



# **Programming in C# with SQL**

## Realization

**Internship**

**Ruben Boone 3APP02**

Academic year 2021-2022

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel



## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION .....</b>	<b>6</b>
<b>2</b>	<b>COMAPNY STRCUTURE .....</b>	<b>7</b>
<b>2.1</b>	<b>3-tier .....</b>	<b>7</b>
2.1.1	Data-layer .....	7
2.1.2	Application- or Business-layer .....	7
2.1.3	Presentation layer.....	7
2.1.4	Helper classes .....	7
<b>3</b>	<b>ASSIGNMENTS .....</b>	<b>8</b>
<b>3.1</b>	<b>EconomatLabels.....</b>	<b>8</b>
3.1.1	Description .....	8
3.1.2	Goal .....	8
3.1.3	Research .....	8
3.1.4	Realization.....	9
<b>3.2</b>	<b>Spoed Etiketten .....</b>	<b>11</b>
3.2.1	Description .....	11
3.2.2	Goal .....	12
3.2.3	Research .....	12
3.2.4	Realization.....	12
<b>3.3</b>	<b>Spool registration .....</b>	<b>13</b>
3.3.1	Description .....	13
3.3.2	Goal .....	13
3.3.3	Realization.....	13
3.3.3.1	step-by-step plan.....	13
3.3.4	Configuration part .....	15
3.3.4.1	Data layer.....	15
3.3.4.2	Application layer .....	16
3.3.4.3	Presentation layer.....	17
3.3.5	Data-in part.....	18
3.3.5.1	Data layer.....	19
3.3.5.2	Application layer .....	20
3.3.5.3	Presentation layer.....	20
3.3.6	Data-out .....	21
3.3.6.1	Data layer.....	22
3.3.6.2	Application layer .....	22
3.3.6.3	Presentation layer.....	23
3.3.7	Excel creation .....	23
3.3.8	Helper classes .....	24
3.3.9	Deploying the software.....	25
<b>3.4</b>	<b>Extra assingments .....</b>	<b>25</b>
3.4.1	QR-code generator .....	25
3.4.2	Datatype change .....	26
3.4.3	Importing intern Excel.....	26
3.4.4	Analyses exercises.....	27
<b>4</b>	<b>SOFT SKILLS .....</b>	<b>28</b>
<b>4.1</b>	<b>Communication.....</b>	<b>28</b>
<b>4.2</b>	<b>Scheduling .....</b>	<b>28</b>



# 1 INTRODUCTION

---

In this paper I am going to prove all that I have learned and created during my internship at the hospital in Geel. Now you might be thinking, why in a hospital? I have chosen for this because of 2 reasons. The first reason is because the hospital in Geel had an internship on offer in programming in C# and SQL. My personal preference lies in creating applications in C#, Java, ... . This does not mean that I do not like working with web-based applications, because with ASP you also have the opportunity to work with this. The second reason was mainly because I wanted to take a look behind the scenes of a hospital. How does it all work there, how much does IT have to do with the current state of affairs in a hospital? I would like to thank my internship mentor Mr. Stefan de Wilde and internship supervisor Hajar Ghaem Sigarchian for the great support and tips during my internship and the hospital for giving me the opportunity to do my internship with them.

## 2 COMAPNY STRCUTURE

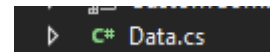
---

### 2.1 3-TIER

The applications in the hospital are built using a "3-tier" structure. So there is a data layer, an application/business layer and a presentation layer.

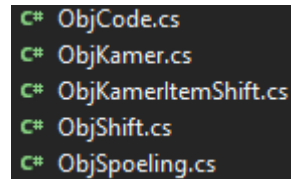
#### 2.1.1 Data-layer

The data layer contains all the SQL queries of the application. This is a static class in the application that can be called anywhere.



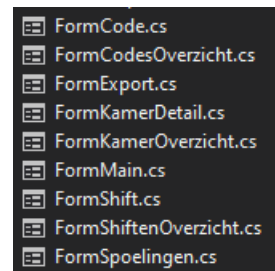
#### 2.1.2 Application- or Business-layer

In these classes the objects are created. These contain all the properties of an object and methods that are needed. Also present here are the various methods to populate the object, store it in the database, modify it in the database and delete it.



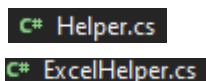
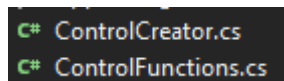
#### 2.1.3 Presentation layer

In this layer all screens are provided, this is the visual layer where all buttons, selection lists, input fields are present. This layer contains very little self written code.



#### 2.1.4 Helper classes

This is not really in the 3-tier system but it is used a lot. An example of a method in such a helper class is to fill a selection list.

