**Opdrachten week 1**

**Opdracht 1: Taakverdeling voor deze week**

Met deze opdracht ga je de taken verdelen. Als je in groepen werkt is het belangrijk goede afspraken te maken. Met B5 ga je iedere week van (dus de eerste les) gezamenlijk het werk verdelen. Dit is altijd opdracht 1 van de week. Dus lees de opdrachten goed door en maak daarna de verdeling.

Deze opdracht doe je gezamenlijk. Je vervangt teamlid 1 t/m 3 door jullie namen. Als iets ontbreekt vul je dit aan.

Rick = Teamlid 1

Ilyas = Teamlid 2

Martijn = Teamlid 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Omschrijving | Datum | Duur | Naam teamlid |
| 1 | Inlezen opdracht | 31-05-2021 | 20min | allemaal |
| 2 | Taakverdeling van de opdracht | 31-05-2021 | 30min | allemaal |
| 3 | Uitzoeken wat de applicatie precies doet | 31-05-2021 | 30min | allemaal |
| 4 | Uitzoeken wat de applicatie precies doet formulier 1 | 31-05-2021 | 10min | Rick |
| 5 | Uitzoeken wat de applicatie precies doet formulier 2 | 31-05-2021 | 10min | Ilyas |
| 6 | Uitzoeken wat de applicatie precies doet formulier 3 | 31-05-2021 | 10min | Martijn |
| 7 | Uitzoeken wat de applicatie precies doet formulier 4 | 31-05-2021 | 10min | Rick |
| 8 | Uitzoeken wat de applicatie precies doet formulier 5 | 31-05-2021 | 10min | Ilyas |
| 9 | Uitzoeken wat de applicatie precies doet samenvoegen | 31-05-2021 | 10min | Martijn |
| 10 | Structuur van de applicatie bepalen |  |  | Allemaal |
| 11 | Eens door de code heen gaan |  |  | Rick |
| 12 | Eens door de code heen gaan |  |  | Ilyas |
| 13 | Eens door de code heen gaan |  |  | Martijn |
| 14 | Fouten in de applicatie |  |  |  |
| 15 | Fouten in de applicatie |  |  |  |
| 16 | Fouten in de applicatie |  |  |  |

**Opdracht 2: Uitzoeken wat de applicatie precies doet en hoe de formulieren samenhangen**

Om een goed beeld te hebben van de applicatie gaan jullie bekijken welke formulieren (5 in aantal) er allemaal al gemaakt zijn. Per formulier maak je dan een beschrijving met wat het hoofddoel is van het formulier. Een beschrijving van een formulier bestaat minimaal uit 5 regels. Deze opdracht doe je gezamenlijk. Let op; om de applicatie te starten heb je een wachtwoord nodig. Om deze te achterhalen zul je moeten uitzoeken wat het wachtwoord is.

**Formulier 1:**

|  |  |
| --- | --- |
| **Formuliernaam** | *Bekijk in Visual studio hoe het formulier word genoemd.* |
| **Omschrijving** | *Bekijk wat het hoofddoel is van het formulier. Probeer met elkaar goed vast te leggen wat het precies allemaal doet zonder in technische details te duidelijken. Gebruik dus zo min mogelijk technische woorden, maar woorden van de gebruiker.* |

**Formulier 2:**

|  |  |
| --- | --- |
| **Formuliernaam** | *Bekijk in Visual studio hoe het formulier word genoemd.* |
| **Omschrijving** | *Bekijk wat het hoofddoel is van het formulier. Probeer met elkaar goed vast te leggen wat het precies allemaal doet zonder in technische details te duidelijken. Gebruik dus zo min mogelijk technische woorden, maar woorden van de gebruiker.* |

**Formulier 3:**

|  |  |
| --- | --- |
| **Formuliernaam** | *KlantForm.CS[Design]* |
| **Omschrijving** | *Dit is de soort van Rekenmachine van de applicatie. Dit scherm laat zien hoeveel er betaald moet worden. Hoeveel geld de kassière heeft ontvangen en wat de kassière uiteindelijk terug moet geven. Links word een tabel weergegeven met alle producten die gescand zijn, met de prijs erachter. Dit wordt automatisch toegepast in de tekstvlakken die rechts staan.* |

