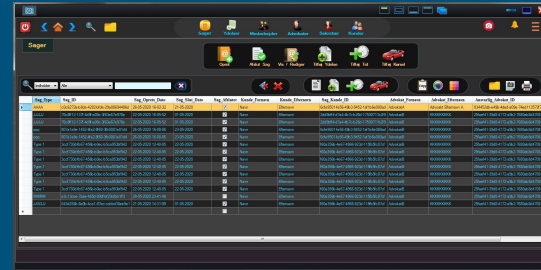


\_\_\_\_\_

Sommer 2020



# ADVOCATE



# Indhold:

---

1. Start af Planlægning
2. Process
3. Start på Udvikling
4. Skemaer / Diagrammer
5. Arkitektur
6. Programmering
7. Database

# Plan / Iterations Plan:

## 1. “Inception” - Iteration 1

---

### 1.1 Strategi

1.2 Valg af Process - Unified Process “UP” - “I vores tilfælde er det UP”

1.3 - Generel Kravspecificering

1.4 - Beskrivelse af Programmet “Hvad skal programmet kunne”

1.5 - Generelle Funktioner - “FURPS+” - “Funktionale. Krav Ikke Funktionale. krav”

1.6 - Use Cases - “Brief”

1.7 - Valg af GUI - “Win Forms” - “I vores tilfælde er det Win Forms”

1.8 - Valg af Database Type - “MMSQL” - “I vores tilfælde er det MMSQL”

1.9 - EER - Diagram - Database - “Start”

1.10 - GUI - “Start”

1.10 - Mini Prototype 1

# Plan / Iterations Plan:

## 2. “Elaboration” - Iteration 1, 2

2.1 - Valg af Arkitektur

2.2 - Package Diagram

2.3 - Proof of Concept - “Lille Prototype 2” - Demo -”GUI-uden mange funktioner”

2.4 - Use Case Diagram

2.5- Domain Model Diagram - “UML - Diagram”

2.6 - Class Diagram - “UML - Diagram” - “Start”

2.7 - SSD - “Start” - Viser Bestemt Scenarie af Use Case af ekstern aktør med “rækkefølge og relationer”

2.8 - SD - “Start” - Viser objekter og klasse som intercter med hinanden

2.9 - Programmering - “Start”

2.10 - GUI Design - “Start”

2.11 - Use Cases - “Casual”

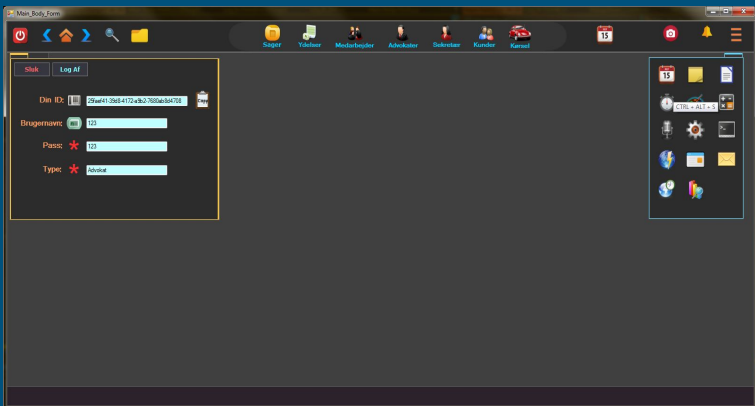
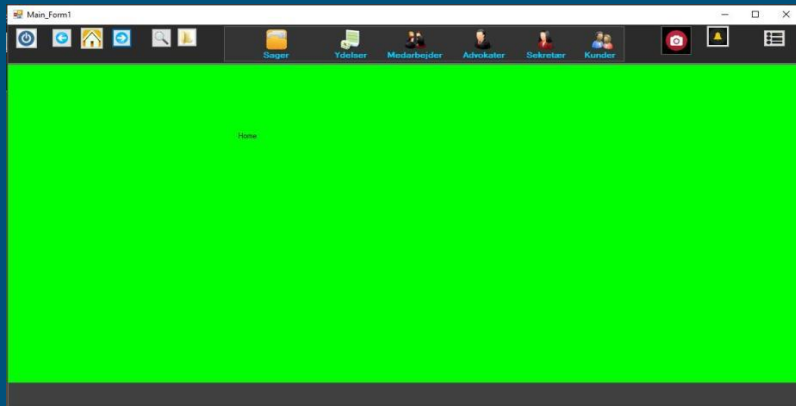
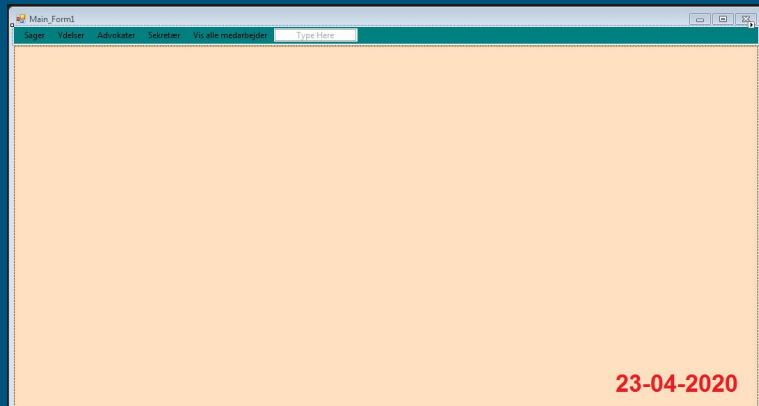
2.12 - Class Diagram - “Casual”

2.13 - Database - “Færdig”

