Elien Knockaert
1NMCT3

Monopoly: Howest edition

Monopoly: Howest edition is een project die zich baseert op de originele Monopoly maar gebruik maakt van RFID kaarten in plaats van briefjes geld.

De belangrijkste sensor is dus de RFID reader. Het is de bedoeling om RFID kaarten in te lezen met de reader en door de verschillende ID's per kaart wordt er een onderscheidt gemaakt of het om een bankkaart, eigendomskaart, station, etc. gaat.

De dobbelsteen is een bakje die kan doorgegeven worden, namelijk de speler die aan de beurt is drukt op een knop en er komt op een 4 digit 7 segment display 2 random getallen tevoorschijn van één tot en met zes. Daarnaast staat er een LCD naast het bord om te zien welke speler welk saldo nog heeft.

Tot slot is er een website waar je een nieuw spel kan starten, daar geef je op wat de namen van de spelers zijn welke pionnen ze gebruiken en wat de groepsnaam is. Op de website kan het spelbord met de legende opgevraagd worden alsook de geschiedenis van het spel zelf.

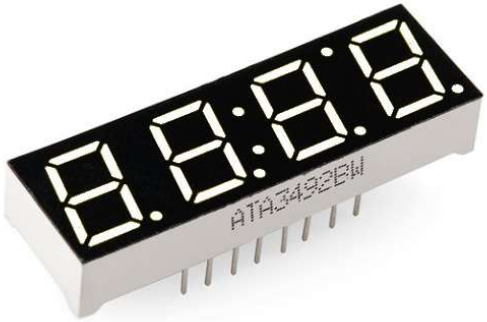
Step 1: Let's Get Started!

In de bijlage vindt je een excel bestand, BOM (bill of materials). Daar staan de materialen die je nodig hebt, waar je ze kan vinden en hoeveel ze kosten. Ook staat er in het bestand hoeveel het eigenlijke project ongeveer zal kosten.

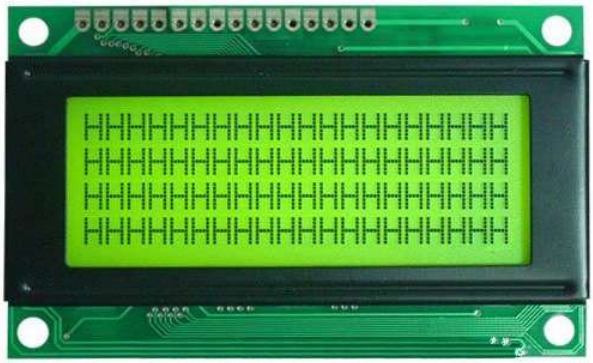
Benodigdheden:

Raspberry PiR

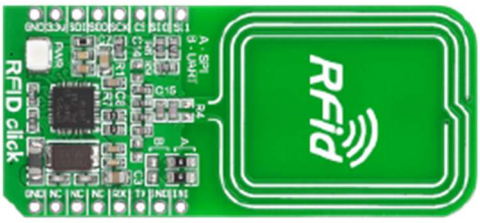
4 digit 7 segment display



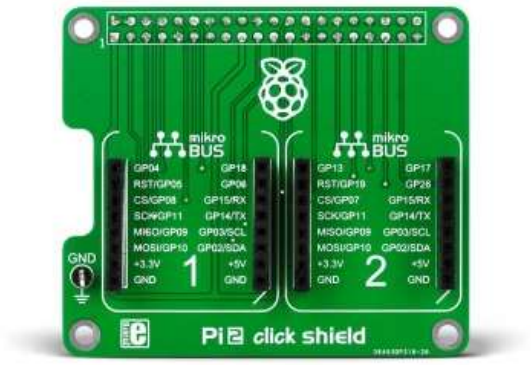
LCD (20 x 4)



Click RFID (RFID reader)

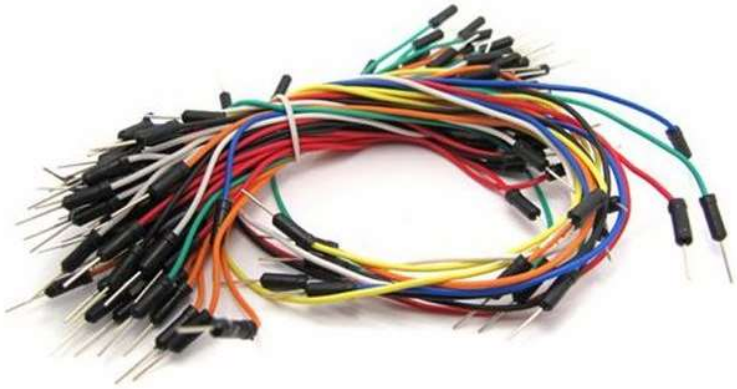


Raspberry Pi Click Shield

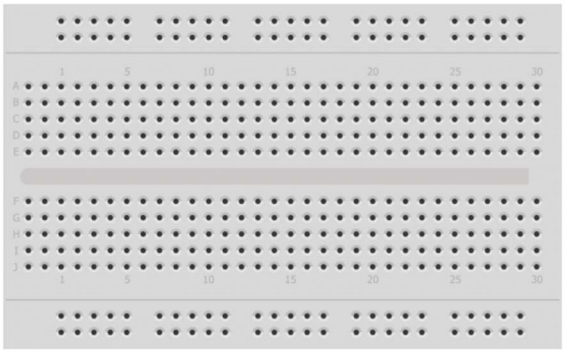


Plexiglas 8mm (50cm op 50cm)