## 3 ASSIGNMENTS

---

### 3.1 ECONOMATLABELS

#### 3.1.1 Description

This is an existing application that reads an Excel file to print labels to stick in the warehouse with the corresponding articles. On such a label is the article number, the quantity of a minimum order, a description, a reference number, a barcode and finally the date and time of the creation of the label. As soon as the barcode is scanned, the article is automatically ready to be ordered.



#### 3.1.2 Goal

The current printing process is not ideal. Once the data is read it is placed into a Word template using mail references, this Word document is then printed. The problem here is that when the printer finishes printing the Word process is not always completed correctly. If a lot of labels are printed it could be that a lot of Word processes are left open and that the computer is slow or even breaks down in the long run.

This gives us the following goals:

1. Look for a way to get to a label without making use of Word.
2. Look for a way to print the label without Word.
3. The barcode will need to be generated in a new way
- 4.

#### 3.1.3 Research

After some searching and the already delivered information found by the mentor, I came up with the following solution.

Word can be replaced by the C# class "PrintDocument". This immediately gagged me the possibility to use a PrintPage, such a printPage corresponds to the page where you normally write the text.

The Barcode can be generated using a Nuget package.

### 3.1.4 Realization

To keep it neat, we chose to create a static class called "Label" where everything related to the label can be placed.

In the new class, the method "PrintEconomaat" is present. It takes a "KeyValuePair" object as a parameter so that we have all the necessary data that needs to be on the label available in the class. Thus we also immediately fill in the barcode and the date with time. Finally, we call the "PrintEtiket()" method to continue.

```
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();
}
```

This is where the "PrintDocument" is created, followed by the creation and padding of the "PrintPage". Because we are not really working in a document and are actually drawing on a sheet we have to draw everything with "Graphics". Thinking about possible future changes, several constant variables have been created so that later changes only have to be made in one place and it is not necessary to search for everything.

```
private static void CreateEtiket(object sender, PrintPageEventArgs printPageEventArgs)
{
    printPageEventArgs.Graphics.DrawString(artikelLabel, DefaultFontBold, textColor, lijn1ArtikelX, lijn1Y);
    printPageEventArgs.Graphics.DrawString(aantallabel, DefaultFont, textColor, lijn1AantalX, lijn1Y);

    if (beschrijving.Length > maxCharacterOnLine)
    {
        printPageEventArgs.Graphics.DrawString(beschrijving.Substring(0, maxCharacterOnLine), DefaultFont, textColor, lijn2BeschrijvingX, lijn2BeschrijvingY);
        printPageEventArgs.Graphics.DrawString(beschrijving.Substring(maxCharacterOnLine), DefaultFont, textColor, lijn2BeschrijvingX, lijn2BeschrijvingY2);
    }
    else
    {
        printPageEventArgs.Graphics.DrawString(beschrijving, DefaultFont, textColor, lijn2BeschrijvingX, lijn2BeschrijvingY);
    }

    printPageEventArgs.Graphics.DrawString(referentieLabel, DefaultFont, textColor, lijn3ReferentieX, lijn3ReferentieY);

    printPageEventArgs.Graphics.DrawImage(barcodeImg, lijn4BarcodeX, lijn4BarcodeY);

    printPageEventArgs.Graphics.DrawString(barcodeCijfers, 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 maxCharachterOnLine = 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;

```

Finally, a printer must be chosen where the document is to be torn off. To think about the future, we thought of doing this with the "App.config" so that the whole application does not have to be re-released and it is possible to work with a different printer in a different location. This also gives us the possibility to provide multiple printers and thus let the user choose which printer to use.

```

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;
}

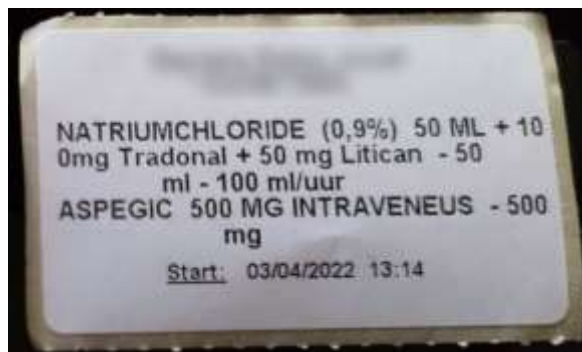
```

## 3.2 SPOED ETIKETTEN

### 3.2.1 Description

For the emergency department one has to print labels to stick on administered medication. This is required by Belgian law.

This is also an existing application but one that is structured slightly differently. Here the data comes through external software called ChipSoft - HiX. HiX provides, in short, information from electronic patient records. The selected patient ID and file number are then forwarded on request to an ASPX web page using arguments in the URL. There all additional information is retrieved using the ID. All the information on the label is placed in a text file and read by in-house made software. This data is then, just like the Economate labels, transferred via mail references to a Word template for subsequent printing.