2.14 - Tests

2.14 - Prototype 3

# Historik af Vores App:



# Plan / Iterations Plan:

## 3. “Construction” - Iteration 3,4,5

### 3.1 - Programming

### 3.2 - Use Cases “Fully Dressed”

### 3.3 - Use Case Diagram “Opdater hvis nødvendig”

### 3.4 - Class Diagram “Slut” -

### 3.5 - Tests

### 3.6- SSD “Slut”

### 3.7 - SD “Slut”

### 3.8 - Programming

### 3.9 - Prototype

### 3.10 - Tests - “Unit Tests”, “Acceptance Testing”

### 3.11 - Tests - “GUI Test”

### 3.12 - Bug Fixing

### 3.13 - Prototype 4

# Plan / Iterations Plan:

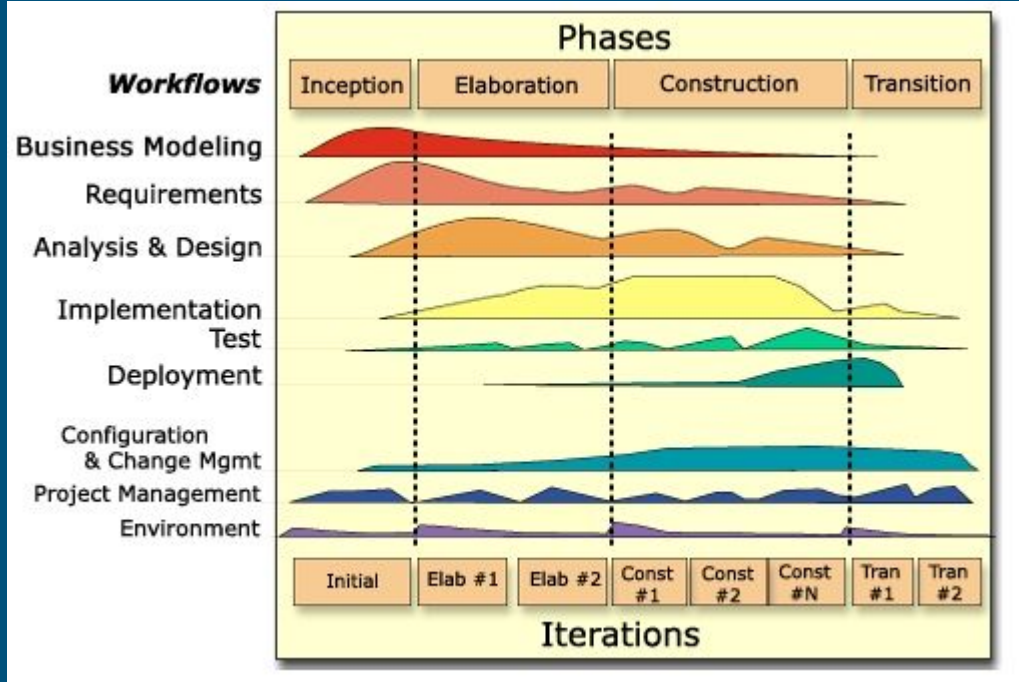
Transition” - Iteration 6

- 4.1 Sidste Tests

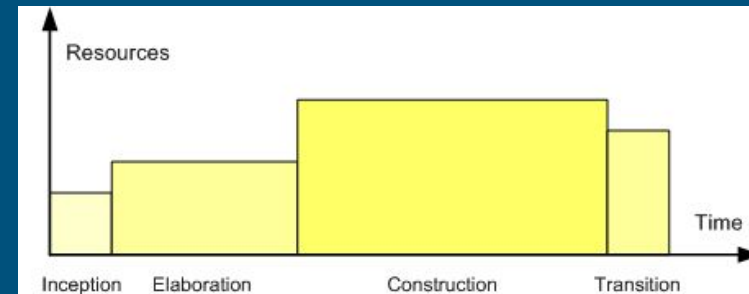
- 4.2 Integrer programmet i kundens miljø

- 4.3 Lær Kunden hvordan programmet skal bruges

# Process: UP - Unified Process



1. **Inception** - Den første Fase som er meget kort hvor man laver **Generelt Planlægning**.
2. **Elaboration** - Generelt - Diagrammer og Programmering
3. **Construction** - Generelt - Programmering og Færdiggøring af Diagrammerne.
4. **Transition** - Sidste Tests og Integrer programmet i kundens miljø.

