IT UNIVERSITY OF COPENHAGEN

# Assignment 2

**Analysis Design and Software Architecture,**
**Bachelor in Software Development,**
**IT University of Copenhagen**

**Hand-in date:** 24/09 2021

**Group members**
Rasmus Ulrik Kristensen
Sarah Cecilie Chytræus Christiansen
David Martin Sørensen

**Emails**
rkri@itu.dk
sacc@itu.dk
daso@itu.dk

## Our github repository:

https://github.itu.dk/daso/Assignment2.git

# Contents

# C#

## Exercise 4

Classes and records are both reference types where a struct is a value type. A value type is passed as a value and a reference type are passed as a reference. A value type does not require instantiation on the heap and is therefor much better when having to create a lot of instances. Structs also do not support inheritance. Records are great for immutable data. And when creating immutable objects the structural equality is that two instances are the same if the have the same data instead of the same reference as it would be with a normal class - it uses value-based equality.

- Class
  Use a class for complex values and methods and where you cannot use a struct or a record to fulfill your needs. Also use a class when you are focusing on the behavior of the type.
- Struct
  Use a struct whenever you do not need a reference type and can settle with a value type and do not need inheritance. A struct should be used when it is very simple like representing a point with (x,y) where the focus is not on the behavior but just the data.
- Record
  Use a record when you want something that should be immutable and you want the value equality instead. It also has great built in formatting for displaying (ToString).

# Software Engineering

## Exercise 1

A use case describes all the possible scenarios to focus on the completeness of the functionality whereas a scenario is an instance of the use case. Scenarios are used for

illustration the common cases and describing some set of concrete actions. To set it into perspective of programming, a use case can be seen as a class (so a general thing) and a scenario as an instance used in a specific situation. You use both constructs under the "Requirements elicitation activities" where you first identify the scenarios and then the use cases.

