Name: _____ **Student#:** _____ **Lec. sec.:** _____

Date: _____

COMP 3111: Software Engineering

Lecture 10 Exercises: Requirements

Exercise 1: When developing software, it might be tempting to include additional features because you think they are ingenious and/or possibly useful to someone now or in the future.

Do you think this is good idea? ☐ Yes ☐ No Explain your answer.

Exercise 2: What, if anything, is wrong with the basic flow of the following use case?

Use-case:	Withdraw Cash from ATM
Actor:	Bank Customer
Preconditions:	Bank customer has a checking or savings account with the bank
Postconditions:	Bank customer has cash or the reason the withdraw request failed
Basic Flow:	1. The bank customer (BC) inserts his/her bank card into the ATM machine and enters his/her PIN number. 2. The BC requests a withdraw transaction from checking and enters the amount. 3. The BC indicates he/she is finished and takes the cash and bank card from the machine.

Exercise 3: Is the following a functional or non-functional requirement?

The system shall be responsive. No query should last longer than 3 seconds without some type of feedback or progress indication sent back to the user.

Exercise 4: There can be defects in requirements the same way there can be defects in code. What is the defect in the following requirements?

The system will inform the student whether he or she passed an assessment. The system shall take two inputs: score and possible points. The system shall report an error if either input is negative or if score is > (possible points). The system will return true if score / (possible points) is ≥ 0.5; else it returns false.

Exercise 1: When developing software, it might be tempting to include additional features because you think they are ingenious and/or possibly useful to someone now or in the future. Do you think this is good idea
☐ Yes
X No Explain your answer.

When developing software it is important to stick to exactly what the requirements of the software are. Adding additional features will in most cases conflict with the current features and will require a lot of effort and time. Therefore it is important to stick to what the project requirements are. If the additional features are really useful it could be good to rethink the requirements and understand why it needs to be added.

Exercise 2: What, if anything, is wrong with the basic flow of the following use case?

The basic flow is missing scenarios. This means that the user inputs something to the system and then the system will respond. In this case, the flow only shows one input from the user perspective and two from the system side.

Exercise 3: Is the following a functional or non-functional requirement? The system shall be responsive. No query should last longer than 3 seconds without some type of feedback or progress indication sent back to the user.

"*system shall be responsive*": Non functional requirement
"*no query should last longer than 3 seconds without some type of feedback or progress indication sent back to the user*": Functional requirement

If both statements would be together it would be a functional requirement because this is something that needs to be developed.

Exercise 4: There can be defects in requirements the same way there can be defects in code. What is the defect in the following requirements? The system will inform the student whether he or she passed an assessment. The system shall take two inputs: score and possible points. The system shall report an error if either input is negative or if score is > (possible points). The system will return true if score / (possible points) is ≥ 0.5; else it returns false.

What should happen if the score and the possible points equal 0? In this case, it is not specified. These are exceptional cases which needs to be specified what the program should react for these cases.