Sektion 1

1 a) Which of the following statements are true? There are exactly two correct answers, wrong answers give negative score. (2)

□ A key difference between a *functional* and *non-functional requirement* is that the latter cannot typically be attributed to a single component, but to the system as a whole.

□ *User requirements* only include functional requirements but exclude non-functional requirements.

□ *Workflow models* used in requirements engineering describe how the system operates in a business context, e.g. within the business processes at a company.

□ *Interviews* during the requirement elicitation process help to formally capture the true needs of the customer, hence eliminating all potential fuzziness in requirements.

1 b) *Scenario*: A web-based library management system (LMS) allows library members to take books on loan. A member can register for the library by providing his/her address and personal number. Then (s)he can search for a book with a given title or from designated authors. The library keeps a limited number of copies for each book. The librarians may order new books by the end of each month. If a copy of the designated book is available for a target period, then it can be taken for loan by a member, and the system records the loan date and return date. One week before the return date, the system sends an automated reminder to the member. If the book is returned by the member after the return date, then the librarian sets a late-return penalty.

Task: Create a *UML Use Case diagram* for the LMS with key actors and use cases. *Give appropriate names for these actors and use cases*, but *no detailed descriptions are needed*. (Logging in and logging out are basic functions, not to be considered as individual use-cases.)
Append a UML Use Case diagram and write in the text which appendix that contains the answer to this question. (4)

B *I* U | ☰ ☰ á

0 / 10000 Word Limit

1 c) *Scenario*: Same as in Problem 1b) above

Task: Write *two user stories* of the system described above. Write *one non-functional requirement* for the system together with a *short recommendation* (1-2 sentences) on which level of testing can it be verified. (4)

B *I* U | ☰ ☰ á

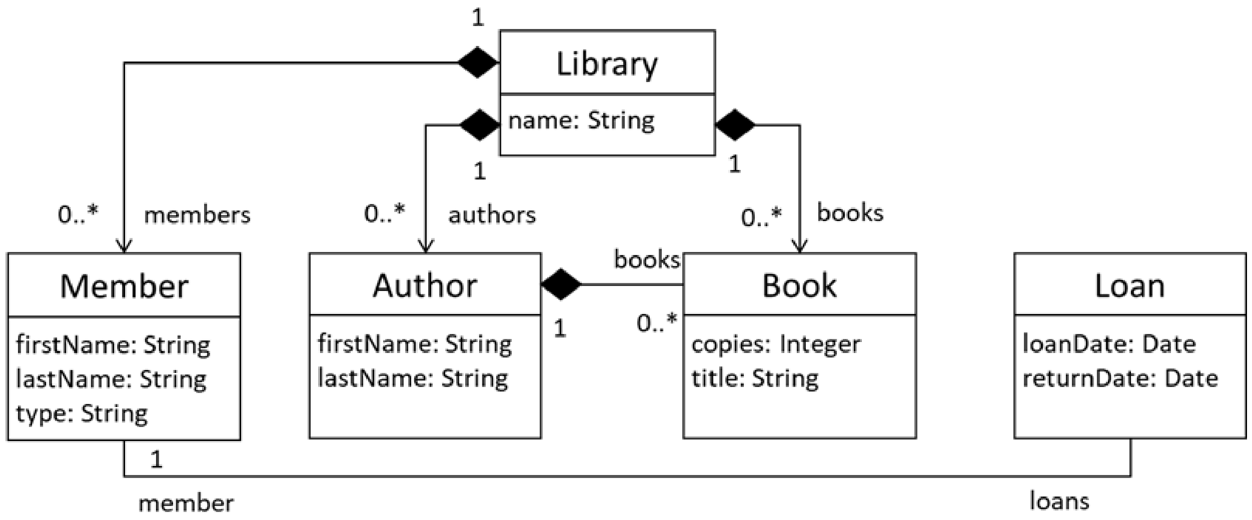
0 / 10000 Word Limit

Sektion 2

2 a) Which of the following statements are true? There are exactly two correct answers, wrong answers give negative score. (2)

- □ The system architecture contains all decisions on which functional component needs to be deployed on what platform element.
- □ In a real software system, several architectural styles are often combined
(e.g. model-view-controller architecture combined with a layered architecture).
- □ The performance of a system can be scaled up by creating a large number of subsystems with redundant functionality.
- □ By introducing a unified interface, the Façade design pattern promotes high coupling between subsystems.

2 b) Carry out a thorough design review of the following domain model captured by a UML Class Diagram (5)
Write down a list of (at most five) recommendations how the design should be improved in constructive way as you would propose as a reviewer as part of a design review.



B *I* U

0 / 10000 Word Limit

2 c) Which *non-functional system characteristics* can be enhanced by redundancy and diversity? Give *two examples* of a software system where the respective non-functional characteristic is important. (3)

B *I* U á

0 / 10000 Word Limit

Sektion 3

3 a) Which of the following statements are true? There are exactly two correct answers, wrong answers give negative score. (2)

- □ White box testing can only reveal extra functionality, but not missing functionality.
- □ Tests for non-functional requirements are typically carried out as part of unit testing.
- □ The role of a stub is to call the function under test and pass test data to it.
- □ A test suite that ensures full branch coverage also ensures full statement coverage.