## Exercise 2

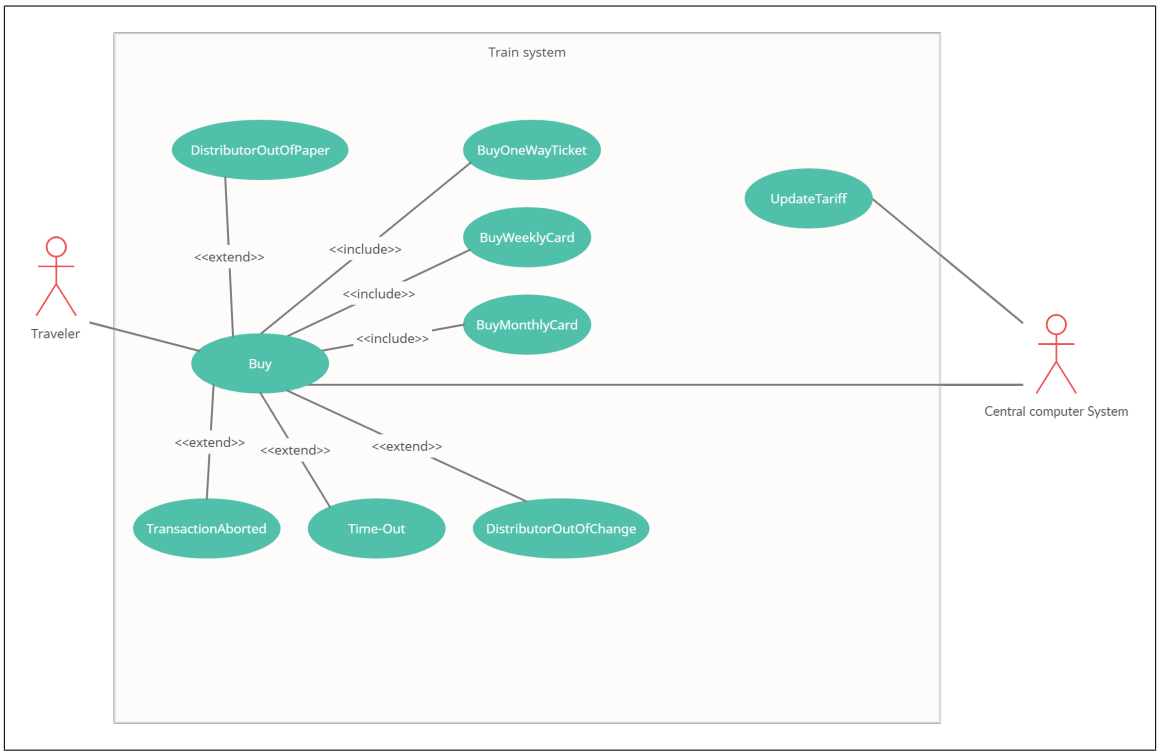


**Figure** 1: Exercise 2 - use case diagram (ticket system)

## Exercise 3

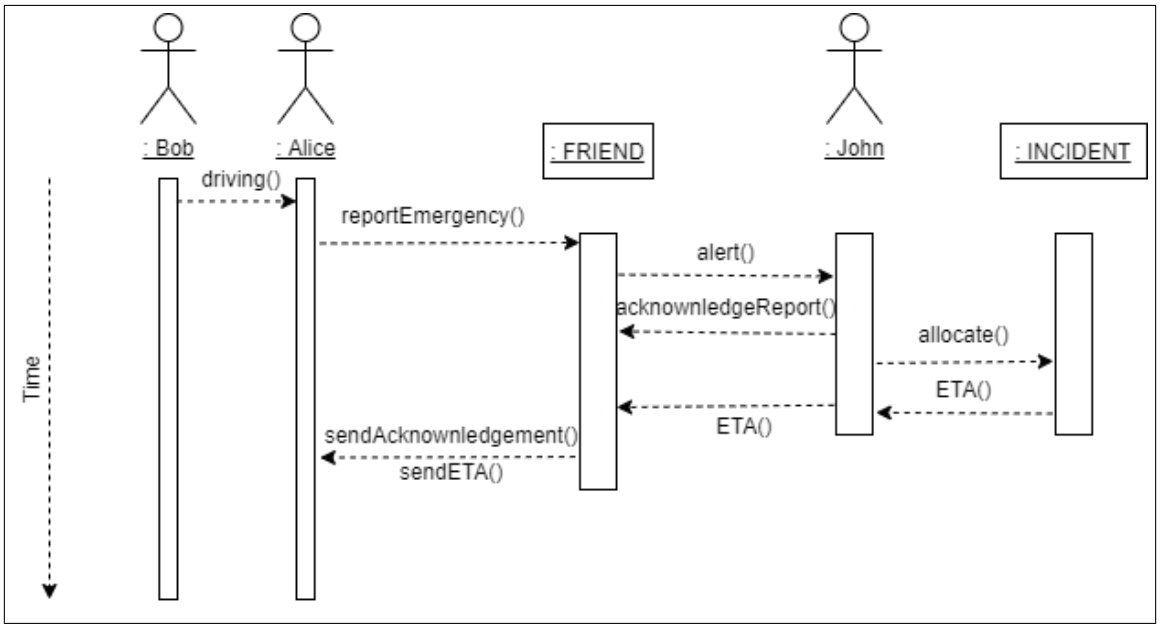


