



Programmeren in C# met SQL

Realisatie

Stage

Ruben Boone 3APP02

Academiejaar 2021-2022

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

INHOUDSTAFEL

1	INTRODUCTIE	6
2	BEDRIJFSSTRUCTUUR	7
2.1	3-tier	7
2.1.1	Data-laag	7
2.1.2	Applicatie- of Business-laag	7
2.1.3	Presentatielaag	7
2.1.4	Helper klassen	7
3	OPDRACHTEN	8
3.1	Economaat Etiketten	8
3.1.1	Beschrijving	8
3.1.2	Doelstelling	8
3.1.3	Onderzoek	9
3.1.4	Realisatie	9
3.2	Spoed Etiketten	11
3.2.1	Beschrijving	11
3.2.2	Doelstelling	12
3.2.3	Onderzoek	12
3.2.4	Realisatie	12
3.3	Spoel registratie	12
3.3.1	Beschrijving	12
3.3.2	Doelstelling	13
3.3.3	Realisatie	13
3.3.3.1	Stappenplan	13
3.3.4	Configuratie deel	15
3.3.4.1	Data-laag	15
3.3.4.2	Applicatielaag	15
3.3.4.3	Presentatielaag	16
3.3.5	Data-in deel	18
3.3.5.1	Data-laag	18
3.3.5.2	Applicatielaag	19
3.3.5.3	Presentatielaag	20
3.3.6	Data-out	20
3.3.6.1	Data-laag	21
3.3.6.2	Applicatielaag	21
3.3.6.3	Presentatie laag	22
3.3.7	Excel maken	22
3.3.8	Extra label	23
3.3.9	Helper klassen	24
3.3.10	Uitbrengen van de software	25
3.4	Extra opdrachten	25
3.4.1	QR-code generator	25
3.4.2	Datatype aanpassing	26
3.4.3	Importeren van de stagiair Excel	26
3.4.4	Analyse oefeningen	30

4	SOFT SKILLS.....	31
4.1	Communicatie.....	31
4.2	Planning	32

1 INTRODUCTIE

In dit document ga ik bewijzen wat ik allemaal geleerd en gecreëerd heb tijdens mijn stage in het ziekenhuis in Geel. Nu denk je misschien, waarom in een ziekenhuis? Daar heb ik voor gekozen door 2 redenen. De eerste reden is omdat het ziekenhuis in Geel een stage in de aanbieding had in het programmeren in C# en SQL. Mijn persoonlijke voorkeur ligt bij het creëren van applicaties in C# , Java, Dit wil niet zeggen dat ik niet graag bezig ben met web gebaseerde applicaties want met ASP heb je ook de mogelijkheid om hierin te werken. De tweede reden was vooral omdat ik ook eens achter de schermen van een ziekenhuis wilde kijken. Hoe gaat het daar allemaal in zijn werking, hoeveel heeft IT te maken met de huidige gang van zaken in een ziekenhuis. Graag wil ik mijn stage mentor Dhr. Stefan de Wilde en stage begeleider Hajar Ghaem Sigarchian bedanken voor de geweldige ondersteuning en tips tijdens mijn stage en het ziekenhuis om mij de mogelijkheid te geven om mijn stage bij hun mogelijk te kunnen maken.

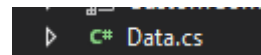
2 BEDRIJFSSTRUCTUUR

2.1 3-TIER

De applicaties in het ziekenhuis worden opgebouwd met behulp van een "3-tier" structuur. Er is dus een data-laag, een applicatie/business-laag en een presentatielaag.

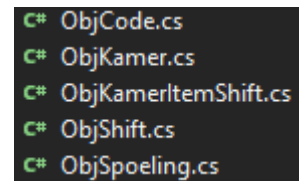
2.1.1 Data-laag

In de data-laag staan alle SQL query's van de applicatie. Dit is een statische klassen in de applicatie die overal kan opgeroepen worden.



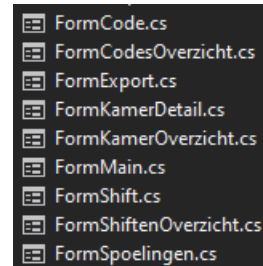
2.1.2 Applicatie- of Business-laag

In deze klassen worden objecten gemaakt. Deze bevatten alle eigenschappen van een object en methodes die nodig zijn. Hier zijn ook de verschillende methodes aanwezig om het object op te vullen, op te slaan in de database, aan te passen in de database en te verwijderen.



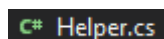
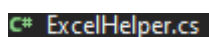
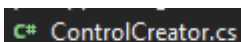
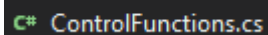
2.1.3 Presentatielaag

In deze laag worden alle schermpjes voorzien, dit is de visuele laag waar alle knoppen, keuze lijsten, invul velden aanwezig zijn. Deze laag bevat zeer weinig zelf geschreven code.



2.1.4 Helper klassen

Dit zit niet echt in het 3-tier systeem maar wordt wel heel veel gebruikt. Een voorbeeld van een methode in zo een helper klasse is het opvullen van een keuze lijstje.

3 OPDRACHTEN

3.1 ECONOMAAT ETIKETTEN

3.1.1 Beschrijving

Dit is een bestaande applicatie die een Excel bestand inleest om etiketten te printen voor in het magazijn te plakken bij de bijbehorende artikels. Op zo een etiket staat het artikel nummer, het aantal van een minimum bestelling, een beschrijving, een referentie nummer, een barcode en tot slot de datum en tijd van het aanmaken van het etiket. Zodra de barcode wordt gescand zal dit artikel automatisch klaargezet worden om besteld te worden.



3.1.2 Doelstelling

Het huidige print proces is niet ideaal. Zodra de gegevens worden ingelezen worden deze met behulp van mail referenties in een Word template geplaatst, dit Word document wordt dan afgedrukt. Het probleem hier is dat zodra de printer klaar is met printen dat het Word proces niet altijd correct afgesloten wordt. Als er veel etiketten worden afgedrukt kan het zijn dat er veel Word processen open blijven staan en dat de computer traag en op lange termijn zelfs uit valt.

Dit geeft ons de volgende doelstellingen:

1. Zoek een manier om tot een etiket lay-out te komen zonder Word te gebruiken.
2. Zoek een manier om dit etiket vervolgens af te drukken.
3. De barcode zal op een nieuwe manier moeten gegenereerd worden.

3.1.3 Onderzoek

Na wat onderzoek werk en de al reeds opgeleverde informatie die de mentor had gevonden ben ik tot de volgende oplossing gekomen.

Word kan vervangen worden door de C# klasse "PrintDocument". Dit gag mij meteen de mogelijkheid om een PrintPage te gebruiken, zo een printPage komt overeen met de pagina waar je normaal de tekst schrijft.

De Barcode kan gegenereerd worden met behulp van een Nuget pakket.

3.1.4 Realisatie

Om het netjes te houden is er gekozen om een statische klasse genaamd "Etiket" te maken waar alles wat te maken heeft met het etiket in te plaatsen.

In de nieuwe klasse is de methode "PrintEconomaat" aanwezig. Deze vraagt een "KeyValuePair" object als parameter zodat we alle benodigde data die op het etiket moet ter beschikking hebben in de klasse. Zo vullen we ook meteen de barcode en de datum met tijd in. Tot slot roepen we dan de "PrintEtiket()" -methode op om verder te gaan.

```
public static void PrintEconomaat(List<KeyValuePair<string, string>> fields)
{
    Zen.Barcode.Code128BarcodeDraw barcode = Zen.Barcode.BarcodeDrawFactory.Code128WithChecksum;

    barcodeCijfers = fields.FirstOrDefault(key => key.Key == "BARCODETEKST").Value.ToString();
    barcodeImg = barcode.Draw(barcodeCijfers, 32);
    artikellabel = fields.FirstOrDefault(key => key.Key == "ARTIKEL").Value.ToString();
    aantallabel = fields.FirstOrDefault(key => key.Key == "AANTAL").Value.ToString();
    beschrijving = fields.FirstOrDefault(key => key.Key == "ARTIKELOMSCHRIJVING").Value.ToString();
    referentielabel = fields.FirstOrDefault(key => key.Key == "REFERENTIE").Value.ToString();
    systime = DateTime.Now.ToString("MM/dd/yyyy HH:mm");

    PrintEtiket();
}
```

