BDSA - Assignment02

Jaria Sally Dumbuya (jard)
Magnus Dalsgaard Larsen (daml)
Magnus E.G. Svensson (mesv)

24. September 2021

1 C#

Github Repository Link

https://github.com/BlueMango10/bdsa-assignment2

Exercise 1

Look at the repository on Github.

Exercise 2

Look at the repository on Github.

Exercise 3

Look at the repository on Github.

Exercise 4

The class and record construct are reference types, which means that they hold the address to an object and not the object itself. While the struct is a value type, which means that it holds the object within and therefore needs memory allocated every time it is called. In addition the record and class construct support inheritance while the struct does not.
Furthermore the key difference between the class and record construct is that the properties/fields of a class can be both immutable and mutable, which means that they can be both changed or locked with the value that the object was initialised with, while the record construct is immutable by default, although

this can be changed with public fields.

As a result of the aforementioned differences a record construct can be used when we are working with a substantial amount of data, that is shared between concurrent programs to prevent different threads from trying to change the same data at the same time. A struct can be used when we are dealing with a small amount of related data for example a point in the plane consisting of one y and x coordinate. Lastly we can use a class when facing no constraints.

2 Software Engineering

Exercise 1

A scenario describes a series of interactions between the user and the system and thereby provides an informal description of a single feature of the system from the perspective of one actor. A use case is an abstraction that describes all possible scenarios/actions between an actor and the system for a functionality and may include other actors. In addition the use case serves as an formal description of a functionality. In conclusion the use case is general while the scenario is a concrete instance of a use case.
Both constructs are used during requirements elicitation. We use scenarios when describing typical functionality that is to be provided by the future system. In addition we can use scenarios to communicate with the user and get a better understanding of the application domain including getting an understanding of the users assumptions about the system.
Use cases can be derived once users and developers have agreed on relevant scenarios, we can then use them to describe the complete functionality and scope of the future system.

Exercise 2

See figure 1.

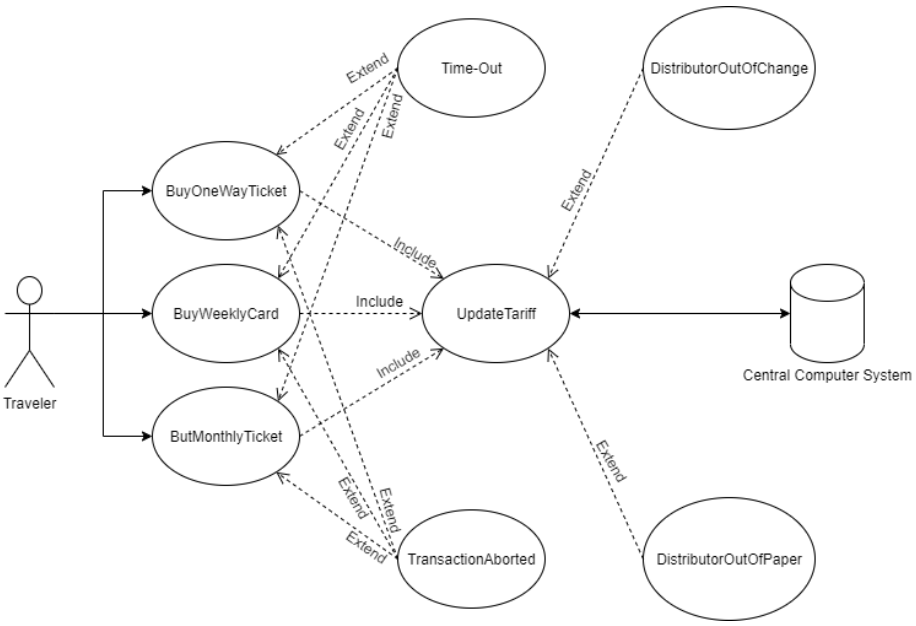

Figure 1: Use case diagram for a ticket distributor for a train system.

Exercise 3

See figure 2.

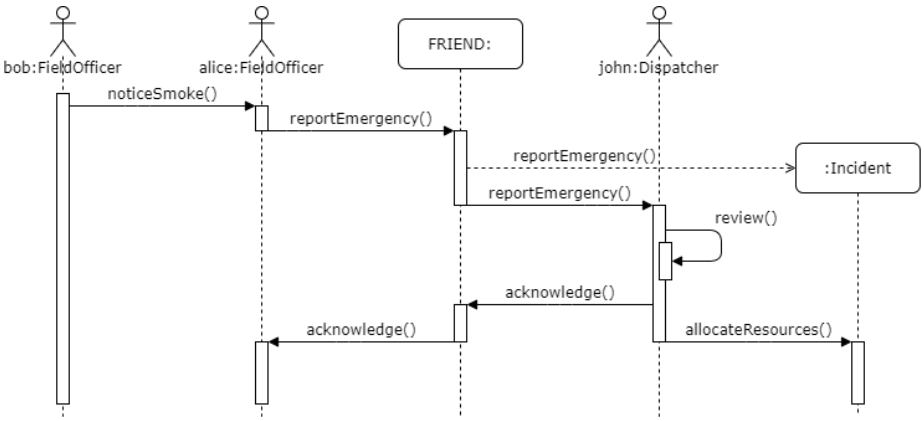

Figure 2: Sequence diagram of the `warehouseOnFire` scenario of Figure 2-21.