**Figure** 2: Exercise 3 - sequence diagram (wareHouseOnFire)

## Exercise 4

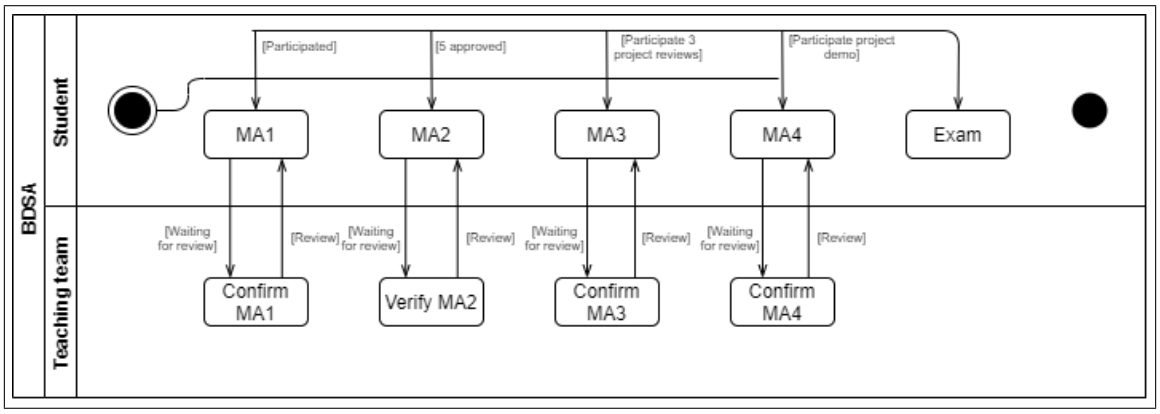


**Figure** 3: Exercise 4 - swimlane diagram (course)

## Exercise 5

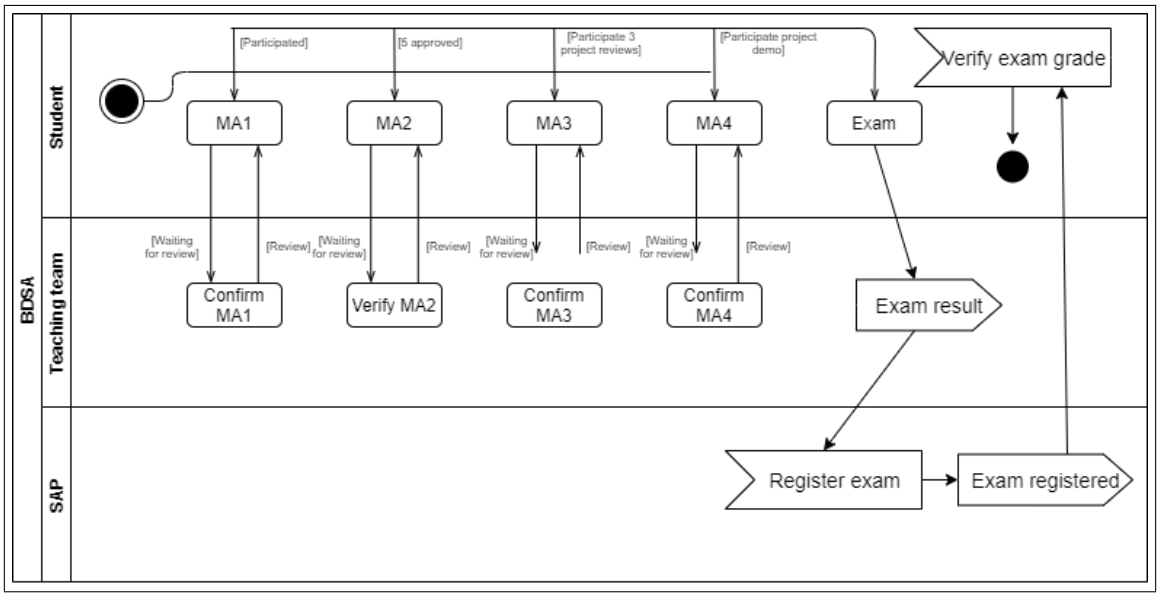


