

# Test af viden og færdigheder

opgave 2022 januar

Frisør Booking System



# Frisør Booking System

## Case beskrivelse

Frisørsalonen 'Hårtotten' har 3 frisører ansat og er åben 6 dage om ugen.

Overordnet skal bookingsystemet kunne håndtere frisører og kunder. Desuden skal systemet kunne holde styr på forskellige bookninger på de forskellige dage.

## Overordnede System

På baggrund af denne overordnede beskrivelse skal du lave tre opgaver, senere kommer flere opgaver til en første version af systemet.

### Opgave 1: User stories (swd)

Du skal skrive 3 user stories med acceptance kriterier til Frisør Booking systemet.

### Opgave 2: Domain model (swd)

Du skal lave en domain model for centrale dele af Frisør Booking systemet, hvor der mindst indgår 5 klasser.

Du kan bruge et digitalt tegne værktøj som MS-Visio eller DrawIO.

### Opgave3: Design klasse diagram (swd)

Du skal nu lave et Design klasse diagram ud fra ovenstående domain model.

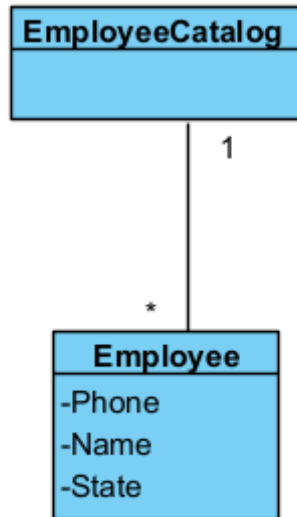
Hvad er forskellen på en Domain Model og et Design Klasse Diagram?

Du kan bruge et digitalt tegne værktøj som MS-Visio eller DrawIO.

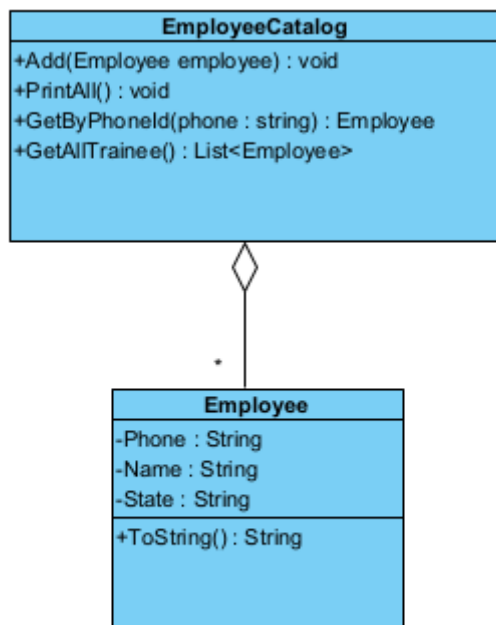
### Første version af Frisør Booking Systemet

Product Owner har besluttet at få lavet en første version af booking systemet, som indeholder håndteringen af Ansatte og dertil er der udarbejdet en domain model og et design klasse diagram

Første version: domain model



Første version: Design Klasse Diagram



#### Opgave 4: Implementer klassen Employee (swc)

Du skal implementere klassen Employee

1. Opret en **console application** (.Net Core) 'BookingApp'
2. Lav en folder (underkatalog) 'model'
3. I 'model' folderen opret en klasse 'Employee'
4. Klassen skal indeholde de tre properties 'Phone', 'Name' og 'State', Hvor state er 'frisør' eller 'elev'.
5. Lav to konstruktører
  - a. En default konstruktør (uden parametre)
  - b. En konstruktør til at sætte de tre properties
6. Lav en ToString metoder, der returnerer property værdierne i en string.

#### Opgave 5: Opret objekter af Employee (swc)

Du skal i Main oprette tre forskellige objekter af klassen Employee, samt udskrive disse på skærmen.

#### Opgave 6: Implementer klassen EmployeeCatalog (swc)

Du skal implementere klasse EmployeeCatalog i folderen 'model'

1. Opret klassen EmployeeCatalog.
2. Klassen skal indeholde en collection til at opbevare ansatte (employees)  
Du vælger selv om du vil benytte List eller Dictionary.
3. Lav en default konstruktør, hvor collection'en initialiseres.
4. Du skal implementere de fire metoder
  - a. Add(Employee newEmployee) som tilføjer en ny ansat til collection'en.
  - b. PrintAll() som udskriver alle ansatte på skærmen en ad gangen.
  - c. GetByPhoneld(String phoneld) som finder og returnere en ansat med den specificerede phoneld  
(hvis der ikke findes en ansat med phoneld'en kastes en exception – dette behandles senere i opgave 10).
  - d. GetAllTrainee() som finder alle elever og returnere dem i en liste.

#### Opgave 7: Anvend EmployeeCatalog (swc)

Du skal anvende klassen EmployeeCatalog i main

1. Du skal i Main oprette et objekt af klassen EmployeeCatalog.
2. Du skal tilføje de tre objekter fra opgave 5 til EmployeeCatalog-objektet.
3. Du skal udskrive alle ansatte til skærmen ved hjælp af EmployeeCatalog-objektet.
4. Du skal finde og udskrive en ansat med et givet phoneld.
5. Du skal finde og udskrive alle ansatte, der er elever.

### Opgave 8: Booking (swd)

Du skal udvide den første version, så det bliver muligt at en kunde kan booke en tid. En 'Customer' har to attributter/properties 'phone' og 'name'. Samt en 'Booking' har en relation til en ansat og til en kunde, samt et 'tidspunkt'.

Du skal i et design klasse diagram vise, hvorledes forholdet mellem 'Employee', 'Customer' og 'Booking' (i diagram form). Du kan udvide design klasse diagrammet for første version, som er vist tidligere.

Du kan bruge et digitalt tegne værktøj som MS-Visio eller DrawIO.

### Opgave 9: Implementer Customer og Booking (swc)

Du skal implementere klassen Customer

1. Lav en klasse Customer
2. lav to properties 'Phone' og 'Name'
3. Lav en konstruktør til at sætte relevante værdier
4. Lav en ToString metode til at returnere et planner objekts værdier.

Du skal implementere klassen Booking

1. Lav en klasse Booking
2. lav en property 'Time'
3. lave de to referencer til Customer hhv. Employee
4. Lav en konstruktør til at sætte relevante værdier
5. Lav en ToString metode til at returnere et planner objekts værdier.

Du skal desuden i programmet oprette et objekt af klassen Customer samt af Booking, samt udskrive begge.

### Opgave 10: Exceptions (swc)

Du skal understøtte at dit program ved fejlsituationer kaster en passende exception.

I metoden GetByPhoneld, den skal kaste en passende exception (du vælger selv hvilken), hvis der ikke findes en ansatte med det specificerede phoneld.

Du skal i programmet vise; Hvordan du kan håndtere, at du kalder metoden med et phoneld, der ikke findes, samt udskrive en passende fejl meddelelse på skærmen.