### 3.2.2 Goal

As with the economy labels, the way of printing and the use of Word must be adjusted. In this command, no bar code should be provided on the label.

### 3.2.3 Research

Since this task is almost identical to the previous task, we can therefore reuse some code in this application. Thus, additional Research should not be done.

### 3.2.4 Realization

Again, a Label class is provided. So again the data is given in a "KeyValuePair" to the class, now the only real difference between these two assignments is just layout of the labels and the constants. So the printing is done just the same. The constants are here placed in the App.config because otherwise with every change in the label the web page has to be re-released and that is not ideal in an urgency department.

```
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 SPOOL REGISTRATION

#### 3.3.1 Description

In the hospital, there are several items that need to be flushed. Such an item may be a shower, sink, toilet, .... These items can be flushed by a different performer. There are currently 3 different types of performers. The nursing staff, the technical staff and of course the cleaning staff. Now we have been asked if an application can be made to improve this process. Currently everything is noted on paper and this takes a lot of time.

#### 3.3.2 Goal

A few things need to be provided.

1. A configuration part in the application so that it can be entered which items are there, which rooms are there, which shifts are there and which shift should flush a particular item in which room.
2. A way for the performers to mark an item as completed.
  - Because the cleaning crew does not have access to a laptop, they will have to indicate this in another way. Especially for them we provide a web page where an item can be marked as completed.
3. A way to export all collected data to an Excel file. The data that can be exported can be filtered in different ways so you can generate specific tables

#### 3.3.3 Realization

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 Configuration part.

##### 3.3.3.1 *step-by-step plan*

###### Database

1. setting up the Tables
  - a. Rooms table
  - a. RoomsItems table
  - a. RoomsItemsShiften table

###### Data-layer

1. Writing query's
  - a. Room
    - i. SELECT Room

- i.SELECT Rooms
  - i.INSERT Room
  - i.UPDATE Room
  - i.DEL Room
- a. RoomItem
  - i.SELECT RoomItems
  - i.INSERT RoomItem
  - i.DELETE RoomItem
  - i.(optioneel) UPDATE RoomItem
- a. RoomItemShift
  - i.SELECT RoomItemShift
  - i.INSERT RoomItemShift
  - i.DELETE RoomItemShift
  - i.(optioneel) UPDATE RoomItemShift
- a. general query's (dropdowns)
  - i.SELECT Items
  - i.SELECT Spool locationen
  - i.SELECT Shiften
  - i.SELECT SpoelUitvoerdersShiften (uitvoerderID vanuit AD)

#### Applicatie/Business-layer

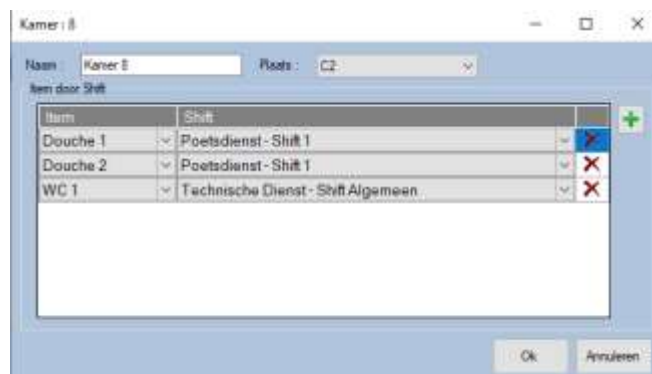
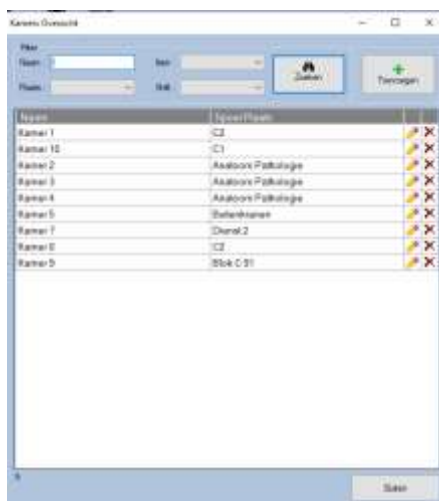
1. Objecten aanmaken
  - a. Spool location
    - i.Properties
      - ID
      - Name
  - a. Item
    - i.Properties
      - ID
      - Name
  - a. Room
    - i.Properties
      - ID
      - Name
      - Spool location - Spool location
      - Items - List<Item>
    - i.Methodes
      - Load - void
      - Save - void
      - Delete - void
      - IsValid - bool (out errormsg)
  - a. RoomItemShift
    - i.Properties
      - ID - int
      - RoomID - int
      - ItemID - int
      - ShiftID - int

#### Presentation layer

1. general
  - a. Ribbon
    - i.Config tab
1. Room overview (config)
  - a. Insert button
  - a. Filters
    - i.Spool location
    - i.Room
    - i.Item
    - i.Shift
    - i.Uitvoerder
  - a. Datagrid
    - i.Roomname
    - i.Delete button
    - i.Detail button

1. Room Details (config)
  - a. Room name
    - i.Label
    - i.Textbox
  - a. Spool location
    - i.Label
    - i.Dropdown met spool locationen
  - a. Datagrid
    - i.Items (dropdown)
    - i.Shift (dropdown)
  - a. Add ItemShift Button
    - i.Button
  - a. Delete ItemShift Button
    - i.Button

### 3.3.4 Configuration part



#### 3.3.4.1 Data layer

We start by getting the database in order; without correct tables, we can hardly proceed. Once the database is in order we can begin our 3-tier system. The first step is to write our database queries so we can retrieve, create, modify or delete our data.

+	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 Application layer

After the Data layer, the object classes are created also called the application or business layer. In the objects we keep track of the properties, e.g. the object "Room" has the properties "Name", "ID", .... Similarly, there is a "Save" method to save an instance of this object to the database or modify it, a "Delete" to delete an instance. An "IsValid" method is also provided to check if the properties meet the predefined requirements. If they do not, they may not be saved or modified.

