they add new functionalities or scale up the system. In certain cases, a scaling up can mean change in the fundamental architecture, ~~but~~ but that should be very expensive and done properly to redesigning the system.

1) c) Architectural View is a view of the system from a central angle. For example, component diagram captures the components of the system and their connections. This allows us to see the interfaces to be implemented and the responsibilities of each component. There are also views capturing the behaviour of the system, the relation between hardware and software components (location and others).

2) d) Consistency is an important feature of the architectural views that ensures that the same system is represented ~~is~~ in each view. Since we want to look at a certain system (to be designed) from different ~~points~~ angles of views, aspects point of view, it is of utmost importance that the components represented in one view should be the same as those represented from another view.

For example



We want to know the structure, behaviour and other aspects of A, B, so we need to capture them in ~~each~~ one diagram. Also, if they are connected sometimes ~~we can~~ in one ~~view~~, they need to be connected ~~in~~ the other views too.

2) i) First, by evaluating architecture we can make sure that it follows and supports the most important business needs. Evaluation allows us to talk about stakeholder's ~~the~~ interests and prioritize them. We can reduce risks by doing so, and make sure that flaws are detected early on.

ii) Later, we can check if the development / implementation actually followed the architecture we aimed for. Thus, we can make sure that the documentation is up to date, or update it if necessary. It is important for later maintenance works to have ~~the~~ appropriate models.

2)

a)

Performance is a driver because the system should be able to display 30 images per seconds for all clients.

1

Portability is another driver because Netflix client should be able to run on ~~any device~~ different kind of devices.

Scalability is a third driver because we expect many users using Netflix ~~in parallel~~, and ~~the user~~ will increase ~~for the~~ with time.

Availability, because it would cause loss of money if many people ~~were not~~ able to use the service for sometime.

b)

Here I will provide scenarios for Performance, ~~and~~ Scalability and Availability because only 2 was asked for, so one more is ok.

	source of stimuli	stimuli	environment	artifact	response	response measure
Performance	user	start video streaming stream on their device	user is the 3rd who started the same movie	Netflix client on mobile	video starts	count image per second on device
Scalability	user	download Netflix on a new device and start watching a video	already several (100 000) devices run Netflix	client - server connection manager	streams on regular new device	# of devices running in parallel without loss in performance
Availability	server failure	downtime	serial 100 000 people watches videos	load manager	switches to another server	down time per server time to recovery

3) a) Publish style seems most appropriate for this scenario.

When a user is interested in a match, they subscribe for that channel (class of information) and receive it whenever there is an update. Information about each match is generated and published, even if no-one is subscribed for that channel at a specific moment.

Similarly, if there is no match of ~~that kind~~, even if a person is subscribed, they get no information. Match statistics generation therefore is independent from user's choices, the message forwarding by channel is done by a broker.

b) Blackboard style seems most appropriate for this scenario. The data is posted to the shared data store by the data collector, and any/each application can access it and also post their findings/results to the same shared data store. Each application can access any data and ante any data (that is usually result of an analysis). At the end of the day, everything will be available in the central server.

c) MVC style seems most fitting for this one. The data acquisition is done in a remote place (village or otherwise) and ~~is~~ saved to a data store, ~~that~~ is then being processed by the Model part of the ~~style~~ architecture. Once it is done, the visualisation is updated in the View part. This ^{visualisation} is most likely be ~~either~~ in the three geographical levels. The Client can decide ~~which~~ ^{which} information to be shown at a certain time point.

4)

8) ~~At~~ I used information ~~hiding~~ ^{at} the User Interface subsystem's connection to Image recognition subsystem. The UI is only aware about the dressing interface that hides the actual implementation of how to dress up a virtual image ~~(that is only)~~ ^(secret methods) which is known by the image recognitions. The secret is how to dress up the image of the person and it is hidden behind ~~and~~ interface called dressing.