Hier wordt het "PrintDocument" aangemaakt, gevolgd door het aanmaken en opvullen van de "PrintPage". Omdat we niet echt in een document werken en eigenlijk aan het tekenen zijn op een blad moeten we alles dus met "Graphics" tekenen. Nagedacht over eventueel latere aanpassingen zijn er verschillende constante variabelen gemaakt zodat er later maar op 1 plek iets moet aangepast worden en niet gezocht moet worden waar alles staat.


```
private static void CreateTitel(object sender, PrintPageEventArgs printPageEventArgs)
{
    printPageEventArgs.Graphics.DrawString(artikelLabel, DefaultFontBold, textColor, Lijn1ArtikelX, Lijn1Y);
    printPageEventArgs.Graphics.DrawString(aantalLabel, DefaultFont, textColor, Lijn1AantalX, Lijn1Y);

    if (beschrijving.Length > maxCharakterOnLine)
    {
        printPageEventArgs.Graphics.DrawString(beschrijving.Substring(0, maxCharakterOnLine), DefaultFont, textColor, Lijn2BeschrijvingX, Lijn2BeschrijvingY);
        printPageEventArgs.Graphics.DrawString(beschrijving.Substring(maxCharakterOnLine), DefaultFont, textColor, Lijn2BeschrijvingX, Lijn2BeschrijvingY2);
    }
    else
    {
        printPageEventArgs.Graphics.DrawString(beschrijving, DefaultFont, textColor, Lijn2BeschrijvingX, Lijn2BeschrijvingY);
    }

    printPageEventArgs.Graphics.DrawString(referentieLabel, DefaultFont, textColor, Lijn3ReferentieX, Lijn3ReferentieY);

    printPageEventArgs.Graphics.DrawImage(barcodeTag, Lijn4BarcodeX, Lijn4BarcodeY);

    printPageEventArgs.Graphics.DrawString(barcodeTijfers, DefaultFont, textColor, Lijn5BarcodeTekstX, Lijn5BarcodeTekstY);
    printPageEventArgs.Graphics.DrawString(sysTime, DefaultFontSmall, textColor, SysTimeX, SysTimeY);
}
```

```
private static readonly Font DefaultFont = new Font("Arial", 7);
private static readonly Font DefaultFontBold = new Font("Arial", 7, FontStyle.Bold);
private static readonly Font DefaultFontSmall = new Font("Arial", 4, FontStyle.Bold);

private static readonly SolidBrush textColor = new SolidBrush(Color.Black);

private static readonly int maxCharakterOnLine = 34;

private const int StartPunt = 110;

private const int Lijn1ArtikelX = StartPunt;
private const int Lijn1AantalX = StartPunt + 80;
private const int Lijn1Y = 0;

private const int Lijn2BeschrijvingX = StartPunt;
private const int Lijn2BeschrijvingY = Lijn1Y + 10;
private const int Lijn2BeschrijvingY2 = Lijn2BeschrijvingY + 10;

private const int Lijn3ReferentieX = StartPunt;
private const int Lijn3ReferentieY = Lijn2BeschrijvingY2 + 10;

private const int Lijn4BarcodeX = StartPunt + 35;
private const int Lijn4BarcodeY = Lijn3ReferentieY + 12;

private const int Lijn5BarcodeTekstX = StartPunt + 55;
private const int Lijn5BarcodeTekstY = Lijn4BarcodeY + 40;

private const int SysTimeX = StartPunt + 135;
private const int SysTimeY = Lijn4BarcodeY + 58;
```

Tot slot moet er nog een printer gekozen worden waar het document afgerukt moet worden. Om weer aan de toekomst te denken is er nagedacht om dit te doen met de "App.config" zodat niet heel de applicatie opnieuw moet worden uitgebracht en eventueel op een andere locatie met een andere printer kan gewerkt worden. Dit geeft ons ook de mogelijkheid om meerdere printers mee te geven en de gebruiker dus kunnen laten kiezen welke printer we gaan gebruiken.

```

private static string GetSelectedPrinter(PrintDocument pDoc)
{
    string printNames = ConfigurationManager.AppSettings["Printers"].ToString();
    string[] printers = printNames.Split(';');

    string selectedPrinter = string.Empty;
    if (printers.Length == 1)
    {
        selectedPrinter = printNames; //Of printers[0] was ook ok
    }
    else
    {
        //Meerder printers in config wil zeggen dat we die negeren.
        //User dient printer zelf te kiezen
        //(kan andere zijn dan in de config staat).
        PrintDialog pd = new PrintDialog();
        pd.Document = pDoc;
        DialogResult dr = pd.ShowDialog();
        if (dr == DialogResult.OK)
        {
            selectedPrinter = pd.PrinterSettings.PrinterName;
        }
    }

    return selectedPrinter;
}

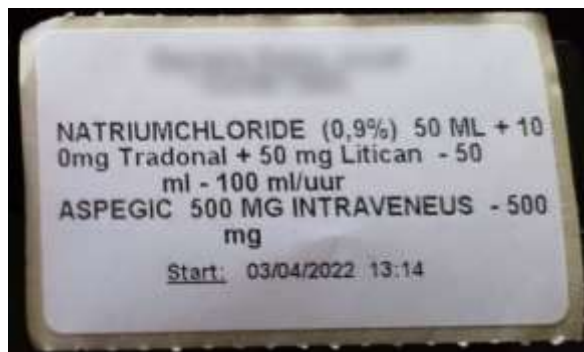
```

3.2 SPOED ETIKETTEN

3.2.1 Beschrijving

Voor de spoeddienst moet men etiketten afdrukken om op toegediende medicatie te plakken. Dit is verplicht vanuit de Belgische wetgeving.

Dit is een ook een bestaande applicatie maar deze zit iets anders in elkaar. Hier komt de data via externe software genaamd ChipSoft – HiX. HiX zorgt kort samengevat voor informatie van elektronische patiënten dossiers. De geselecteerde patiënt ID en dossier nummer wordt dan op aanvraag doorgestuurd naar een ASPX-webpagina met behulp van argumenten in de URL. Daar word dan alle extra informatie opvragen met behulp van de ID. Al de informatie die dan op het etiket moet word in een tekstbestand geplaatst en ingelezen door in huis gemaakte software. Deze data komt dan net zoals de Economaat etiketten via mail verwijzingen terecht in een Word template om vervolgens afgedrukt te worden.



3.2.2 Doelstelling

Net zoals bij de economaat etiketten moet de manier van printen en het gebruik van Word aangepast worden. In deze opdracht moet er geen barcode voorzien worden op het etiket.

3.2.3 Onderzoek

Omdat deze opgave bijna identiek is met de vorige opdracht kunnen we dus wat code hergebruiken in deze toepassing. Extra onderzoek moet dus niet gebeuren.

3.2.4 Realisatie

Ook hier is er een Etiket klasse voorzien. De data wordt dus ook weer in een "KeyValuePair" naar de klasse gegeven, het enige echte verschil tussen deze twee opdrachten is nu alleen maar lay-out van de etiketten en de constanten. Het printen gebeurt dus net hetzelfde. De constanten zijn hier in de App.config geplaatst omdat anders bij elke aanpassing in het etiket de webpagina opnieuw uitgebracht moet worden en dat is niet ideaal in een spoed afdeling.

```
private static void CreateEtiketSpoed(object sender, PrintPageEventArgs printPageEventArgs)
{
    printPageEventArgs.Graphics.DrawString(CenterText(patientNaam), DefaultFont, textColor, NaamX, NaamY);
    printPageEventArgs.Graphics.DrawString(CenterText(patientGebDatum), DefaultFont, textColor, GeboorteDatumX, GeboorteDatumY);

    extraPosition = startHoogteMedicatie;
    UseExtraLine(printPageEventArgs, hoofdProduct);

    if (bijkomendProduct != " ")
    {
        UseExtraLine(printPageEventArgs, bijkomendProduct);
    }

    extraPosition -= margeSysTime;

    printPageEventArgs.Graphics.DrawString("Start: ", DefaultFontBoldUnderline, textColor, StartX, ProdY - extraPosition);
    printPageEventArgs.Graphics.DrawString(systime, DefaultFontSmallBold, textColor, TimeX, ProdY - extraPosition);
}
```

3.3 SPOEL REGISTRATIE

3.3.1 Beschrijving

In het ziekenhuis zijn verschillende items die gespoeld moeten worden. Zo een item kan een douche, wasbak, toilet, Deze items kunnen worden gespoeld door een verschillende uitvoerders. Er zijn momenteel 3 verschillende soorten uitvoerders. De verpleging, de technische dienst en uiteraard ook de poetsploeg. Nu is er gevraagd of er geen applicatie kan gemaakt kan worden om dit proces te verbeteren. Momenteel word alles genoteerd op papier en dit vraagt enorm veel tijd.