```
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 *Presentation layer*

Finally, we end with the presentation layer, these are our screens with our button pen, choice lists, checkboxes, ... . These controls all contain their own events that hardly contain real code but rather refer to a method in another place. For example, there are several "Helper" classes that are privileged, filling a combo box.

A presentation layer also usually contains the same methods. For example, in the event when the window is loaded we have our "Init" methods.

These methods are executed only once, so the datagrid is prepared preferentially. This means that columns that should not be visible are made invisible. Another important method is the "InitControls" method, this method is used more often. Every time a button is pressed this method can be called. This method ensures that all controls in the window are set correctly. A small example of this is that the Room picker may not be used until a shift is selected.

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

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

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

    ControlCreator.CreateSpoelPlaatsenComboBox(plaatsComboBox, addEmptyRow);
    ControlCreator.CreateSpoelItemsComboBox(itemsComboBox, addEmptyRow);
    ControlCreator.CreateShiftenComboBox(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();
}
```

For example, each method has its own functionality so they can be reused anywhere and therefore only perform 1 thing.



```
4 references
private void FillDataGridViewOverzicht()
{
```

These steps also recur in Data-in and Export. Many of the methods can often be adopted with minor modifications.

### 3.3.5 Data-in part

The first screenshot shows the 'Spiegelings' application with a calendar for May 2022. The date '10' is selected. The second screenshot shows a list of items to be played, including 'Douches 1', 'WC 1', and 'Lunch'. The third screenshot shows the same list with checkboxes for 'Gespeeld' (Played).

Below the screenshots are two detailed views of the calendar and item list.

**Calendar View (Left):**

mei 2022							jun
ma	di	wo	do	vr	za	zo	
23	24	25	26	27	28	1	
2	3	4	5	6	7	8	
9	10						

Shift :

Kamer :

**Item List View (Right):**

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

Item is nog niet opgeslagen

The data-in part consists of 2 parts. The Windows form application and the ASP web page. Because not all functionalities work or are present in the web page, some things have to be done differently.

### 3.3.5.1 Data layer

The WHERE in the SQL query contains 3 pieces. The date of the flush, what shift the user is on and what Room the user is in to request the items that need to be flushed.

If no flush has been done yet then the application shows the configured items. As soon as an item with the correct Room, shift and date combination is found it will be displayed.

```
public static IQueryable GetSpoolings(DateTime? date, int? shiftId, int? roomId)
{
    SqlConnection sqlcon = new SqlConnection(databaseConnectionString);
    sqlcon.Open();

    string selectString = @"
        (SELECT COUNT(*)
         FROM Spoolings
         WHERE RoomID = @RoomId and ShiftID = @ShiftId
          AND (DATEPART(yy, @Spoolings.DateTime) = @Year
          AND DATEPART(mm, @Spoolings.DateTime) = @Month
          AND DATEPART(dd, @Spoolings.DateTime) = @Day)) > 0)