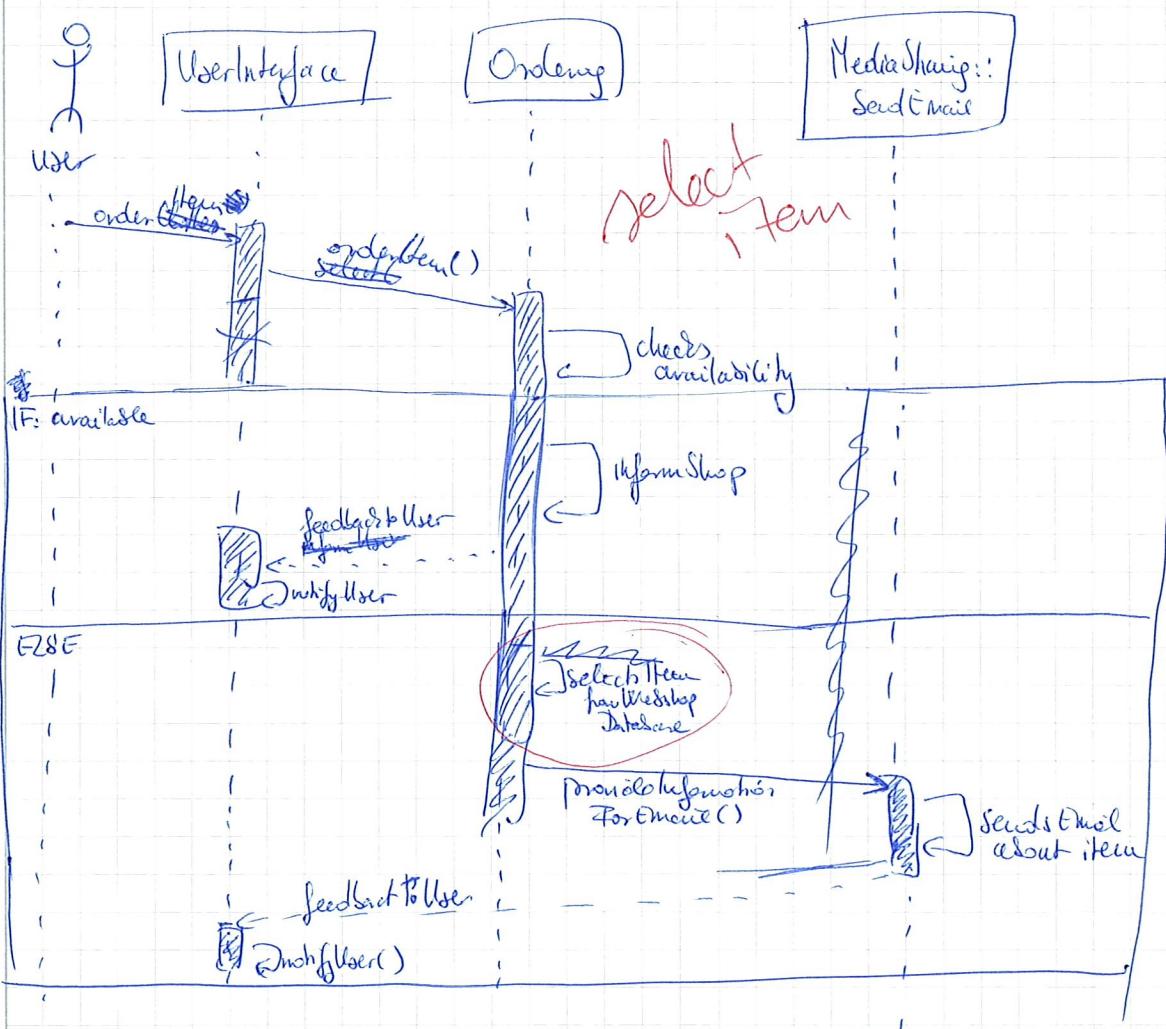
2)

8) To fulfill high level of security, that is easiest to implement with one well-protected server, the ~~avata~~ performance of the system decreased, because all requests will go through a security check and arrive to a single server. ~~It is also~~ This also creates a single point of failure. But security is a driver to the system's architecture, so other qualities can be sacrificed for it (to some extent).

~~10~~
~~10~~
~~10~~

e) ii) Order a piece of clothing

In case we are in a shop



Description:

~~When ordering, the user can be prompted by UI~~

The user can select item to be ordered via UI. UI forwards to ~~Ordering~~ the selected item. Ordering checks its availability in the external IT system, either ~~and informs the shop or selects the item from an online webshop~~ and informs the shop or ~~selects the item from an online webshop~~ and sends the details to the user's email address.