3.3.2 Doelstelling

Er moeten een paar dingen voorzien worden.

1. Een configuratie deel in de applicatie zodat er kan ingevoerd worden welke items er zijn, welke kamers er zijn, welke shiften er zijn en welke shift een bepaald item in welke kamer moet spoelen.
2. Een manier voor de uitvoerders om een item te markeren als voltooid.
 - Omdat de poetsploeg geen toegang heeft tot een laptop zullen ze dit op een andere manier moeten aanduiden. Speciaal voor hen voorzien wij een webpagina waar een item als voltooid kan gemarkeerd worden.
3. Een manier om alle verzamelde data te exporteren naar een Excel bestand. De data die geëxporteerd kan worden kan gefilterd worden op verschillende manieren zodat je specifieke tabellen kan genereren

3.3.3 Realisatie

Om te beginnen is er eerst een stappenplan gemaakt. Op deze manier kan er gestructureerd gewerkt worden aan de applicatie. Om op een logische manier te beginnen starten we met het configuratie deel.

3.3.3.1 *Stappenplan*

Databank

1. Tabellen in orde brengen
 - a. Kamers tabel
 - a. KamersItems tabel
 - a. KamersItemsShiften tabel

Data-Laag

1. Query's schrijven
 - a. Kamer
 - i.SELECT kamer
 - i.SELECT kamers
 - i.INSERT kamer
 - i.UPDATE kamer
 - i.DEL kamer
 - a. KamerItem
 - i.SELECT KamerItems
 - i.INSERT KamerItem
 - i.DELETE KamerItem
 - i.(optioneel) UPDATE KamerItem
 - a. KamerItemShift
 - i.SELECT KamerItemShift
 - i.INSERT KamerItemShift
 - i.DELETE KamerItemShift
 - i.(optioneel) UPDATE KamerItemShift

- a. Algemene query's (keuzelijsten)
 - i.SELECT Items
 - i.SELECT Spoelplaatsen
 - i.SELECT Shiften
 - i.SELECT SpoelUitvoerdersShiften (uitvoerderID vanuit AD)

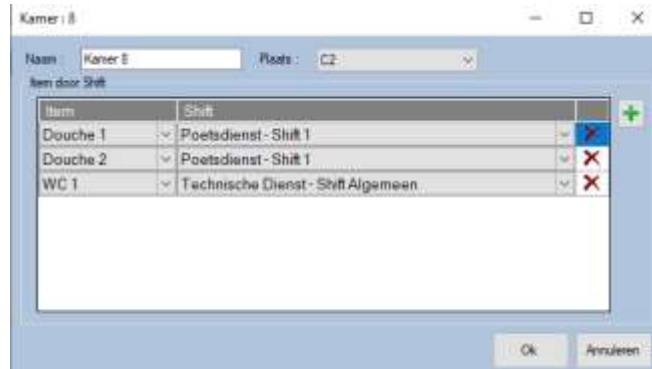
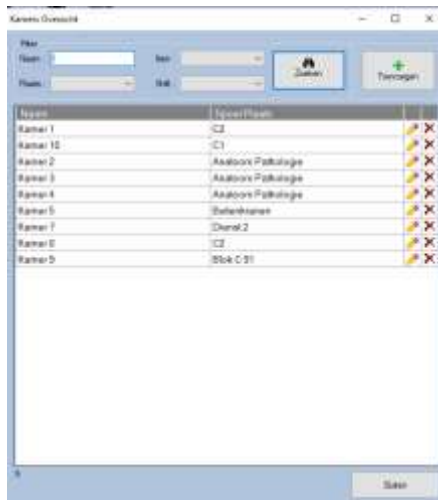
Applicatie/Business-laag

- 1. Objecten aanmaken
 - a. SpoelPlaats
 - i.Properties
 - ID
 - Naam
 - a. Item
 - i.Properties
 - ID
 - Naam
 - a. Kamer
 - i.Properties
 - ID
 - Naam
 - Spoelplaats - Spoelplaats
 - Items - List<Item>
 - i.Methodes
 - Load - void
 - Save - void
 - Delete - void
 - IsValid - bool (out errormsg)
 - a. KamerItemShift
 - i.Properties
 - ID - int
 - KamerID - int
 - ItemID - int
 - ShiftID - int

Presentatielaag

- 1. Algemeen
 - a. Ribbon
 - i.Config tab
- 1. Kamer overzicht (config)
 - a. Insert knop
 - a. Filters
 - i.Spoelplaats
 - i.Kamer
 - i.Item
 - i.Shift
 - i.Uitvoerder
 - a. Datagrid
 - i.Kamernaam
 - i.Delete knop
 - i.Detail knop
- 1. Kamer Details (config)
 - a. Kamer naam
 - i.Label
 - i.Textbox
 - a. Spoelplaats
 - i.Label
 - i.Dropdown met spoelplaatsen
 - a. Datagrid
 - i.Items (dropdown)
 - i.Shift (dropdown)
 - a. Add ItemShift Button
 - i.Button
 - a. Delete ItemShift Button
 - i.Button

3.3.4 Configuratie deel



3.3.4.1 Data-laag

We starten met het in orde maken van de database. Zonder correcte tabellen kunnen we moeilijk verder. Zodra de database in orde is kunnen we beginnen aan ons 3-tier systeem. De eerste stap is om onze database query's te schrijven zodat we onze data kunnen ophalen, aanmaken, aanpassen of verwijderen.

+	db	dbo.KamerItemShift
+	db	dbo.Kamers
+	db	dbo.Spoelingen
+	db	dbo.SpoelItems
+	db	dbo.SpoelPlaatsen
+	db	dbo.SpoelUitvoerders
+	db	dbo.SpoelUitvoerdersShiften

3.3.4.2 Applicatielaag

Na de data-laag worden de object klassen aangemaakt ook wel de applicatie- of de business-laag genoemd. In de objecten houden we de eigenschappen bij, bv. Het object "Kamer" heeft de eigenschap "Naam", "ID", Zo is er ook een "Save" methode om een instantie van dit object op te slaan in de database of aan te passen, een "Delete" om een instantie te verwijderen. Er wordt ook een "IsValid" methode voorzien om te kijken of de eigenschappen voldoen aan de vooropgestelde eisen. Als dit niet zo is mag er niet opgeslagen of aangepast worden.

```

public class ObjKamer
{
    14 references
    public int? Id { get; set; }
    7 references
    public string Naam { get; set; }
    6 references
    public int SpoelPlaatsId { get; set; }

    public List<ObjKamerItemShift> KamerItemShiften = new List<ObjKamerItemShift>();

    1 reference
    public ObjKamer(int? id, string naam, int spoelPlaatsId)
    {
        Id = id;
        Naam = naam;
        SpoelPlaatsId = spoelPlaatsId;
    }
}

```

```

1 reference
public bool Load()
{
    bool succeeded = false;

    try
    {
        DataRow dr = Data.GetKamer((int)Id);
        Naam = dr["Naam"].ToString().Trim();
        SpoelPlaatsId = (int)dr["spoelplaatsid"];
    }
    catch (Exception error)
    {
        string errorMessage = error.Message;
    }

    return succeeded;
}

```

```

public bool IsValid(out string error)
{
    bool valid = true;
    error = string.Empty;

    if (Naam == "" || Naam == null)
    {
        valid = false;
        error += "- Het veld \"Naam\" is niet ingevuld.\n";
    }

    if (SpoelPlaatsId == 0)
    {
        valid = false;
        error += "- Er is geen spoelplaats geselecteerd.\n";
    }

    string errorText = string.Empty;
    foreach (ObjKamerItemShift kamerItemShift in KamerItemShiften)
    {
        kamerItemShift.IsValid(out errorText);
    }
    error += errorText;

    return valid;
}

```

3.3.4.3 Presentatielaag

Tot slot eindigen we met de presentatielaag, dit zijn onze schermen met onze knoppen, keuzelijsten, checkboxes, Deze controls bevatten allemaal hun eigen events die nauwelijks echte code bevatten maar eerder verwijzen naar een methode op een andere plaats. Zo zijn er verschillende "Helper"-klassen die bevoordeeld, een combo box opvullen.

Een presentatielaag bevat meestal ook dezelfde methodes. Zo hebben we in het event wanneer het venster geladen word onze "Init"-methodes.