**Figure** 4: Exercise 5 - swimlane diagram (course) - extended

## Exercise 6

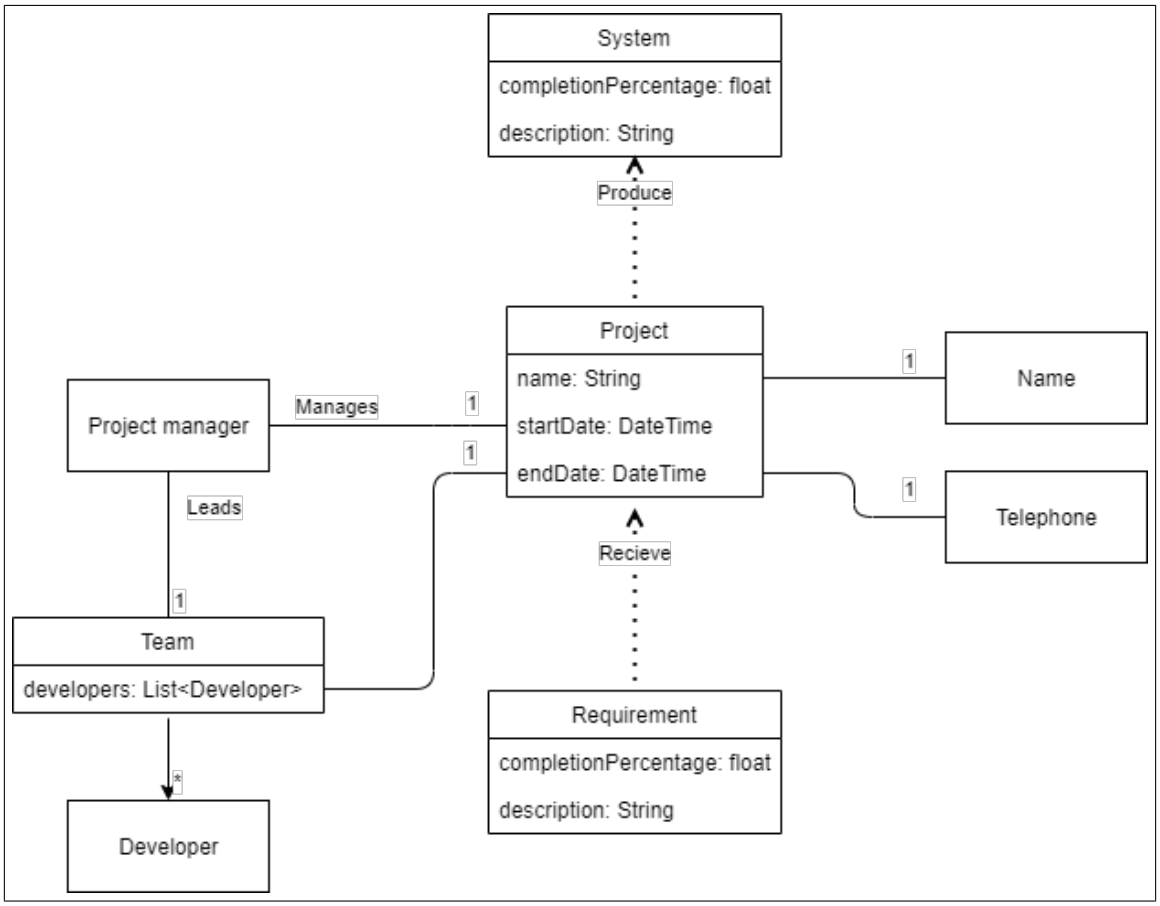


**Figure** 5: Exercise 6 - class diagram (Project)

## Exercise 7