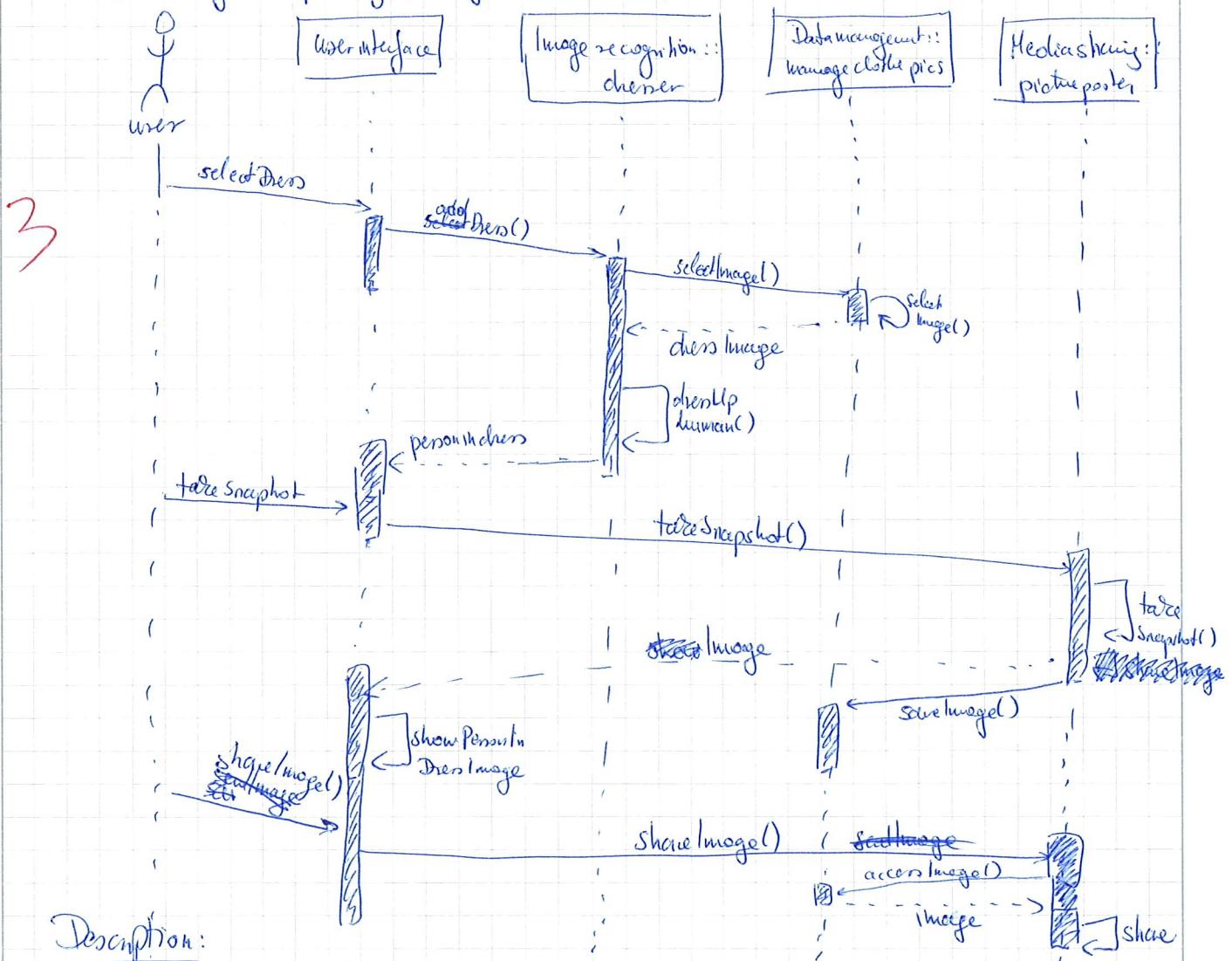
In any case, it notifies the user about the status of the order through UI.

d) description:

Data management ss. access the database (not shown) with two different components, one for clothes, the other for human data. DM ss provides an interface that other subsystems (in the application logic layer) are using. Image recognition opens up the virtual image, so it needs to be connected to UI and DM. MediaShang is also connected to UI and DM, as well as to ordering ss because all communication to external systems. MediaShang is responsible for managing emails. Ordering ss can communicate with shops in order to set an invoice for the user. MediaShang ss is accessing an interface to social media and video call and mailing systems (external).

e)

i) Try on a piece of clothing and take snapshot



Description:

User Interface manages user requests and sends to the selected dress's ID or similar to Image recognition's dress component. This component access the dress from the database via DM's special component. Once the dress is selected and rendered on the person by IR, the person can prompt the UI to take a snapshot. MediaShang is responsible for this and they're connected by UI. The user can choose how to share image, and the saved image can be shared to by MS.