**Formulier 4:**

|  |  |
| --- | --- |
| **Formuliernaam** | *Bekijk in Visual studio hoe het formulier word genoemd.* |
| **Omschrijving** | *Bekijk wat het hoofddoel is van het formulier. Probeer met elkaar goed vast te leggen wat het precies allemaal doet zonder in technische details te duidelijken. Gebruik dus zo min mogelijk technische woorden, maar woorden van de gebruiker.* |

**Formulier 5:**

|  |  |
| --- | --- |
| **Formuliernaam** | *Bekijk in Visual studio hoe het formulier word genoemd.* |
| **Omschrijving** | *Bekijk wat het hoofddoel is van het formulier. Probeer met elkaar goed vast te leggen wat het precies allemaal doet zonder in technische details te duidelijken. Gebruik dus zo min mogelijk technische woorden, maar woorden van de gebruiker.* |

**Opdracht 3: Structuur van de applicatie bepalen**

Als je begint aan een bestaand project is het zaak om eerst te weten hoe het programma is opgebouwd. In de vorige opdracht hebben we naar de individuele formulieren gekeken. Nu gaan we kijken naar de samenhang. Met deze opdracht ga je daarmee aan de slag.

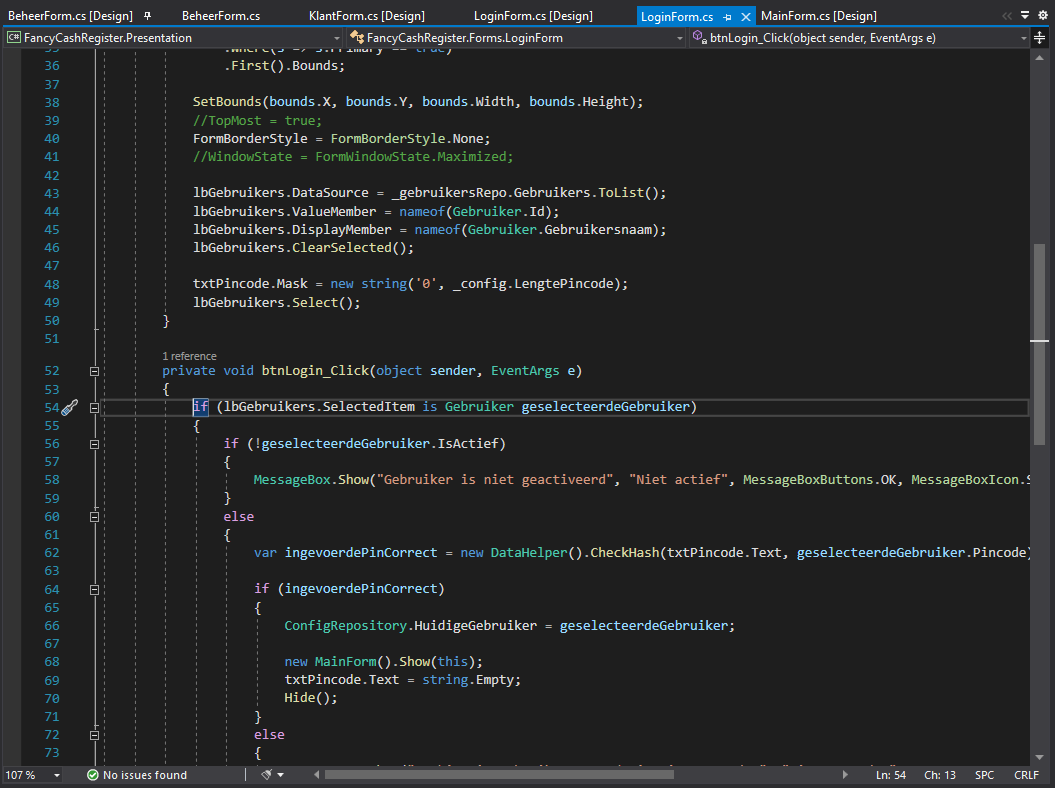
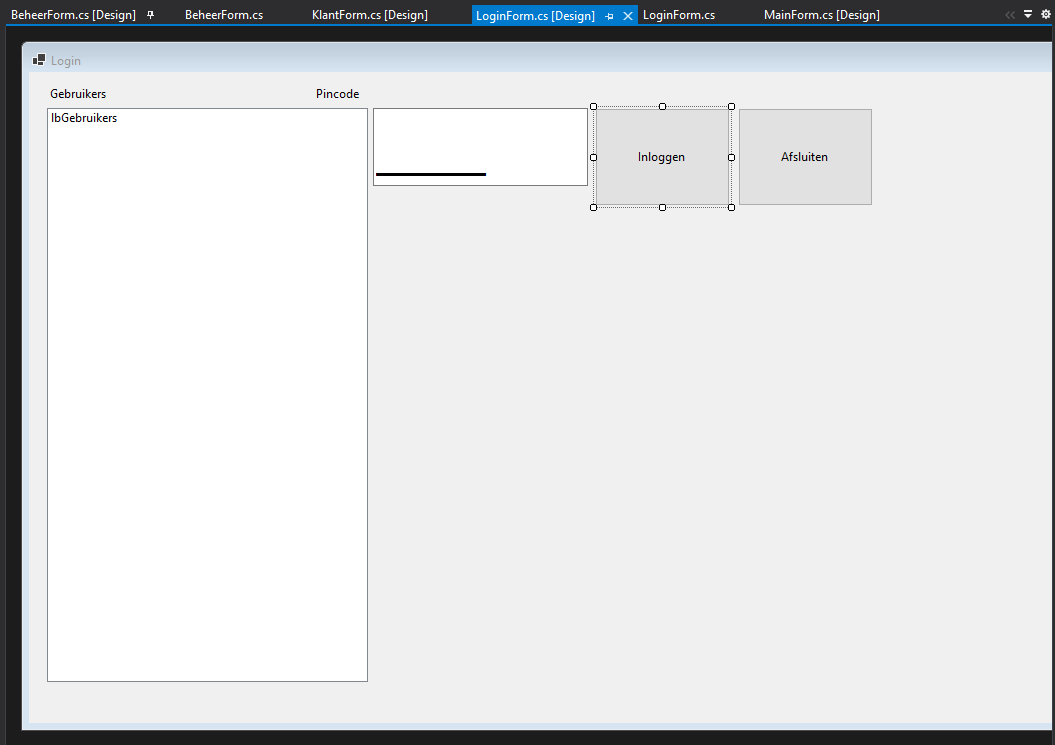
Om de samenhang van de formulieren goed te begrijpen ga je kijken hoe de formulieren elkaar aanroepen. Maak van hiervan een sitemap. Hieronder zie je het eerste opzet.