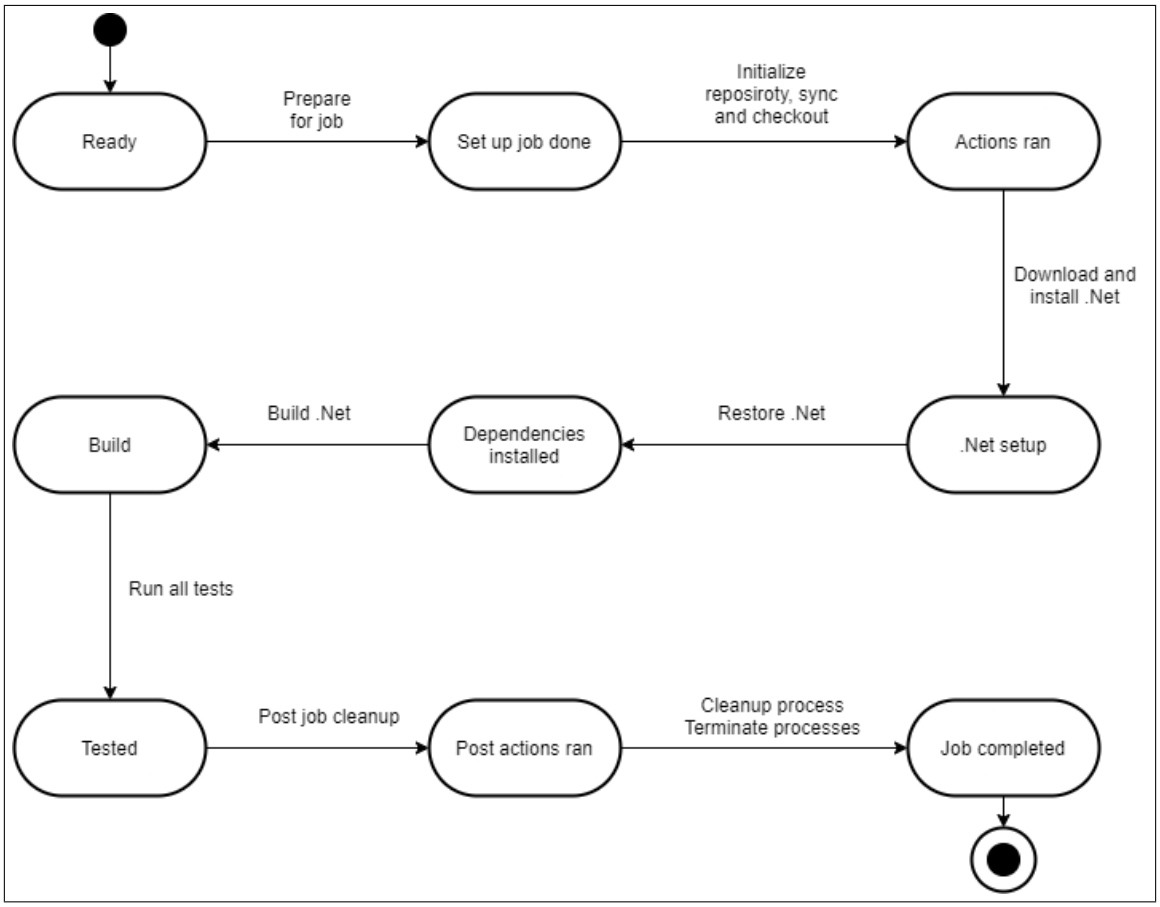


**Figure** 6: Exercise 7 - state diagram (github)

## Exercise 8

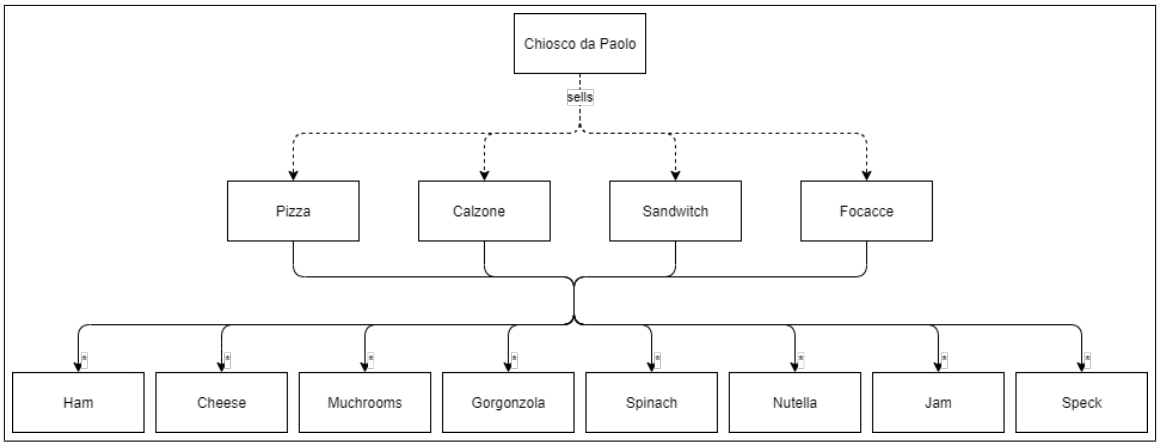


**Figure** 7: Exercise 8 - class diagram (Chiosco da Paolo)