Tekst Calc4You Project Scrum Groep 5

Inhoud

[Taakverdeling Dia’s 2](#_Toc27382894)

[Begin 2](#_Toc27382895)

[Ontwerp 2](#_Toc27382896)

[GitHub 2](#_Toc27382897)

[Sprints 3](#_Toc27382898)

[Rekenmachine code 4](#_Toc27382899)

[Taakverdeling 5](#_Toc27382900)

[Database 5](#_Toc27382901)

[Gezichtsherkenning 5](#_Toc27382902)

[Verbeterpunten 5](#_Toc27382903)

[Samenvatten 5](#_Toc27382904)

# Taakverdeling Dia’s

Julian : Begin + Sprints

Jesse : Ontwerp + Taakverdeling

Lucas : GitHub + Database

Sam : Rekenmachine code + Gezichtsherkenning

Bram : Verbeterpunten + Samenvatten

# Begin

# Ontwerp

# GitHub

**Lucas** : In de warming up week werd de GitHub niet gebruikt omdat niemand wist hoe het allemaal precies werkte. Toen we waren begonnen met de eerste sprints kwamen we er achter dat het delen van bestanden niet gebeurde of het ging via een omweg. Na een aantal video’s op YouTube kwam ik er achter wat de basis commands waren en wat ze precies deden. Na het zien van de video’s had ik een GitHub repository aangemaakt. Toen het allemaal goed werkte op mijn laptop heb ik de basis commands voorgelegd aan de groep. De eerste paar dagen verliepen niet zo soepel omdat bijvoorbeeld de commands in de verkeerde volgorde werden uitgevoerd. Toen heb ik een GitHub readme aangemaakt met alle commands en de juiste volgorde ervan. Na de 2e week kwam ik erachten dat de .vs mapjes bij elke computer anders was. Hierdoor moest je bij elke push of pull de online map en de lokale map gaan mergen. Dit koste veel tijd en moeite dus heb ik een .gitignore toegevoegd. In dit bestand geef je aan wat GitHub niet moet pushen of pullen. Ook heb ik tussentijds een readme toegevoegd aan de home page van onze repository waarop alle informatie staat over ons project.

# Sprints

**Julian** : We zijn in Sprint 2 begonnen met het project Calc4You, we moesten aan een aantal functionaliteiten voldoen:

1. De applicatie moet kunnen delen, vermenigvuldigen, optellen en aftrekken.

2. De huidige invoer moet gewist kunnen worden.

3. Een positief getal moet kunnen worden omgezet naar een negatief getal.

4. Euro's moeten kunnen worden omgerekend naar dollars

5. Decimaal getal moet kunnen worden omgerekend naar een binaire waarde.

6. De modulo (rest na deling) moet uitgerekend kunnen worden.

7. Decimaal getal moet kunnen worden omgerekend naar een hexadecimale waarde

Op te leveren producten:

• Complete user stories van alle gemaakte onderdelen.

• Een correct werkende applicatie waarin alle functionaliteiten zijn verwerkt.

• Alle functionaliteiten moeten in aparte methodes zijn verwerkt.

• Nette code die goed gedocumenteerd is.

• Een repository in GITHUB.

• Een persoonlijke private repository waarin je een backup voor codes hebt staan.

DEEL 2 CALC4YOU uitgebreid

1. Een gemaakte berekening moet kunnen worden opgeslagen in de database.

2. De laatste 10 opgeslagen berekeningen moeten kunnen worden opgehaald uit de database. De gebruiker moet een van deze berekeningen kunnen selecteren om deze opnieuw te laten berekenen.

3. Pas de applicatie zo aan dat je toegang krijgt tot de applicatie wanneer je 15 jaar of ouder bent. De leeftijd wordt gecontroleerd doormiddel van gezichtsherkenningssoftware. Volg hiervoor de tutorials die zijn opgenomen bij het lesmateriaal.

