

BDSA Assignment 2

Mille M. Z. Loo (milo), Ahmed Bassam Galal (ahga), Michael Tran(mhtr)

September 23, 2021

Link to the github repo: <https://github.com/Mhtr/Assignment2>

1 C#

Class:

A class would be used to model real life objects such as a person. It's able to have properties and methods that describes the object. Properties for a person could be name, surname, age and birthdate which you can use to create an instance of a person. Classes also allows you to use things such as inheritance to create child and parent classes, where the fields and methods might only vary slightly. This is useful to reduce code duplication when worked with related objects.

Struct:

A struct is a lightweight datatype compared to classes. Structs are value types which takes less memory than reference types which a class is. So in applications where you manage big amounts of data, it can pay off to save the data as structs instead of classes. On the flip side, structs do not support inheritance, so it does not have all the same options which classes has. Additionally structs does not support default constructors. So in systems where the data has a high level of complexity, classes can be more useful than structs

Record:

A record is an immutable data type meaning that it cannot mutate or change after it has been created. This would be useful if you were working with a database, with external data. You might want to work on the data without actually changing it. The record class also follows value based equality semantics making you able to compare two record constructs with the "==" operator. If the two records have the same values, this would output "true", whereas if you had compared two classes this would have been "false"

2 Software Engineering

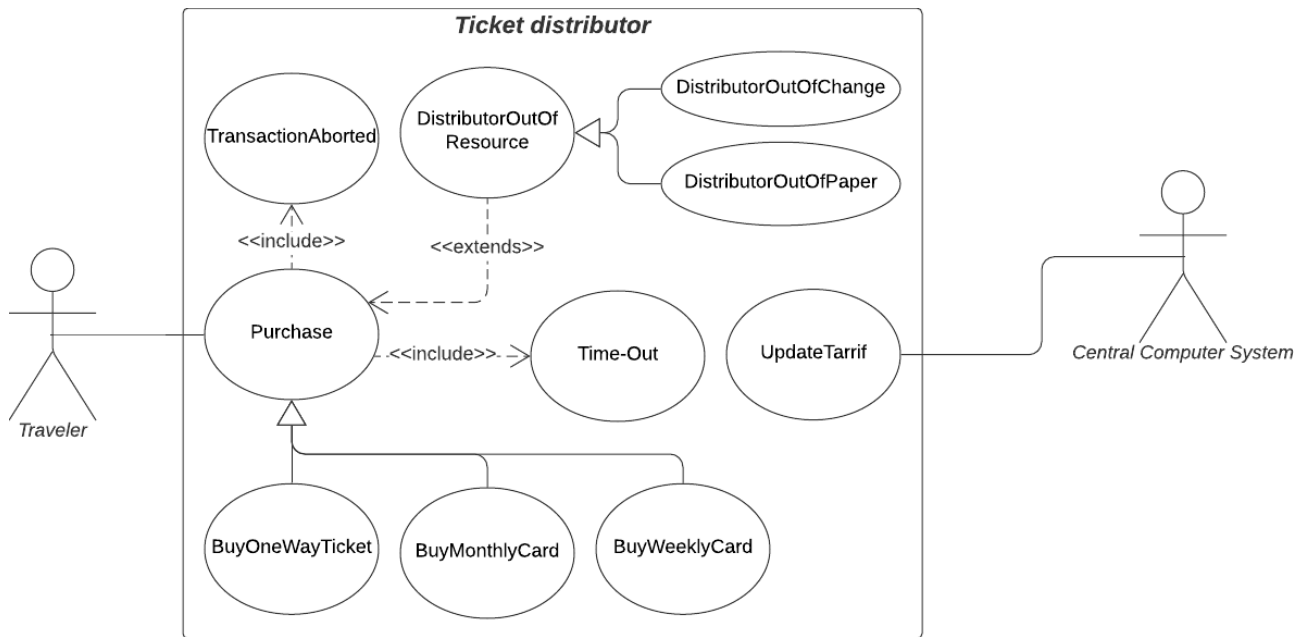
2.1 exercise 1

What is the difference between a scenario and a use case? When do you use each construct?

A **use case** refers to an abstraction that describes all possible scenarios involving a certain functionality, whilst a **scenario** is a specific instance of a use case, describing certain actions. Scenarios are used as examples for illustrating common cases, thereby making the focus of the scenario understandability, whereas use cases are used to describe all possible cases and focus more on the complete overview.