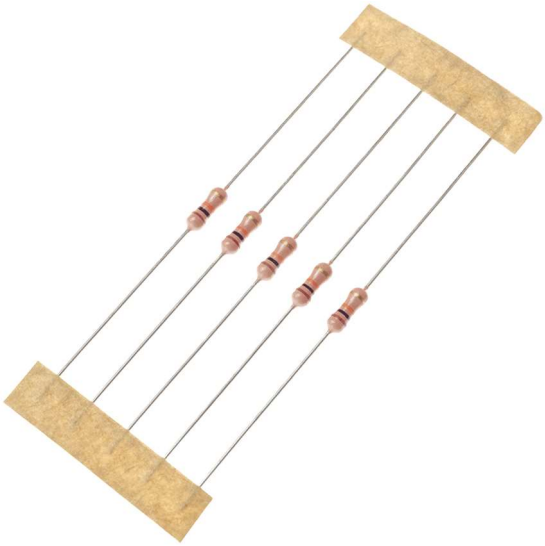
Jumper cables



Breadboard



4 x 10KΩ resistors

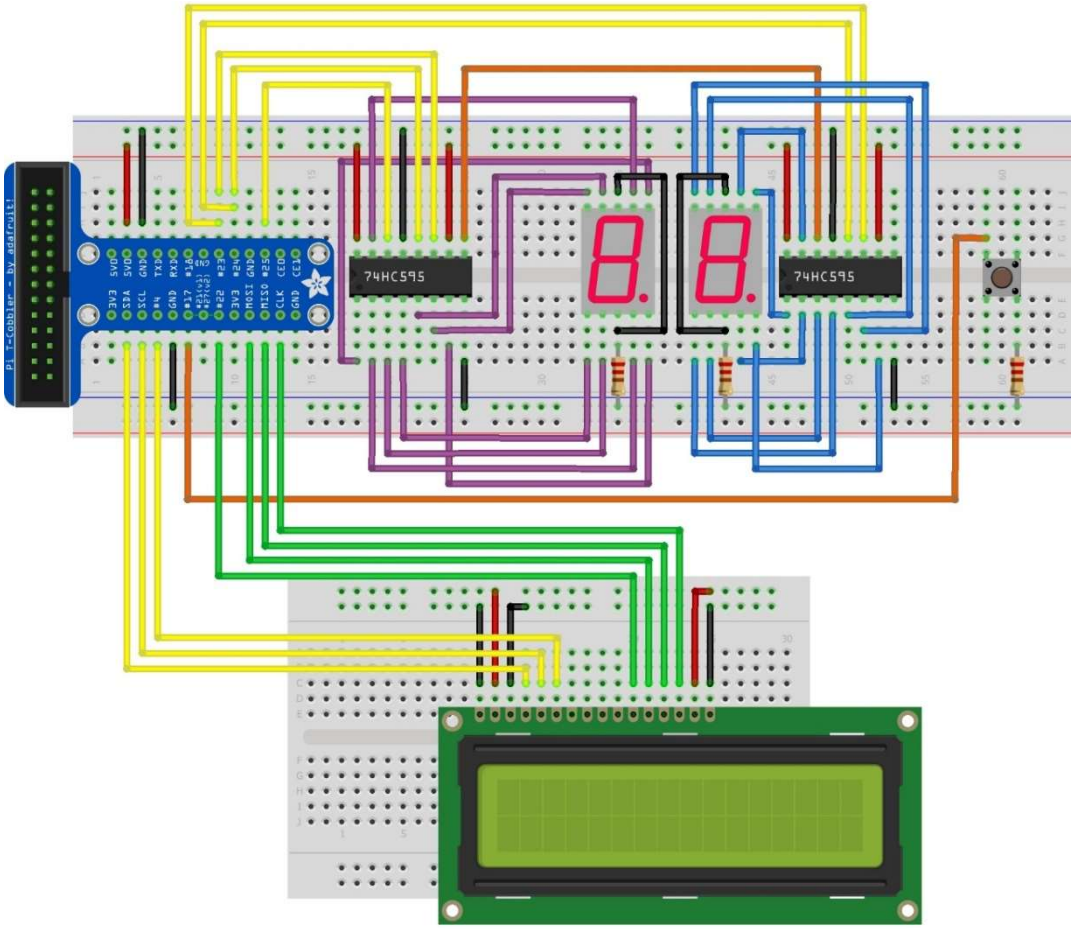


8 x 220Ω resistors

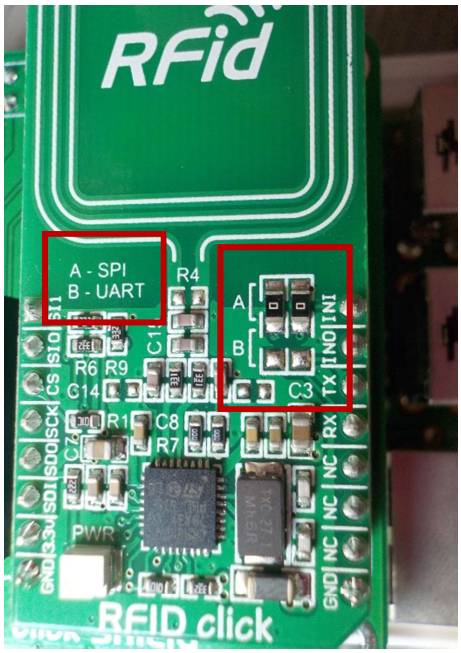
SN74HC595N Shift Register

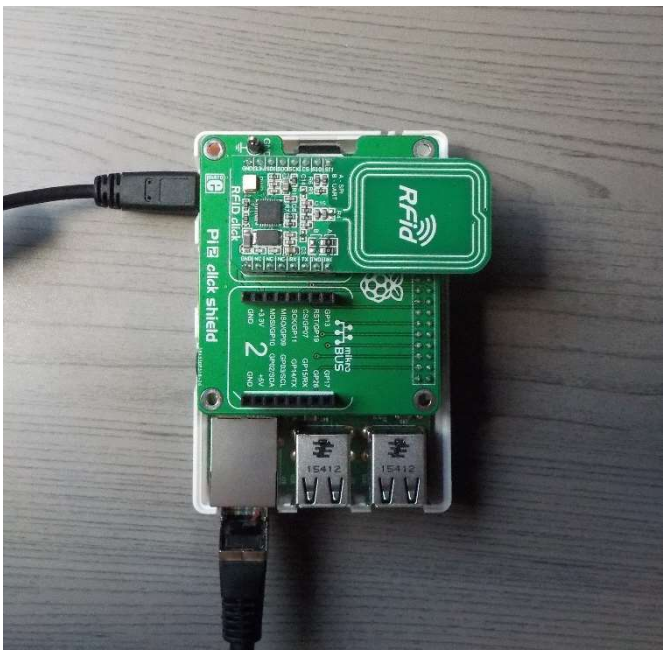
Step 2: Let's Start Wiring

Aan de hand van onderstaande fritzing schema's kunnen de verschillende componenten aan elkaar worden gehangen.



Opmerking: De Click RFID heeft 2 modes, namelijk SPI en UART.