(a)

Client-server at style

(b)

- Image recognition subsystem: Responsible for ~~rec~~ recognizing the shape of the user in front of the mirror and dress up accordingly.

- Data management subsystem: Deals with the selection of clothing, access image of selected item and stores user information

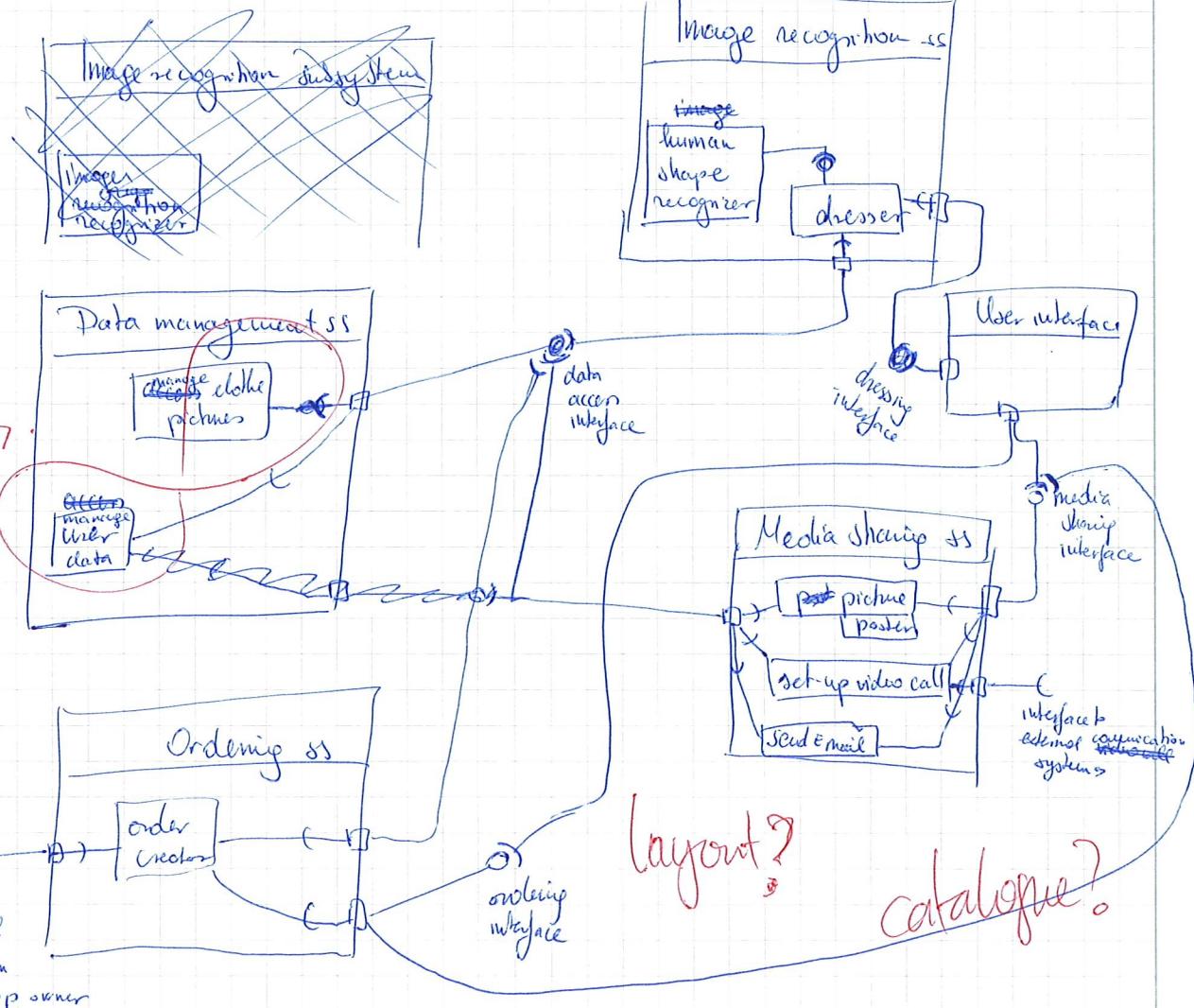
- Video calling subsystem: Set up video-call, take picture and communicate with external system. Posting picture to user's facebook or email it to them.

