**Introduction :**

Dans le cadre du projet Fas, après plusieurs changements d’idées, nous avons décidé d’avoir comme objectif de créer un système de surveillance géré par un site internet.  
Sur le site un utilisateur peut s'inscrire et se connecter pour accéder aux différents Raspberry enregistrer sur son compte.

Chaque Raspberry est composé d'une caméra et d'un détecteur de distance afin de photographier en fonction de paramètres.

Ainsi l'utilisateur peut gérer les paramètres de ses raspberry sur le site en entrant :

* une indication qui permet de savoir si la caméra doit capturer lorsque la distance mesurée avec le capteur de distance est *inférieur ou supérieur* à la distance entrée.
* une *distance* qui sera comparé à la distance mesurée et qui sera lié à l’attribut inférieur ou supérieur pour enregistrer des photos.
* l'*état du programme* pour lancer ou à l’inverse arrêter le programme de surveillance.
* une *adresse mail* sur laquelle sera envoyé le mail pour prévenir que le système a pris une ou des photos.
* un *temps de latence* minimum entre chaque mail pour éviter les spams.

Les photos prises sont envoyées sur la page web du raspberry avec comme nom la date et l’heure précise de leur enregistrement. L’utilisateur peut gérer les photos prises en les supprimant si besoin.

**Mode d’emploi :**

Étapes préliminaires:

* Connecter la camera au raspberry
* Connecter le shield du raspberry
* Connecter le module hr-sr04 au port D4 du shield

Puis, alimentez le raspberry, et faites ces étapes :

1. Connectez le raspberry à internet (wifi ou ethernet)
2. Allez sur <https://www.onama.fr/Projet_Fas/Connexion_Raspberry.php>
3. Saisissez l’adresse mail suivante : [fabazad@live.fr](mailto:fabazad@live.fr)

mdp: raspberry1

En effet, le raspberry a été enregistré avec ce compte là.

4) Attendre que l’etat du raspberry soit : En ligne. (il s’active automatiquement au démarrage du raspberry)

5) Si au bout de 5min, le raspberry est toujours hors ligne, alors c’est qu’il y a un problème avec la connexion internet. Soyez sur qu’elle fonctionne et qu’il est bien connecté à celle ci en wifi ou ethernet.

b) Retentez de redémarrer le raspberry.

6) Bravo, le système de surveillance est activé.

7) Réglez vos paramètres sur le site afin qu’il corresponde à vos attentes.

Package particulier : sous Raspbian, nous avons installé GrovePi pour pouvoir utiliser le shield.

**Moyens matériels :**

* Raspberry Pi 3 : Le raspberry nous permettait une plus grande plage d’action qu’un arduino en particulier pour communiquer grâce au FTP du site et en utilisant des requêtes (POST ici). De plus nous avions déjà tous les deux manipuler du Arduino et nous voulions découvrir une nouvelle technologie.
* capteur ultrason : Le capteur ultrason nous semblait être la solution la plus adaptée pour répondre à des conditions de capture de photo pour une caméra de surveillance.
* Camera pi NoIR : Nous voulions une caméra capable de prendre des photos dans des endroits obscures étant donné que les caméras de sécuritées sont très souvent utilisées dans des endroits obscurs.
* site web : Le site web permet d’avoir une interface d’utilisation pour l’utilisateur qui n’a qu'à brancher son raspberry pour l'utiliser. De plus il pourra surveiller le surveiller depuis tout endroit et matériel recevant internet.

Pour la programmation des scripts raspberry nous avons utiliser Python.

Pour le site web nous avons utilisé comme technologie : HTML, CSS, PHP, MYSQL, JS, JQUERY

**Moyens humains :**

Nous étions deux pour le projet : Xeryus Mauguy et Fabien Turgut.

Au début du projet nous étions donc tous les deux à chercher l’idée qui nous intéresse, nous sommes passés par plusieurs début de projet mais vite stopper car nous nous rendions compte de la trop grande ambition du projet ou à l’inverse le peu d'intérêt qu’il avait. Nous sommes ensuite concentré sur les documents qui nous aidera à programmer notre raspberry toujours à deux.

Puis c’est au moment de la production du code que nous avons commencé à séparer le travail pour être plus efficace.

Xeryus s’est dirigé vers le raspberry avec de la programmation en python et de la ligne de commande Linux.

Fabien lui s’est dirigé vers la création du site, enfin l'implémentation de cette section de site sur un site déjà créé pour éviter de payer un hébergeur et d’avoir un css déjà avancé.

Nous devions souvent avancer ensemble vu que nos deux interfaces devaient communiquer. Ce qui a entraîné que si l’un de nous était en avance il venait aider l’autre pour être plus efficace. Cela nous a aussi permis de toucher aux deux parties et de comprendre tout le fonctionnement.

Cela jusqu’à la fin du projet.

**Perspectives :**

Nous avons réussis à réaliser les principales fonctionnalitées de notre projet.

Cependant, malheureusement, nous avons dû passer à une caméra pi qui n’est pas NoIR car l’un de nous a égaré la première achetée et nous avons acheté une autre avec le budget que nous avions, l’utilisation des deux caméras étant le même.