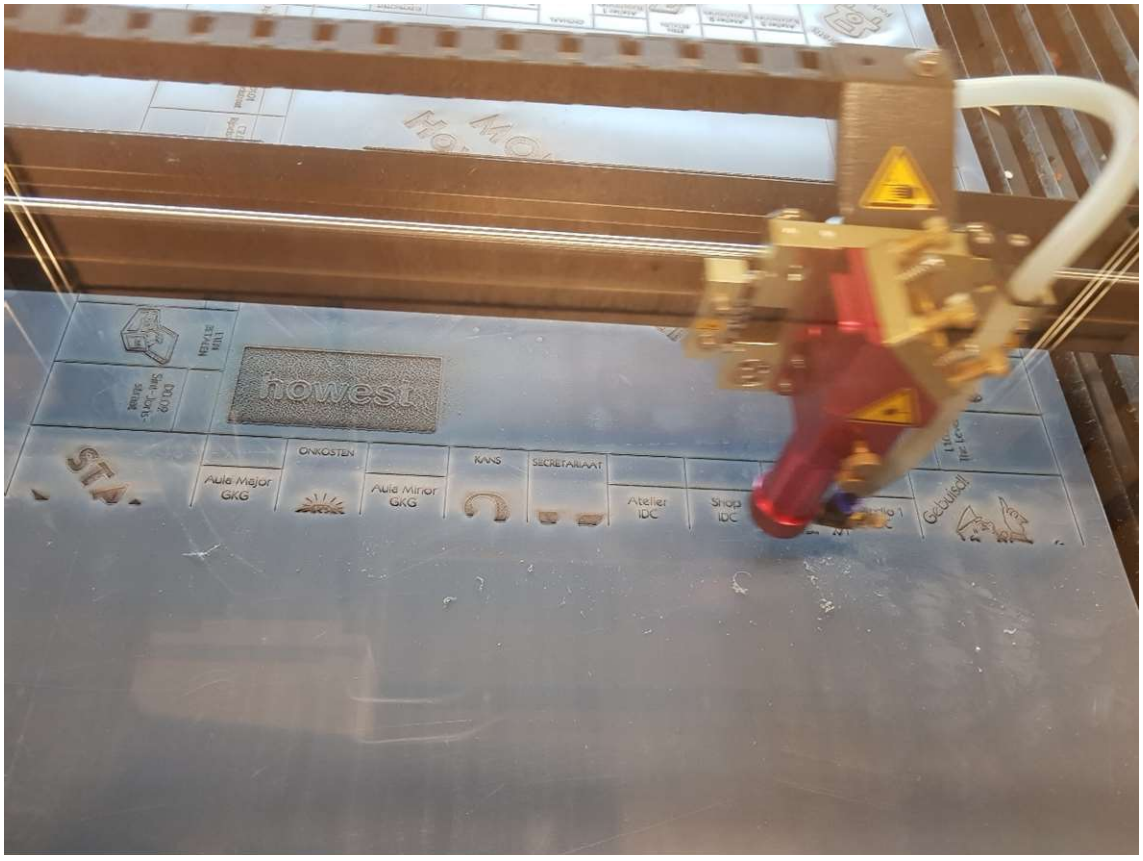


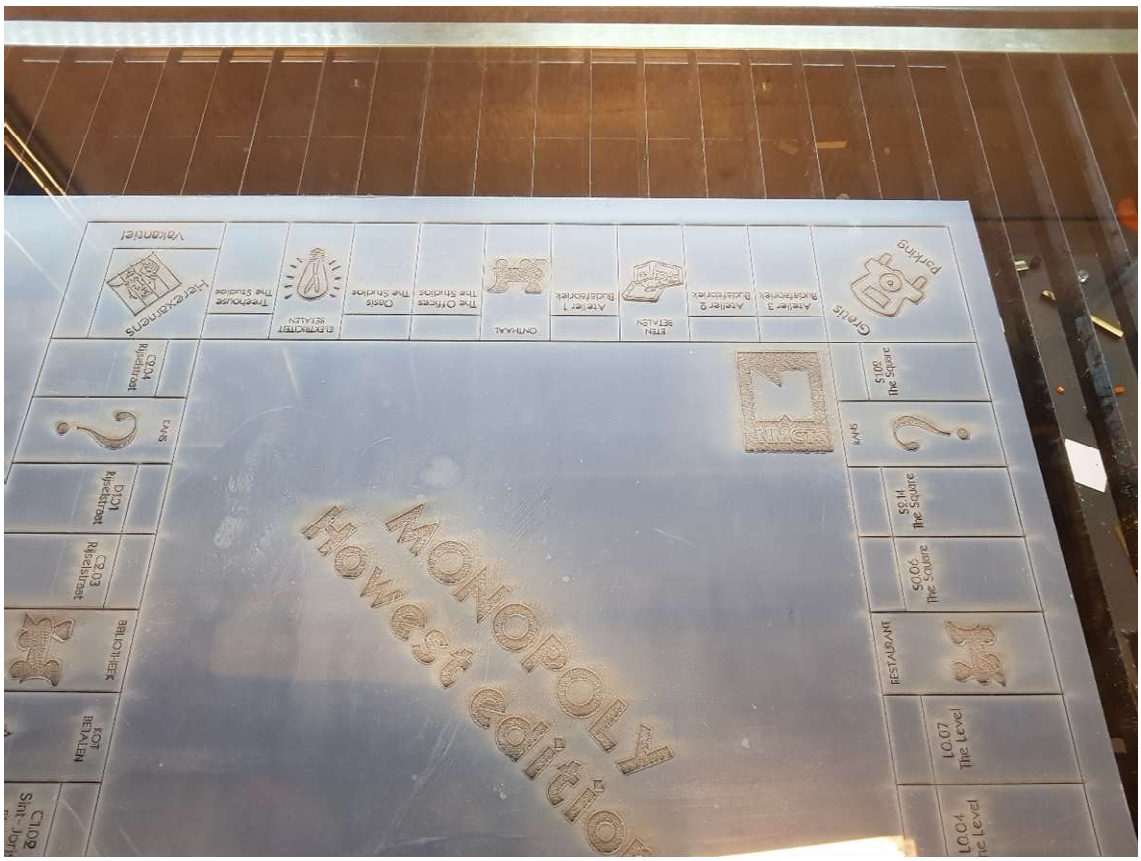


Step 3: Laser cutting!

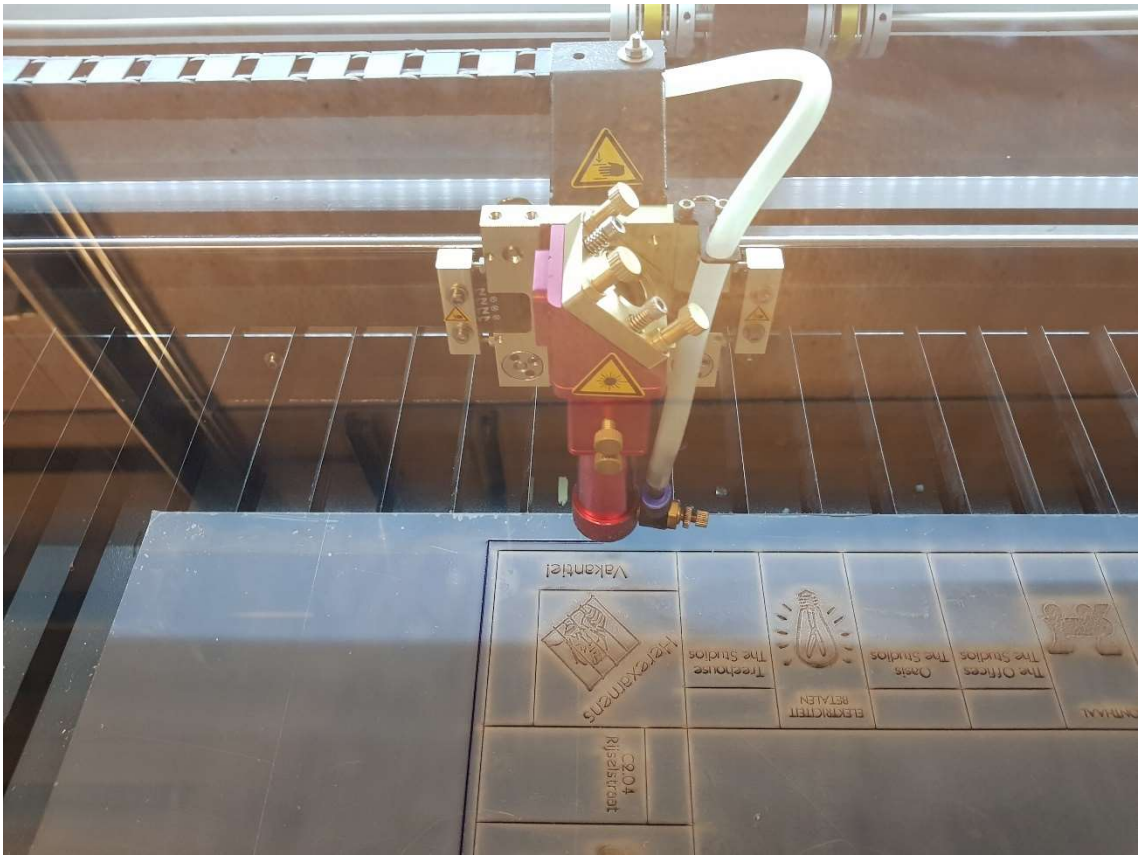
Door middel van de lasercutter, plexiglas en een illustrator file kunnen we het plexiglas graveren en snijden.