- Ordering subsystem: Creates order and sends relevant information to user and company.

- User interface: allows user to select, browse and communicate with the system

4

Should be
separate
components



value?

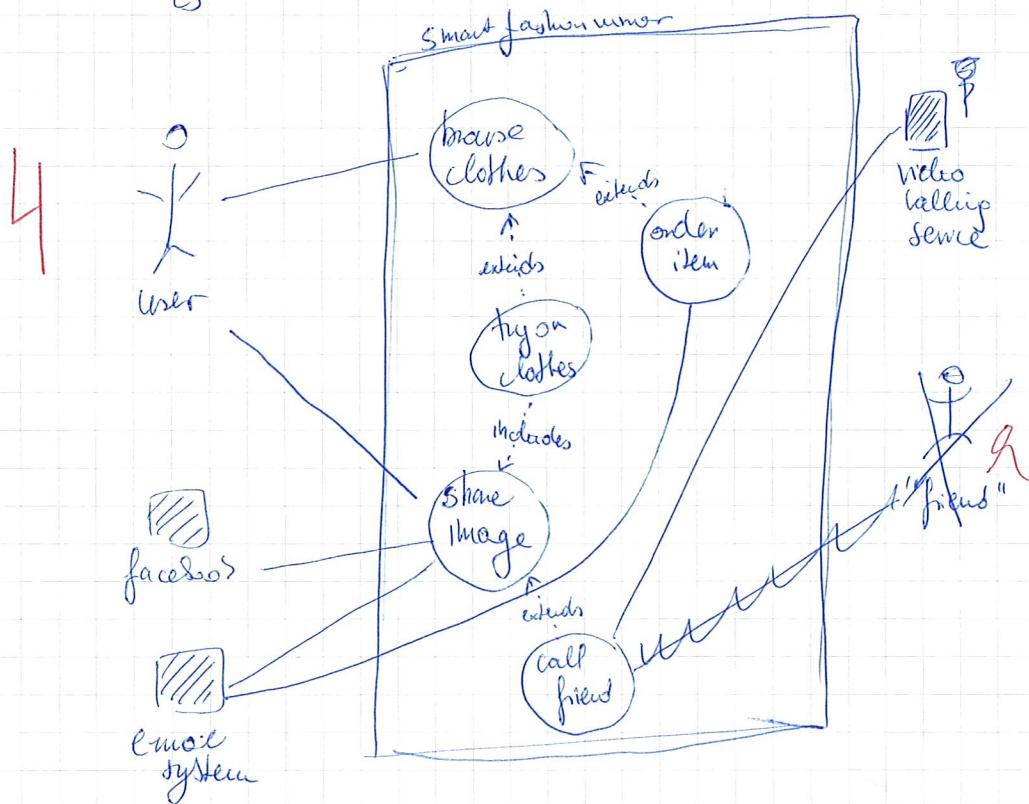
1 1/2 a) Security is very important as the system will ~~know~~ have a representation of the human in front of the mirror so that it can dress it up. Also, the system connects to user's facebook account. Additionally, it will store more authors for the user and ~~the~~ the user's connections.

Availability of the database with clothing items, so the users can browse seamlessly among these.

Therefore I chose Client-Server style for my architecture

2 2 b) i) In accordance with GDPR, the stored data should be anonymised, as images of humans are inherently un-anonymisable, these should be deleted after the process is ended.
ii) Communication, e.g. email, through ~~any~~ any channel should be encrypted and ~~secure~~ secured with encryption.

c)



As mentioned, user should be able to perform the tasks shown. Additionally, external systems such as facebook, email system and video calling system should be accessible for some of the tasks. This is because stakeholders