1 Theory I

For each of the statements below, please mark whether it is true or false:
(+1 for correct answer, no change for wrong answer)

An object is an instance of a certain class.

○ False

○ True ✓

A class diagram show all objects that can be created by every class.

○ True

○ False ✓

A use case diagram show how to use a certain class.

○ True

○ False ✓

A class diagram show all attributes of every class, but not the values of any attribute.

○ True ✓

○ False

A class diagram show how classes and objects collaborate.

○ True

○ False ✓

Interaction diagrams show the methods that objects call on other objects.

○ True ✓

○ False

You make one interaction diagram for each system event.

○ False

○ True ✓

Design patterns describe how you solve common interactions with the users of the system in the use cases.

○ True

○ False

Maximum marks: 8

2 Theory II

For each of the statements below, please mark whether it is true or false:
(+1 for correct answer, no change for wrong answer)

What you learn when you create interaction diagrams is used to create class diagrams.

○ True ✓

○ False

You are not allowed to start designing a class diagram until you have made a proper domain model.

○ False ✓

○ True

A package contains classes and other packages.

○ False

○ True ✓

A package must be independent and is not allowed to use other packages.

○ True

○ False ✓

Information from use cases may be used for the domain model.

○ True ✓

○ False

A system is not ready for delivery unless all use cases are fully implemented.

○ False ✓

○ True

A system sequence diagram show how one use case is connected to the next use case.

○ False ✓

○ True

Maximum marks: 7

3 GRASP Patterns

For each of the statements below, please mark whether it is true or false:
(+1 for correct answer, no change for wrong answer)

A Class can be both a Creator and a Controller.

○ False

○ True ✓

There can only be one single information expert in a system.

○ True

○ False ✓

High cohesion implies that every class should have as few and well defined responsibilities as possible.

○ False

○ True ✓

Low coupling implies that you should strive to have as few and "loose" associations as possible between classes in a system.

○ True ✓

○ False

A creator is a class to create random numbers.

○ False ✓

○ True

Controllers can call other controllers.

○ True ✓

○ False

According to Pure Fabrication, classes that are part of the pattern Abstract Factory are not allowed to do anything else.

○ True

○ False ✓

Polymorphism means that you have several classes that implement the same method, but in different ways.

○ False

○ True ✓

You first pick a GRASP pattern that you will then use as a governing principle when creating the rest of the class diagram.

○ True

○ False ✓

Maximum marks: 9

4 Design Patterns

For each of the statements below, please mark whether it is true or false:
(+1 for correct answer, no change for wrong answer)

A Strategy pattern consist of at least three classes with the roles Context, AbstractStrategy, and ConcreteStrategy.

○ False

○ True ✓

With the design pattern Strategy you want to be able to solve a specific task in different ways, so you need to have different implementations and let the compiler choose which one to use.

○ True ✓

○ False

Singleton uses Pure Fabrication.

○ False ✓

○ True

Abstract Factory is used to create the right type of object given a specific context, where the rest of the system does not need to know the exact type of the object.

○ True ✓

○ False

The design pattern Abstract Factory is just a special case of the design pattern Strategy.

○ True ✓

○ False

The design pattern Strategy uses Observer when you want to do something that depends on the current strategy.

○ True

○ False ✓

Singleton means you are only allowed to call the class once.

