**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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | 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 | Teamlid 1 |
| 5 | Uitzoeken wat de applicatie precies doet formulier 2 | 31-05-2021 | 10min | Teamlid 2 |
| 6 | Uitzoeken wat de applicatie precies doet formulier 3 | 31-05-2021 | 10min | Teamlid 3 |
| 7 | Uitzoeken wat de applicatie precies doet formulier 4 | 31-05-2021 | 10min | Teamlid 1 |
| 8 | Uitzoeken wat de applicatie precies doet formulier 5 | 31-05-2021 | 10min | Teamlid 2 |
| 9 | Uitzoeken wat de applicatie precies doet samenvoegen | 31-05-2021 | 10min | Teamlid 3 |
| 10 | Structuur van de applicatie bepalen |  |  | Allemaal |
| 11 | Eens door de code heen gaan |  |  | Teamlid 1 |
| 12 | Eens door de code heen gaan |  |  | Teamlid 2 |
| 13 | Eens door de code heen gaan |  |  | Teamlid 3 |
| 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** | *loginForm.cs* |
| **Omschrijving** | *Hier kunnen we als gebruiker ons account selecteren. Op het moment dat er een gebruiker is geselecteerd weet het programma welk wachtwoord hij moet checken. Het wachtwoord kunnen we invoeren in de box met de zwarte regelen, het wacht woord is 4 cijfers lang en word met een “\*\*\*\*” weer gegeven. Vervolgens klik je op inloggen en gaat ie de gebruiker inloggen. Om de applicatie af te sluiten drukken we op de knop “afsluiten”.* |

**Formulier 2:**

|  |  |
| --- | --- |
| **Formuliernaam** | *klantForm.cs* |
| **Omschrijving** | *Op het klanten form zien we een aantal dingen: een (lege) lijst waarin transactie komen, verschillende plaatjes de om de zoveel seconde verandere, en 3 boxen waarin een bedrag in gevoerd kan worden. Het laatste werkt nog niet!* |

**Formulier 3:**

|  |  |
| --- | --- |
| **Formuliernaam** | *mainForm.cs* |
| **Omschrijving** | *Dit is het algemene ui van de apllicatie, hier worden de meeste dingen gedaan. Dit scherm is op te delen in vier stukken:*  *-categorien;*  *-producten;*  *- de knoppen en de vakken voor de bedragen;*  *-producten in order(in de bestelling).*  *categorien:*  *het stuk op het scherm dat gaat over de categorieen wordt in geladen vanuit de database. In de database staat een aparte tabel die specifiek gaat over de verrshillende categorien. De gebruiker kan een van deze selecteren om vervolgens de producten uit die categorie te kunnen zien.*  *Producten:*  *Dit blok kunnen we ook weer opdelen in 3 stukken:*  *- categorie-id*  *Dit is een nummer die meegegeven word om later te kunnen bepalen waarbij het product hoor, want appeltaart en koffie gaan goed samen maar vallen niet onder de zelfde soort producten bijv. en wordt ook opgehaald uit de database*  *-Naam*  *Dit spreekt voor zich, in deze colom staat de naam van het product en wordt ook opgehaald uit de database*  *-stukprijs*  *In deze colom staat de prijs van het product en wordt ook opgehaald uit de database*  *de knoppen en de vakken voor de bedragen:*  *in dit stuk van dit scherm word weergegeven wat de naam van de gebruiker is, daaronder staat het aantal producten dat de gebruiker heeft geselecteerd, daaronder staat de prijs per stuk, daaronder staat het totaal bedrag van de geselecteerde producten zodat je kan zien of je bijv genoeg geld hebt voor de aankoop.*  *Dan hebben we nog 3 knoppen over. Dit zijn de knoppen om de producten toe tevoegen en weg te halen uit de lijst met de bestelling van de gebruiker. Ook hebben we nog een beheer knop, hier later meer over, zie formulier 4*  *-producten in order(in de bestelling):*  *Dit is het laatste blok van dit scherm en deze is vrij vanzelf spreekend. Dit stuk laat zien welke producten de gebruiker heeft geselecteerd en laat alle bijbehorende informatie zien zoals: aantal, totaalprijs, naam van de gebruiker en de prijs per stuk. Onder dit blok word de totaal prijs weergegeven en kun je er voor kiezen om te klikken op “betalen”.* |

