Weergeven van notificaties op de sensehat.

Camille Van Damme
Internet of Things
2019 - 2020
New Media Development
Bachelor in de grafische en digitale media
Arteveldehogeschool

Productiedossier:

Discover:

Briefing:

Er werd van ons verwacht dat we een IoT-creatie konden maken met de aangeleerde en nieuwe / trending technologieën. Hiervoor moesten we zelf een project verzinnen om uit te werken.

Define:

Planning:

De opdracht moest af zijn en ingediend voor 23:59 op 05/06/2020.
Ik had geen vaste planning bijgehouden. Maar ik weet wel dat ik uitkeek naar het project.

Analyse:

Voor deze opdracht had ik zelf al iets geprobeerd met PyAutoGUI. Dit vond ik te traag dus wou ik Selenium een kans geven.

Journey Maps:

Voor dit project had ik al iets klein geprobeerd met PyAutoGUI. Het leek mij leuk om een script te kunnen schrijven om iets te automatiseren. Maar na mijn eigen project met PyAutoGUI vond ik dat dit te traag was. Het was alleszins te zwaar voor de Raspberry Pi. Bij het onderzoeken van het automatiseren was ik ook Selenium tegengekomen. Omdat ik hier nog geen ervaring mee had maar toch iets mee wou maken bedacht ik een project waar ik selenium zou kunnen gebruiken. Na het bekijken van YouTube video's over Selenium leek mij dit niet zo moeilijk.
Ik was begonnen met het maken van een script dat keek of je een bericht had op Facebook. In het begin was ik bang dat ik hier op het probleem ging stuiten dat Facebook een authenticatie ging gebruiken tegen bots. Gelukkig heb ik hier geen problemen mee gehad. Het ging vlot tot toen mijn werkende code niet meer werkte. Hier heb ik een paar dagen aan proberen prutsen om dan uiteindelijk op de conclusie te komen dat ik bij Facebook het full XPath moest kopiëren in plaats van het gewone XPath. Eenmaal dit gefixt was had ik het probleem dat ik niet uit de titel van de pagina een gewoon nummer kon halen. Dit heb ik dan moeten oplossen door het nummer uit de messenger badge te halen. Hier heb ik toch ook even op gevloekt.

Na het maken van het script van Facebook was het veel copy-paste bij Discord. Hier had ik dan weer het probleem dat ik niet wist, waar ik kon zien waar de user een bericht had. Uiteindelijk heb ik dit kunnen oplossen door te kijken naar het favicon icon.

Inspiration:

Mijn inspiratie is ontstaan een paar jaar geleden door r/place. Dit was een project van Reddit waar men gebruik maakten van een canvas van 1000/1000 pixels. De gebruikers hadden de keuze uit 16 kleuren en konden kiezen waar ze hun pixel zette op de canvas maar hierna moesten ze telkens 5 minuten wachten. Mensen begonnen samen te werken om een kunstwerk op dit canvas te hebben. Maar er begon al snel rivaliteit en men begon andermans zaken te vernielen. Hierdoor begonnen mensen scripts te schrijven die hun tekeningen beschermden en telkens terug kleurden.
Op dat moment dacht ik, dit ga ik nooit kunnen.
Dus toen we bezig waren met python leek mij de overstap naar automatisatie niet zo moeilijk.

Design:

Doordat mijn project zo gefocust is op automatisatie heb ik niet kunnen designen. Ik heb wel in Aseprite de 2 logo's op het grid gemaakt.

 en 

Deliver:

Handleiding:

Voor je dit kan runnen op je raspberry pi moet je eerst de git repository clonen (https://github.com/camivand4/ioteindwerk2019-2020.git).

Als dit gelukt is voer je deze commands uit:

- pip install selenium
- sudo apt-get install chromium-chromedriver
- py -m pip install pyautogui

Als dit gelukt is verander je het env.text bestand naar een env.py bestand en vul je de juiste info in waar "CHANGEHERE" staat.

Eenmaal klaar voer je het facebook.py of het discord.py script uit.

Deploy:

Dit is hoe het eruit zou zien voor Discord:
https://www.youtube.com/watch?v=tNnmPiRWSZA&feature=youtu.be.

De code kan gevonden worden op:
https://github.com/camivand4/ioteindwerk2019-2020/blob/master/env.txt.