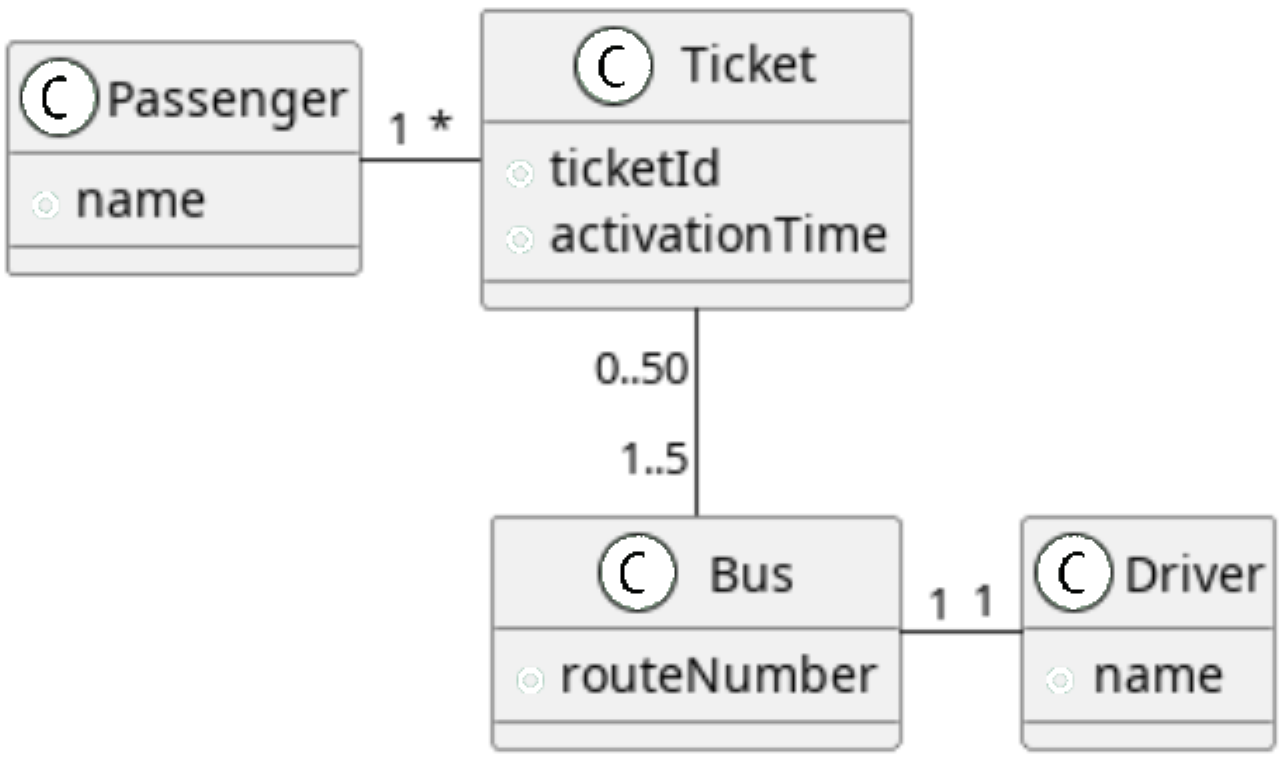
○ True

○ False

Maximum marks: 7

5 Class Diagram

Consider the following class diagram:



The class diagram describes part of a system for bus traffic.

For each of the statements below, please mark whether the diagram supports the statement (true) or does not support the statement (false).
(+1 for correct answer, no change for wrong answer)

b1:Bus is driven by happy:Driver

○ False

○ True ✓

deadbeat:Driver rides with the bus b1:Bus, but does not drive.

○ True

○ False ✓

charlie:Passenger has t1:Ticket.

○ True ✓

○ False

charlie:Passenger is planning to use t1:Ticket to first go into town with b1:Bus, and then continue to the next town with b2:Bus.

○ False

○ True ✓

dave:Passenger has t2:Ticket to tr1:Train.

○ True

○ False ✓

in order to get high cohesion, there should be a separate class Name to represent the attribute "name" which is found both in Passenger and Driver.

○ True

○ False ✓

happy:Driver and charlie:Passenger are best friends so they talk to each other when happy sees charlie on the bus.

○ True

○ False ✓

There is no way for charlie:Passenger to know which :Bus which will take him into tow.

○ False

○ True ✓

When a :Ticket has been activated on a :Bus, you have 24 hours to get to where you are going.