Uitwerking (aanvullen):

LoginForm

MainForm

TIP:



Hier zie de code dat er voor zorgt dat de mainform word aangemaakt

Dubbel klik op de volgende knop op het loginform

[Voorbeeld sitemaps](https://www.google.com/search?q=sitemap+examples&sxsrf=ALeKk010fDg6mzwSPVmDztMEesPx_SPx_w:1622367082559&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjwvpqtjPHwAhVM8BQKHVCGAiQQ_AUoAXoECAEQAw&biw=1718&bih=1287)

**Opdracht 4: Eens door de code heen gaan**

We weten nu wat de formulieren globaal doen en hoe deze samenhangen. Nu gaan we kijken naar wat specifieke code. Er zullen delen zijn die je niet nog niet beheerst maar wel zal begrijpen wat het doet. Als programmeur (en zeker niet alleen als je begint) lopen we daar best vaak tegen aan. En als de methode of class goed is gemaakt hoeven we ook niet echt details te weten. Eigenlijk hoeven we alleen te weten wat we in de methode moeten stoppen en wat er dan uitkomt. Hoe iets in detail gebeurd is dan niet belangrijk. Bijvoorbeeld Console.WriteLn(“Hello World”); weten we niet hoe de methode het voor elkaar krijgt om deze letters op het beeld te tonen. Maar we weten dat de methode een string moet ontvangen en deze dan vervolgens op de positie van cursor op het beeld zet.