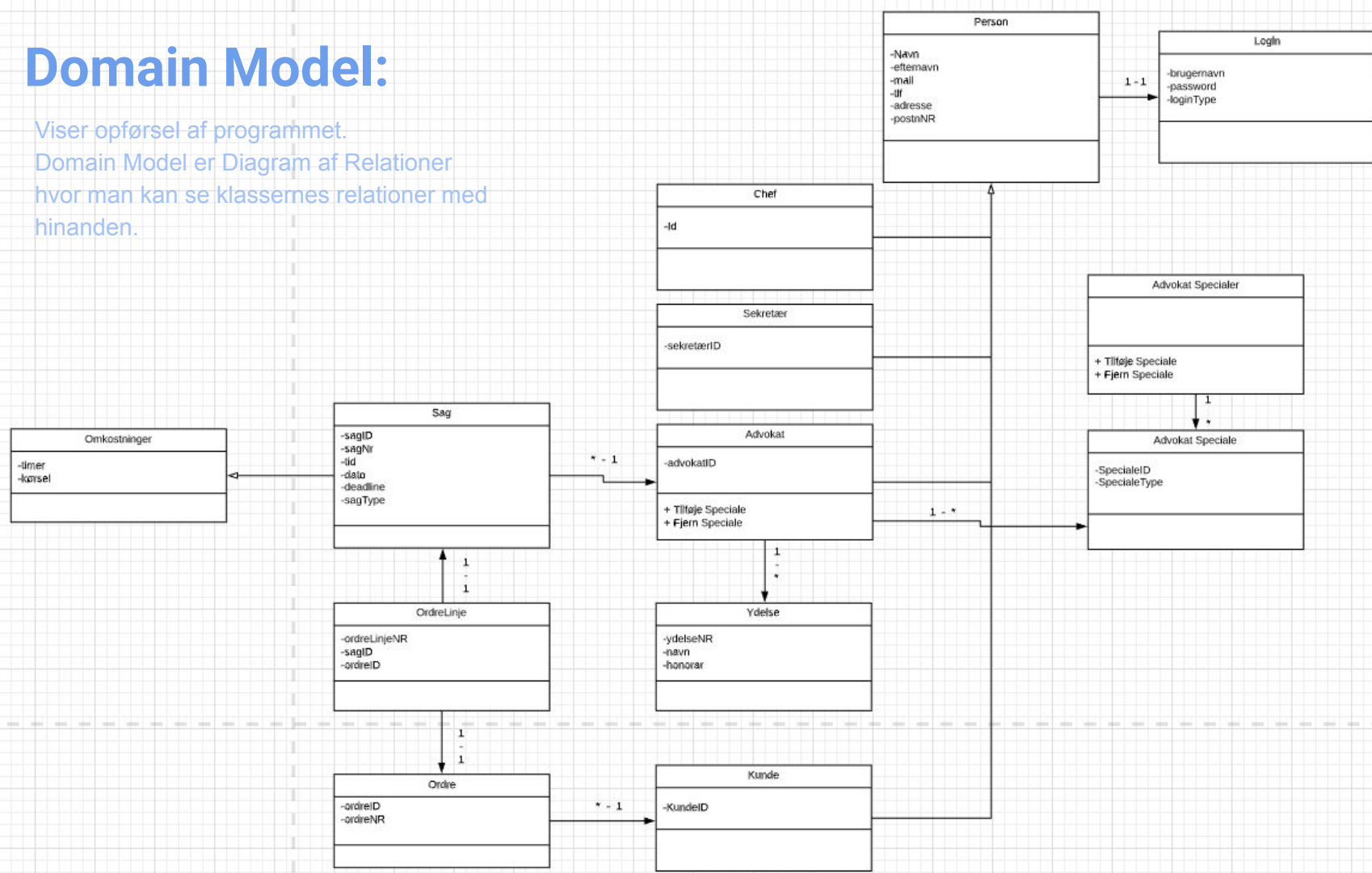




# Domain Model:

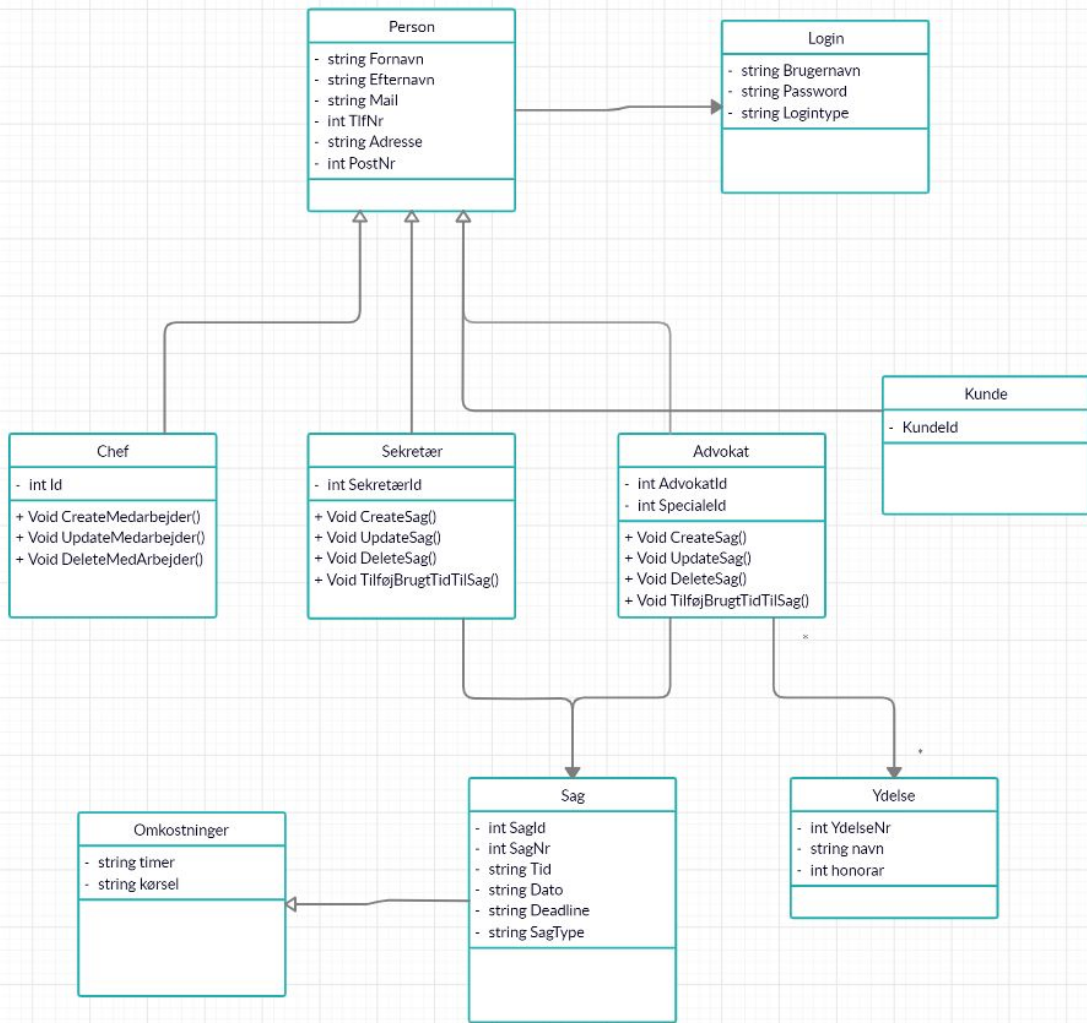
Viser opførsel af programmet.

Domain Model er Diagram af Relationer  
hvor man kan se klassernes relationer med  
hinanden.



