Assignment 3

Anton Vitus Sohl Andersen, Tore Asbjørn Tonnisen Kjelds and Mikkel Bistrup Andersen

September 2022

Link to GitHub repository: https://github.com/tkjelds/assignment-03

Contents

1 Exercise 1

- **What level of detail should UML models have?**

  UML models should be an abstract and not necessarily complete representation of a system. Consequently a UML model purposely leaves out details to simplify a system thus making it easier to understand.

- **What is the difference between structure diagrams and behavior diagrams in UML?**

  A structure diagram is a model of the components in a system and its relations. This model is used when designing the system architecture.

  One example of a structure diagram is class diagrams. These diagrams describe the classes and their relations between each other in a system. The classes is represented as an rectangle and lines between these boxes represent a relation. It is also possible to show multiplicities by using numbers or a *. You annotate this at the end of each line. Class diagrams can be of different levels of detail. When you a showing relations between the classes, it is best practise to represent classes in a simple form like explained above. You can however add attributes and methods to the individual classes if more detail is needed.

  Another example of a structure diagram is a hierarchy diagram. This diagram is used to show as the name say the hierarchy also known as inheritance of classes in a system. The classes is represented in rectangles with its attributes and methods. The hierarchy is shown with an arrowhead pointing up to the more general class.

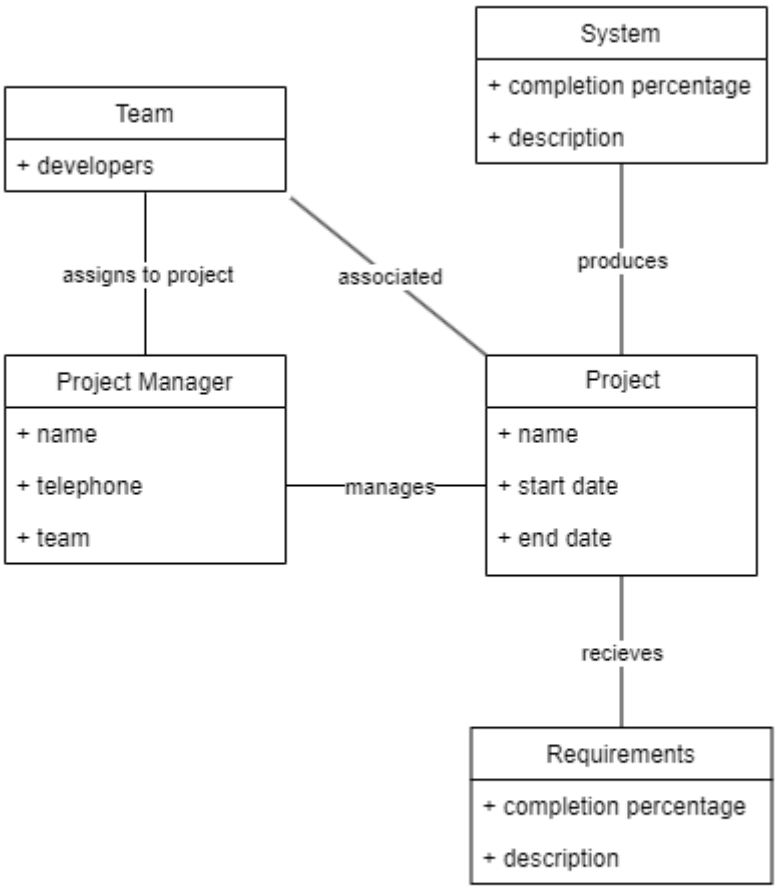
  A behavior diagram is a model of the behavior of a running system. These behaviors show what happens or what is supposed to happen when the system is responding to a stimulus from its environment. These stimulus can be data input or event triggers.

  One example is data-driven models. These models show the sequence of actions involved in processing input data and generating an output. This model can be represented using a activity diagram where the chain of processing steps and data flowing between these steps is shown. Processing steps is represented as rounded rectangles and data is represented as rectangles.

  Another example is event-driven models. These models show how a system responds to events. Event-driven models can be represented using a state diagram. This diagram show states and events in a system that cause transitions from one state to another state. Rounded rectangles represent system states. Labeled arrows represent stimuli that force a transition from one state to another state. The stating point is indicated with a filled circle and the end point is indicated with a double filled circle. The states may include a description of the actions in a state.