3 b) *Scenario*: A company is paying bonuses to their employees if they work more than a year and the company made a profit in the given year. Moreover, an employee (with at least one year of employment) receives individual bonus if he or she worked in a project that was successfully completed.
Task: Create a test table to test a business functionality that controls the paying of (company and individual) bonuses. (5)

B *I* U

0 / 10000 Word Limit

3 c) What is the difference between *Continuous Integration* and *Continuous Delivery*? Explain it in 1-2 sentences.
Name *two expected benefits* and *two main challenges* for each of them. (3)

B *I* U

0 / 10000 Word Limit

Sektion 4

4 a) Which of the following statements are true? There are exactly two correct answers, wrong answers give negative score.(2)

- □ The classical *Waterfall life-cycle model* can be a good alternative for fixed-priced contracts.
- □ An *agile method* constantly adapts the requirements to fit the system being developed.
- □ The definition of an *iterative life-cycle model* is that components are developed and integrated with a fully working sub-system.
- □ A potential *drawback of an iterative life-cycle model* is that extra administrative overhead for planning and coordination is added.

4 b) Describe the following properties of Kanban in 1-2 sentences: *Kanban board*, *lead time*, and *work in progress*. Give one expected advantage of using Kanban. (4)

0 / 10000 Word Limit

4 c) Suppose you are developing software in time-boxed iterations. Name each of the *four dependent project parameters* and describe if it is locked or free to change. Give *two examples* of what you can do if one of the (free) parameters changes. (4)

0 / 10000 Word Limit

Sektion 5

5 a) Which of the following statements are true? There are exactly two correct answers, wrong answers give negative score. (2)

□ In an inspection, it is good if the inspectors have very similar competence to give a statistical ground for process improvement.

□ An audit is a software review with no written documents; the participants discuss about the ideas of the software

□ You normally record data about the inspection itself in order to improve the expensive inspection process.

□ Both the author and the inspection leader take part in the exit-and-follow-up phase of the inspection process

5 b) Suggest *two different software metrics* for measuring the *maintainability* of software. Don't forget to briefly motivate (in 2-3 sentences) why the metrics can give information of the maintainability. (4)

B *I* U á

0 / 10000 Word Limit

5 c) Scenario: Life isn't easy: Your end-users complain that even though your software is failure-free and has a good looking GUI, you miss certain features (functions) that "everyone" in their company know are sometimes needed. Your employees are really appreciating your personal feedback, but it comes sporadically and is based on your feelings only. As a complement, they would like to have more regular and objective feedback.

Task: Describe a *CMMI process area* that you think will help you the most. A description is typically 5-6 sentences that covers *Introductory notes* and/or *Specific goals*. Also, describe how this will help your company. (4)

B *I* U á

0 / 10000 Word Limit

Sektion 6

6 a) *Scenario*: A graduate student application management system (GRADS) helps collect and review applications from prospective graduate students from all around the world to graduate programs offered on different levels (MSc vs. PhD), in different curricula (e.g., Computer Science, Software Engineering) and at a given starting time (e.g. Fall 2023). Prospective graduate students (MSc or PhD students) create a personal profile (with their name and citizenship) and then upload their application, which must contain (a) their language exam score and (b) their official transcripts including their aggregated score. When submitting their application, students specify their preferred supervisors. GRADS checks if all minimum criteria of graduate admission are fulfilled and sends automated emails to prospective students to complete missing information. Professors then rank applications of those students in GRADS who meet the minimum criteria and who selected them as a preferred supervisor by assigning a numeric score. A student may be admitted to the graduate program if he or she is selected by at least one professor, otherwise his/her application is rejected.

Task: Draw a *domain model in the form of a UML Class diagram* for the GRADS system showing the *domain classes* and their relationships as well as potential *generalizations*. Specify *multiplicities* for your *associations and compositions*, and give them *meaningful names*. Provide a *total of 5 key attributes* mentioned in the description above *together with their type*. Append the UML Class diagram and write in the text which appendix that contains the answer to this question. Give a *brief textual justification* (1-2 sentence) of your *key design decisions*. (10)