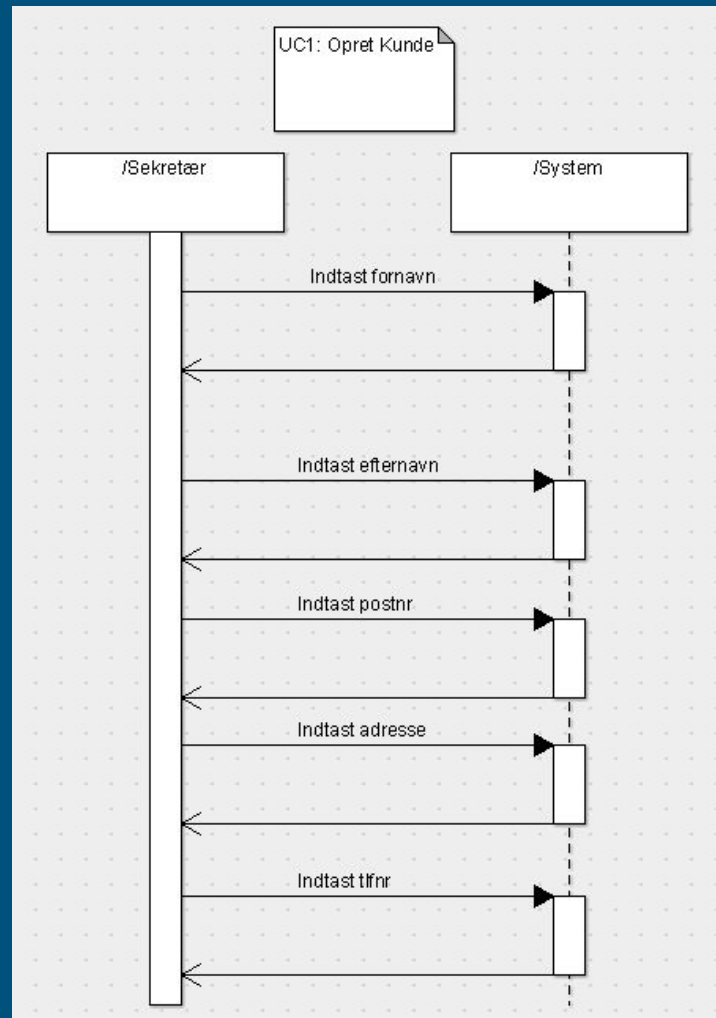
# Class Diagram:

Viser struktur af systemet ved at vise klasser, deres attributter, metoder og relationer mellem klasserne