2 Exercise 2

Figure 1: A class diagram based on the specifications.



3 Exercise 3

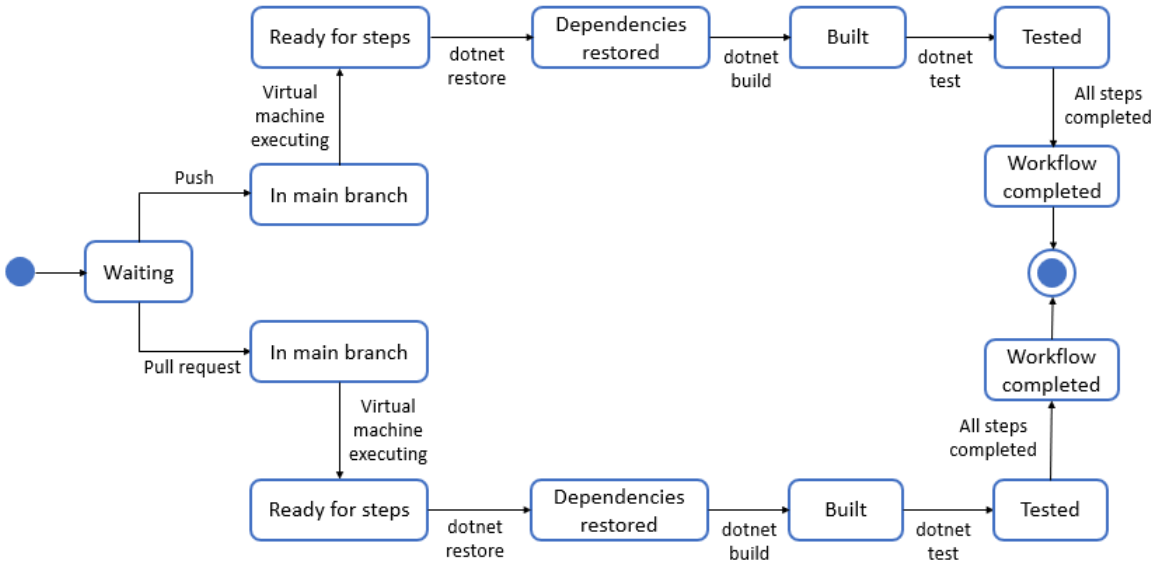


Figure 2: A state diagram of GitHub action configuration from assignment 0

The yml file from assignment 0

```
name: .NET on:
  push:
    branches: [ "main" ]
  pull_request:
    branches: [ "main" ]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
    - uses: actions/checkout@v3
    - name: Setup .NET
      uses: actions/setup-dotnet@v2
      with:
        dotnet-version: 6.0.x
    - name: Restore dependencies
      run: dotnet restore
    - name: Build
      run: dotnet build --no-restore
    - name: Test
      run: dotnet test --no-build --verbosity normal
```

4 Exercise 4

Figure 3: A state diagram of a cloned repo with a file edit.

