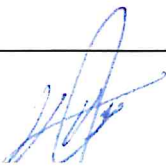


CHALMERS

EXAMINATION / TENTAMEN

Course code/kurskod		Course name/kursnamn		
DIT344		Fundamentals of Software Architecture		
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad	Grade Betyg
575		2022-01-05		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 ifylles av lärare.
No/nr		
1 ✓	15	
2 ✓	18	
3 ✓	14	
4 ✓	14	Total 80
5 ✓	19	
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
Bonus: poäng		
Total examination points Summa poäng på tentamen	80	

EDA360/DIT344 Fundamentals of Software Architecture

Re-Sit Exam January 5, 2022

Wednesday, January 5, 2022

Examiner Eric Knauss 0317721080

Contact person during exam Sam Jobara 0766084815

General information

This is a closed book and notes exam. You are not allowed to have anyone helping you with answers. If you plagiarize or have very close to the same answer, on the free-text/essay type questions or parts of questions, this grounds for failing the exam.

The contact person during the exam is Sam Jobara will be available during the entire exam period, feel free to call if you have a specific question that relates to exam understanding.

The exam has 5 questions, 3-essay, and 2-multiple choices. Giving 4-hours is more than what you need for this exam.

Please, make note that your answers to the essay questions should not exceed the words limit stated per question, and the multiple choice questions should be answered directly on the last column of the table labeled answers.

Each question shows how many points it has for each section. I strongly advise you to answer all question, there is no penalty for answering wrong, by answering all question you may get at least some partial points.

You may not need extra papers to answer this exam. I have provided sufficient space for your answers after each question.

With regard to exam grading, we follow the universities standard grade assignment:

For Chalmers students:

0	<=	points	<	50	=	Fail (U)
50	<=	points	<	65	=	3
65	<=	points	<	80	=	4
80	<=	points	<=	100	=	5

For GU students:

0	<=	points	<	50	=	U (Fail)
50	<=	points	<	80	=	G (Pass)
80	<=	points	<=	100	=	VG

15

Question-1 22p

General question about Software architecture (answer maximum 200 words)

(a) Explain how a software architecture can help in improving communication between stakeholders. 4p

Software architecture requires stakeholders to agree on the different qualities of a system, as well as the Architectural Significant Requirements. Thus, it creates the need and room for stakeholders to define these in relation to the architectural decisions of the system.

Archit. Views suiting each stakeholder role

(b) Explain how software architecture can help in managing risks and liability in developing a system. 6p

Using the Architectural TradeOff Analysis Method, architectural decisions are analysed and risks are identified regarding the requirements and qualities fulfillment. It can then be part of a Risk management plan with categorisation of risk and plan to define a strategy. cost, schedule, security etc

(c) Explain how modularity of software architecture improves the evolvability and variability of a software system. 6p

Having a modular architecture implies that single responsibility and separation of concern of components are respected. This leads to low coupling between modules which makes it easy to add a new module, change a module to a system, improving the factor its evolvability and variability.

(d) Explain how software architecture design can be compatible with Agile development process. 6p

Software architecture decision should be incremental and focus on the most important requirements (ASR). Not too many decision should be made upfront in order to allow adaptability to changes.

Software architecture decision should be assessed using ATAM and Quality Attribute Scenarios and discussed iteratively with stakeholders. A tight communication between development team and architect following DevOps principles described previously.

Question-2 18p

(9 multiple choices, total 18 pts) *Architectural styles and design tactics*

14

*In these question at least one statement is correct.**If both statements 1&2 are correct, then answer A**If only Statement-1 is correct and Statement-2 is wrong, then answer B**If only Statement-1 is wrong and Statement-2 is correct, then answer C**(Please note that a partially correct statement is still considered incorrect one)*