Deze methodes worden maar 1 keer uitgevoerd, zo wordt bevoordeeld de datagrid klaargemaakt. Daarmee wordt bedoeld dat kolommen die niet zichtbaar moeten zijn onzichtbaar gemaakt. Nog een belangrijke methode is de "InitControls"-methode, deze methode wordt vaker gebruikt. Elke keer als er op een knop gedrukt word kan deze aangeroepen worden. Deze methode staat namelijk voor het juist zetten van alle controls in het venster. Een klein voorbeeld hiervan is dat de kamer keuzelijst niet mag gebruikt worden zolang er geen shift gekozen is.

```

public void InitForm()
{
    InitFormText();
    InitComboBoxes();

    InitFilterDefaults();
    kamerDataGridView.Columns.Clear();
}

```

```

private void InitComboBoxes()
{
    bool addEmptyRow = true;
    bool useSpoelUitvoerID = false;

    ControlCreator.CreateSpoelPlaatsenComboBox(plaatsComboBox, addEmptyRow);
    ControlCreator.CreateSpoelItemsComboBox(itemComboBox, addEmptyRow);
    ControlCreator.CreateShiftComboBox(shiftComboBox, addEmptyRow, useSpoelUitvoerID);
}

```

```

// Add new kamer
1 reference
private void insertButton_Click(object sender, EventArgs e)
{
    FormKamerDetail kamerDetail = new FormKamerDetail();
    kamerDetail.ShowDialog();

    FillDataGridViewOverzicht();
}

// Search with new filters
1 reference
private void zoekButton_Click(object sender, EventArgs e)
{
    FillDataGridViewOverzicht();
}

// Exit
1 reference
private void buttonSluiten_Click(object sender, EventArgs e)
{
    Close();
}

```

Zo heeft elke methode zijn eigen functionaliteit waardoor deze overall opnieuw kunnen gebruikt worden en dus ook maar 1 ding uitvoeren.

```

// Add the data grid
4 references
private void FillDataGridViewOverzicht()
{
}

```

Deze stappen komen ook terug bij de Data-in en de export. Veel van de methodes kunnen vaak overgenomen worden mits kleine aanpassingen.

3.3.5 Data-in deel

The first screenshot shows the 'Spoelingen' window with a date of 'dinsdag 10 mei 2022' and a shift of 'Poetsdienst - Shift 1'. The 'Kamer' is set to 'Kamer 8'. The 'Te spoelen items' list is empty.

The second screenshot shows the same window with the 'Te spoelen items' list populated with three items: 'Douches 1', 'WC 1', and 'Lavelin'. The 'Douches 1' item is selected.

The third screenshot shows the 'Spoelingen' window with the 'Te spoelen items' list populated with three items: 'Douches 1', 'Douches 2', 'WC 1', and 'Lavelin'. The 'Douches 1' item is selected.

Below the screenshots are two calendar views for May 2022. The left calendar shows the date '10' selected. The right calendar shows the date '10' selected and the 'Shift' set to 'Poetsdienst - Shift 1' and 'Kamer' set to 'Kamer 8'. Below the calendars is a table with the following data:

	Item	Gespeeld
Wijzigen	Douches 1	<input type="checkbox"/>
Opslaan	Douches 2	<input checked="" type="checkbox"/>

Item is nog niet opgeslagen

Het data-in deel bestaat uit 2 delen. De Windows form applicatie en de ASP webpagina. Omdat niet alle functionaliteiten werken of aanwezig zijn in de webpagina moeten sommige dingen anders gebeuren.

3.3.5.1 Data-laag

De WHERE in de SQL query bevat 3 stukken. De datum van de spoeling, welke shift de gebruiker is en in welke kamer de gebruiker zich bevind om de items die gespoeld moeten worden op te vragen.

Als er nog geen spoeling gedaan is dan toont de applicatie de geconfigureerde items. Zodra er een items met de juiste kamer, shift en datum combinatie gevonden worden zullen deze getoond worden.

```

public class Spoeling : ISpoelingRepository
{
    public Spoeling()
    {
        // ...
    }

    public async Task<IEnumerable<Spoeling>> GetAll()
    {
        string sql = @"
            SELECT * FROM Spoeling
            WHERE 1=1
            AND ( @ID IS NULL OR ID = @ID )
            AND ( @KamerID IS NULL OR KamerID = @KamerID )
            AND ( @ItemID IS NULL OR ItemID = @ItemID )
            AND ( @ShiftID IS NULL OR ShiftID = @ShiftID )
            AND ( @Gespoeld IS NULL OR Gespoeld = @Gespoeld )
            AND ( @Datum IS NULL OR Datum = @Datum )
            AND ( @KamerID IS NULL OR KamerID = @KamerID )
            AND ( @ItemID IS NULL OR ItemID = @ItemID )
            AND ( @ShiftID IS NULL OR ShiftID = @ShiftID )
            AND ( @Gespoeld IS NULL OR Gespoeld = @Gespoeld )
            AND ( @Datum IS NULL OR Datum = @Datum )
        ";

        return await _dbContext.Spoeling.FromSqlRaw(sql).ToListAsync();
    }

    public async Task<Spoeling> GetById(int id)
    {
        return await _dbContext.Spoeling.FirstOrDefaultAsync(s => s.ID == id);
    }

    public async Task<Spoeling> GetById(int id, int kamerID, int itemID, int shiftID, bool gespoeld, DateTime datum)
    {
        return await _dbContext.Spoeling.FirstOrDefaultAsync(s => s.ID == id && s.KamerID == kamerID && s.ItemID == itemID && s.ShiftID == shiftID && s.Gespoeld == gespoeld && s.Datum == datum);
    }

    public async Task<Spoeling> Save(Spoeling spoeling)
    {
        if (spoeling.ID == null)
        {
            // ...
        }
        else
        {
            // ...
        }
    }

    public async Task<Spoeling> Delete(int id)
    {
        // ...
    }
}

```

3.3.5.2 Applicatielaag

De applicatie laag heeft dezelfde structuur als het object van de kamers.

```

public class ObjSpoeling
{
    // ...
    public int? ID { get; set; }
    public int? KamerID { get; set; }
    public int? ItemID { get; set; }
    public int? ShiftID { get; set; }
    public bool Gespoeld { get; set; }
    public DateTime? Datum { get; set; }

    public ObjSpoeling(int? id, DateTime? datum, int? shiftID, int? kamerID, int? itemID, bool gespoeld)
    {
        ID = id;
        Datum = datum;
        ShiftID = shiftID;
        KamerID = kamerID;
        ItemID = itemID;
        Gespoeld = gespoeld;
    }

    public bool Save(out string errorTxt)
    {
        bool succeeded = false;
        errorTxt = string.Empty;
        int affected = 0;

        if (succeeded == IsValid(out errorTxt))
        {
            if (ID == null)
            {
                ID = Data.InsertSpoeling((DateTime)Datum, (int)ShiftID, (int)KamerID, (int)ItemID, Gespoeld);
                affected = 1;
            }
            else
            {
                affected = Data.UpdateSpoeling((int)ID, (DateTime)Datum, (int)ShiftID, (int)KamerID, (int)ItemID, Gespoeld);
            }
        }
        else
        {
            succeeded = false;
        }

        if (affected == 1)
        {
            succeeded = true;
        }
        else
        {
            // ...
        }
    }
}

```

3.3.5.3 Presentatielaag

In dit schermpje wordt de methode “InitControls” vaak gebruikt, namelijk als er een shift wordt geselecteerd, een kamer wordt geselecteerd en in het openen van dit scherm.

```

private void InitControls()
{
    long? ShiftId = Helper.GetIDFromComboBoxValue(ShiftComboBox);
    if (ShiftId != null)
    {
        ShiftComboBox.Enabled = true;
    }
    else
    {
        ShiftComboBox.Enabled = false;
        ShiftComboBox.SelectedIndex = -1;
    }

    long? Kamer = Helper.GetIDFromComboBoxValue(KamerComboBox);
    if (Kamer != null)
    {
        KamerComboBox.Enabled = true;
        KamerComboBox.SelectedIndex = 0;
        KamerComboBox.Focus();
    }
    else
    {
        KamerComboBox.Enabled = false;
        KamerComboBox.SelectedIndex = -1;
    }

    long? SpoelId = Helper.GetIDFromComboBoxValue(SpoelComboBox);
    if (SpoelId != null)
    {
        SpoelComboBox.Enabled = true;
        SpoelComboBox.SelectedIndex = 0;
        SpoelComboBox.Focus();
    }
    else
    {
        SpoelComboBox.Enabled = false;
        SpoelComboBox.SelectedIndex = -1;
    }
}