Eerst en vooral wordt alles gegraveerd.



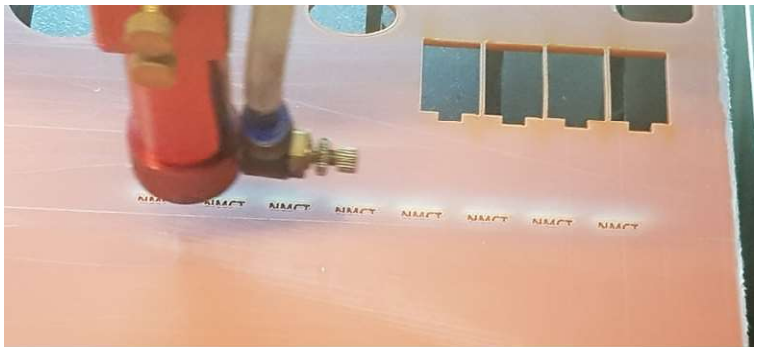


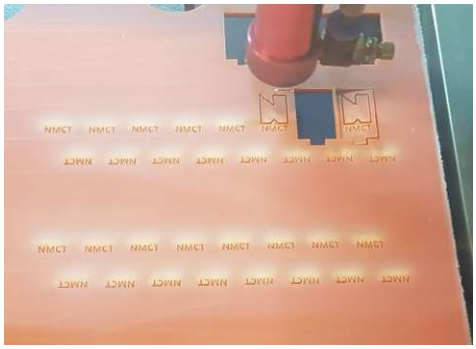
Daarna wordt het bord uitgesneden.



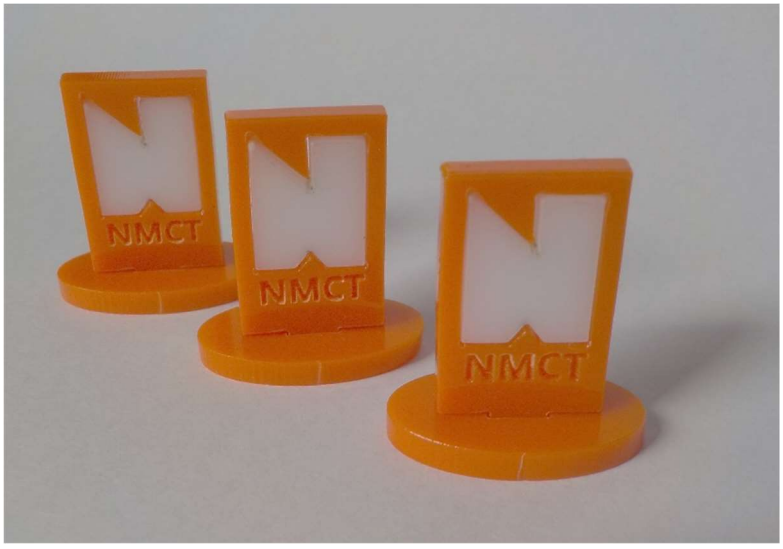
Met als resultaat:

Hetzelfde met de pionnen.





Met als resultaat:



Step 4: Programming

Het programmeren zelf gebeurt via Python (https://www.python.org/)

RFID reader:

```
import serial
import time
try:
    while True:
        ser = serial.Serial('/dev/serial1', 38400,
timeout=2)
        print(ser.name)
        print(ser.isOpen())
        print(ser.readlines())
        time.sleep(1)

except KeyboardInterrupt:
    print('done')
```