Qs	Statement-1	Statement-2	Answer
1	There are two types of architecture styles: Monolithic (like Event-driven architecture and Pipeline architecture), and Distributed (like Microservices architecture, and Layered architecture)	A Monolithic architecture has single deployment unit of all code and distributed architecture has multiple deployment units connected through access protocols	C
2	In Blockchains, Smart contracts act as a complement, or substitute, for legal business agreements. They are self-executing computer code that directly control some aspects of condition-based transactions.	A smart contract also capable of automatically executing and enforcing an agreement. Smart contracts are immutable (unchangeable record) and are enforced by the system itself.	A
3	In Microservices architecture style, the sidecar component can handle all the operational concerns across microservices to benefit from coupling together. Thus, when upgrading the monitoring tool, only the standard sidecar can be upgraded, and each microservice receives that new functionality.	In Microservices architecture style, the common sidecar components connect to form a consistent operational interface across all microservices, while preserving a degree of decoupling for each microservice.	A
4	Monolithic architecture styles have an advantage over Distributed architecture styles related to the communication network speed limitation across processing modules.	It is common for a Distributed architecture to measure network latency in microsecond, while for Monolithic architecture it may be measure by milliseconds.	B
5	The advantages of Event-Driven Architecture Style-Broker topology, that it offers highly coupled event processors, high performance and great recoverability, due to its unique error handling ability.	The mediator topology of event-driven architecture addresses some of the shortcomings of the broker topology. The simple mediator can handle the event itself or forward it to another, more complex, event mediator.	C
6	Publish-Subscribe architecture can promote decoupling, scalability, extendibility, and improved security. However, it has some drawbacks, such as the need to guarantee delivery, and performance problems with messages.	You would use Publish-Subscribe architecture when data is not long-lived, multiple consumers are interested in updates, and it helps to dynamically change topology of producers and/or consumers	B X
7	In Peer-to-Peer Mediated architecture a client-server setup is used to control operations. Only one peer at a time can logon to a central server that manages the file and user databases.	In Peer-to-Peer Mediated architecture a client-server setup is used to control operations. Any peer at a time can logon to a central server that manages the file and user databases.	C
8	Blackboard Style is recommended if representation & management of data is a central issue and data is long-lived. However, it is not recommended if order of computation – can't be determined a-priori or highly irregular	It is easy in Blackboard Style to add components, execute in parallel fashion, easily change functionality of components. But it offers poor security because all processes share the same data.	A X
9	In Client-Server architecture style, thin client offers low load on network, simple server, and security focus is kept at the server, this design is more suitable for web-applications	In Client-Server architecture style, thick client makes significant processing at client side, where all of the application logic is implemented at the server side.	B

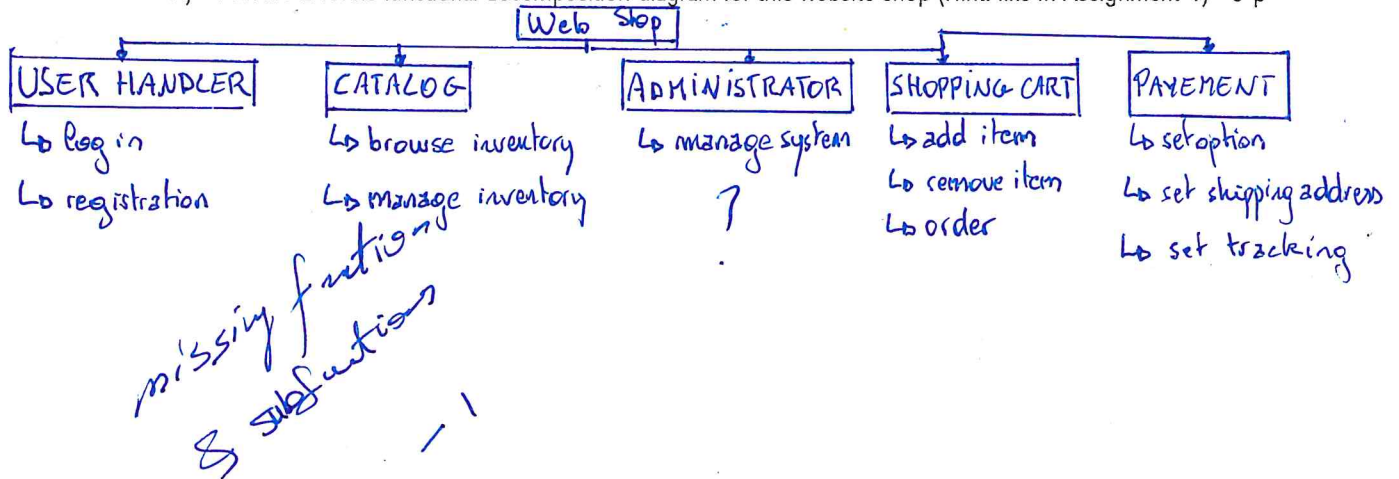
Question 3 22-points (answer in max. 350 words)