```

3.3.6 Data-out

Export Overzicht

Filter: Spoelplaats: Shift: Kamer: Begin datum: maandag 2 mei 2022 Einde datum: zondag 8 mei 2022 Zoeken Reset filter Exporteer naar Excel

Shift	Spoelplaats	Kamer	Wisseljaar	Spoeltype	Gevoeld	Datum
Postdienst - Shift 1	Buizenruimte	Kamer 1	W19-2022	Douches 2	<input checked="" type="checkbox"/>	2/05/2022 12:10
Postdienst - Shift 1	Buizenruimte	Kamer 5	W19-2022	Lavabo	<input checked="" type="checkbox"/>	2/05/2022 12:10
Postdienst - Shift 1	Buizenruimte	Kamer 5	W19-2022	WC 1	<input checked="" type="checkbox"/>	2/05/2022 12:10
Postdienst - Shift 1	C2	Kamer 5	W19-2022	Douches 1	<input checked="" type="checkbox"/>	2/05/2022 12:10
Postdienst - Shift 1	C2	Kamer 5	W19-2022	Douches 2	<input checked="" type="checkbox"/>	2/05/2022 12:10
Postdienst - Shift 6	Anatomie Pathologie	Kamer 2	W19-2022	Douches 1	<input checked="" type="checkbox"/>	2/05/2022 8:37
Postdienst - Shift 6	Anatomie Pathologie	Kamer 2	W19-2022	Douches 2	<input checked="" type="checkbox"/>	2/05/2022 8:37
Postdienst - Shift 6	Anatomie Pathologie	Kamer 4	W19-2022	WC 2	<input checked="" type="checkbox"/>	2/05/2022 8:37

8

Als het veld is vullen op een export

Als het scherm wordt geopend kan er gefilterd worden op verschillende eigenschappen van een spoeling. Bijvoorbeeld: toon mij alle douches die niet gespoeld zijn tussen deze bepaalde tijd.

Export Overzicht

Filter: Spoelplaats: Shift: Kamer: Begin datum: maandag 20 mei 2022 Einde datum: zondag 8 mei 2022 Zoeken Reset filter Exporteer naar Excel

Shift	Spoelplaats	Kamer	Wisseljaar	Spoeltype	Gevoeld	Datum
Postdienst - Shift 1	C2	Kamer 5	W19-2022	Douches 1	<input checked="" type="checkbox"/>	1/04/2022
Postdienst - Shift 1	C2	Kamer 5	W19-2022	Douches 1	<input type="checkbox"/>	6/04/2022
Postdienst - Shift 1	C2	Kamer 5	W17-2022	Douches 1	<input type="checkbox"/>	28/04/2022 9:48
Postdienst - Shift 1	C2	Kamer 5	W18-2022	Douches 1	<input type="checkbox"/>	29/04/2022 9:44
Postdienst - Shift 1	C2	Kamer 5	W18-2022	Douches 1	<input type="checkbox"/>	27/04/2022 13:28
Postdienst - Shift 6	Anatomie Pathologie	Kamer 2	W19-2022	Douches 1	<input type="checkbox"/>	14/04/2022 9:46
Postdienst - Shift 6	Anatomie Pathologie	Kamer 2	W17-2022	Douches 1	<input type="checkbox"/>	26/04/2022 9:49
Postdienst - Shift 6	Anatomie Pathologie	Kamer 2	W18-2022	Douches 1	<input type="checkbox"/>	26/04/2022 9:44

Na het opvragen van de gewilde data kan er gevraagd worden om deze data te exporteren naar een Excel. Er wordt door behulp van een laadbalkje getoond hoe ver het genereren van de Excel staat.

Schrijven van data op rij 8 van 18

	A	B	C	D	E	F	G	H
1	Shift	SpoelPlaats	Kamer	Week - Jaar	SpoelItem	Gespoeld	Datum	
2	Poetsdienst - Shift 1	C2	Kamer 8	W15 - 2022	Douche 1	False	5/04/2022 0:00:00	
3	Poetsdienst - Shift 1	C2	Kamer 8	W15 - 2022	Douche 1	False	6/04/2022 0:00:00	
4	Poetsdienst - Shift 1	C2	Kamer 8	W16 - 2022	Douche 1	True	14/04/2022 9:46:15	
5	Poetsdienst - Shift 1	C2	Kamer 8	W16 - 2022	Douche 1	True	15/04/2022 9:21:16	
6	Poetsdienst - Shift 1	C2	Kamer 8	W17 - 2022	Douche 1	True	19/04/2022 11:04:18	
7	Poetsdienst - Shift 1	C2	Kamer 8	W17 - 2022	Douche 1	False	20/04/2022 9:48:59	
8	Poetsdienst - Shift 1	C2	Kamer 8	W17 - 2022	Douche 1	True	21/04/2022 0:00:00	
9	Poetsdienst - Shift 1	C2	Kamer 8	W18 - 2022	Douche 1	False	25/04/2022 9:44:17	
10	Poetsdienst - Shift 1	C2	Kamer 8	W18 - 2022	Douche 1	False	27/04/2022 13:38:21	
11	Poetsdienst - Shift 1	C2	Kamer 8	W18 - 2022	Douche 1	True	29/04/2022 0:00:00	
12	Poetsdienst - Shift 1	C2	Kamer 8	W19 - 2022	Douche 1	True	2/05/2022 12:13:11	
13	Poetsdienst - Shift 6	Anatoom Pathologie	Kamer 2	W16 - 2022	Douche 1	True	13/04/2022 7:58:01	
14	Poetsdienst - Shift 6	Anatoom Pathologie	Kamer 2	W16 - 2022	Douche 1	False	14/04/2022 9:46:15	
15	Poetsdienst - Shift 6	Anatoom Pathologie	Kamer 2	W16 - 2022	Douche 1	True	15/04/2022 9:21:16	
16	Poetsdienst - Shift 6	Anatoom Pathologie	Kamer 2	W17 - 2022	Douche 1	False	20/04/2022 9:48:59	
17	Poetsdienst - Shift 6	Anatoom Pathologie	Kamer 2	W18 - 2022	Douche 1	False	25/04/2022 9:44:17	
18	Poetsdienst - Shift 6	Anatoom Pathologie	Kamer 2	W18 - 2022	Douche 1	True	27/04/2022 13:50:09	
19	Poetsdienst - Shift 6	Anatoom Pathologie	Kamer 2	W19 - 2022	Douche 1	True	2/05/2022 8:37:21	
20								
21								
22								

3.3.6.1 Data-laag

De data wordt opgevraagd met de volgende query. De kolommen worden opgemaakt door middel van het joinen van meerdere tabellen.

```

SELECT (SELECT ShiftID FROM Shift WHERE ShiftID = 1) AS ShiftID, (SELECT SpoelItemID FROM SpoelItem WHERE SpoelItemID = 1) AS SpoelItemID, (SELECT WeekJaarID FROM WeekJaar WHERE WeekJaarID = 1) AS WeekJaarID, (SELECT KamerID FROM Kamer WHERE KamerID = 1) AS KamerID, (SELECT SpoelPlaatsID FROM SpoelPlaats WHERE SpoelPlaatsID = 1) AS SpoelPlaatsID, (SELECT Datum FROM Datum WHERE Datum = '2022-04-05 00:00:00') AS Datum
FROM Shift
JOIN SpoelItem ON ShiftID = SpoelItemID
JOIN WeekJaar ON WeekJaarID = WeekJaarID
JOIN Kamer ON KamerID = KamerID
JOIN SpoelPlaats ON SpoelPlaatsID = SpoelPlaatsID

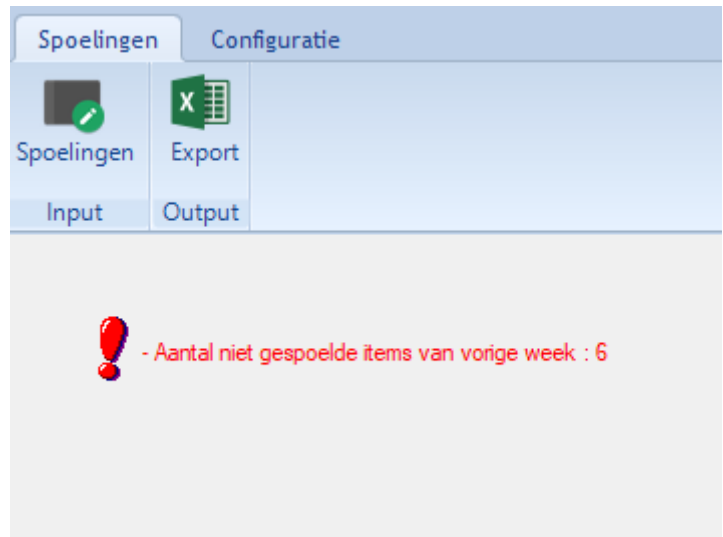
```

3.3.6.2 Applicatielaag

Voor de export is er geen object aangemaakt dus is er geen applicatielaag aanwezig voor dit deel.