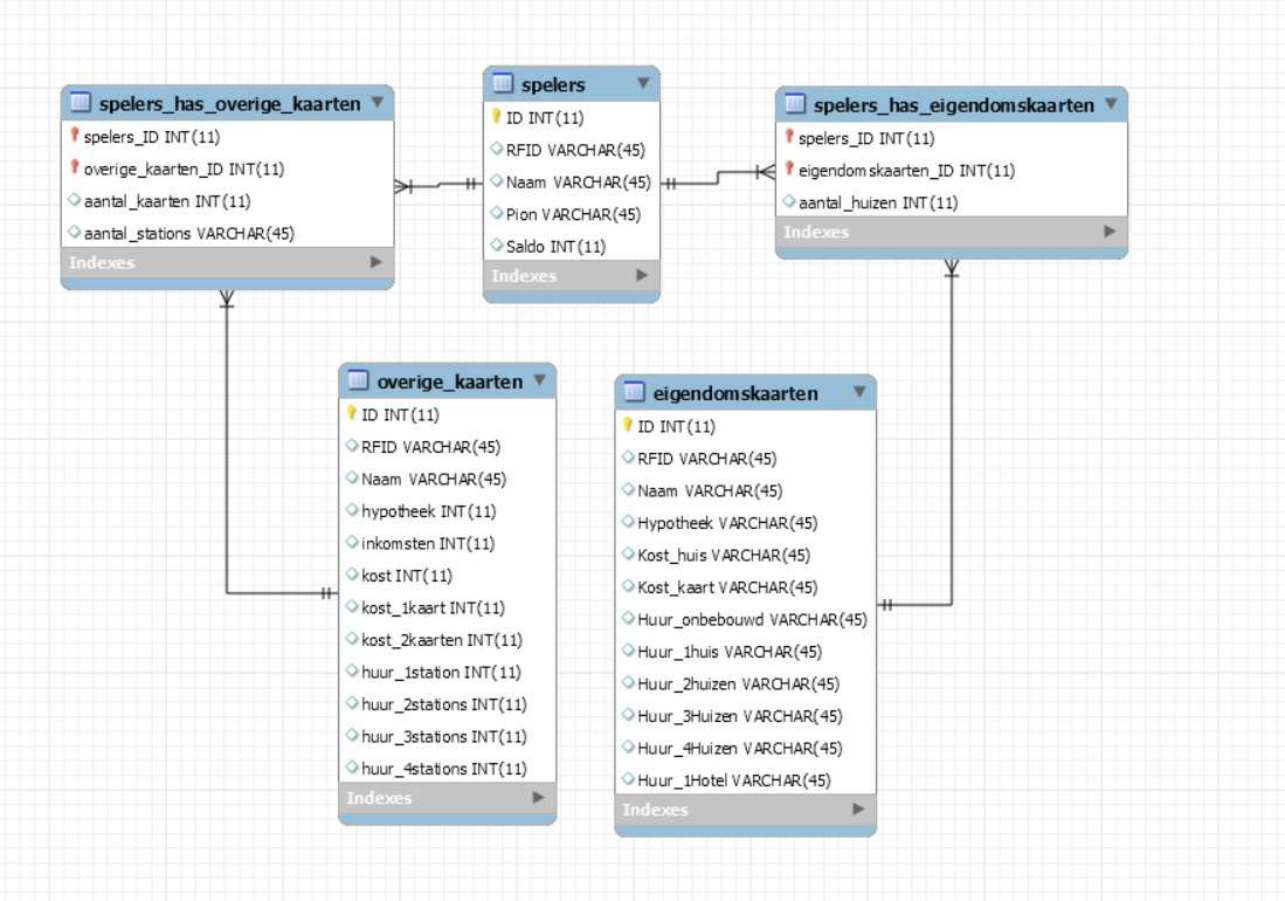
LCD:

```
class LCD:
    wait = 1e-3

    def __init__(self, RS=3, RW=15, E=14, D0=12, D1=16, D2=20, D3=21, D4=23,
D5=24, D6=25, D7=8):
        self.RS = RS
        self.RW = RW
        self.E = E
        self.D0 = D0
        self.D1 = D1
        self.D2 = D2
        self.D3 = D3
        self.D4 = D4
        self.D5 = D5
        self.D6 = D6
        self.D7 = D7

    def setup(self):
        DataBits = [self.D0, self.D1, self.D2, self.D3, self.D4, self.D5,
self.D6, self.D7]
        GPIO.setwarnings(False)
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(self.RS, GPIO.OUT)
        GPIO.setup(self.RW, GPIO.OUT)
        GPIO.setup(self.E, GPIO.OUT)
        for i in range(8):
            GPIO.setup(DataBits[i], GPIO.OUT)

    def Enable(self):
        GPIO.output(self.E, 1)
        time.sleep(LCD.wait)
        GPIO.output(self.E, 0)
        time.sleep(LCD.wait)

    def reset(self):
        LCD.write4bits(self, int("0011", 2))
        LCD.write4bits(self, int("0011", 2))
        LCD.write4bits(self, int("0011", 2))
        LCD.write4bits(self, int("0010", 2))

    def initLCD4bit(self):
        LCD.reset(self)

        LCD.write4bitsTwice(self, 0x2C, 0)
        LCD.write4bitsTwice(self, 0x0F, 0)
        LCD.write4bitsTwice(self, 0x01, 0)
        LCD.write4bitsTwice(self, 0x06, 0)

    def write4bitsTwice(self, value, rs_mode):
        GPIO.output(self.RS, rs_mode)
        DataBits = [self.D0, self.D1, self.D2, self.D3, self.D4, self.D5,
self.D6, self.D7]
        time.sleep(LCD.wait)

        for i in range(4):
            GPIO.output(DataBits[i + 4], 1 if ((value >> i + 4) & 1) > 0
            else 0)
        LCD.Enable(self)
        for i in range(4):
            GPIO.output(DataBits[i + 4], 1 if ((value >> i) & 1) > 0
            else 0)
        LCD.Enable(self)
```

```
    def write4bits(self, value):
        GPIO.output(self.RS, 0)
        DataBits = [self.D0, self.D1, self.D2, self.D3, self.D4, self.D5,
self.D6, self.D7]
        time.sleep(LCD.wait)

        for i in range(4):
            GPIO.output(DataBits[i + 4], 1 if ((value >> i) & 1) > 0
            else 0)
        LCD.Enable(self)

    def printMsg(self, msg):
        for l in msg:
            LCD.write4bitsTwice(self, ord(l), 1)

    def message(self):
        self.printMsg('Speler 1: €5000')

        self.write4bitsTwice(0xC0, 0)
        self.printMsg("Speler 2: €14000")

        self.write4bitsTwice(0x94, 0)
        self.printMsg('Speler 3: €2200')

        self.write4bitsTwice(0xD4, 0)
        self.printMsg('Speler 4: €1900')

    def __str__(self):
        pass


try:
    run = LCD()
    run.setup()
    run.reset()
    run.initLCD4bit()
    run.message()
    while True:
        pass
except (KeyboardInterrupt, SystemExit):
    destroy = LCD()
    destroy.reset()
    GPIO.cleanup()
```