        SELECT @Spoolings.ID, @RoomID as Room, @Spoolings.Spinner, @Spoolings.ID as SpoolingID,
              @Spoolings.FromDeviceID as FromDeviceID, @Spoolings.ToDeviceID as ToDeviceID
        FROM @Spoolings INNER JOIN
              @SpoolingsFromDevices ON @Spoolings.FromDeviceID = @SpoolingsFromDevices.ID INNER JOIN
              @SpoolingsToDevices ON @Spoolings.ToDeviceID = @SpoolingsToDevices.ID INNER JOIN
              @Rooms ON @Spoolings.RoomID = @Rooms.ID
        WHERE DATEPART(yy, @Spoolings.DateTime) = @Year
          AND DATEPART(mm, @Spoolings.DateTime) = @Month
          AND DATEPART(dd, @Spoolings.DateTime) = @Day
          AND RoomID = @RoomId
          AND ShiftID = @ShiftId

        ELSE
        SELECT @RoomID as ID, @RoomID as Room, @Spoolings.ID as SpoolingID, @Spoolings.FromDeviceID as FromDeviceID, @Spoolings.ToDeviceID as ToDeviceID,
              @Spoolings.ID as ID, @Spoolings.ID as SpoolingID
        FROM @Spoolings
        WHERE @RoomID = @RoomId
          AND @Spoolings.FromDeviceID = @FromDeviceID";

    SqlCommand selectCommand = new SqlCommand(selectString, sqlcon);

    if (date != null)
    {
        DateTime dateTime = (DateTime)date;

        string year = dateTime.ToString("yyyy");
        string month = dateTime.ToString("MM");
        string day = dateTime.ToString("dd");

        selectCommand.Parameters.AddWithValue("year", year);
        selectCommand.Parameters.AddWithValue("month", month);
        selectCommand.Parameters.AddWithValue("day", day);
    }
    else
    {
        selectCommand.Parameters.AddWithValue("year", DBNull.Value);
        selectCommand.Parameters.AddWithValue("month", DBNull.Value);
        selectCommand.Parameters.AddWithValue("day", DBNull.Value);
    }

    selectCommand.Parameters.AddWithValue("RoomID", DatabaseHelper.GetIdFromRoom(roomId));
    selectCommand.Parameters.AddWithValue("ShiftID", DatabaseHelper.GetIdFromShift(shiftId));
}
```

### 3.3.5.2 Application layer

The application layer has the same structure as the Rooms object.

```
public class ObjSpooling
{
    //reference
    public int? ID { get; set; }
    //reference
    public int? KamerID { get; set; }
    //reference
    public int? ItemID { get; set; }
    //reference
    public int? ShiftID { get; set; }
    //reference
    public bool Genspoolid { get; set; }
    //reference
    public DateTime? Datum { get; set; }

    //reference
    public ObjSpooling(int? id, DateTime? datum, int? shiftID, int? kamerID, int? itemID, bool genspoolid)
    {
        ID = id;
        Datum = datum;
        ShiftID = shiftID;
        KamerID = kamerID;
        ItemID = itemID;
        Genspoolid = genspoolid;
    }

    //reference
    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.InsertSpooling((DateTime)Datum, (int)ShiftID, (int)KamerID, (int)ItemID, Genspoolid);
                affected = 1;
            }
            else
            {
                affected = Data.UpdateSpooling((int)ID, (DateTime)Datum, (int)ShiftID, (int)KamerID, (int)ItemID, Genspoolid);
            }
        }
        else
        {
            succeeded = false;
        }

        if (affected == 1)
        {
            succeeded = true;
        }
        else
        {
            succeeded = false;
        }
    }
}
```

### 3.3.5.3 Presentation layer

The "InitControls" method is often used in this screen, namely when a shift is selected, a Room is selected, and in opening this screen.

```
//reference
private void InitControls()
{
    long? idShift = Helper.GetIDFromCboSelectedValue(shiftComboBox);
    if (idShift != null)
    {
        kamerComboBox.Enabled = true;
    }
    else
    {
        kamerComboBox.Enabled = false;
        kamerComboBox.SelectedIndex = -1;
    }

    long? idKamer = Helper.GetIDFromCboSelectedValue(kamerComboBox);
    if (idKamer != null)
    {
        teSpoolenItemsGridView.Enabled = true;

        saveAndNextButton.Enabled = true;
        okButton.Enabled = true;
    }
    else
    {
        teSpoolenItemsGridView.Enabled = false;
        teSpoolenItemsGridView.DataSource = null;

