

Introduction

If you feel that any of the problems require information that you do not find in the text, then you should

- Document the necessary assumptions
- Explain why you need them

You need to put all your answers to the following problems 1 to 4 in a .pdf file and upload the .pdf file to Inspera before the examination time expires. Your answers to problems 1 to 4 should be brief and to the point.

Problem 1 – Use case (5 points)

Suppose you will develop an online pizza order system. Based **on the use case template taught in this course, write a use case** "order pizza online."

Ordering pizza online means that:

- The user can search pizza types online
- The user can choose the type and the amount of the pizza to order
- The user can specify the delivery address and time
- The user can provide payment card information
- The system can communicate with the bank to check the correctness of the payment card information and charge the card
- The system can issue a receipt to the users
- The system can inform the delivering company of the user's order so that the delivering company can come to pick up the pizza for delivery

Problem 2 – Goal-oriented requirement engineering (5 points)

Based on this course's goal-oriented requirement engineering lecture, the task is to make a goal model of the "autonomous bus" control system using soft goal types. The goal model should have at least three layers.

(Note: You can use whatever software to draw the goal model. Handwriting drawing is not acceptable).

Problem 3 – Software testing (10 points)

Suppose you will lead the system test of an e-voting system. Your task is to explain to your top-level managers the purpose of each type of the following system test and give them at least one test case example.

- Scalability test
- Stress test
- Load and stability test
- User acceptance test
- Operational acceptance test

Problem 4 – Code review and testing (10 points)

Use code smells your group identified and refactored in the exercises as examples to explain code refactoring.

- You need to give at least four code smells your group identified through code review as examples. Code smells of different categories are preferred.
- List the code of the four code smells before and after refactoring and explain the purpose of each refactoring.

**DO NOT FORGET TO ANSWER THE CLOSED ENDED
QUESTIONS (20 POINTS)**