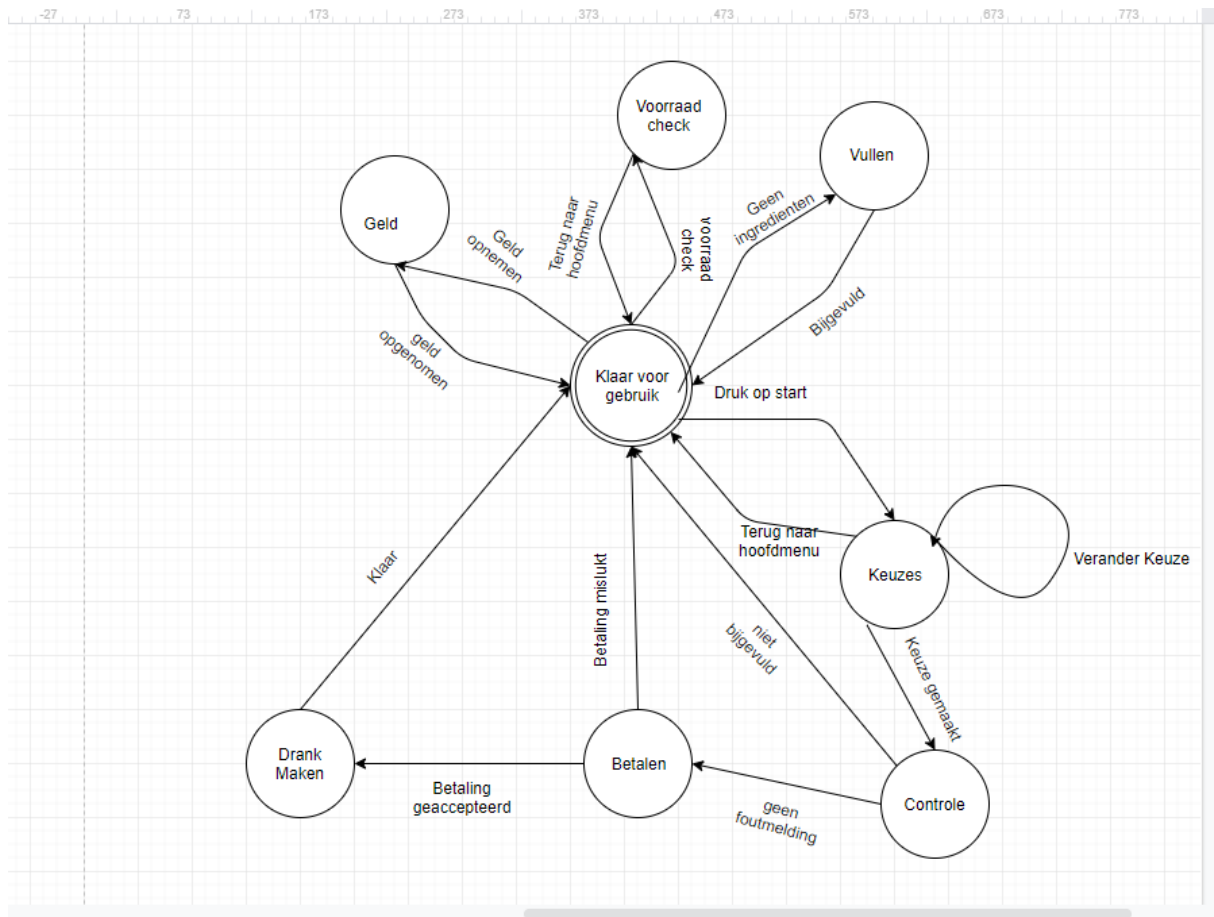


Koffieapparaat

Frank Spakman

1720733

Docent: Nick Roumimper



Het model start bij 'Klaar voor gebruik', dat is het startmenu, vanuit het startmenu kan je: Geld opnemen, voorraad checken, de voorraad bijvullen en natuurlijke producten kopen als klant. Na elke van deze acties word je terug naar het startmenu gebracht. Natuurlijk is de keuze er ook om een drankje te kopen, als de klant een keuze heeft gemaakt en toch een andere wil is dat mogelijk. Ik heb alleen nog last min een verificatie toegevoegd aan de machine zodat klanten niet bij het geld opnemen en overzicht kunnen.

```
# Mogelijkheden die het koffie apparaat kan uitvoeren
if self.actie == "kopen":
    self.kopen()
elif self.actie == "vullen":
    verificatie = input("Voer de verificatiecode in: ") # verificatie code voor de
    if verificatie == "0123":
        self.vullen()
    else:
        print("Verificatie mislukt. Toegang geweigerd.")
elif self.actie == "geld opnemen":
    verificatie = input("Voer de verificatiecode in: ")
    if verificatie == "0123":
        self.opnemen()
    else:
        print("Verificatie mislukt. Toegang geweigerd.")
elif self.actie == "overzicht":
    verificatie = input("Voer de verificatiecode in:")
    if verificatie == "0123":
        self.overzicht()
    else:
        print("Verificatie mislukt. Toegang geweigerd.")
elif self.actie == "exit":
    self.running = False
else:
    print("Ongeldige keuze. Probeer opnieuw.")
```

Als de klant een definitieve keuze heeft gemaakt wordt er gekeken of er genoeg ingrediënten zijn, zo niet dan komt er een foutmelding en word je terug naar het hoofdmenu gebracht.

```
def beschikbaar_check(self): # kijkt of er genoeg materialen zijn om het te maken.
    self.not_available = "" # door checken of de waarde niet onder de 0 gaat.
    if self.water - self.verminder[0] < 0:
        self.not_available = "water"
    elif self.melk - self.verminder[1] < 0:
        self.not_available = "melk"
    elif self.bonen - self.verminder[2] < 0:
        self.not_available = "bonen"
    elif self.cacao - self.verminder[3] < 0:
        self.not_available = "cacao"
    elif self.bekers - self.verminder[4] < 0:
        self.not_available = "bekers"
    elif self.saldo - self.verminder[6] < 0:
        self.not_available = "saldo"

    if self.not_available != "": # als iets onder de 0 is na verminderen
        print(f"Sorry, niet genoeg {self.not_available}!")
        return False
    else: # als er van alles genoeg is.
        print("Ik heb genoeg ingrediënten, het komt eraan!")
        return True
```

Als er geen problemen zijn, wordt er betaald als de transactie niet goed gaat, ga je ook weer terug naar het hoofdmenu. Na het betalen wordt uw drankje voor u gemaakt en ga je terug naar het hoofdmenu.