        saveAndNextButton.Enabled = false;
        okButton.Enabled = false;
        IsInitDataGridView = false;
    }
}
```

### 3.3.6 Data-out

Export Overzicht

Filter:   
 Spreektaal:   
 Shift:   
 Kamer:   
 Begin datum: maandag 1 mei 2022   
 Eind datum: zondag 8 mei 2022   
 Ziekenhuis:   
 Exporteer naar Excel

Shift	Spreektaal	Kamer	Werk_jaar	Spreektaal	Geslacht	Tijdstip
Postdienst - Shift 1	Babekkenen	Kamer 1	W18 - 2022	Douches 2	<input checked="" type="checkbox"/>	2-05-2022 12:13
Postdienst - Shift 1	Babekkenen	Kamer 5	W18 - 2022	Lavabo	<input checked="" type="checkbox"/>	2-05-2022 12:13
Postdienst - Shift 1	Babekkenen	Kamer 5	W18 - 2022	WC 1	<input checked="" type="checkbox"/>	2-05-2022 12:13
Postdienst - Shift 1	C2	Kamer 8	W18 - 2022	Douches 1	<input checked="" type="checkbox"/>	2-05-2022 12:13
Postdienst - Shift 1	C2	Kamer 8	W18 - 2022	Douches 2	<input checked="" type="checkbox"/>	2-05-2022 12:13
Postdienst - Shift 5	Anatomie Pathologie	Kamer 2	W18 - 2022	Douches 1	<input checked="" type="checkbox"/>	2-05-2022 9:37
Postdienst - Shift 5	Anatomie Pathologie	Kamer 2	W18 - 2022	Douches 2	<input checked="" type="checkbox"/>	2-05-2022 9:37
Postdienst - Shift 5	Anatomie Pathologie	Kamer 4	W18 - 2022	WC 2	<input checked="" type="checkbox"/>	2-05-2022 9:37

8

Aan het wachten op een export


When the screen is opened you can filter by different properties of a flush. For example: show me all showers that have not been flushed between this particular time.

Export Overzicht

Filter:   
 Spreektaal:   
 Shift:   
 Kamer:   
 Begin datum: maandag 26 mei 2022   
 Eind datum: zondag 8 mei 2022   
 Ziekenhuis:   
 Exporteer naar Excel

Shift	Spreektaal	Kamer	Werk_jaar	Spreektaal	Geslacht	Tijdstip
Postdienst - Shift 1	C2	Kamer 8	W15 - 2022	Douches 1	<input checked="" type="checkbox"/>	5-04-2022
Postdienst - Shift 1	C2	Kamer 8	W15 - 2022	Douches 1	<input type="checkbox"/>	6-04-2022
Postdienst - Shift 1	C2	Kamer 8	W17 - 2022	Douches 1	<input type="checkbox"/>	26-04-2022 9:48
Postdienst - Shift 1	C2	Kamer 8	W18 - 2022	Douches 1	<input type="checkbox"/>	26-04-2022 9:44
Postdienst - Shift 1	C2	Kamer 8	W18 - 2022	Douches 1	<input type="checkbox"/>	27-04-2022 12:38
Postdienst - Shift 5	Anatomie Pathologie	Kamer 2	W16 - 2022	Douches 1	<input type="checkbox"/>	14-04-2022 9:46
Postdienst - Shift 5	Anatomie Pathologie	Kamer 2	W17 - 2022	Douches 1	<input type="checkbox"/>	26-04-2022 9:49
Postdienst - Shift 5	Anatomie Pathologie	Kamer 2	W18 - 2022	Douches 1	<input type="checkbox"/>	26-04-2022 9:44

After requesting the desired data, you can be asked to export the data to an Excel file. A loading bar shows the progress made in generating the Excel.

Schrijven van data op rij 8 van 18 

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

### 3.3.6.1 Data layer

The data is retrieved with the following query. The columns are formatted by joining multiple tables.

```

SELECT (SELECT ShiftName FROM Shift WHERE ShiftName = ShiftName) AS ShiftName,
       (SELECT SpoelPlaats FROM SpoelPlaats WHERE SpoelPlaats = SpoelPlaats) AS SpoelPlaats,
       (SELECT Kamer FROM Kamer WHERE Kamer = Kamer) AS Kamer,
       (SELECT WeekJaar FROM WeekJaar WHERE WeekJaar = WeekJaar) AS WeekJaar,
       (SELECT SpoelItem FROM SpoelItem WHERE SpoelItem = SpoelItem) AS SpoelItem,
       (SELECT Gespoeld FROM Gespoeld WHERE Gespoeld = Gespoeld) AS Gespoeld,
       (SELECT Datum FROM Datum WHERE Datum = Datum) AS Datum
FROM Shift, SpoelPlaats, Kamer, WeekJaar, SpoelItem, Gespoeld, Datum
WHERE ShiftName = ShiftName AND SpoelPlaats = SpoelPlaats AND Kamer = Kamer AND WeekJaar = WeekJaar AND SpoelItem = SpoelItem AND Gespoeld = Gespoeld AND Datum = Datum

