

PRODUCTIEDOSSIER

SECURE PI

INHOUDSOPGAVE

1. Discover	4
2. Define	5
2.1 Analyse	5
2.2 Noodzakelijke soft- en hardware	5
2.3 Inspiratie	6
3. Design	6
4. Develop	7
5. Deliverables	8
6. Timesheet	9
7. Referentielijst	9

DISCOVER

Ons concept

Bij aanvang van ons project hadden we meerdere ideeën voor ogen. De inspiratie van het project kwam van bij Senne thuis. Hij heeft namelijk een security camera maar die had nog geen bewegingsdetectie, met ons project zou dit mogelijk worden. We dachten aan allerlei toepassingen die we met de Raspberry PI konden realiseren. Uiteindelijk besloten we voor de “Security Camera” te kiezen, gezien dit enerzijds een uitdagend project was & anderzijds weinig extra materiaal vergde. Het concept is simpel : de enige benodigdheden zijn een RaspBerry PI, een simpele USB-camera en een werkende internetverbinding. Het doel is om elke detectie van de camera (die zou duiden op een beweging) te registreren en deze data zo te verspreiden naar de eindgebruiker toe.

De verspreiding gebeurt via verscheidene platformen :

1. Mail met foto van detectie als bijlage
2. Online Dashboard met tijdstippen van detecties dat up-to-date is
3. Discord Notificatie
4. Visuele Notificatie a.d.h.v. SenseHat
5. Geluidsnotificatie via aux naar luidspreker (aangesloten op Raspberry PI)

DEFINE

Analyse

De concurrentie bij dergelijk project is vanzelfsprekend groot, gezien een groot deel van de bevolking beschikt over een camera (GSM, Laptop, etc) en een computer. Software-programmeurs spelen hier op in en maken allerlei freeware, die vaak op basis van upgrade-pakketten winstgevend worden. Een simpele Google Search leverde meer dan 100 verschillende programma's op. Deze programma's waren echter vaak uitsluitend voor Windows/Mac of mobiele toestellen (iOS/Android). Voor andere besturingssystemen was dit eerder beperkt.

Competitieve Analyse

Voor onze competitieve analyse kozen we als drie belangrijkste competitors voor "Agent DVR", Blue Iris & Ivideon Server. Het merendeel van deze programma's bestaat uit een gratis versie met uiteraard wat beperkingen. Om gebruik te kunnen maken van de online services (web toegang, SMS, Twitter, email notificaties, toegang vanaf mobiel toestel) is vaak een abonnement vereist.

	SecurePI	Agent DVR	Blue Iris	Ivideon Server
Online dashboard	JA	JA	JA	JA
Gratis	JA	JA	NEEN	NEEN
Multi Camera Support	JA	JA	JA	JA
# Meldings-mogelijkheden	5 (Discord, Dashboard, Sensehat, Email, Geluid, Custom commands)	5 (Email, SMS, Dashboard, Geluid, Custom commands)	4 (Email, Geluid, Mobile App & Dashboard)	2 (Mobile App & Email)

Noodzakelijke soft- en hardware

Hardware

1. Raspberry PI met SenseHat & aangesloten luidspreker
2. USB Camera

Software

1. motionEye software
2. Aanvullende packages voor de installatie van MotionEye & correcte verbinding met de camera

Inspiratie

Voor dit project haalden we onze inspiratie uit enkele Youtube video's die zelf aan security-systeem maakten aan de hand van sensoren. Aangezien we beschikten over een Raspberry PI en een USB-camera besloten we dan maar om onze "eigen" versie te creëren van een security-systeem die ons op verschillende manieren op de hoogte hield van mogelijke gebeurtenissen. De inspiratie hiervoor haalden we ook uit onze directe omgeving. Gezien ieder van ons wel beschikt over een luidspreker / kleine muziekbox en ons communicatiemiddel bij voorkeur Discord betreft, kozen we ervoor om deze twee ook te integreren in ons project.

DESIGN



Latest movement recorded at:
18:08:15 - 20/05/20

Movement in the last 48 hours

Movement spotted at 17:59:10 - 20/05/20

Movement spotted at 18:06:56 - 20/05/20

Movement spotted at 18:07:25 - 20/05/20

[Check on discord !](#)

[Check your e-mail !](#)

DEVELOP

In deze functie (niet volledig weergegeven) worden alle events uit de Firestore opgehaald en verwijderd wanneer deze ouder zijn dan 48h. Op deze manier blijft de Firestore data beperkt en overzichtelijk.