Step 5: Database

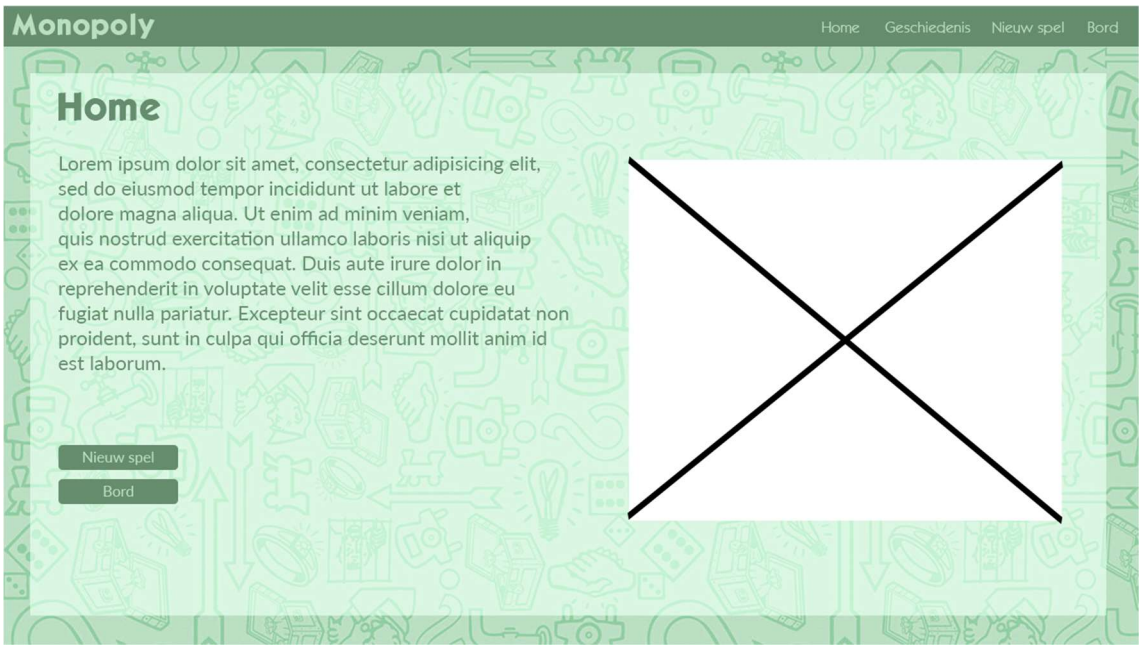
We maken gebruik van MySQL (https://www.mysql.com/) om een database te maken. In dit project wordt gebruik gemaakt van 5 tabellen, namelijk ‘spelers’, ‘eigendomskaarten’, ‘overige_kaarten’, ‘spelers_has_eigendomskaarten’ en ‘spelers_has_overige_kaarten’. Daarin worden alle gegevens opgeslagen van de kaarten (staat vast) en de spelers die het spel spelen (wordt na ieder spel gewist).