On aurait pu rajouter une coque design et pratique pour notre système. Nous aurions ainsi un système complet, solide et fiable.

Enfin, nous aurions pu permettre que ce système soit autonome et rapidement connecté au wifi sans l’utilisation de ligne de commande. Cela signifi qu’il serait opérationnel sur n’importe quelle zone disposant d’une connexion internet.

Pour cela, nous rajouterons en paramètre utilisateur la possibilité d’ajouter le nom de la wifi (ainsi que son code) auquel le raspberry doit se connecter.

A chaque démarrage, le raspberry, se connecterais au wifi connu.

Une batterie externe (5V) pourrait être ajoutée au systeme afin qu’il ne nécessite pas d’une prise de courant à proximité pour etre en etat de fonctionnement. Il serait donc nomade.

Actuellement, il est possible de couper le systeme de video a distance mais le système reste toujours sous tension. Il peut ainsi être activé à distance.

Si l’utilisateur souhaite eteindre son système afin d'économiser de la batterie par exemple, nous pourrions rajouter au système interrupteur permettant de couper l’alimentation (au lieu d’enlever la batterie directement).

Enfin, nous aurions aimé une visionneuse d’image plus complète avec zoom sur image et pouvoir passer à la suivante dans l’overlay.

**Question de marché :**

Le système a été imaginé et conçu afin d'être souple d’utilisation et ainsi toucher le plus large public possible.

Ainsi, tout potentiel client désirant mettre sous surveillance une zone avec des contraintes de distance pourra être satisfait par ce système.

Exemples variés :

Il pourrait y avoir un marché par exemple pour la surveillance des bébés quand ils dorment. Avec la camera Pi noir, on aurait de plus belles images dans le noir et l’utilisateur sera prévenu si l’enfant sort de son lit.

Un autre exemple d’utilisation pourrait être de savoir si une personne est rentrée dans une pièce ou non. Grâce à la photo, on pourrait reconnaître l’identité de la personne.

Enfin un raspberry placé dans un tiroir pourra aussi prendre en photo toute personne mal intentionné qui fouille dans celui-ci

**Cout du projet :**

* 30€ pour le raspberry.
* 30€ la camera Pi Noir.
* 5€ le module ultra-son.
* 5€ le domaine du site.
* 4€/mois hébergement du serveur.
* Sans compter la main d'oeuvre de deux experts IG qui auraient besoin d’argent car ce sont actuellement encore de pauvres étudiants.

Cout total du projet : ~75€

**Code :**

Sites utilisés (sources) :

* <https://docs.python.org/2/library/time.html>
* <http://stackoverflow.com/questions/13890935/does-pythons-time-time-return-the-local-or-utc-timestamp>
* <http://docs.python-requests.org/en/master/>
* <https://openclassrooms.com/courses/utiliser-le-module-ftp-de-python>
* <http://easydomoticz.com/forum/viewtopic.php?t=1061>
* <https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspistill.md>
* <https://docs.python.org/2/library/json.html>
* <https://openclassrooms.com/courses/lister-le-contenu-d-un-dossier-avec-php>
* <http://php.net/manual/fr/ref.filesystem.php>
* <https://api.jquery.com/jquery.get/>
* <http://php.net/manual/fr/function.time.php>

Pour obtenir la fonction qui retourne la distance du capteur ultrason nous avons avons utilisé un bout de programme déjà existant dans le pack grove pi. au lien suivant : /home/fabien/GrovePi/Software/Python/grove\_ultrasonic.py !!!!!!!!

programmation python :

fonction getParametre(adresseMac : String) -> String

#donnée : adresse mac du raspberry

#résultat : String correspondant à un objet json. Cet objet contient les paramètres du programme envoyés par le site après une requête utilisant l’adresse Mac.

procedure envoiePhoto(adresseMac : String) -> void

#donnée : adresse mac du raspberry

#résultat : envoie une photo prise par le camera pi au serveur FTP dans le dossier qui correspond à l’adresse Mac du raspberry

fonction getMAC(interface : String) -> String

#donnée : eth0 du raspberry

#résultat : String contenant l’adresse mac du raspberry

Toutes ces fonctions sont ensuite utilisées dans le programme principal

programmation php, js

Le site du raspberry est une portion de site utilisé sur un site déjà existant. Le principal du css provient du style du site.

La connexion à la base de donnée est aussi contenu dans un fichier appartenant au site complet où l’on peut aussi retrouver les fonctions de connexions au site principal.

Pour le peu de page du site nous avons utilisé un modèle CV : Controleur, Vue.

Un contrôleur de page étant appelé dans une vue.

fonctions js

procedure openOverlay(imgHTML : elemDOM) -> void

#donnée : image cliquable de la visioneuse de la page

#résultat : ouvre dans l’overlay cette image

procedure closeOverlay() -> void

#résultat : Ferme l’overlay

fonction supprimer(elt : elemDOM) ->Bool

#donnée : lien cliquable sous l’image de la visioneuse de la page

#résultat : Supprime du FTP et de la page l’image sélectionné et annule, grâce au renvoie Faux de la fonction, le liende redirection pour une interface plus fluide.