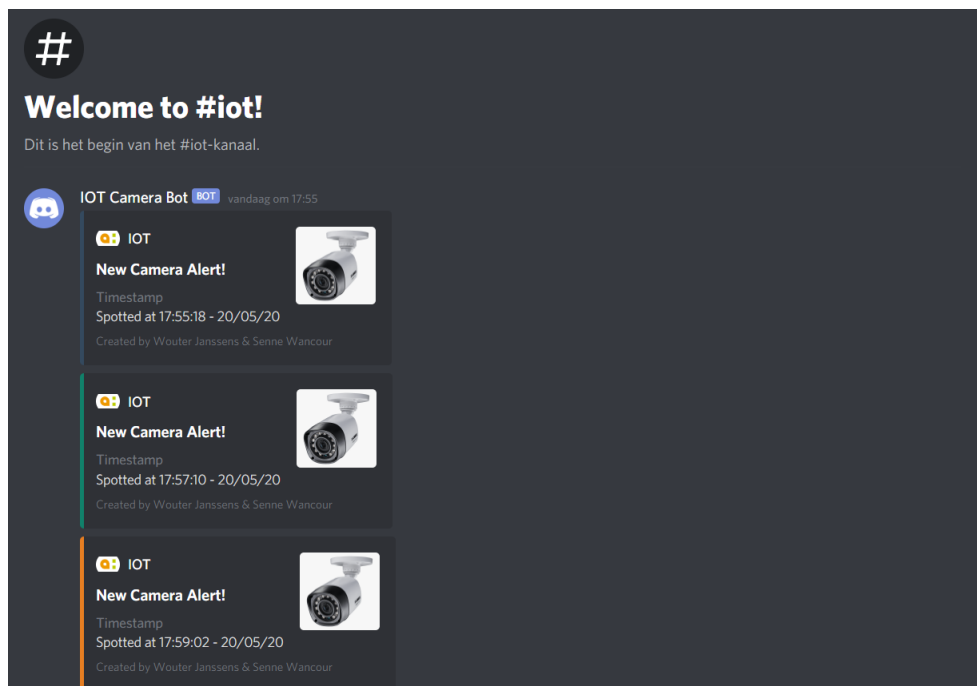
```
async function getDataEvents() {

    const eventlist = document.getElementById('eventlist');

    const data = firebase.firestore().collection('data');
    await data.onSnapshot(function(querySnapshot){
        eventlist.innerHTML = null;
        if(!querySnapshot.empty){
            querySnapshot.docs.forEach(doc => {
                if(doc.exists){
                    const tsToday = Math.round(new Date().getTime()/1000);
                    const tsYesterday = tsToday - (48 * 3600);
                    console.log('verval: ' + tsYesterday);
                    const spottedDate = doc.data().spottedDate;
                    console.log('spotted at: ' + spottedDate);
                    if (spottedDate > tsYesterday){
                        const liTime = document.createElement('li');
                        liTime.className = "li-small";
                        liTime.innerHTML = `Movement spotted at ${doc.data().timestamp}`;
                        eventlist.appendChild(liTime);

                    } else {

                        firebase.firestore().collection("data").doc(doc.id).delete().then(function() {
                            console.log("Document successfully deleted!");
                        }).catch(function(error) {
                            console.error("Error removing document: ", error);
                        });
                    }
                } else {
                    eventlist.innerHTML = null;
                }
            });
        }
    });
};
```



woutjans1@student.arteveldehs.be

aan mij ▾

Motion has been detected by camera "MovementDetector/raspberrypi" at 2020-05-20 18:07:17 (local time).

motionEye.



DELIVERABLES

Stappen om het project na te maken zijn als volgt.

1. Installeren van MotionEye en de bijhorende packages op de Raspberry PI
2. Connecteren van de USB Camera, luidspreker en SenseHat
3. MotionEye starten via de commandline (commando: `sudo systemctl start motionEye`)
4. Surfen naar eigen ip-adres (commando: `hostname -I`) gevolgd door poort '8765'
5. Python script verbinden met motionEye en de instellingen van motionEye configureren naargelang de gewenste acties die ondernomen worden
6. Verbinden van het script met een online dashboard, bij voorkeur door gebruik te maken van Firestore & Github pages.

TIMESHEET

Bepalen onderwerp	30min
Installeren MotionEye & packages	10min
Oplossen errors installatie (verouderde documentatie)	+ - 5 uur
Schrijven code (Firebase - Dashboard, Discord Module, Email Notificaties per SMTP ...)	4 uur
Testen camera & oplossen errors bij uitvoeren van script (verouderde documentatie)	+ - 5 uur
Testing & opstellen dossier - video-opname van de uitwerking	+ - 5 uur

REFERENTIELIJST

<https://github.com/ccrisan/motioneyeos/releases>

<https://github.com/ccrisan/motioneye/wiki/Install-On-Raspbian>