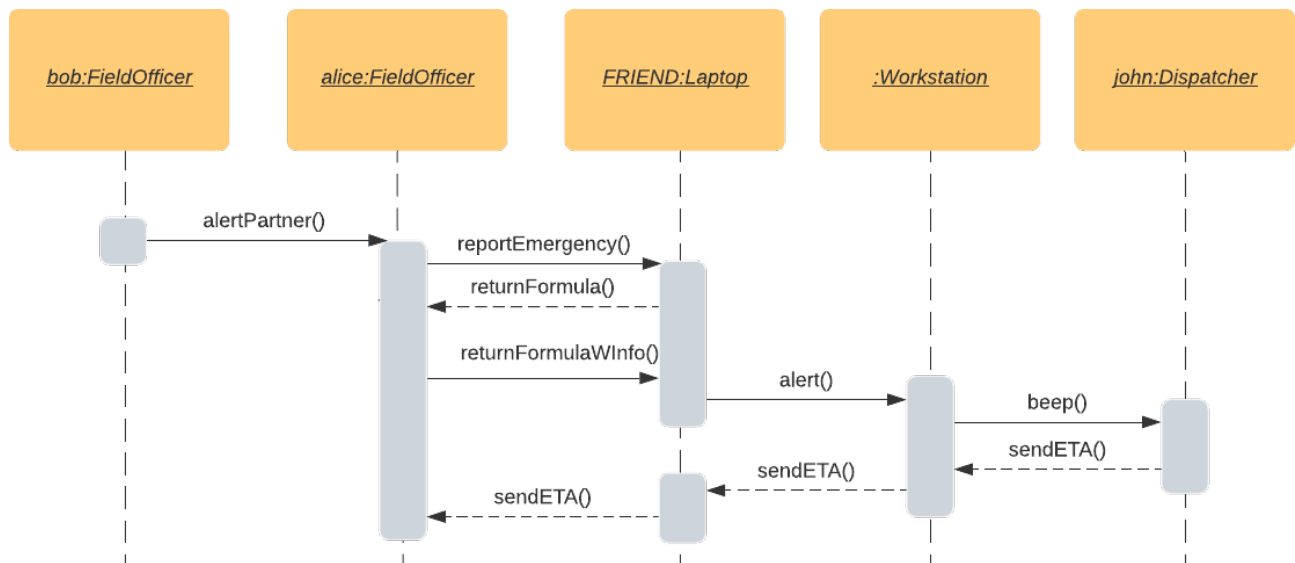
2.2 exercise 2

Draw a use case diagram for a ticket distributor for a train system. The system includes two actors: a traveler, who purchases different types of tickets, and a central computer system, which maintains a reference database for the tariff. Use cases should include: **BuyOneWayTicket**, **BuyWeeklyCard**, **BuyMonthlyCard**, **UpdateTariff**. Also include the following exceptional cases: **Time-Out** (i.e., traveler took too long to insert the right amount), **TransactionAborted** (i.e., traveler selected the cancel button without completing the transaction), **DistributorOutOfChange**, and **DistributorOutOfPaper**.



2.3 exercise 3

Draw a sequence diagram for the warehouseOnFire scenario of Figure 2-21. Include the objects **bob**, **alice**, **john**, **FRIEND**, and instances of other classes you may need. Draw only the first five message sends.



2.4 exercise 4/5

This year, the Analysis, Design, and Software Architecture course has re-introduced the group project and the entire process for accessing the exam with regards to the mandatory activities has changed. You have been tasked with drawing a UML diagram to depict the new process. Looking at the slides from the first lecture, you have found all the information you needed to complete the task. Diligently, you have summarized all of them in the below bullet point list:

- After a student enrolls in the course, four mandatory activities (MA) require completion before being allowed to take the exam:
 - MA1 requires a student to participate to an exam simulation;
 - MA2 requires a student to submit and get approved 5 weekly;
 - MA3 requires a student to participate to three project reviews;
 - MA4 requires a student to participate to the project demo.
- Participation to MA1 needs to be confirmed by the teaching team.
- Each weekly activity submission needs to be verified by the teaching team.
- Participation to MA3 needs to be confirmed by the teaching team.
- Participation to MA4 needs to be confirmed by the teaching team.

Note: for the purpose of this task you must assume the following:

- The re-execution of mandatory activities must not be modelled. For instance, failing to participate to the exam simulation would require a student to take the second run of such activity. You must not model this detail.
- The number of Weekly activities can be arbitrary.
- Responsibility for the execution of activities must be modelled.
- No time constrain must be included, hence, a student could spent a lifetime completing the mandatory activities and project reviews before being accepted to the exam.