3.3.6.3 Presentatie laag

Net zoals bij het configuratie en data-in deel is de structuur van de presentatielaag hier hetzelfde. Het enige opmerkbare in dit stuk is dat we gebruik maken van een enumeratie voor de status van de items. Omdat er in het hoofdscherm van de applicatie een informatieve label staat waar dubbel op geklikt kan worden en direct gefilterd kan worden op niet gespoelde items van de vorige week, moet deze ook direct worden ingevuld worden in de filters.



```
2 references
public FormExport(Enums.SpoelStatuses? spoelStatusFilter, bool isFillMode = false)
{
    InitializeComponent();
    IsFillMode = isFillMode;
    SpoelStatusFilter = spoelStatusFilter;
}
```

3.3.7 Excel maken

Het Excel bestand wordt opgemaakt in een statische klasse. De Excel klasse heeft ook zijn eigen helper-klasse, hierdoor kan er bijvoorbeeld een methode aangemaakt worden die de rand van een bepaalde range in het bestand dik maakt. Zo zijn er verschillende methodes gemaakt om de achtergrond van kleur te veranderen, tekst decoratie toe te voegen en verschillende soorten randen aan te maken.

```
public static void ColorBackgroundCell(int row, int col, Color color, Excel.Worksheet worksheet)
{
    worksheet.Cells[row, col].Interior.Color = color;
}

// New Test in cell Bold
public static void AddBoldStyleCell(int row, int col, Excel.Worksheet worksheet)
{
    worksheet.Cells[row, col].Font.Bold = true;
}

// New Test in cell Italic
public static void AddItalicStyleCell(int row, int col, Excel.Worksheet worksheet)
{
    worksheet.Cells[row, col].Font.Italic = true;
}

public static void ColorBooleanCell(int row, int col, string data, Excel.Worksheet worksheet)
{
    switch (data)
    {
        case "Value":
            ColorBackgroundCell(row, col, Color.Salmon, worksheet);
            break;
        case "True":
            ColorBackgroundCell(row, col, Color.PaleGreen, worksheet);
            break;
        default:
            break;
    }
}
```

```
public static void DrawThickBorder(int startRow, int startCol, int endRow, int endCol, Excel.Worksheet sheet)
{
    Excel.Range ranges = sheet.Range[sheet.Cells[startRow, startCol + 1], sheet.Cells[endRow, endCol]].Cells;
    ranges.BorderAround2(Excel.XlLineStyle.xlContinuous, Excel.XlBorderWeight.xlThick, Excel.XlColorIndex.xlColorIndexAutomatic, Color.Black);
}
```

3.3.8 Extra label

Tegen het einde van mijn stage, kreeg ik de vraag of ik een label kon toevoegen zoals het "niet gespoelde items" label. Deze keer wilde men zien hoeveel kamers er gedurende een week werden overgeslagen. Daarom werd de volgende query in het bestaan geroepen:

```
SELECT HuidigeKamersVorigeWeekJaar.KamerID, HuidigeKamersVorigeWeekJaar.SpoelDag
FROM (SELECT TOP (100) PERCENT HuidigeKamers.KamerID, VorigeWeekJaar.SpoelDag
      (SELECT ID AS KamerID
      FROM dbo.Kamers) AS HuidigeKamers CROSS JOIN
      (SELECT SpoelDag
      FROM (SELECT TOP, CONVERT(VARCHAR, DATEPART(ww, CONVERT(datetime, '20220509')) + CONVERT(VARCHAR, DATETIME(yyyy, CONVERT(datetime, '20220509')))) AS SpoelDag: wv dt)
      AS VorigeWeekJaar
      ORDER BY VorigeWeekJaar.SpoelDag: wv HuidigeKamersVorigeWeekJaar.SpoelDag: wv dt)
      (SELECT DISTINCT TOP (100) PERCENT KamerID, CONVERT(VARCHAR(10), CONVERT(datetime, Datum))) + CONVERT(VARCHAR, DATEPART(yyyy, CONVERT(datetime, Datum)))
      AS SpoelDag
      FROM dbo.Spoelingen
      (SELECT TOP, CONVERT(VARCHAR, DATEPART(ww, CONVERT(datetime, Datum))) + CONVERT(VARCHAR, DATEPART(yyyy, CONVERT(datetime, Datum))) = CONVERT(int, CONVERT(VARCHAR, DATEPART(ww, CONVERT(datetime, '20220509')))) + CONVERT(VARCHAR, DATEPART(yyyy, CONVERT(datetime, '20220509'))))
      ORDER BY SpoelDag: wv SpoelingenSpoelDag: wv HuidigeKamersVorigeWeekJaar.KamerID: wv SpoelingenSpoelDag: wv HuidigeKamersVorigeWeekJaar.SpoelDag: wv dt)
WHERE (SpoelingenSpoelDag: wv dt) IS NULL)
```

De query bestaat uit verschillende delen:

1. Alle geconfigureerde kamers

```
SELECT ID AS KamerID
FROM dbo.Kamers
```

2. De query om het weeknummer en de jaarcombinatie van de vorige

```
SELECT SpoelDag
FROM (SELECT CONVERT(INT, CONVERT(VARCHAR, DATEPART(wv,
CONVERT(datetime, '20220509')))) + CONVERT(VARCHAR, DATEPART(yyyy, CONVERT(datetime,
'20220509')))) AS SpoelDag) AS dt
```

3. Alle kamers met de weeknummercombinatie

```
SELECT TOP (100) PERCENT dbo.HuidigeKamers.KamerID, dbo.VorigeWeekJaar.SpoelDag
FROM dbo.HuidigeKamers CROSS JOIN
      dbo.VorigeWeekJaar
ORDER BY dbo.VorigeWeekJaar.SpoelDag
```

4. Alle spoelingen met de weeknummercombinatie

```
SELECT DISTINCT TOP (100) PERCENT KamerID, CONVERT(VARCHAR(10), CONVERT(VARCHAR(10),
DATEPART(wv, CONVERT(datetime, Datum))) + CONVERT(VARCHAR, DATEPART(yyyy,
CONVERT(datetime, Datum)))) AS SpoelDag
FROM dbo.Spoelingen
WHERE (CONVERT(INT, CONVERT(VARCHAR, DATEPART(wv, CONVERT(datetime, Datum))) +
CONVERT(VARCHAR, DATEPART(yyyy, CONVERT(datetime, Datum)))) = CONVERT(int,
CONVERT(VARCHAR, DATEPART(wv, CONVERT(datetime,
'20220509')))) + CONVERT(VARCHAR, DATEPART(yyyy,
CONVERT(datetime, '20220509'))))
ORDER BY SpoelDag
```

Als we dit alles samenvoegen krijgen we alle kamers die gespoeld moeten worden, als we dan een "WHERE" toevoegen waar de kamer "NULL" is krijgen we alle spoelingen die overgeslagen worden.