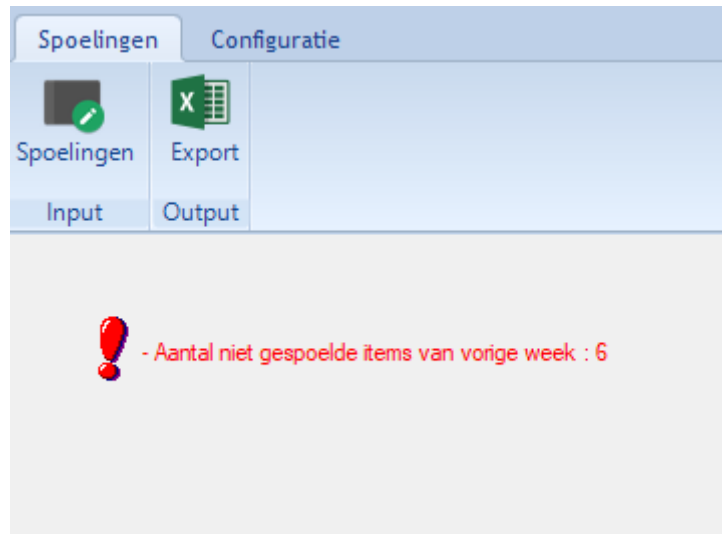
```

### 3.3.6.2 Application layer

For the export, no object is created so there is no Application layer present for this part.

### 3.3.6.3 Presentation layer

As with the configuration and data-in section, the structure of the presentation layer is the same here. The only noticeable thing in this piece is that we use an enumeration for the status of the items. Since there is an informational label in the main screen of the application that can be double clicked and directly filtered for items not flushed from the previous week, it should also be directly populated in the filters.



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

### 3.3.7 Excel creation

The Excel file is created in a static class. The Excel class also has its own helper class, which allows, for example, the creation of a method to thicken the border of a certain range in the file. Several methods have been created to change the background color, add text decoration and create different types of borders.

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

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

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

public static void ColorBackgroundCell(int row, int col, string data, Excel.Worksheet worksheet)
{
    switch (data)
    {
        case "Value":
            ExcelHelper.ColorBackgroundCell(row, col, Color.Salmon, worksheet);
            break;
        case "True":
            ExcelHelper.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 Helper classes

As already mentioned, several helper classes have been created. For example, there is a helper to populate drop-down lists.

```

//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";
}

```

Thus, there are also certain functions that only users in a certain Active Directory group are allowed to perform. Therefore, there is a method that looks at which AD group the user is from and what things this user may and may not see and/or perform.

```

//reference
public static void SelectADMembership()
{
    foreach (int group in Enum.GetValues(typeof(Enum.ADGroups)))
    {
        string name = Enum.GetName(typeof(Enum.ADGroups), 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;
    }
}

```

For example, the application uses a datagrid a lot. To give the columns a nice header, we have a function that makes the name nice for the user.