**Formulier 4:**

|  |  |
| --- | --- |
| **Formuliernaam** | *beheerForm.cs* |
| **Omschrijving** | *Het beheer scherm is voor de meeste gebruikers niet beschikbaar of in ieder geval niet te gebruiken. Dit secherm kunnen we ook weer opdelen in verschillende delen, we hebben 3 taps: 1 gebruikers, 2 producten, 3 applicatie*  *Op dit moment werkt alleen de gebruikers tab nog. Dit is dan ook de gene waar ik het over ga hebben*  *We zien hier in princiepe 2 stukken:*  *-de knoppen aan de linker kant*  *De knop terug*  *De knop afmelden. Om met een andere gebruiker in te loggen*  *De knop afsluiten. Om het programma af te sluiten*  *-en het blok met gebruikers informatie*  *Hier zien we wat meer intressante dingen. Als we van links naar recht gaan kijken zien we eerst weer een lijstje met gebruikers. Zodra we dit scherm openen dan zien we onze eigen gebruiker met al zijn info. Zodra we op een andere gebruiker klikken kunnen we zien wat hij of zij kan doen en wat de gebruiker zijn functie. Hier kunnen mogelijk beperkingen worden aan gegeven of informatie veranderd worden. Als we iets veranderen dan klikken we op “opslaan” en dan worden onze nieuwe gegevens aangepast.* |

**Formulier 5:**

|  |  |
| --- | --- |
| **Formuliernaam** | *betalenForm.cs* |
| **Omschrijving** | *Dit is het laatste form wat ik ga om schrijven. dit scherm gaat over het betalen van de producten. Als we gaan kijken hebben we weer een lijst met producten die worden weergegeven. Daar naast van boven naar beneden hebben we het totaal bedrag van de de geselecteerde producten, daaronder geven we aan hoeveel geld er gegeven is en dan daaronder word berekend hoeveel de gebruiker terug krijgt.*  *Verder kunnen we ook een kortings code gebruiken en die toevoegen* |

**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

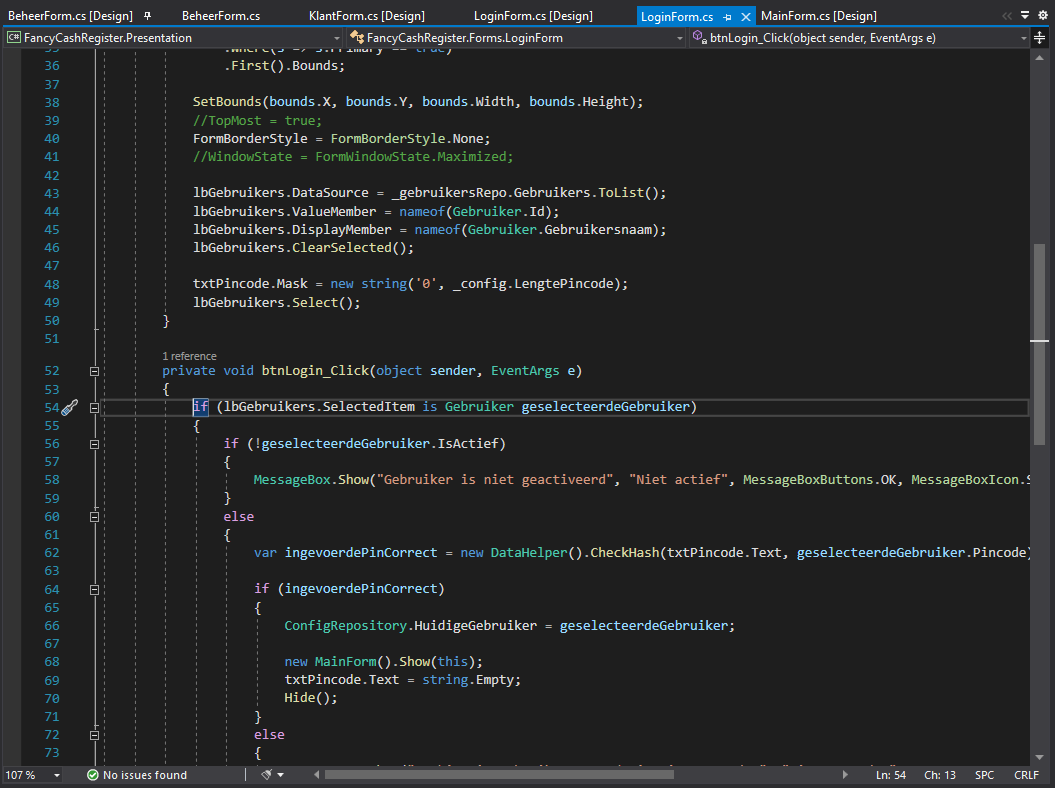
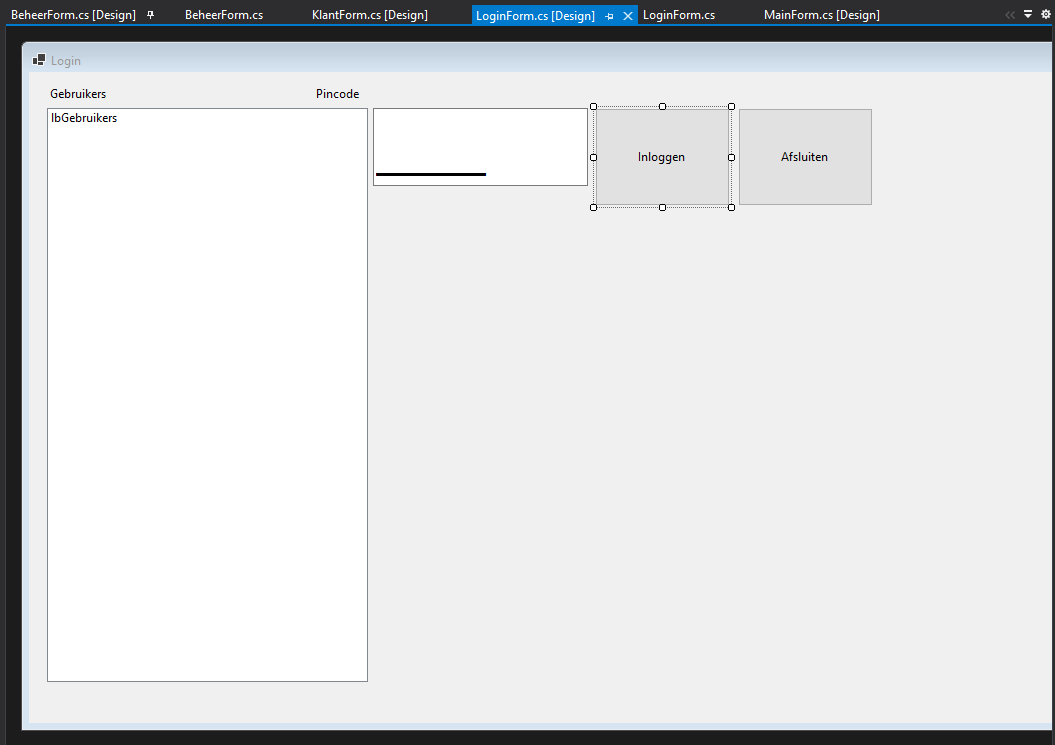
MainForm