3.3.9 Helper klassen

Zoals al vermeld zijn er verschillende helper klassen gemaakt. Zo is er een helper om keuzelijsten op te vullen.

```

//reference
public static void CreateShiftenComboBox(ComboBox cmb, bool addEmptyRow, bool useSpoelUitvoerderID)
{
    DataTable dtShift;

    if (useSpoelUitvoerderID)
    {
        dtShift = Data.GetSpoelUitvoerdersShiften(helper.SpoelUitvoerderID, addEmptyRow);
    }
    else
    {
        dtShift = Data.GetSpoelUitvoerdersShiften(null, addEmptyRow);
    }

    cmb.DataSource = dtShift;
    cmb.DisplayMember = "SpoelUitvoerdersShift";
    cmb.ValueMember = "ID";
    cmb.DropDownWidth = 300;
}

//reference
public static void CreateKamersComboBox(ComboBox cmb, int? idShift, bool addEmptyRow)
{
    int? ID = null;
    int? spoelPlaatsID = null;
    string kamerNaamFilter = string.Empty;
    int? spoelItemID = null;

    cmb.DataSource = Data.GetKamers(ID, spoelPlaatsID, kamerNaamFilter, spoelItemID, idShift, addEmptyRow);
    cmb.DisplayMember = "naam";
    cmb.ValueMember = "id";
}

```

Zo zijn er dus ook bepaalde functies die alleen gebruikers in een bepaalde Active Directory groep mogen uitvoeren. Daarom is er een methode die kijkt van welke AD-groep de gebruiker zit en welke dingen deze gebruiker dan wel en niet mag zien en/of uitvoeren.

```

//reference
public static void SelectADMembership()
{
    foreach (int group in Enum.GetValues(typeof(Enum.ADGroep)))
    {
        string name = Enum.GetName(typeof(Enum.ADGroep), group);
        bool isMember = IsMemberOf(Environment.UserDomainName, Environment.UserName, name);

        if (isMember)
        {
            if (name == "U.SpoelRegistratie_PowerUsers")
            {
                IsPowerUser = true;
            }
            else
            {
                SpoelUitvoerderID = group;
                break;
            }
        }
    }
}

//reference
private static bool IsMemberOf(string domain, string userName, string groupName)
{
    PrincipalContext ctx = new PrincipalContext(ContextType.Domain, domain);

    // find a user
    UserPrincipal user = UserPrincipal.FindByIdentity(ctx, userName);

    // find the group in runtime
    GroupPrincipal group = GroupPrincipal.FindByIdentity(ctx, groupName);

    // get members
    object foundUsers = null;
    if (user != null && group != null)
    {
        foundUsers = group.GetMembers(true).Where(p => p.SamAccountName.Equals(user.SamAccountName, StringComparison.InvariantCultureIgnoreCase)).FirstOrDefault();
    }

    // check if user is member of that group
    if (foundUsers != null)
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

Zo wordt er in de applicatie veel gebruik gemaakt van een datagrid. Om de kolommen een mooie hoofding te geven hebben we een functie die de naam mooi maakt voor de gebruiker.

```

6 references
public static string BeautifyTableName(string DatabaseTabel, bool enkelvoud)
{
    string newName = DatabaseTabel[0].ToString().ToUpper() + DatabaseTabel.Substring(1);

    if (enkelvoud)
    {
        if (newName.Substring(newName.Length - 2).Contains("en"))
        {
            newName = newName.Substring(0, newName.Length - 2);
        }
        else if (newName.Substring(newName.Length - 1).Contains("s"))
        {
            newName = newName.Substring(0, newName.Length - 1);
        }
    }

    return newName;
}

```

Zo zijn er nog veel meer kleine methodes die simpele taken uitvoeren zonder dubbele code te hoeven schrijven.

3.3.10 Uitbrengen van de software

Dit is niet iets wat ik zelf mocht doen maar wel bij aanwezig mocht zijn. De webpagina wordt uitgebracht via IIS en de Forms applicatie wordt op het portaal gezet dat het ziekenhuis gebruikt om software uit te delen.

3.4 EXTRA OPDRACHTEN

De hoofdopdrachten waren rapper klaar dan voorzien maar dat heeft er niet voor gezorgd dat we stil zitten. Zo heb ik mij beziggehouden met kleine opdrachtjes die nog moeten gebeuren.

- QR-code generator
- Datatype aanpassing van een kolom in de database
- Importeren van een Excel bestand om simpel nieuwe stagiair accounts aan te maken.
- Kleine analyse oefeningen

3.4.1 QR-code generator

De dienst communicatie gebruikt graag QR-codes. En om niet telkens een website te moeten gebruiken werd er gevraagd of er geen QR-generator kon gemaakt worden.

De mogelijkheid is voorzien om de QR-code op te slaan of om direct op het klembord te plaatsen.

Er zit ook een controle op dat als er geen "https://" meegegeven wordt dat deze automatisch wordt toegevoegd aan de link.



3.4.2 Datatype aanpassing

In de database stond een kolom als een VARCHAR aangeduid en deze moest naar een INT worden veranderd. Dit kan niet zomaar gedaan worden omdat in de bestaande applicatie nog niks aangepast was.

Daarom moest ik een structurele manier van aanpak zoeken om geen enkele stap over te slaan zodat de applicatie nog perfect werkt na de aanpassing in de database. Zo ben ik begonnen in de data-laag, zo kon ik met de referenties van die methodes simpel zien waar deze gebruikt worden en deze konden stap per stap aangepast worden. Om voor de mentor een duidelijk overzicht te geven over wat er allemaal aangepast is staat bij elke aanpassing in de code een commentaar lijn met mijn naam en de datum van de aanpassing. Zo kan deze simpel terug gevonden worden met de "Find" mogelijkheid in Visual studio. Als extra heb ik een document opgesteld waar ook alle aanpassingen staan genoteerd.

Dit document genaamd "DataType aanpassing" kan terug gevonden worden op het portfolio.

3.4.3 Importeren van de stagiair Excel

Omdat er elk jaar wel stagiairs zijn en dit niet gaat over een of twee stagiairs wordt er voorzien dat de school een Excel template kan invullen en dat deze in een applicatie kan geplaatst worden zodat de accounts die normaal handmatig gemaakt moesten worden automatisch gegenereerd kunnen worden.

Mijn mentor stuurde me een mailtje met alle nodige velden in de Excel en heeft mij het document laten opstellen. Zo zijn er velden die verplicht moeten worden ingevuld, velden die optioneel zijn en velden die voor het personeel van het ziekenhuis zelf is. Belangrijk was dat het duidelijk was voor beide partijen wie wat waar moet invullen. Zo zijn er van alle

trucjes in Excel gebruikt om de hoofding te vergrendelen en zijn er keuze lijstjes gemaakt zodat de juiste waardes worden ingevuld.

[illegible]

Voor het ziekenhuis staan de velden eigenlijk al vast, het personeel moet hier alleen nog kiezen uit een keuzelijst. Deze keuze lijstjes zijn opgemaakt in een tweede blad dat onzichtbaar gemaakt is voor de scholen.

Type	Department	SubType	Student	Dienst	Faculteit/Dienst	Amendement	
Student	Ankers	Actiefuyske	VerpleegkundigActiefuyske	VerpleegkundigDienst	Actiefuyske	PacemakersDienst	Amendement
		Arts	Dokter Stager	Student Junior Medischtoeliefer	Dokter Directie Verpleging	Binnen Afdeling	Anaesth
	Paramedici	Dokter Assistent	Student Med Medischtoeliefer	Chirurgisch Dagziekenhuis	Dokter Anesthetist	Dienst Ziekenhuis	Artsdienst
		Verpleegkundige	Dokter Assistent	Student Med Medischtoeliefer	Diabetes Consultatie	Dokter Anesthesioloog	Dokter
	Verpleegkundige	Diabetes	Student Med Medischtoeliefer	Diabetes Consultatie	Dokter Anesthesioloog	Endoscopie	Biochemische Dienst
		Dienst 1	Student Med Medischtoeliefer	Diabetes Consultatie	Dokter Endoscopie	Gepanen	Biochemie
		Dienst 10	Student Med Medischtoeliefer	Diabetes Consultatie	Dokter Gynaecologie	Labo	Directie
		Dienst 2	Student Med Medischtoeliefer	Diabetes Consultatie	Dokter Heelkunde	Medische Registratie	Directie Secretariaat
		Dienst 4	Student Med Medischtoeliefer	Diabetes Consultatie	Dokter Inwendige	Nucleaire	Economie
		Dienst 5	Student Med Medischtoeliefer	Diabetes Consultatie	Dokter Internis Labo	Palliatief Support Team	Financieel Manager
		Dienst 6	Student Med Medischtoeliefer	Diabetes MFC	Dokter MFC	Physiokn	Financieel Dienst
		Dienst 7	Student Med Medischtoeliefer	Diabetes Heelkunde	Dokter Heelkunde	Postdentist	PersonneelDienst
		Dienst 8	Student Med Medischtoeliefer	Diabetes Neurochirurgie	Dokter Neurochirurgie	Poli Gynaecologie	Informatie
		Dienst 9	Student Med Medischtoeliefer	Diabetes Neurologie	Dokter Neurologie	Poli Heelkunde - Orthopedie	Informatieverlichting
		Dienst Internis	Student Med Medischtoeliefer	Diabetes Nucleaire	Dokter Nucleaire	Poli Inwendige	Isolatie
		Dienst OK	Student Med Medischtoeliefer	Diabetes Oefeningslogie	Dokter Oefeningslogie	Poli Neum	Levenskoper
		Dienst Pediatris	Student Med Medischtoeliefer	Diabetes Oncologie	Dokter Oncologie	Poli Oreo	Magasin
		Dienst GP	Student Med Medischtoeliefer	Diabetes ORL	Dokter ORL	Poli ORL	Opname
		Dienst Tappet	Student Med Medischtoeliefer	Diabetes Otorhinologie	Dokter Otorhinologie	Poli Pediatris	Overzichttafel
		Dienst Vaginale	Student Med Medischtoeliefer	Diabetes Pediatris	Dokter Pediatris	Poli Urologie	Tafelkast - Gachstaf
		Louwacht	Student Med Medischtoeliefer	Diabetes Radiologie	Dokter Radiologie	Psychologie	Technische Dienst
			Oncologisch Dagziekenhuis	Diabetes Radiologie	Dokter Radiologie	Radiologie	YTO
Maaglabo	Student Med Medischtoeliefer			Diabetes Spandevelatie	Dienst Revalidatie	Communicatie	
	Dokter Malenier	Diabetes Stomatologie	Dokter Stomatologie	Secretariaat Vaginale Hoes			
		Geriatrisch Dagziekenhuis	Dokter Urologie	Sociale Dienst			
	Medische Square Intake Deciden	Physiotherapie	Student Med Medischtoeliefer	Physiotherapie	Standaard		
		Diabetes Pneumologie	Dokter Pneumologie	Tafelkast			
		Diabetes Geriatrische	Dokter Geriatrische	Versier			
		Dienst Gastro-Enterologie	Student Med Medischtoeliefer	Wondzorg			
				Dienst Ego			
				Poly Gastro			

Op vraag van de mentor is er een handleiding gemaakt zodat er later iets aangepast kan worden. Zo is het simpel om mijn instellingen ongedaan te maken, iets aan te passen en terug in te stellen.

De import zal worden gedaan in een reeds bestaande applicatie genaamd "Nieuwe Medewerkers". Ik heb 2 extra klassen toegevoegd genaamd "ExcelImportHelper", dit is waar mijn functies die helpen zullen gaan zoals "GetCellValue" die mij de waarde van een specifieke cel zal geven.