```

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;
}

```

Thus, there are many other small methods that perform simple tasks without having to write duplicate code.

### 3.3.9 Deploying the software

This is not something I was allowed to do myself but was allowed to be present for. The web page is released through IIS and the Forms application is put on the portal that the hospital uses to distribute software.

## 3.4 EXTRA ASSINGMENTS

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 change in a column in the database
- Importing a Excel file to create new intern accounts.
- Small analyses exercises

### 3.4.1 QR-code generator



[illegible]

For the hospital, the fields are actually already fixed, here the staff only has to choose from a selection list. These choice lists are laid out in a second sheet that has been made invisible to the schools.

Type	Department	SubType	Dienst
Student	Andere	ArtsSubtype	VerpleegkundigeDienst
	Arts	Dokter Stager	Adjunct Directie Verpleging
Paramedici	Paramedici	Dokter Assistent	Chirurgisch Daghospitaal
	Verpleegkundige	Student Met Medicatiebeheer	Diabetes Consultatie
			Dialyse
			Dienst 1
			Dienst 10
			Dienst 2
			Dienst 4
			Dienst 5
			Dienst 6
			Dienst 7
			Dienst 8
			Dienst 9
			Dienst Intensieve
			Dienst OK
			Dienst Pediatrie
			Dienst SP
			Dienst Spoed
			Dienst Vliegende
			Loopwacht
			Oncologisch Daghospitaal
			Slaaplabo
			Dienst Materniteit
			Geriatrisch Daghospitaal
			Mobiele Equipe kritieke Diensten
			Dokters Allogie
			Dokters Anaesthesie
			Dokters Anathomopathologie
			Dokters Dermatologie
			Dokters Endoscopie
			Dokters Gynaecologie
			Dokters Heelkunde
			Dokters Inwendige
			Dokters Klinisch Labo
			Dokters MKG
			Dokters Nefrologie
			Dokters Neurochirurgie
			Dokters Neurologie
			Dokters Nucleaire
			Dokters Oftalmologie
			Dokters Oncologie
			Dokters ORL
			Dokters Orthopedie
			Dokters Pediatrie
			Dokters Radiologie
			Dokters Revalidatie
			Dokters Spoedgevallen
			Dokters Stomatologie
			Dokters Urologie
			Psychiatrie
			Dokters Pneumologie
			Dokters Geriatrie
			Dienst Gastro- enterologie
			Anapath
			Dienst Zingeving
			Dietisten
			Endoscopie
			Gipskamer
			Labo
			Medische Registratie
			Nucleaire
			Palliatief Support Team
			Pijnkliniek
			Poetsdienst
			Poli Gynaecologie
			Poli Heelkunde - Orthopedie
			Poli Inwendige
			Poli Neuro
			Poli Onco
			Poli ORL
			Poli Pediatrie
			Poli Urologie
			Psychologen
			Radiologie
			Dienst Revalidatie
			Secretariaat Vliegende Ploeg
			Sociale Dienst
			Sterilisatie
			Tabakologen
			Vervoer
			Wondzorg
			Dienst Ergo
			Poly Gastro
			Accreditatie
			Apotheek
			Archief
			Biotechnische Dienst
			Boekhouding
			Directie
			Directie Secretariaat
			Economaat
			Financieel Manager
			Financiële Dienst
			Personeelsdienst
			Informatica
			Informatieveiligheid
			Keuken
			Linnenkamer
			Magazijn
			Opname
			Preventieadviseur
			Tarificatie - Facturatie
			Technische Dienst
			VTO
			Communicatie

At the request of the mentor, a manual was created so that something can be adjusted later. So it is simple to undo my settings, adjust something and set it back.

### 3.4.4 Analyses exercises

The mentor sometimes sent an email with a text that needed to be analyzed. He then expected a diagram or clarification about what is present of objects, what properties these objects have, ... .

## 4 SOFT SKILLS

### 4.1 COMMUNICATION

During my internship, I did not work from home for a single day. If there was a problem I could just stand up and tell the mentor. If he had time, he would come and think with me or administer help. If the mentor was not at the office, he could always be reached via email. We also had 2 team meetings to see how things were going.



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.

### 4.2 SCHEDULING

An Excel file was created in which I could make a schedule. This way the mentor could always see what I was doing and what I planned to do

Item	Uitvoeren	toet		2020/03/03	2020/03/04	2020/03/05	2020/03/06	2020/03/07	2020/03/08
3-tier architecture & Etlten		2		Voorontwerp deel 1	Bevallen	Barcode test	Barcode test	Barcode test	Barcode test
Aanpak op Thomas More		6		Voorontwerp deel 2	3-tier architecture & Etlten	Barcode test	Barcode test	Barcode test	Barcode test
Barcode controle app		4		Naamgeving deel 1	Opdracht leren kennen / verspreiden	Barcode test	Barcode test	Barcode test	Barcode test
Barcode test		32		Naamgeving deel 2	Opdracht leren kennen / verspreiden	Barcode test	Barcode test	Barcode test	Barcode test
Barcode test in E-commerce/Etlten		6							
Barcode test documenteren		2		Voorontwerp deel 1	Barcode test	Barcode test	Barcode test	Barcode test	Barcode test
Barcode test toepassen op speediketten		8	30	Voorontwerp deel 2	Barcode test	Barcode test	Barcode test	Barcode test	Barcode test
Eng dag		8		Naamgeving deel 1	Opdracht leren kennen / verspreiden	Barcode test	Barcode test	Barcode test	Barcode test
Eng Free		2		Naamgeving deel 2	Opdracht leren kennen / verspreiden	Barcode test	Barcode test	Barcode test	Barcode test
configuratie/klanten informatie maken		30							
configuratie/klanten informatie maken		6		Voorontwerp deel 1	Barcode test	Barcode test	Barcode test	Barcode test	Barcode test
Documenten voor Thomas More opstellen		30		Voorontwerp deel 2	Barcode test	Barcode test	Barcode test	Barcode test	Barcode test
Herstructureren expert & kleine details		16	24	Naamgeving deel 1	Opdracht leren kennen / verspreiden	Barcode test	Barcode test	Barcode test	Barcode test
Import Excel nieuwe medewerkers		12	8	Naamgeving deel 2	Opdracht leren kennen / verspreiden	Barcode test	Barcode test	Barcode test	Barcode test
Maakmaken/deployen speedikette		0							

next. There was also a column with all the subjects. Here I could visualize how long I expected to work on them and how long I actually worked on them.