Exercise 4

See figure 3.

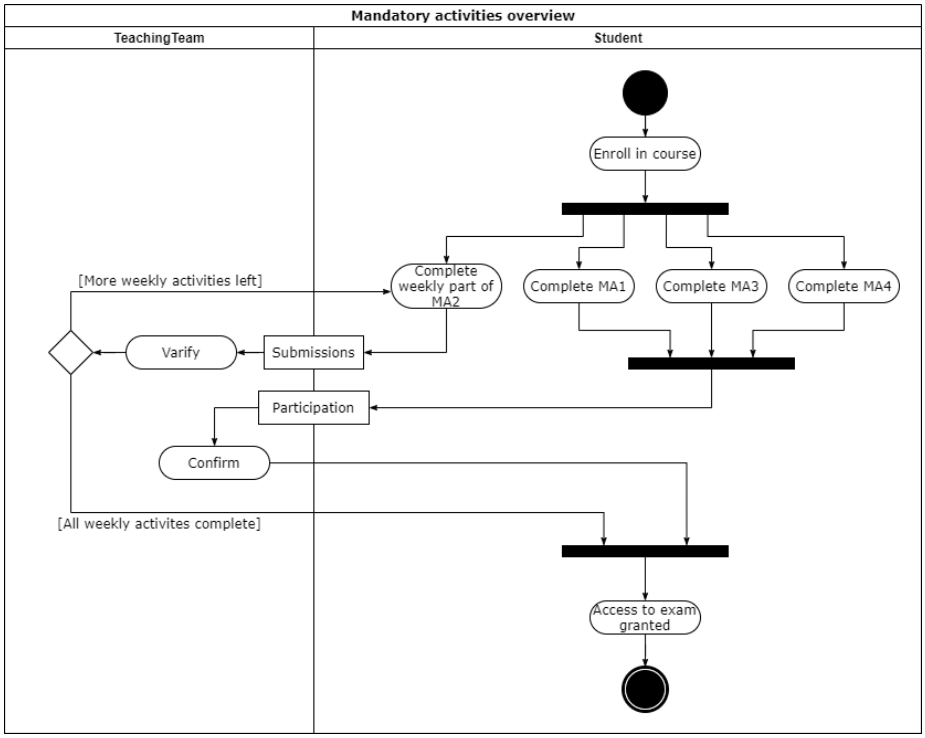

Figure 3: Activity diagram describing the mandatory activity requirements for accessing the exam.
`MA1` is an exam simulation, `MA2` is 5 weekly assignments, `MA3` is three project reviews and `MA4` is a project demo.

Exercise 5

See figure 4.