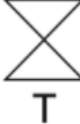
Using the solution for Exercise 4 and making use of event-based action (see below), include the process that describes:

- the teaching team sending the exam results to study administration (SAP);
- SAP registering the exams in the system;
- the students verifying their exam grade.

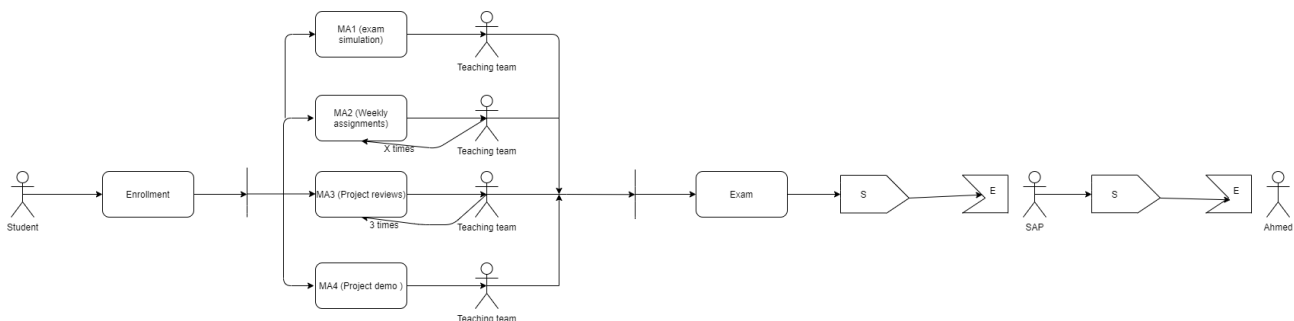
Event-based actions (from Section 7.2.1 in [1])

Event-based actions enable objects and signals to be transmitted to receiver objects. They allow you to distinguish between different types of events. You can use an **accept event action** to model an action that waits for the occurrence of a specific event. The notation element for an accept event action is a "concave pentagon"---a rectangle with a tip that points inwards from the left. If the event is a time-based event, you can use an **accept time event action**, whereby in this case, the notation is an hourglass.

To send signals, you can use **send signal actions**. Send signal actions are denoted with a "convex pentagon"---a rectangle with a tip that protrudes to the right.

Send signal action		Transmission of a signal to a receiver
Asynchronous accept (time) event action	 or 	Wait for an event E or a time event T

[1] Seidl, Martina, Marion Scholz, Christian Huemer, and Gerti Kappel. UML@ classroom: An introduction to object-oriented modeling. Springer, 2015.



2.5 exercise 6

Draw a class diagram that models the following specifications:

- Each project has a name, a start date, and an end date.
- Each project is associated to a project manager, a name, a telephone, and a team.
- The project manager manages (by starting and terminating) a project and leads the team associated with the project.
- The project receives as input the requirements and produces a system. Both requirements and the system have a completion percentage and a description.
- Each team is composed by developers.