18

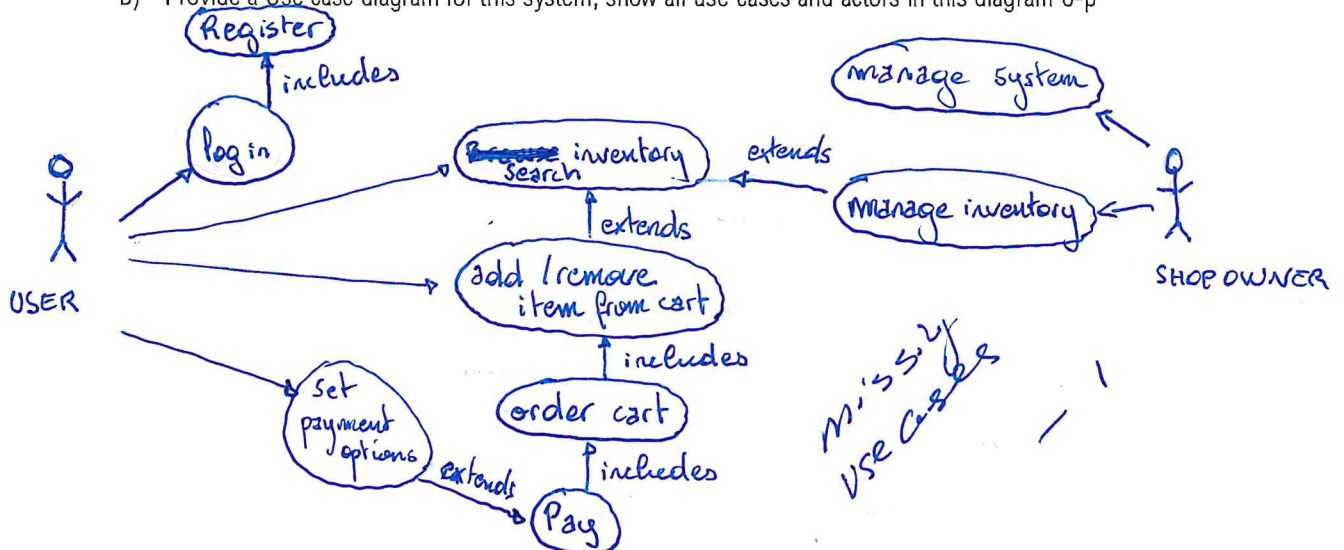
You are to design a commerce website for shopping with the following features:

Catalog of items with inventory management and search capability, customer registration and login security, Administrator management, adding and removing items at a shopping cart, Order with payment options, Shipping and tracking as defined by customer.

A) Provide 2-levels functional decomposition diagram for this website shop (Hint: like in Assignment-1) 6-p



B) Provide a Use case diagram for this system, show all use cases and actors in this diagram 6-p



C) What is the most suitable style or mix of styles would you recommend for this system architecture, explain the motives and the architecture drivers that made you select such style(s). 5-p

For this case, a client server based architecture in order to access database quickly as well as having a possibility to have different client or account type accessing the same resources. We can also consider some service based architecture, to deal with the payment for instance which could be outsourced. The client Service based architecture would allow the system to be more fault tolerant whereas the client server style would make the core of the system cheap and easy to maintain.

not good choice

D) What design tactics or decisions would you suggest to improve performance, and security of the system 5p

I would probably have a distributed database system to increase the performance having the load distributed based on a specific book and not on server load. This would allow to adjust server capacity as well as lower the charge on database.

I would also make authentication separated from load balancer and make it required in order to access books.

Question 4 18p

In these question at least one statement is correct.

