Bijlage 1.0 Behoefte

Uitwerking van de wensen rond de applicatie KentekenCheck

## Algemene omschrijving

De parkdirectie heeft signalen gekregen dat er auto’s met een vals kenteken op het terrein hebben gereden. Wat die op het terrein deden, is niet bekend, maar dat het ongewenst is, mag duidelijk zijn.

De parkbeheerder heeft aan jouw manager gevraagd of er een systeem gemaakt kan worden dat voertuigen bij de slagboom controleert op kenteken versus type auto en signaleert als die niet overeenkomen.

Aan jou wordt gevraagd een *proof of concept* van dit systeem te bouwen.

## Probleemstelling

Auto’s met een vals kenteken moeten gesignaleerd worden voor zij het terrein opkomen. Op die manier kan de beveiliger eventueel besluiten de toegang te ontzeggen en/of de politie in te schakelen.

## Doel

Een applicatie met de volgende werking: er staat een auto aan de slagboom. Op dat moment moeten het kenteken, het merk, het type en de kleur van de auto geregistreerd worden. Met deze data moet in een bestand van de Rijksdienst voor het Wegverkeer (RDW) worden gecontroleerd of het kenteken hoort bij het merk, het type en de kleur van de auto. Als dit afwijkt, dan moet de beveiliger een signaal krijgen.

## Beknopte beschrijving van de wensen

1. Maak gebruik van een bestaand IoT-device dat op basis van AI, het kenteken, het merk, het type en de kleur de auto kan herkennen. Dit IoT device schrijft data weg in het bestand *GescandeData.csv* in de database *Poortwachter*. Dit bestand is jouw uitgangspunt.
2. In dezelfde database *Poortwachter* is een bestand *RDW.csv*. Dat is het bestand waarin je de kentekens en gegevens controleert.
3. Het programma is een applicatie op het werkstation bij de toegangspoort. Die applicatie heeft toegang tot een map met de genoemde bestanden.
4. Geef een melding dat het programma gestart is.
5. Als een auto voor de slagboom staat met een correct kenteken, dan moet het kenteken met daarachter *OK* worden afgedrukt.
6. Als een auto voor de slagboom staat met een onjuist kenteken, dan moet het kenteken met daarachter *VERSCHIL* worden afgedrukt met daaronder een duidelijk en net lijstje van de geobserveerde eigenschappen versus de geregistreerde eigenschappen.
7. Bij het krijgen van het teken *EINDE* moet een melding gegeven worden dat het programma gaat stoppen. Het aantal gecontroleerde kentekens en de aantallen *OK* en *VERSCHIL* moeten getoond worden. Vervolgens stopt het programma.

## Randvoorwaarden

1. De tool moet in Python gebouwd worden.
2. De interactie met de gebruiker gaat via het CMD-scherm (CUI).
3. De output komt ook op het CMD-scherm.
4. De werkmap voor alle opdrachtbestanden is *$home\KentekenCheck*.
5. De applicatie moet draaien op jouw (virtuele) werkstation (*PC008*).
6. Het hele bestand met kentekens moet correct verwerkt worden.