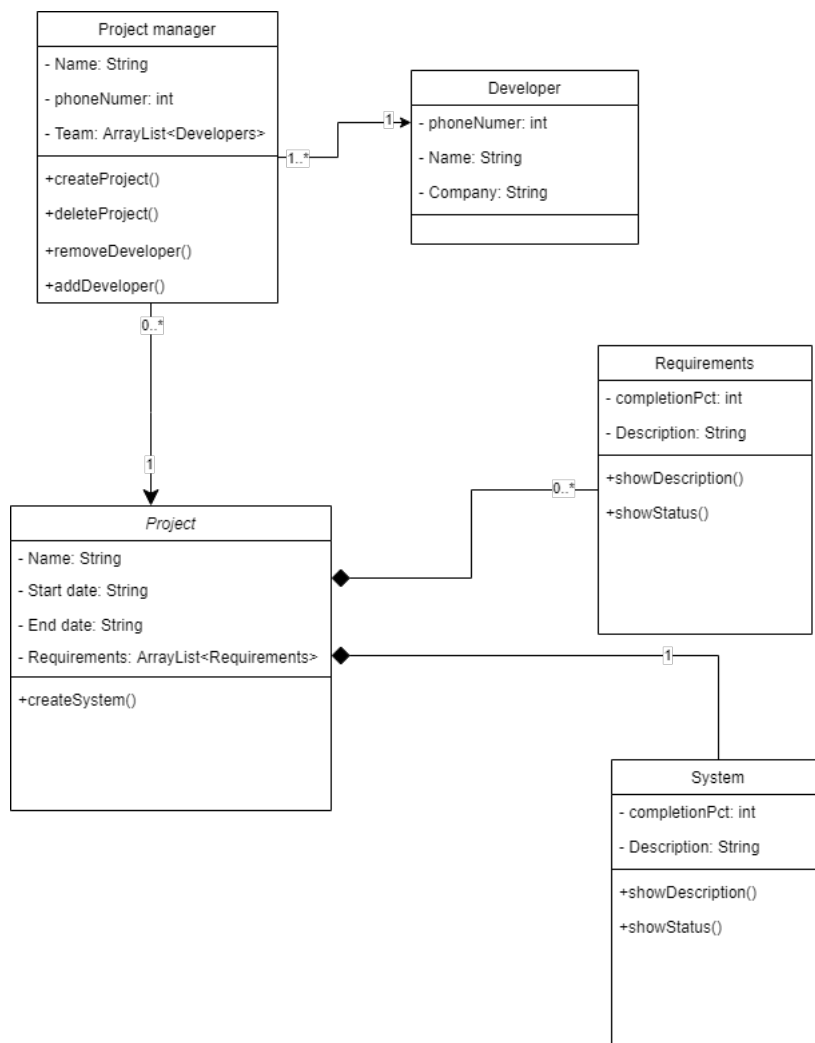


Figure 1: UML class diagram (Project)

2.6 exercise 7

Draw a state machine diagram that models your GitHub action configuration; include all triggers that you have defined.

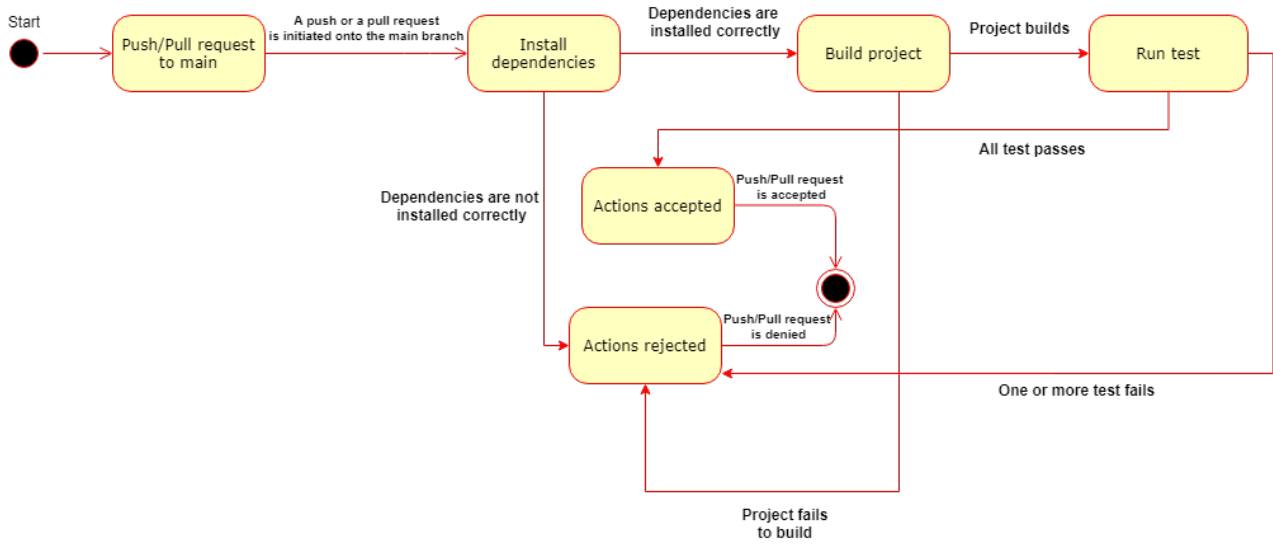


Figure 2: State machine diagram

2.7 exercise 8

You are asked to design a software system to support the **Chiosco da Paolo** new fast food take away business concept. You have had the chance to have a chat with Paolo and he told you all about the offering that the **Chiosco da Paolo** will have at the opening. Ranting about how these italians always want options and exceptions, you try to wrap your head around a class diagram you are drawing to model all the notes you took when Paolo was presenting the food. Among your notes:

- the place is meant to sell food. Paolo mentioned: pizza, calzone, toast, sandwich, focacce, ...
- Paolo also mentioned that somehow he can do all the above with an assortment of ingredients, which can all be combined at the wimps of the customer. He mentioned: ham, cheese, muchrooms, gorgonzola, spinach, speck, nutella, jam, ...
- Paolo indicated that he wants the customer to be able to say that they want double, triple, ... amount of each ingredient

Draw a class diagram representing the different types of offering that can be ordered at **Chiosco da Paolo**.