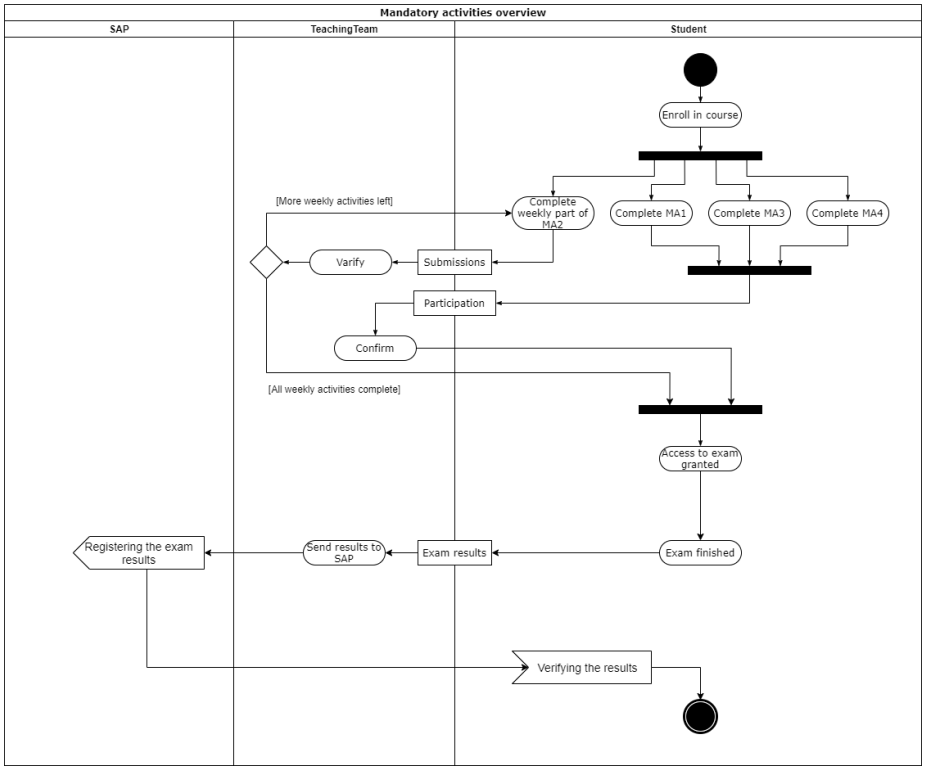

Figure 4: Activity diagram describing the mandatory activity requirements for accessing the exam and the exam verification process.

Exercise 6

See figure 5.

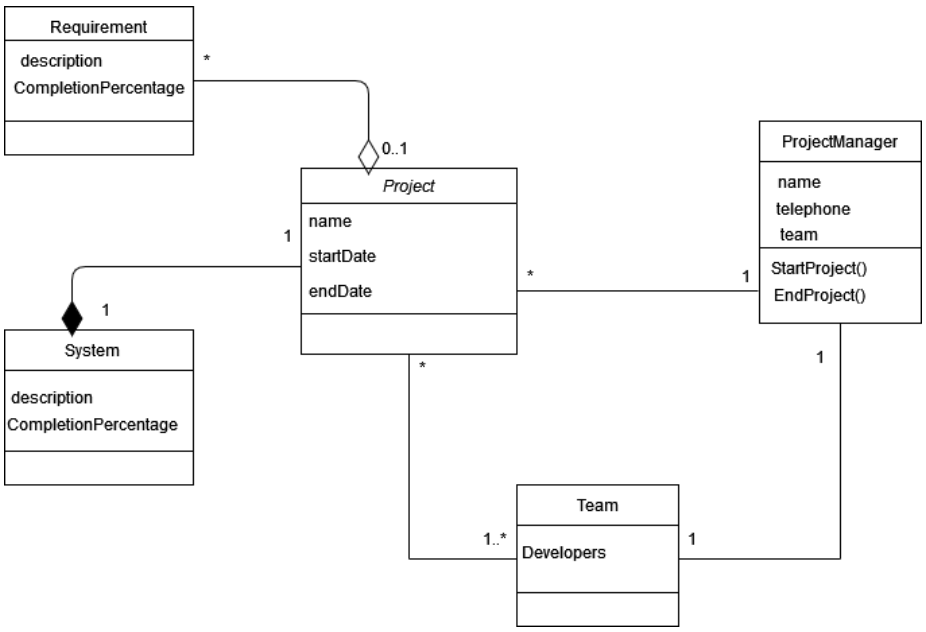

Figure 5: Class diagram modelling a set of requirements

Exercise 7

See figure 6.

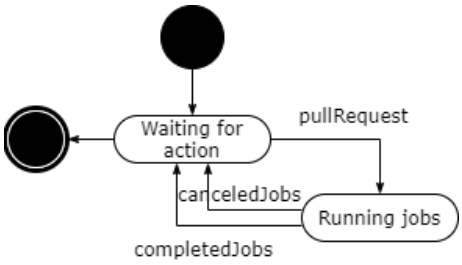

Figure 6: State machine describing our github actions setup.

Exercise 8

See figure 7.

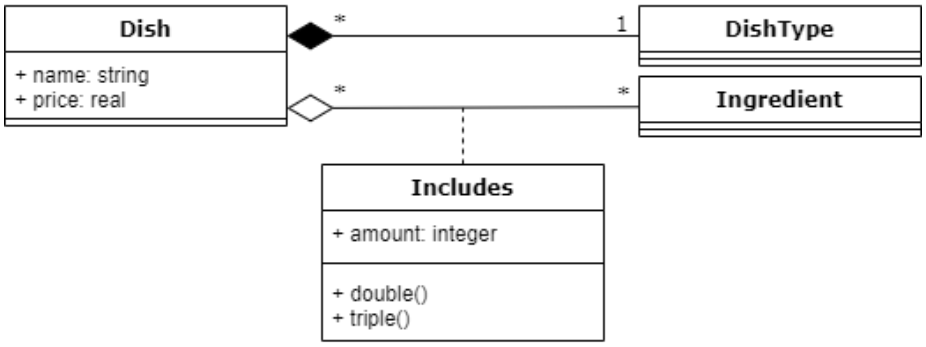

Figure 7: A class diagram representing the different types of dishes that can be ordered at Chiosco da Paolo.