0 / 10000 Word Limit

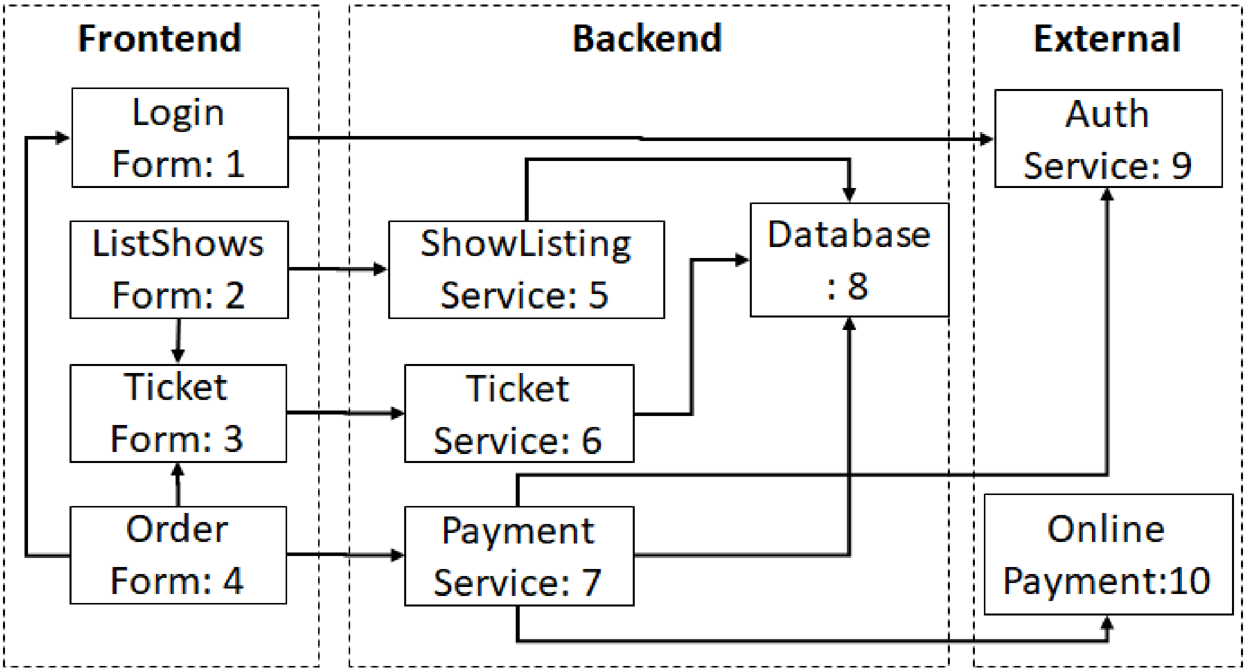
6 b) *Scenario*: as above in Problem 6 a)

Task: Describe the state-based behavior of the *GraduateApplication* class by a *UML Statemachine* diagram. (The *GraduateApplication* class of the GRADS system represents the application created or submitted by a graduate student.)
List all the *triggering events* that need to be handled by the statemachine. Append the *UML Statemachine* diagram and write in the text which appendix that contains the answer to this question. (10)

0 / 10000 Word Limit

Sektion 7

7. Consider the following block diagram representing the architecture and dependencies of a multi-tier software system (where an arrow from component A to component B means that A calls / uses B).



Task: Assuming that unit testing has been successfully completed, propose and briefly explain an *integration strategy* which (a) minimizes the number of stubs and (b) postpones integration of external services to a later stage of integration (10)

For each integration stage, specify clearly *what components are tested together at the given stage*, and *what stubs are needed.* Moreover, define what is the *integration order* between the stages! (Each component is uniquely identified by a number next to it, feel free to use those numbers as abbreviations to help describe your strategy in a more compact way. For example, **Stage #1: {1, 6, 10}** means that in Stage #1, you test components 1, 6 and 10 together)

0 / 10000 Word Limit

Sektion 8

8. In the list below you find the twelve principles of agile software development from the Agile manifesto. Select five of them and write down how they can be realized in SCRUM or eXtreme Programming (XP). For SCRUM, you define which artifacts, roles, or meetings that are involved. For XP, you mention the practice (rule) that you are referring to. Your motivations shall be thorough, 5-10 sentences per agile principle. (10)

The twelve principles behind agile software development from the Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly

B I U á

0 / 10000 Word Limit

Sektion 9

9. *Scenario:* Suppose that you've got the job to develop a software system for a computerized examination room for Linköping University. The examinations are taken by students in a specially equipped room as soon as they feel ready. Biological sensors determine the identity of the students, and electronic sensors aim to ensure that no communication equipment is brought into the room. Most exam exercises are graded automatically. There is grading support for essay questions using topic and phrase extraction from answers. The requirements on security and reliability are high.

You and your four team members start the development project on 8 January 2024 and start alpha testing on 26 August 2024. You need to use formal procurement procedures to buy the hardware. This includes publishing specifications, waiting three weeks for bids, evaluating bids formally, publishing the evaluation result, providing an appeal period, and writing a contract with the chosen vendor.

Since this project is supported by EU funding, you get payed in EUR and need to outsource some of the development to another EU country.

Task: Make a list of *five relevant risks* that need to be monitored. Use your risks to demonstrate how you can calculate the *risk magnitude indicator*. Moreover, make a *plan for each identified risk*. Select your risks so that you can *give examples of four different types of risk planning*: Risk avoidance, risk transfer, risk mitigation, and contingency plan definition.

Hint: You don't have to give examples of all types of risk planning for each risk. It is sufficient if all types of risk planning occur in your entire solution. So Risk #1 demonstrates risk avoidance, Risk #2 highlights risk transfer, etc. (10)

0 / 10000 Word Limit

Sektion 10

Don't write anything. The administrator will fill in your registered bonus credits for autumn 2023 here.

0 / 10000 Word Limit