Documentatie Python automatisatie

**Python packages**

Te installeren packages:

* PyAutoGUI: Controle muis en keyboard.
* Pytesseract: wrapper van Tesseract OCR engine van Google.
* PIL: Image processing.
* PyQt5: GUI van het programma.
* PyInstaller: Converteren python files naar één uitvoerbare “.exe” file.

**Algemeen**

Python files voor de test omgeving zijn: RunTest.py, mainTest.py, navigatieTestomgeving.py, OCRtest.py en checklistGevarenTestomgeving.py

Python files voor de productie omgeving zijn: Run.py, main.py, navigatieProductieomgeving.py, OCRproductie.py en checklistGevarenProductie.py

De verschillen tussen de python files voor de productie- en test omgeving zijn miniem, deze staan expliciet in het document “verschil productie- en test omgeving.docx”

Bij de code staat ook uitgebreid uitleg. In dit document worden de grote lijnen verder toegelicht.

**Werking script**

Run.py

De Python file “Run.py” omvat de lay-out voor de GUI. Deze GUI wordt gebruikt om de gebruiker te informeren over de nodige voorbereiding voor dat het script uitgevoerd mag worden.

Main.py

Deze file bevat de hoofdfunctie die de functies van de andere python files combineert om voor een vlotte werking te zorgen.

Navigatie.py

Deze Python file omvat verschillende functies die verantwoordelijk zijn voor het navigeren binnen H@W.

* Openhw(): deze functie opent H@W van het bureaublad en navigeert binnen H@W tot “overzicht bedrijfsbezoeken” voor een gegeven tijdperiode. Alle bedrijfsbezoeken zullen dan tevoorschijn komen te staan.
* openVerslag(): deze functie wordt nadien opgeroepen om te klikken op de juiste bedrijfsbezoek. Deze wordt gedaan op basis van een if-else statement. De Y waarde voor het (x,y) coördinaat zal aangepast worden op basis van de huidige iteratie. Vanaf de 39ste record is dat niet meer nodig omdat H@W dan vanzelf de volgende record op het scherm weergeeft.

Table

Description automatically generated



Vervolgens zal de functie nagaan als er een checklijst voor de bedrijfsgevaren aanwezig is in “Checklist”, indien dit zo is moet er niks gedaan worden en sluit het venster, anders hervat de functie zich.

Ten slotte zal het programma navigeren naar “bewaarde documenten”, sorteren van Z-A en nagaan als er een verslag van het verkennend bedrijfsbezoek aanwezig is. Indien er geen verslag aanwezig was zal de referentie van het bedrijfsbezoek genoteerd worden in “bedrijfsbezoek.txt”, deze kan dan erna opgevraagd worden om na te gaan welke bedrijfsbezoeken niet volledig in orde zijn.

* openNieuwelijst(): deze functie zal opgevraagd worden wanneer er een verslag aanwezig is. Deze functie navigeert naar “Checklist” en creëert een nieuwe checklijst voor de bedrijfsgevaren.
* bedrijfsbezoekenNoteren(): deze functie navigeert naar “Algemeen” en noteert de referentie van het bedrijfsbezoek in een .txt file.

OCR.py

Binnen deze file staan functies die op bepaalde gebieden van het scherm, een screenshot neemt en de informatie eruit extraheert aan de hand van Pytesseract. De verkregen informatie wordt dan terug meegegeven aan navigatie.py of main.py.

De screenshots die worden genomen wordt gesignaleerd aan de hand van een rode rechthoek.

* retrieveAantalRecords(): Deze functie neemt een foto van het aantal records wanneer de tijdperiode wordt ingegeven om de gewenste bedrijfsbezoeken te verkrijgen. Hieruit wordt een getal verkregen, op basis daarvan wordt het aantal iteraties van de for loop ingesteld.

**Opmerking**: Deze functie werkt in zeldzame gevallen niet optimaal omdat de rechter balk in de weg staat van het getal. Indien het script erdoor niet zou werken probeer dan de tijdperiode groter of kleiner te nemen.



* wordOCR(): Dit is een grote functie die veel functionaliteiten aanbiedt, deze zou eventueel opgesplitst kunnen worden als dat makkelijker zou zijn. Deze functie wordt opgeroepen wanneer een verslag van het verkennend bedrijfsbezoek wordt geopend (Word document). Het zal eerst nagaan of het document NL, FR of ABC+ is en op basis daarvan naar de juiste plaats binnen het document navigeren.

Daarna zullen twee foto’s genomen worden van de gevaren die in het Word document staan en de tekst geëxtraheerd met behulp van Pytesseract.

Vervolgens wordt en aparte GRIM code toegekend voor het gevaar “ Bedelving, verdrinking en pletgevaar” omdat deze hetzelfde is als die van “Nood- en vluchtwegen”.

Ten slotte wordt er een check uitgevoerd om na te gaan of er wel degelijk gevaren staan in het Word document, op basis daarvan zal de functie de gevonden GRIM codes returnen of melden dat het niet in orde is. Indien het niet in orde was zal de main.py file voor deze bedrijfsbezoeken ook de referentie laten noteren in “bedrijfsbezoeken.txt”.

* checkAanwezigheidGevarenlijst(): Neemt een screenshot binnen “Checklist” van de gegeven checklijsten en gaat na als er als een checklijst is voor de bedrijfsgevaren of ABC+.

Graphical user interface, application

Description automatically generated

* checkAanwezigheidVerslag(): gaat nagaan of er een verslag aanwezig is binnen “bewaarde documenten”. Hier wordt vanuit de assumptie gegaan dat het verslag bovenaan zal komen te staan nadat het gesorteerd wordt van Z-A. Uit ervaring met de data manueel in te putten heb ik opgemerkt dat dit zo zal zijn voor ~95%-98% van de gevallen (schatting op basis van ervaring).

Graphical user interface, application, Word

Description automatically generated

* checkTaalverslag(): neemt een foto van de eerste pagina van het verslag van het verkennend bedrijfsbezoek (Word document) en op basis van een keyword gaat het nagaan of het FR, NL of ABC+ is.

Graphical user interface, text, application

Description automatically generated

* retrieveBedrijfsbezoek(): Gaat binnen “Algemeen” van het bedrijfsbezoek de referentie van het bedrijfsbezoek extraheren. Deze string wordt aangepast zodanig dat enkel de cijfers overgehouden worden zodat er geen vreemde tekens overgenomen worden.

Graphical user interface, application

Description automatically generated

checklistGevarenProductie.py

Deze file bevat een hoofdfunctie “gevarenAanvinken(codes)” die verwijst naar elke mogelijk categorie van bedrijfsgevaren. In elk van de categorieën wordt chronologisch over elk mogelijk gevaar nagegaan of deze voorkomt in de codes die we hadden geëxtraheerd van het verslag van het verkennend bedrijfsbezoek. Deze manier is tijdsintensief, maar het garandeert dat er geen gevaren overgeslagen worden. Zo is er geen afhankelijkheid van hoe de gevaren noteert waren in het Word document, want ze zijn vaak niet chronologisch binnen de Word documenten.

Binnen elke categorie zal er eerst één maal geklikt worden op de eerste rij zodat alle rijen vergroot worden en er verder geen moeilijkheden ondervonden worden bij het klikken op andere gevaren.

Muis.py

Deze file werd louter gebruikt om de coördinaten te verkrijgen van de huidige locatie van de muis om dan te gebruiken voor de .click() functies en screenshot functionaliteiten.