```
10 references
public static string GetCellValue(int col, int row)
{
    var value = excelWorksheet.Cells[row, col].Value;
    if (value == null)
    {
        return null;
    }

    string result = value.ToString();
    if (!string.IsNullOrEmpty(result))
    {
        result = result.Trim();
    }
    return result;
}
```

De andere klasse is de "ExcelImport" klasse. Hier is de hoofdstroom van wat er moet gebeuren als een gebruiker op de import knop drukt. De eerste stap is controleren of de persoon die het Excel bestand heeft ingevuld een manier heeft gevonden om onze beveiliging van de header te breken. Daarom controleren we nog steeds of alles op de juiste plaats staat.

```
1 reference
public static bool CheckHeaderCells(Excel.Worksheet sheet)
{
    bool succeeded = false;

    foreach (VerplichteKolommen col in Enum.GetValues(typeof(VerplichteKolommen)))
    {
        string cellValue = GetCellValue((int)col, rowHeader);

        cellValue = BeautifyCellValue(cellValue);

        if (cellValue != col.ToString())
        {
            return succeeded;
        }
    }

    foreach (OptioneleKolommen col in Enum.GetValues(typeof(OptioneleKolommen)))
    {
        string cellValue = GetCellValue((int)col, rowHeader);

        cellValue = BeautifyCellValue(cellValue);

        if (cellValue != col.ToString())
        {
            return succeeded;
        }
    }

    foreach (ZiekenhuisKolommen col in Enum.GetValues(typeof(ZiekenhuisKolommen)))
    {
        string cellValue = GetCellValue((int)col, rowHeader);

        cellValue = BeautifyCellValue(cellValue);

        if (cellValue != col.ToString())
        {
            return succeeded;
        }
    }

    succeeded = true;

    return succeeded;
}
```

```

ExcelImporterHelper.setLabel("...Excel nakijken", label);

if (ExcelImporterHelper.CheckHeaderCells(ExcelImporterHelper.excelWorksheet))
{
    int lastDataRow = ExcelImporterHelper.SearchLastDataRow(ExcelImporterHelper.excelWorksheet);
    int totalRows = lastDataRow - ExcelImporterHelper.rowDataFirst;

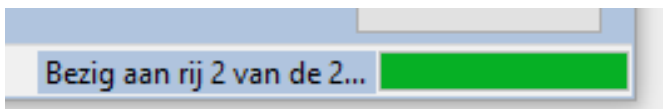
    ExcelImporterHelper.InitProgressBar(progressBar, totalRows);

    if (lastDataRow != 4)
    {
        for (int row = ExcelImporterHelper.rowDataFirst; row <= lastDataRow; row++)
        {
            ExcelImporterHelper.setLabel($"...Bezig aan rij {row - ExcelImporterHelper.rowDataFirst} van de {totalRows}", label);

            string succeedCheck = ExcelImporterHelper.GetCellValue(1, row);
            if (succeedCheck != "OK")
            {
                // ...
            }
        }
    }
}

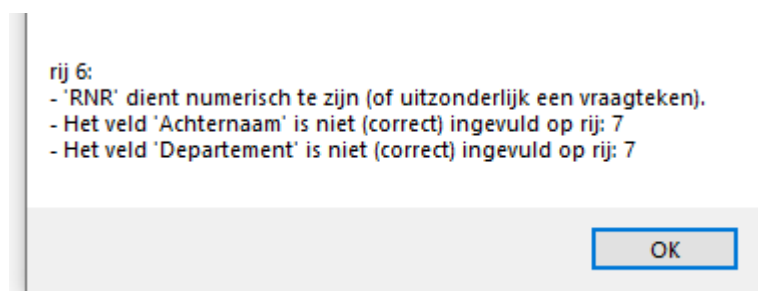
```

Omdat de gebruiker Excel niet te zien krijgt (omdat we het onzichtbaar op de achtergrond openen), heb ik een voortgangsbalk en status label aan het formulier toegevoegd. Op deze manier weet de gebruiker dat er iets gebeurt.



Als een verplicht veld niet is ingevuld in het template bestand, geeft de applicatie een foutmelding aan het einde van het importeren van het bestand. Het stopt niet zodra 1 rij niet correct is ingevuld. Indien er een fout optreedt, zal er een mail doorgestuurd worden naar de bug-email van het ziekenhuis zodat zij weten dat er iets mis is en de gebruiker kunnen helpen indien nodig.

Er is voorzien in een validatie op de verplichte velden en een validatie op het type van de eigenschappen. Zo moet de eigenschap "RNR" een numerieke waarde zijn.



	...	?	...	Student
--	-----	---	-----	---------



Ik heb ook iets voorzien dat een markering plaatst in de eerste cel van de rij, zodat als er iets fout gaat, je gewoon hetzelfde bestand kunt gebruiken en het zal alle rijen overslaan die de gemarkeerde cel bevatten, zodat we niet veel fouten krijgen bij dubbele invoer.

5			
6	OK	RubenTest	Boone
7		StefTest	
8			

Als de cel "OK" bevat, wordt de rij overgeslagen, de rode en groene kleur zijn er zodat de gebruiker kan zien welke rijen gelukt zijn en welke niet.

3.4.4 Analyse oefeningen

De mentor stuurde soms een mail met een tekstje dat moest geanalyseerd worden. Hij verwachtte dan een schema of verduidelijking over wat er aanwezig is van objecten, welke eigenschappen deze objecten hebben,

4 SOFT SKILLS

4.1 COMMUNICATIE

Tijdens mijn stage heb ik geen enkele dag van thuis uit gewerkt. Als er een probleem was kon ik gewoon rechtstaan en dit aan de mentor vertellen. Als hij dan tijd had, kwam hij even meedenken of hulp toe dienen. Als de mentor niet aanwezig was op kantoor dan was hij altijd te bereiken via email. Zo hebben we ook 2 teams gesprekken gehad om de stand van zaken te bekijken.



Dag Stefan

Wat ik tot nu toe voor elkaar gekregen heb in de webpagina:

- De kalender heeft een default value van vandaag.
- Er kan geen datum in de toekomst gekozen worden.
- De shift dropdown (combobox) toont de juiste shifts (Door de AD-groepen)
- De juiste kamers worden getoond in hun dropdown
- De datagrid komt tevoorschijn met de juiste data, de overbodige kolommen worden onzichtbaar gemaakt

Het probleem waar ik nu al eventjes mee vast zit is dat de datagrid read only is. De datagrid bevat geen .ReadOnly property, heb al redelijk wat Google searches gedaan, maar zonder succes.

aken. Op
wat ik
t alle
chte hier
heb.

[illegible]