Toch zijn er ook methode die we wel moeten kunnen begrijpen binnen een applicatie. Hieronder zijn een aantal methoden, stukjes code etc uit de applicatie. Aan jullie als groep om te kijken hoe deze werken en hiervan te beschrijven wat er gebeurd. Ieder gaat individueel aan de slag. Uiteindelijk geven jullie elkaar uitleg van wat je stukje code precies doet.

1. De methode Reset()

Beschrijf wat de onderstaande functie doet. Je kunt de functie terugvinden op BetalenForm.cs. Vind je deze functie misschien ook op andere formulieren terug? Waarom is het handig om een dergelijke methode te hebben per formulier?

public void Reset()

{

txtTeBetalen.Text = $"{0:c2}";

txtOntvangen.Text = $"{0:c2}";

txtTeruggave.Text = $"{0:c2}";

}

Uitwerking:

1. Een if-statement

Soms lopen we tegen methodes, classes aan die we nog niet kennen. Nu kunnen we eenvoudig informatie opzoeken met behulp van visual studio zelf. Zoek het onderstaande code op. Je vindt de code in BeheerForm.cs regel 123. Plaats je cursor op SelectedItem en druk op F1. Vervolgens kom je uit op de microsoft site waarin de methode word uitgelegd.

Aan jullie de opdracht om de onderstaande if statement uit te schrijven in gewone taal. Hou ook rekening met de AND en OR die worden gebruikt.

if (lbGebruikers.SelectedItem != null ||

string.IsNullOrWhiteSpace(txtPincode.Text) == false

&& (txtPincode.Text.Length < \_config.LengtePincode

|| Regex.IsMatch(txtPincode.Text, "[^0-9]")))

{

MessageBox.Show("Pincode voldoet niet aan de eisen", "Pincode ongeldig", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

Uitwerking:

1. Inloggen van een gebruiker

Als een gebruiker wilt inloggen drukt hij op de knop btnLogin. De onderstaande code word dan uitgevoerd. Voor deze methode geldt dat je regel voor regel moet beschrijven wat er nu exact gebeurd. Maak gebruik van de Microsoft website voor extra ondersteuning.

private void btnLogin\_Click(object sender, EventArgs e)

{

if (lbGebruikers.SelectedItem is Gebruiker geselecteerdeGebruiker)

{

if (!geselecteerdeGebruiker.IsActief)

{

MessageBox.Show("Gebruiker is niet geactiveerd", "Niet actief", MessageBoxButtons.OK, MessageBoxIcon.Stop);

}

else

{

var ingevoerdePinCorrect = new DataHelper().CheckHash(txtPincode.Text, geselecteerdeGebruiker.Pincode);

if (ingevoerdePinCorrect)

{

ConfigRepository.HuidigeGebruiker = geselecteerdeGebruiker;

new MainForm().Show(this);

txtPincode.Text = string.Empty;

Hide();

}

else

{

MessageBox.Show("Combinatie gebruikersnaam / pin niet gevonden", "Niet gevonden", MessageBoxButtons.OK, MessageBoxIcon.Stop);

}

}

}

}

Uitwerking:

**Opdracht 5: Fouten in de applicatie**

Je hebt misschien al gemerkt wanneer je het programma start dat er nog fouten in zitten. Aan jullie de taak deze vast te leggen.

Het vastleggen doe je van een fout doe je in de onderstaande tabellen. Beschrijf in totaal 3 fouten.

Fout 1:

|  |  |
| --- | --- |
| Formulier |  |
| Handelingen vooraf aan de foutmelding |  |
| Foutmelding |  |
| Analyse |  |

Fout 2:

|  |  |
| --- | --- |
| Formulier |  |
| Handelingen vooraf aan de foutmelding |  |
| Foutmelding |  |
| Analyse |  |

Fout 3:

|  |  |
| --- | --- |
| Formulier |  |
| Handelingen vooraf aan de foutmelding |  |
| Foutmelding |  |
| Analyse |  |