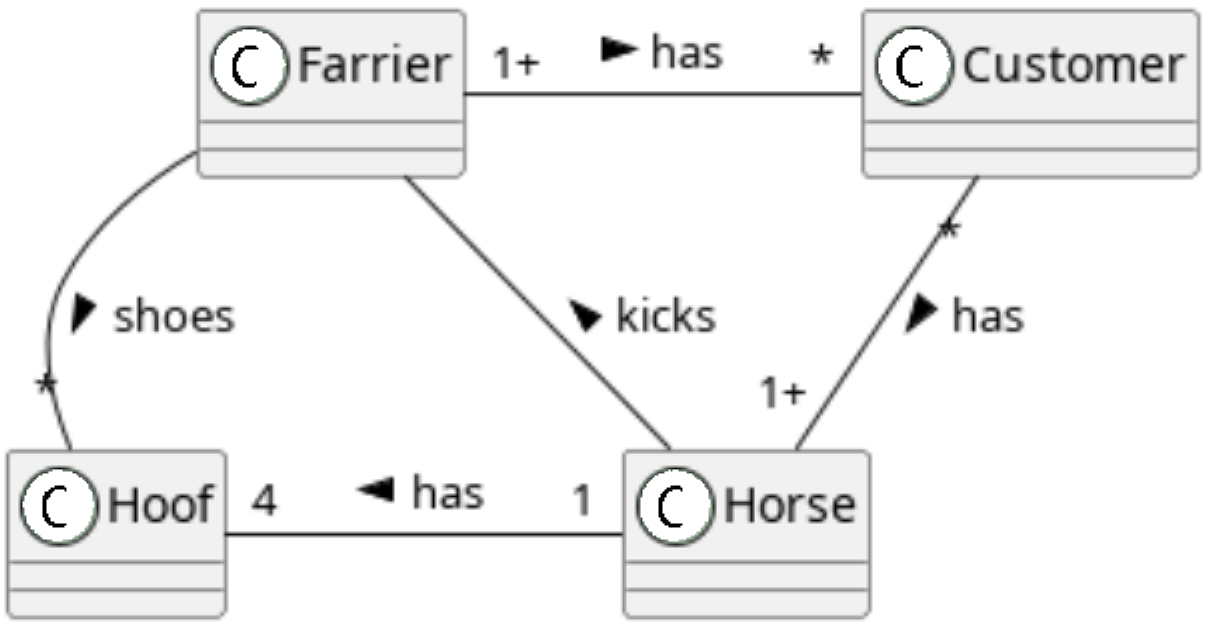
○ False ✓

○ True

Maximum marks: 9

6 Class Relations

Consider the following class diagram:



The class diagram show the relations between a Farrier, their customers (Horse), and the customer's owners (Customer).

For each of the statements below, please mark whether the diagram supports the statement (true) or does not support the statement (false).
(+1 for correct answer, no change for wrong answer)

A Farrier does not need to have any customers.

○ True ✓

○ False

It is undefined how many Horses kick their Farrier (but is hopefully close to zero)

○ False

○ True ✓

Bert:Customer owns horace:Horse and rosa:Horse

○ True ✓

○ False

A Farrier shoes horses (Horse)

○ False ✓

○ True

As a result of an accident, lukas:Horse only has three Hoofs.

○ True

○ False ✓

jakob:Farrier can shoe the hoofs of any number of horses.

○ True ✓

○ False

wellington:Customer only owns copenhagen:Horse

○ False

○ True ✓

erica:Customer only has an association with jakob:Farrier

○ True ✓

○ False

filippa:Customer, on the other hand, has an association both with jakob:Farrier and knut:Farrier

○ True ✓

○ False

jakob:Customer shoes his own horses.