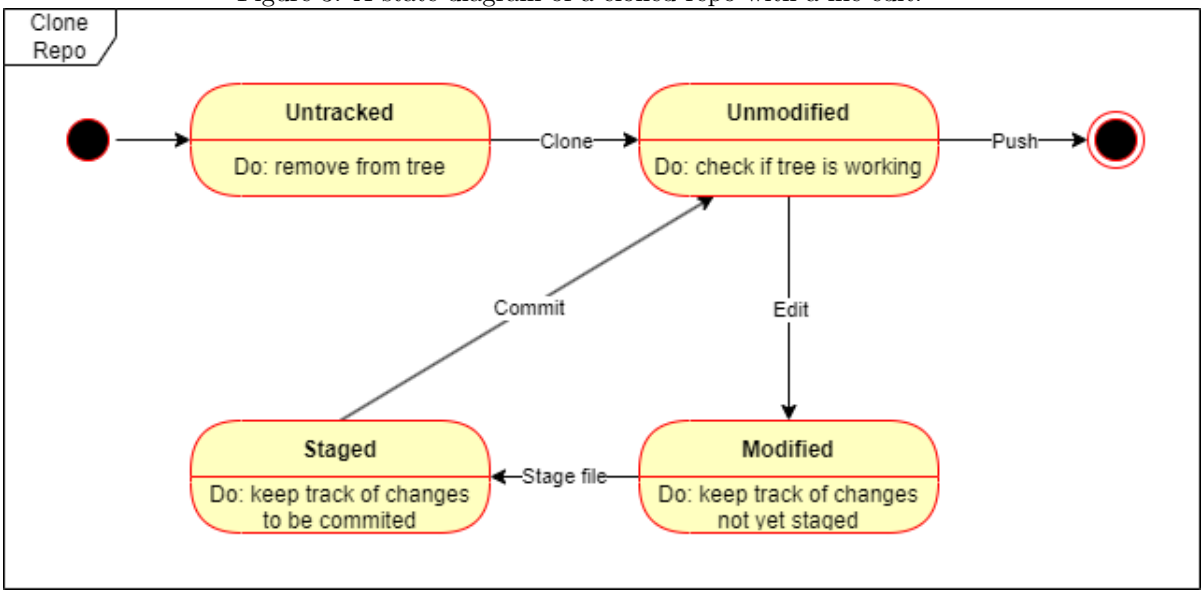
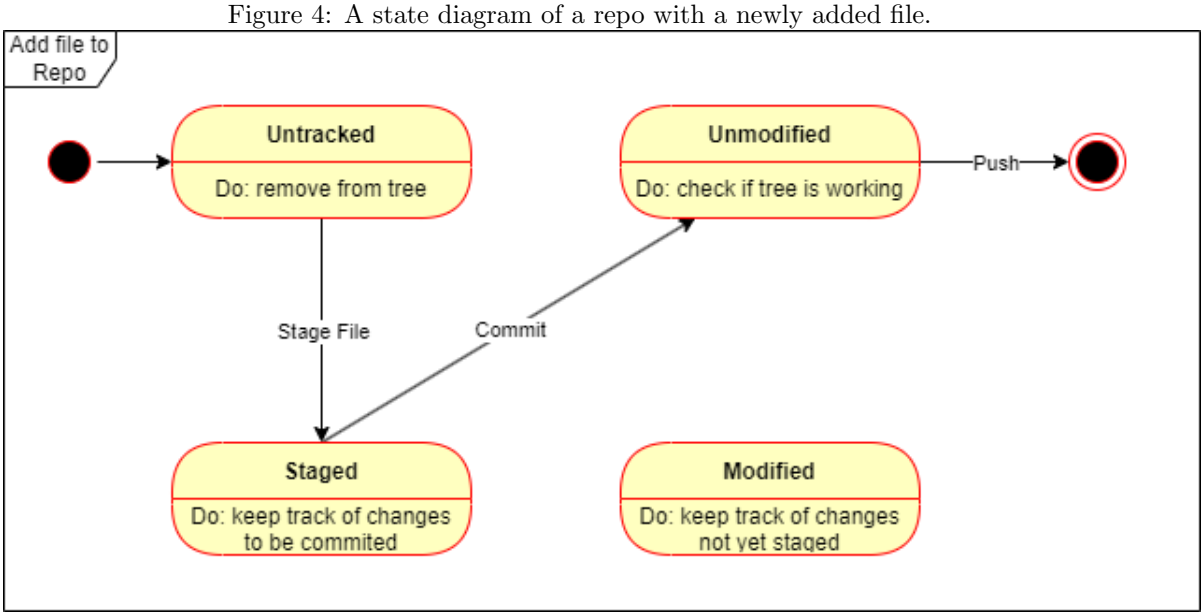
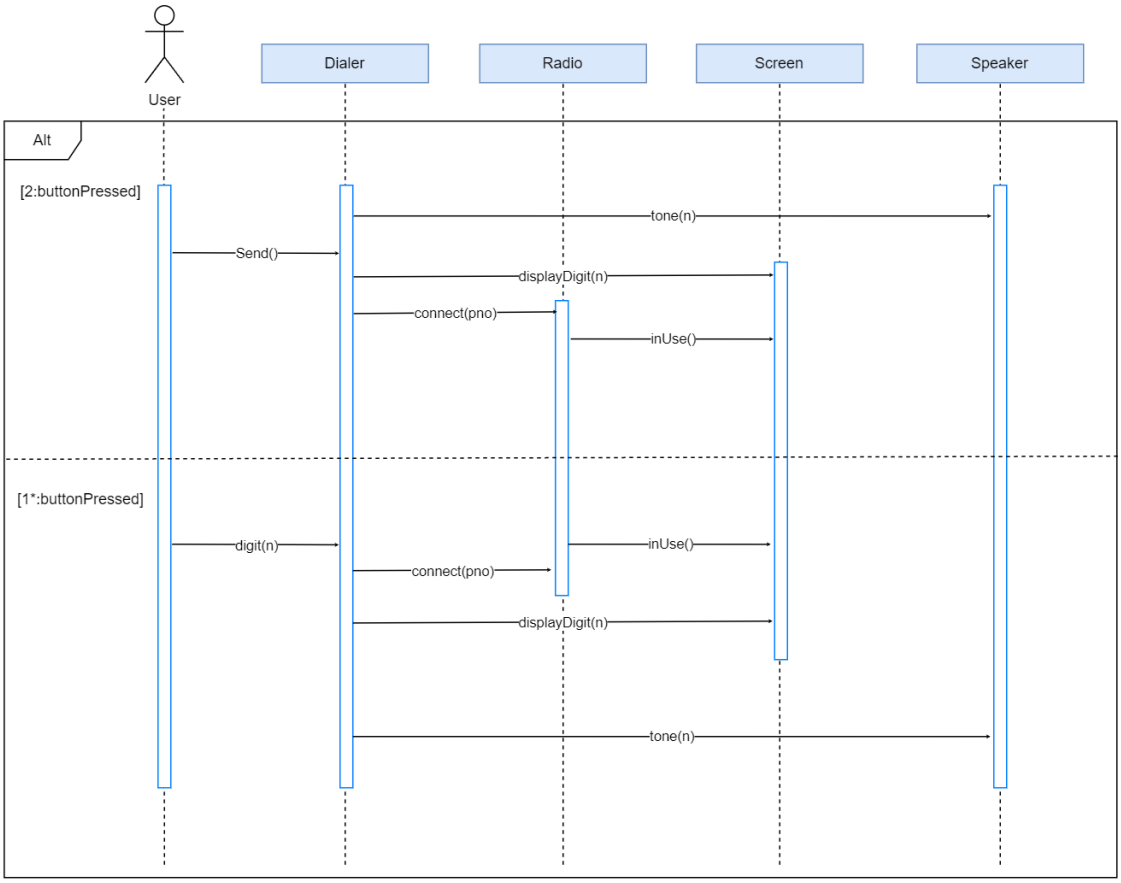