# Rekenmachine code

**Sam** : De code voor de Rekenmachine werkt als volgt:

Als de gebruiker op een getal klikt wordt de functie van die knop aangeroepen. Die functie zorgt ervoor dat het getal waarop is geklikt de ‘invoer’ wordt. Als er dan op een operatie zoals: plus, min, keer, delen is geklikt wordt de invoer opgeslagen in de variabele ‘getal 1’ en de operatie (+ - \* /) wordt dan opgeslagen in de variabele ‘operatie’ en daarna wordt de variabele ‘invoer’ weer leeg gemaakt.

Als de gebruiker dan op een tweede getal klikt wordt dat getal opnieuw opgeslagen in de variabele ‘invoer’. Wanneer er op de is-knop wordt gedrukt slaat het programma de invoer op in de variabele ‘getal 2’.

Als je kommagetallen wilt gebruiken in het programma kan dat ook. Wanneer de gebruiker op de komma-knop klikt wordt in de variabele ‘invoer’ een punt neergezet en wordt daarna ge-convert naar een double.

Als er op de is-knop is geklikt wordt probeert het programma van de variabelen ‘getal 1’ en ‘getal 2’ double’s te maken. Dit doet het door “TryParse” te gebruiken. Hierdoor worden de string’s ‘getal 1’ en ‘getal 2’ omgezet naar de double’s ‘nummer 1’ en ‘nummer 2’.

Wanneer dat is gelukt controleert het programma welke operatie er is gebruikt door het ‘if’ statement te gebruiken. Als hij de goede operatie heeft gevonden voert hij de som uit. Bijvoorbeeld als de gebruiker op de plus-knop heeft geklikt maakt het programma de som ‘nummer 1’ + ‘nummer 2’.

Ook kan er met deze rekenmachine getallen naar binair, hexadecimaal, en dollars worden omgerekend.

Als de gebruiker op de binair-knop klikt wordt de invoer omgezet naar een string waarbij er maar 2 getallen mogen worden gebruikt. 0 en 1.

Bij hexadecimaal werkt dat ongeveer hetzelfde maar dan met 16 getallen.

En bij de euro naar dollar knop wordt de invoer naar een double ge-convert en daarna keer de wisselkoers gedaan.

# Taakverdeling

# Database

**Lucas** : We gebruiken een lokale database om onze informatie op te slaan. In onze code staan 3 verschillende connection strings. De eerste connection string zit in een methode zodat we de lijst kunnen refreshen als er aanpassingen zijn in de database met simpelweg een lijn met code. De tweede connection string zit in de “=” knop, als je op de knop klikt voeg hij de berekening en de uitkomst toe aan de SQL Database table. De laatste connection string is niet echt noodzakelijk. We hebben een leeg knop toegevoegd die leegt de hele database table en dus ook de lijst. Toen de lijst en de database waren gesynchroniseerd. Kwam ik erachter dat ze niet op de goede volgorde stonden. Door een automatische ID Identity Specification maakt de bij elke database row een nieuwe ID aan doormiddel van nummers. Toen heb ik alleen nog een ORDER BY statement toegevoegd.

# Gezichtsherkenning

**Sam** : Als de gebruiker de rekenmachine opstart krijgt hij/zij een scherm te zien met twee knoppen: Kies foto en Verder. De verder knop is standaard uitgeschakeld en schakelt alleen aan wanneer de gebruiker ouder is dan 15. Dit kan het programma herkennen wanneer de gebruiker een foto instuurt via de Kies foto knop. Als dat is gebeurt wordt er als eerst gekeken of het bestand wil een goed .png of .jpg bestand is. Daarna wordt er een request gemaakt naar de servers van Microsoft. Via de ‘requestParameters’ variabele kunnen we aangeven wat voor een details we willen weten. Zoals: leeftijd, emoties, make-up, geslacht, enz. Dat wordt dan in een JSON string teruggegeven. Het programma heeft alleen maar het getal nodig uit die string. Het krijgt dat door de string te parsen. Als dat getal boven de 15 is wordt de verder knop ingeschakeld.

# Verbeterpunten

# Samenvatten