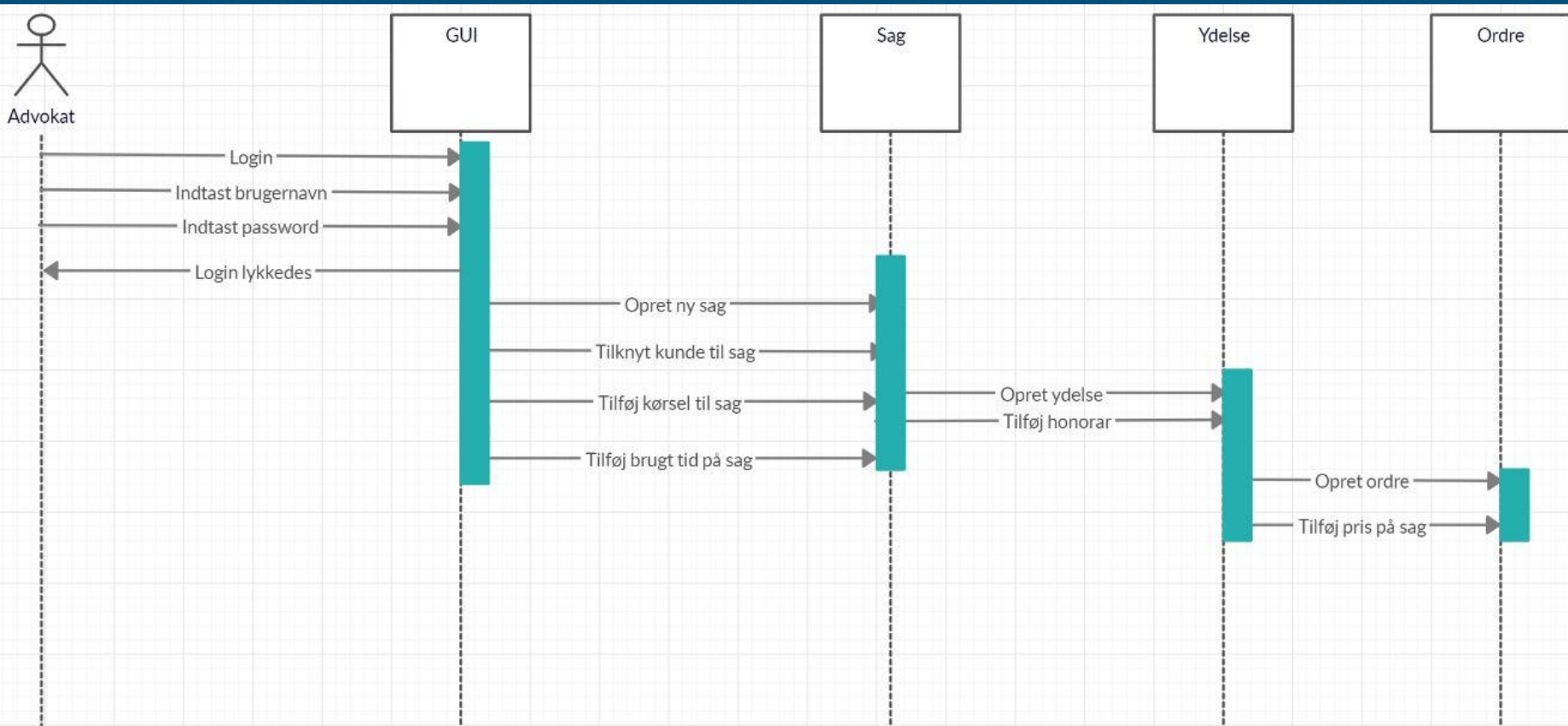


# SSD:

Viser bestemt scenarie af en Use Case

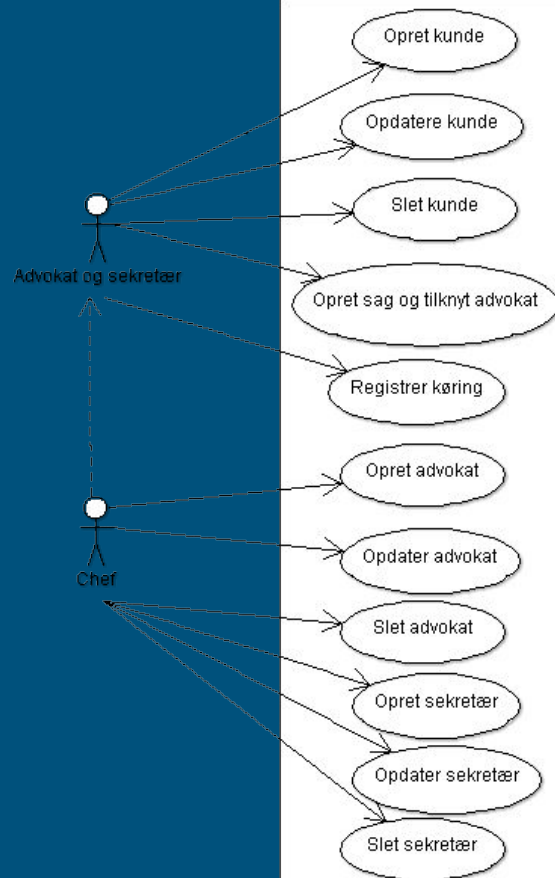


# SD: Viser objekter og klasse som interacter med hinanden



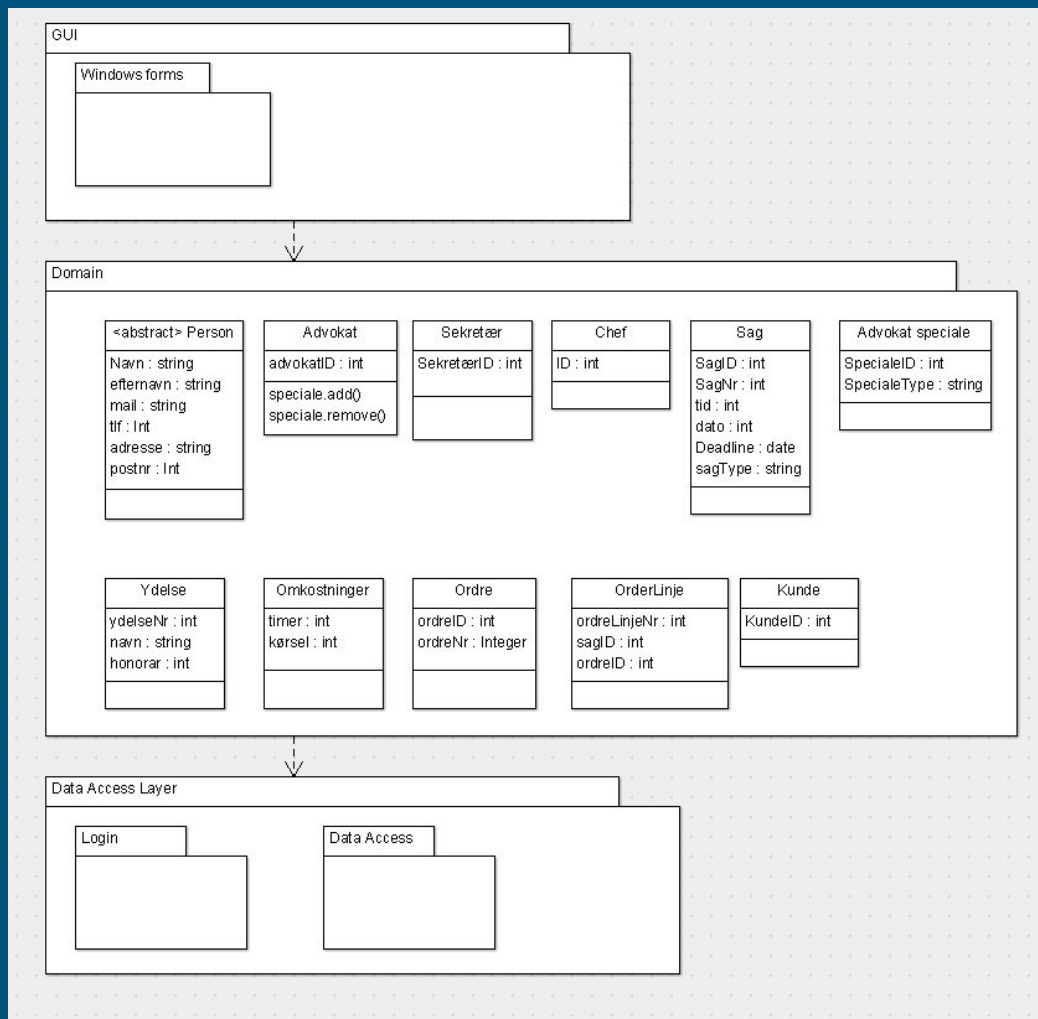
# Use Case Diagram

Use Case Diagram er en repræsentation af brugerens interaktion med systemet.