Figure 4: A state diagram of a repo with a newly added file.



5 Exercise 5

Figure 5: A Sequence diagram made from collaboration diagram



6 Exercise 6

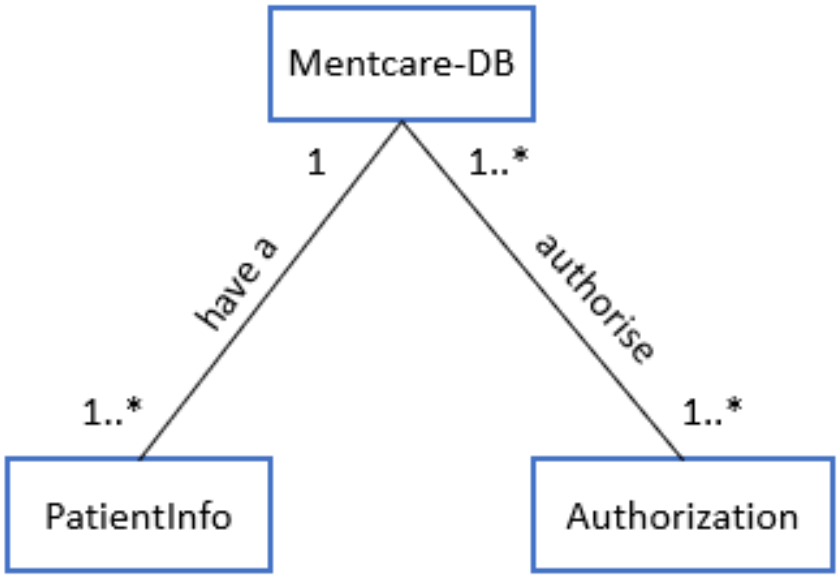


Figure 6: A UML class diagram that models the structural information given in Fig. 5.7