betalenFrom

BeheerFormk

klantenForm

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:

Alle 3 variabelen worden naar 0 gezet om ze te reseten

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. Als het label gebruiker niet gelijk is aan niks **of**
2. datatype string methode is niet niks of witruimte in textbox pincode met daar in tekst gelijk staat aan niet waar **en**
3. tekst pincode tekst en daarvan de lengte kleiner is dan de geconfigureerde lengte van de pin code **of**
4. matched de info in textbox pincode en daarvan de tekst tussen 0 t/m 9 **dan doe:**
5. toon een bericht in een pop up venster die zegt: “Pincode voldoet niet aan de eisen", "Pincode ongeldig",” in het venster zie je een knop met ok erop en een error icoon
6. 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:

1. wanneer btnLogin\_Click word ingedrukt

2. als de geselecteerde gebruiker in de database voor komt

3 als de gebruiker niet actief is dan krijgt hij het bericht “Gebruiker is niet geactiveerd", "Niet actief”

4. anders gaat het programma kijken of de pincode goed is

5. als hij goed is gaat het programma een anderen form openen omdat je bent ingelogd

6. als de pincode fout is gaat hij dit bericht showen: “Combinatie gebruikersnaam / pin niet gevonden", "Niet gevonden”

**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 | Het programma crashed wanneer je betalen klikt |
| Handelingen vooraf aan de foutmelding | Het programma crashed gewoon |
| Foutmelding | System.argumentNullException: ‘value cannot be null. Arg\_ParamName\_Named’ |
| Analyse | We moeten deze value later iets ander maken dan null zodat het programma weer werkt |

Fout 2:

|  |  |
| --- | --- |
| Formulier | Kortings code werkt nog niet |
| Handelingen vooraf aan de foutmelding | Is nog niet gemaakt |
| Foutmelding | - |
| Analyse | Moeten we nog in bouwen |

Fout 3:

|  |  |
| --- | --- |
| Formulier | Producten laden niet in bij beheer |
| Handelingen vooraf aan de foutmelding | Kijken of hij de goede gegevens pakt uit de database |
| Foutmelding | geen |
| Analyse | Moeten we nog gaan maken |