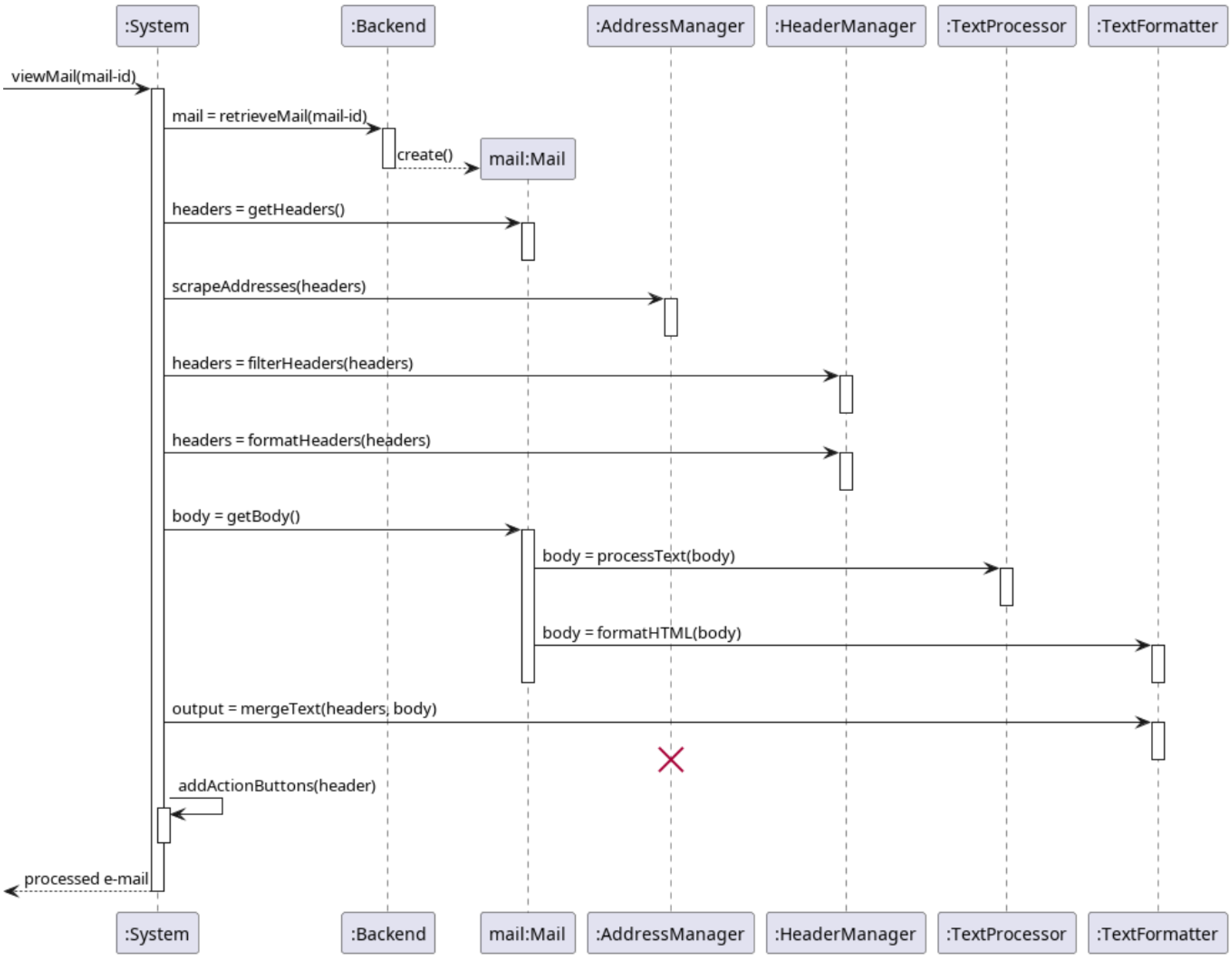
○ False ✓

○ True

Maximum marks: 10

7 Interaction Diagrams

Consider the following sequence diagram:



The sequence diagram describes a part of a mail program, specifically what happens when you want to view a specific mail.

For each of the statements below, please mark whether the diagram supports the statement (true) or does not support the statement (false).
(+1 for correct answer, no change for wrong answer)

:System contains the methods retreiveMail(), getHeaders(), scrapeAddresses(), filterHeaders(), formatHeaders(), getBody(), mergeText(), and addActionButtons().

○ True

○ False ✓

The call to addActionButtons(header) must go to some other object than :System.

○ False ✓

○ True

:Backend only knows how to retrieve one mail, but nothing about the contents of the mail.

○ True ✓

○ False

the class System is a controller for everything that should be done to a mail before it is viewed.

○ True ✓

○ False

According to high cohesion and low coupling, it is mail:Mail that should make sure that headers are formatted and filtered (and not :System).

○ True ✓

○ False

The variable "headers" is stored in :HeaderManager.

○ False ✓

○ True

:AddressManager is stalled and dies by the big X.

○ True

○ False ✓

the class Mail is information expert on everything specific to a certain mail.

○ False

○ True ✓

The classes AddressManager and HeaderManager have these names because both of them inherit from the base class Manager.

○ False ✓

○ True

The class Mail is a controller for what needs to be done with the body of a mail.

○ False

○ True

Maximum marks: 10

ℹ Grade Limits

The grade limits for this exam are:

Grade	Percent	Points
MAX	100%	60
A	90%	54
B	80%	48
C	70%	42
D	65%	39
E	60%	36

Good luck!