# Package Diagram:

Viser overblik over arkitekturen



# Arkitektur:

## 1. 3-lags Arkitektur

## 2. Hvert lag har sin Namespace

## 3. Underlaget kender ikke til den ovenfor og den ovenfor kender kun den nedenunder

```
using Domain;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Media;
using System.Reflection;
using System.Runtime.InteropServices;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace View_GUI
{
    4 references | Stefan Biserov Popov, 7 hours ago | 1 author, 8 changes
    public partial class Main_Body_Form : Form
    {
        //-----FORM--Border---TitleBar-----Settings-----SATRT:-----
        // For the Title Bar make it Draggable with The Form so you can drag the form
        public const int WM_NCLBUTTONDOWN = 0xA1;
        [DllImport("User32.dll")]
        1 reference | Stefan Biserov Popov, 2 days ago | 1 author, 1 change
        public static extern bool ReleaseCapture();
        [DllImport("User32.dll")]
        1 reference | Stefan Biserov Popov, 2 days ago | 1 author, 1 change
        public static extern int SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Domain
{
    4 references | Stefan Biserov Popov, 18 days ago | 2 authors, 2 changes
    class Kunde:Person
    {
        2 references | Stefan Biserov Popov, 18 days ago | 2 authors, 2 changes
        public Kunde() //Constructor
        {
        }
    }
}
```

```
namespace DAL
{
    18 references | Stefan Biserov Popov, 18 hours ago | 1 author, 2 changes
    class DB_Connection_String
    {
        17 references | Stefan Biserov Popov, 18 hours ago | 1 author, 2 changes
        public string DBConnectionString { get; set; } = "";

        private static DB_Connection_String ConnString_Instance = null;

        // Constructor is protected
        1 reference | Stefan Biserov Popov, 8 days ago | 1 author, 1 change
        protected DB_Connection_String()
        {
        }

        // Get Singleton Instance
        7 references | Stefan Biserov Popov, 8 days ago | 1 author, 1 change
        public static DB_Connection_String Get_Connection_String_Instance()
        {
            if(ConnString_Instance == null)
            {
                ConnString_Instance = new DB_Connection_String();
            }

            return ConnString_Instance;
        }
    }
}
```

1. GUI - Brugergrænseflade
2. Domain
3. DAL
4. Threads - "Verdens Ur"
5. Singleton
6. Facade
7. Collections



```
1 reference | Stefan Biserov Popov, 11 days ago | 1 author, 2 changes
public Statistics_Facade()
{
    Load_Data = new DB_Loader();
}
```

```
// Get Statistics - MAIN
// reference | Steffen Biserow Popow, 10 days ago | 1 author, 1 change
public void Get_Statistics()
{
    Get_Inkomst_Data_From_DB(); // Get Inkomst
    Get_Me_Salary(); // Get Me_Salary
    Calculate_Profit(); // Calculating
```

```
// Get Data - Inkomst
1 reference | Stefan Blacov Popov, 10 days ago | 1 author, 2 changes
private void Get_Inkomst_Data_From_DB()
{
    Indkomst = Load_Data.PopulateTextbox(Indkomst_Query);
}
```

```
// Me_Salary - "Len"
1 reference | Stefan Biserov Popov, 10 days ago | 1 author, 2 changes
private void Get_Me_Salary()
{
    Me_salary_Time = Load_Data.PopulateTextbox(Me_Salary_Query);
}
```

```
// Calculate Profit
1 reference | Stefan Biserov Popov, 10 days ago | 1 author, 2 changes
private void Calculate_Profit()
{
    if(Me_salary_Time != "" && Indkomst != "")
    {
        Me_Total_Salary = (decimal.Parse(Me_salary_Time) * Hour_salary).ToString();
        Profit = (decimal.Parse(Indkomst) - (decimal.Parse(Me_Total_Salary) + Fees)).ToString();
    }
}
```

```

1 Update Clock - Time
1 reference | Stefan Biserov Popov, 9 days ago | 1 author, 1 change
public void Update_Time()
{
    lock(lock_Obj)
    {
        while(true) // The Thread is Background Thread and will be disposed on Form Close
        {
            string SF_Time = DateTime.Now.AddHours(-9).ToShortTimeString(); // San Francisco
            string KB_Time = DateTime.Now.ToShortTimeString(); // København - Time
            Update_View(SF_Time, KB_Time);
            Thread.Sleep(60000);
        }

        void Update_View(string SF, string KB)
        {
            Clock_Delegate c_delegate = new Clock_Delegate(Update_Clocks);
            Invoke(c_delegate, SF, KB);
        }

        void Update_Clocks(string SF_Clock, string KB_Clock)
        {
            San_Francisco_Time_textBox.Text = SF_Clock; // San Francisco
            KB_clock_textBox.Text = KB_Clock; // København - Time
        }
    }
}