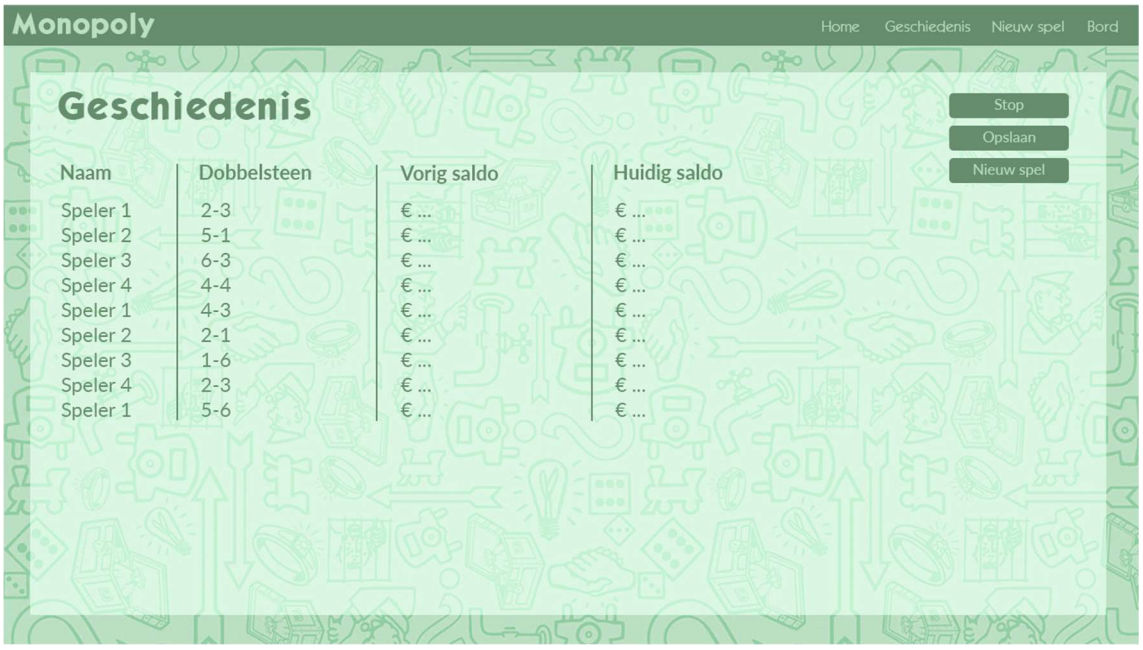


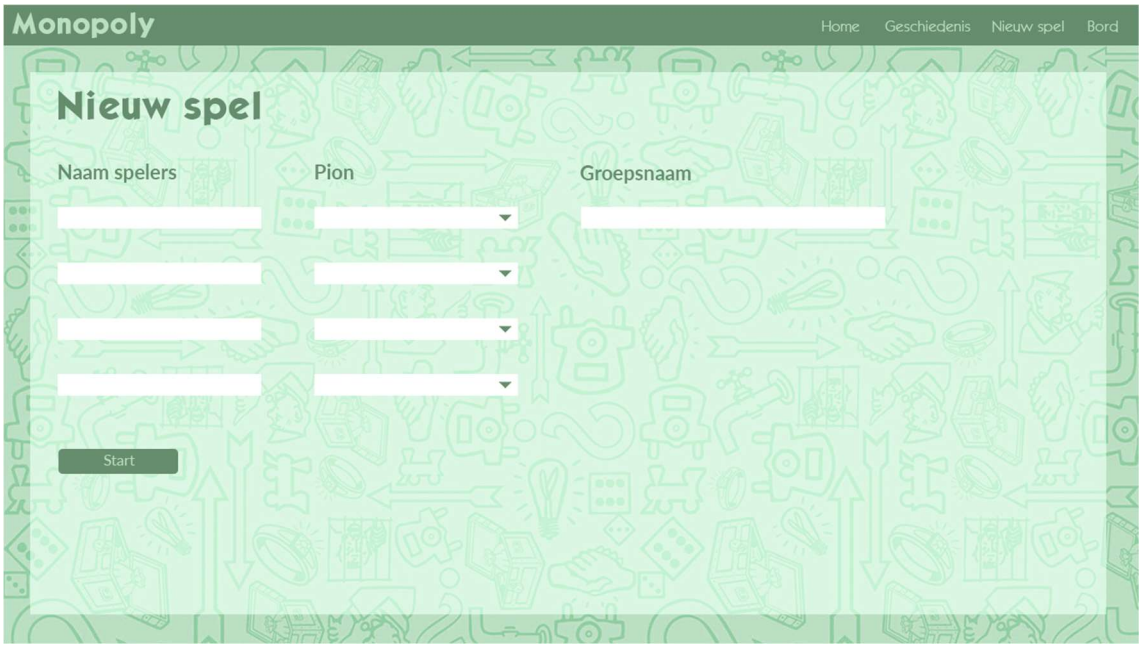
Step 6: Website

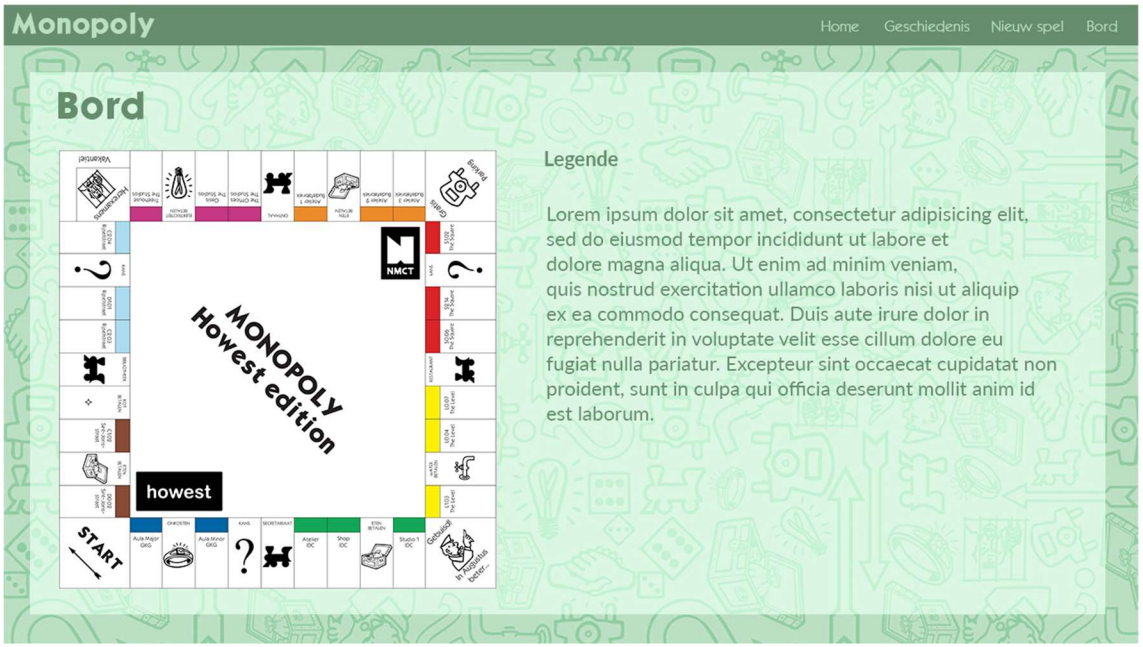
Maak een website waar de gegevens uit de database in een tabel terug kunnen worden gegeven. Dit kan met behulp van Flask in Python. (http://flask.pocoo.org/)

De website die ik gemaakt heb voor dit project ziet er als volgt uit:









Link naar GitHub: https://github.com/elienknockaert/Monopoly