If both statements 1&2 are correct, then answer A

If only Statement-1 is correct and Statement-2 is incorrect, then answer B

If only Statement-2 is correct and Statement-1 is incorrect, then answer C

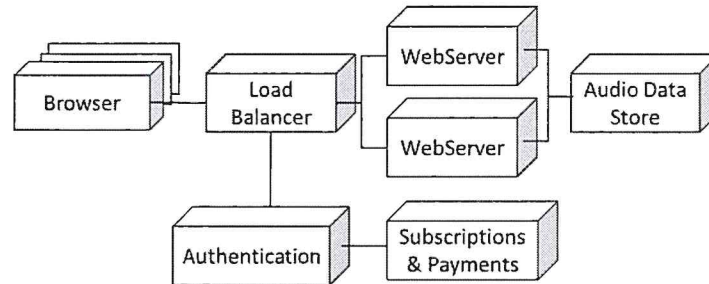
(Please note that a partially correct statement is still considered incorrect one)

14

Qs	Statement-1	Statement-2	Answer
1	Information Hiding is about keeping all information about a module private to the module, to minimize coupling, a programmer will be able to use a class without need to know how methods are implemented.	Coherence is about Keeping things together that belong together, by grouping code that contributes to a single task within a single component.	A
2	Separation of Concerns advantages: more duplication that helps maintainability. The increase in maintainability and extensibility can also slow down the adoption rate of the system.	A decoupled component has a single set of cohesive responsibilities often result in more extensibility points. The decoupling with a single purpose leads to components which are more reusable.	C
3	Separation of Concerns emphasizes separating a program into distinct sections, such that each section addresses a separate concern. A concern is a set of information that affects the code of the program.	Divide and Conquer principle is the same as Separation of Concerns, in the since that it divides each concern to allow faster and easier design approach of smaller concerns.	C X
4	In general, the goals of software modularity encompass the goals behind both encapsulation and information hiding practices.	Modularity allows you to swap out software classes with others. However, modularity decoupling would be a challenge due to code elaborate changes.	B
5	There is coupling between two classes A and B if: B has a method which references a third-class C, and A calls a service of an object in C.	There is coupling between two classes A and B if: B is subclass of A, and A has an attribute that refers to B.	C
6	An example of divide and conquer, the Quicksort algorithm recursively divides a list of values to be sorted into two smaller lists, in which one sub list has the smaller values and the other the bigger values.	In divide-and-conquer you define a method to decompose the complex problem into smaller parts. However, you need later to combine the solutions of the parts to obtain the solution of the original problem	A
7	Design goals for Layering is to divide system into layers of abstraction, modularity, portability, and non-overlapping of Concerns.	Layering provides a cohesive set of services at an abstraction level: layer n-1 is allowed to use services of layer n (not vice versa) alternative: bridging layers: layer n may use layers <n	B
8	Details of the data representation and algorithms should be hidden from users at the interface level. This is achieved when the interface of a component exposes what it does, but not how it does this.	It is a good design practice to separate issues that are not related and handle them in separate parts, and to assign a single responsibility to a single component or class.	A
9	A benefit of Low Coupling/Dependencies is that more interconnections between modules reduce the chance that a fault in one module will cause a failure in other modules, which enhances robustness	With low Coupling/Dependencies, the modules are easier to replace, reduce the chance that changes in one module cause problems in other modules, which enhances reusability, and it also improves testability.	A X

Question 5 20p

Consider a Web System for streaming audiobooks online. The software architect has designed a deployment structure of the system as shown below:



(a) Give three specific example architectural drivers for the audiobook streaming system. 6p

- 1) A solid authentication process / payment process.
 - 2) A good quality of service using the load balancer
 - 3) System is easy to maintain and fault tolerant.
- high availability & performance

(b) Provide at least 4 quality attribute that are most critical for this architecture. 6p

- Efficiency with the bandwidth and resource consumption for audio streaming
 - Security with the authentication and payment process.
 - ~~Reliability~~ Availability with the service being provided at any time, no matter the traffic. (explains the use of load balancer)
 - ~~Usability: system should be~~
 - Extensibility: system should be extended easily if user number increase (adding a web server for instance)
- (c) Provide two most suitable architecture styles for this solution, explain briefly why you would choose them. 8p

The first architecture style would be ~~service based~~ / micro services style. It allows the system to share responsibilities and ensure a quality of service since based on what service is required (listening, subscribe, pay) a different part of the system would be used. It also increases the maintainability and scalability since it is easy to add a service / change a component without impacting others.

The second style could be an ~~publish subscribe~~ event driven architecture with the load balancer acting as an event manager and other component as event handler. It follows the same logic as the micro service style.