```

```
namespace DAL
{
    18 references | Stefan Biserov Popov, 18 hours ago | 1 author, 2 changes
    class DB_Connection_String
    {
        17 references | Stefan Biserov Popov, 18 hours ago | 1 author, 2 changes
        public string DBConnectionString { get; set; } = "";

        private static DB_Connection_String ConnString_Instance = null;

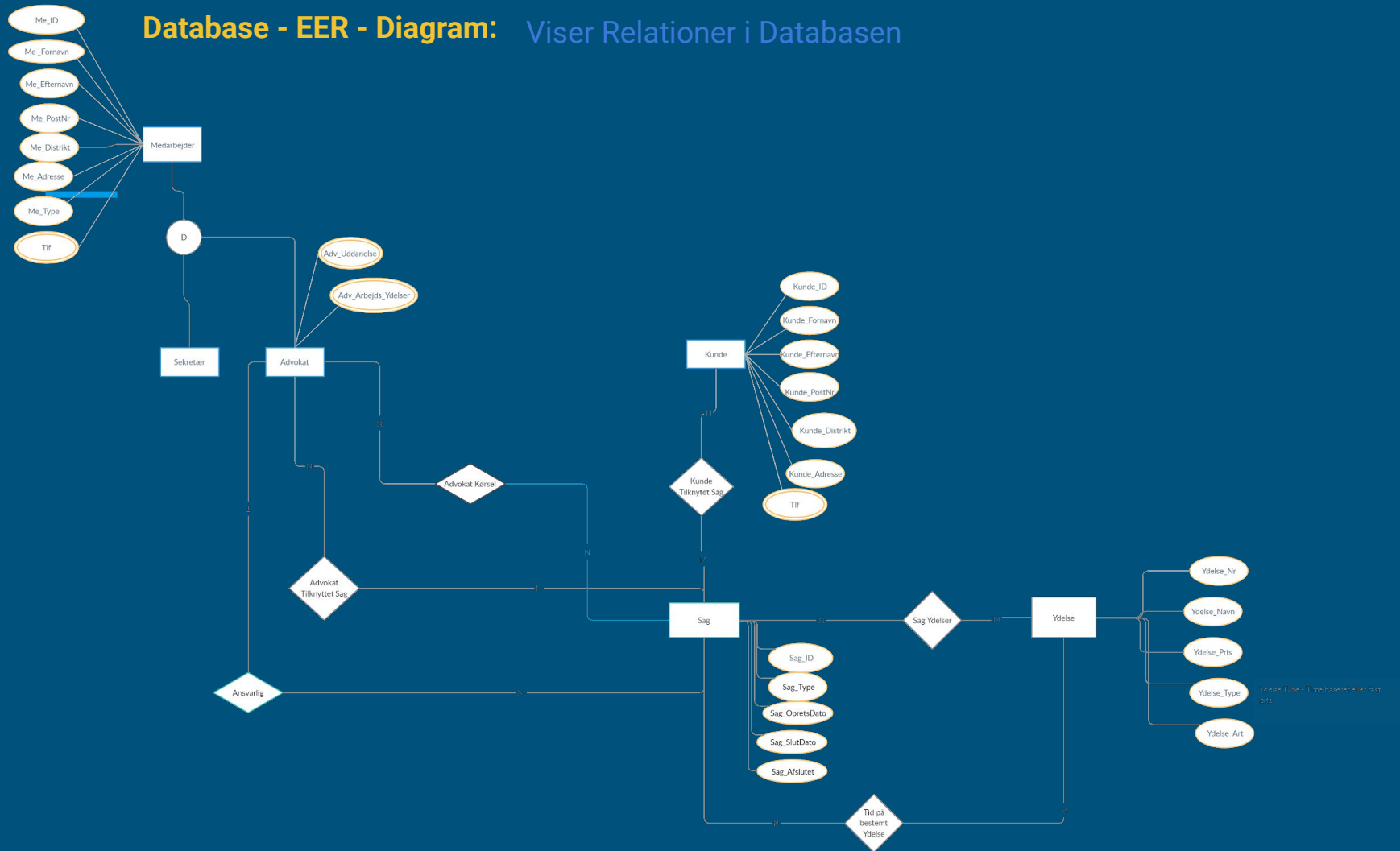
        // Constructor is protected
        1 reference | Stefan Biserov Popov, 8 days ago | 1 author, 1 change
        protected DB_Connection_String()
        {
        }

        // Get Singleton Instance
        7 references | Stefan Biserov Popov, 8 days ago | 1 author, 1 change
        public static DB_Connection_String Get_Connection_String_Instance()
        {
            if(ConnString_Instance == null)
            {
                ConnString_Instance = new DB_Connection_String();
            }

            return ConnString_Instance;
        }
    }
}
```

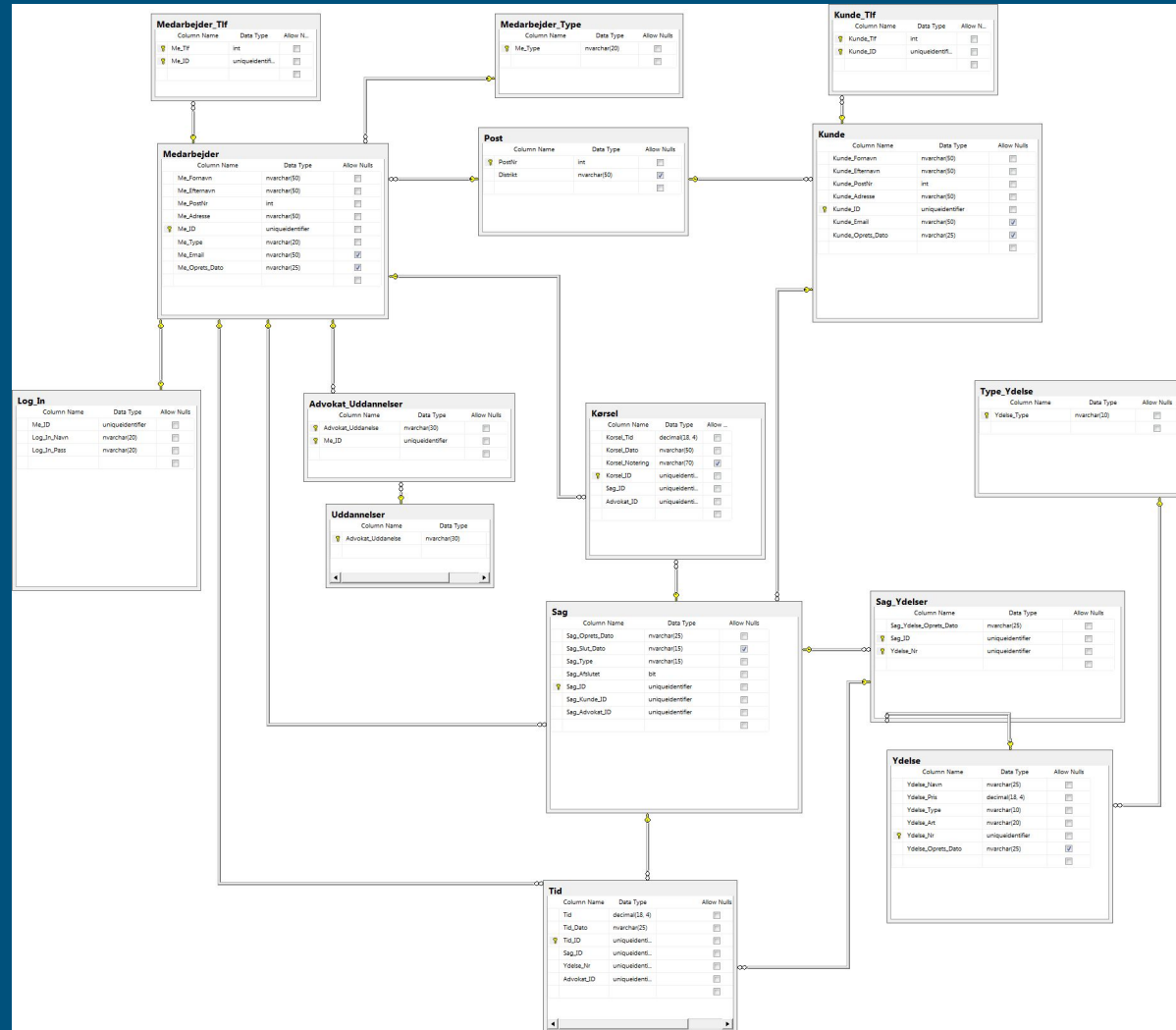


# Database - EER - Diagram: Viser Relationer i Databasen



# Database - Design Diagram:

## Viser selve overblik af Databsen



## Database - SQL:

### Script for at Søge i Sag:

```
"Select S.Sag_Type, S.Sag_ID, S.Sag_Oprets_Dato, S.Sag_Slut_Dato, S.Sag_Afslutet, K.Kunde_Fornavn, K.Kunde_Efternavn, S.Sag_Kunde_ID, M.Me_Fornavn AS Advokat_Fornavn, M.Me_Efternavn AS Advokat_Efternavn, S.Sag_Advokat_ID AS Ansvarlig_Advokat_ID, T.Tid, T.Tid_Dato From Sag As S Full Join Tid AS T On S.Sag_ID = T.Sag_ID Inner Join Kunde As K On S.Sag_Kunde_ID = K.Kunde_ID Inner Join Medarbejder AS M ON S.Sag_Advokat_ID = M.Me_ID Where S.Sag_Type {SearchOptions} OR S.Sag_ID {SearchOptions} OR S.Sag_Oprets_Dato {SearchOptions} OR S.Sag_Slut_Dato {SearchOptions} OR K.Kunde_Fornavn {SearchOptions} OR K.Kunde_Efternavn {SearchOptions} OR S.Sag_Kunde_ID {SearchOptions} OR M.Me_Fornavn {SearchOptions} OR M.Me_Efternavn {SearchOptions} OR S.Sag_Advokat_ID {SearchOptions} OR T.Tid {SearchOptions} OR T.Tid_Dato {SearchOptions};"
```

### Script for at Løbe Sag: - Kan afprøves i Programmet

```
Select S.Sag_Type, S.Sag_ID, S.Sag_Oprets_Dato, S.Sag_Slut_Dato, S.Sag_Afslutet, K.Kunde_Fornavn, K.Kunde_Efternavn, S.Sag_Kunde_ID, M.Me_Fornavn AS Advokat_Fornavn, M.Me_Efternavn AS Advokat_Efternavn, S.Sag_Advokat_ID AS Ansvarlig_Advokat_ID, T.Tid, T.Tid_Dato From Sag As S Full Join Tid AS T On S.Sag_ID = T.Sag_ID Inner Join Kunde As K On S.Sag_Kunde_ID = K.Kunde_ID Inner Join Medarbejder AS M ON S.Sag_Advokat_ID = M.Me_ID
```

If er brugt til at søge i string eller int værdier sådan at der ikke kommer fejl. Da hvis man søger en nvarchar i en int row få man fejl i Programmet da Databasen kan ikke søge en string i int Row.

```
"IF(ISNUMERIC('{search_textBox.Text}') = 0) BEGIN Select K.Kunde_Fornavn, K.Kunde_Efternavn, K.Kunde_PostNr, P.Distrikt, K.Kunde_Adresse, K.Kunde_Email, T.Kunde_Tlf, K.Kunde_ID, K.Kunde_Oprets_Dato From KUNDE AS K FULL JOIN Kunde_Tlf As T ON K.Kunde_ID = T.Kunde_ID Full Join Post AS P ON K.Kunde_PostNr = P.PostNr Where K.Kunde_Fornavn {SearchOptions} OR K.Kunde_Efternavn {SearchOptions} OR P.Distrikt {SearchOptions} OR K.Kunde_Adresse {SearchOptions} OR K.Kunde_Email {SearchOptions} OR K.Kunde_ID {SearchOptions} OR K.Kunde_Oprets_Dato {SearchOptions} END ELSE BEGIN Select K.Kunde_Fornavn, K.Kunde_Efternavn, K.Kunde_PostNr, P.Distrikt, K.Kunde_Adresse, K.Kunde_Email, T.Kunde_Tlf, K.Kunde_ID, K.Kunde_Oprets_Dato From KUNDE AS K FULL JOIN Kunde_Tlf As T ON K.Kunde_ID = T.Kunde_ID Full Join Post AS P ON K.Kunde_PostNr = P.PostNr Where K.Kunde_Fornavn {SearchOptions} OR K.Kunde_Efternavn {SearchOptions} OR K.Kunde_PostNr {SearchOptions} OR P.Distrikt {SearchOptions} OR K.Kunde_Adresse {SearchOptions} OR K.Kunde_Email {SearchOptions} OR T.Kunde_Tlf {SearchOptions} OR K.Kunde_ID {SearchOptions} OR K.Kunde_Oprets_Dato {SearchOptions} END;"
```

# Udvikling:

---

1. Anvende Planen
2. Lave Use Cases / Diagrammer
3. Forstår hvad der skal til at få de Krav implementeret
4. Starte på at Implementer
5. Teste undervejs
6. Source Control "Github"
7. Følge den bestemt Arkitektur
8. Finde på nogle nye funktioner "Snakke med kunden om det"
9. Gøre programmet brugervenlig
10. Prototyper
11. Dokumenter
12. Opdater Dokumentation undervejs
13. Følge planen og strategien som var valgt i starten