



# Rapport

## Projet IOT – Cage connectée

Haseeb JAVAID  
Mathieu JUGI  
Clément ZHOU

Groupe WITHER

Année 2019/2020

# Sommaire

<b>Introduction</b>	<b>3</b>
<b>Description du projet</b>	<b>4</b>
<b>Etat de l'art</b>	<b>4</b>
<b>Modélisation</b>	<b>8</b>
Diagramme des cas d'utilisations	8
Diagramme d'activité	9
<b>Logiciels utilisés</b>	<b>10</b>
<b>Matériels utilisés</b>	<b>12</b>
<b>L'application web</b>	<b>14</b>
Diagramme de navigation	14
Maquette de l'application WEB	15
Base de données	18
<b>Partie fonctionnelle</b>	<b>19</b>
<b>Problèmes rencontrés</b>	<b>26</b>
<b>Répartition des tâches</b>	<b>27</b>
<b>Conclusion</b>	<b>28</b>

# Introduction

Nous avons commencé à chercher une idée de projet lorsque nous n'étions que deux dans le groupe, mais suite à un changement d'option nous avons accueilli un troisième membre.

La cage connectée ne fut pas notre premier choix, en effet, notre première idée était de réaliser un pot de plante connecté. Il aurait permis la maintenance de la plante peu importe où se trouve son propriétaire, avec un arrosage automatique et une acquisition de la température. Malheureusement, un autre groupe eut la même, où du moins une très similaire et nous avons choisi de changer de projet.

Puisque l'un d'entre nous venait d'adopter un lapin à ce moment-là, la nouvelle idée nous ait venu tout de suite, une cage qui permettrait la survie de l'animal lorsque ses maîtres ne sont pas à la maison. De ce fait, le projet, adapté à une vraie cage, pourrait servir nos propres besoins.

Ce projet a aussi un but de prévention et de faire ouvrir les yeux aux personnes qui abandonnent leurs animaux pour partir en vacances, de leur montrer que ce n'est pas la seule solution, plusieurs choix s'offrent à nous si l'on tient vraiment à son animal de compagnie.

# Description du projet

Selon une enquête de la FACCO, 49.9 % des foyers français possèdent au moins un animal de compagnie, soit presque un sur deux. On sait aujourd'hui, qu'il y a 100 000 abandons d'animaux domestiques en France par an, dont 60 000 durant l'été. C'est dans ce contexte que nous est venu l'idée d'une cage connectée qui sera moins coûteuse qu'un "petsitter" pour les particuliers. D'autre part, cette cage pourra aussi être intéressante pour décharger les plus gros organismes tels que l'ASPA, des tâches quotidiennes redondantes comme le nourrissage des animaux et le contrôle de leur état.

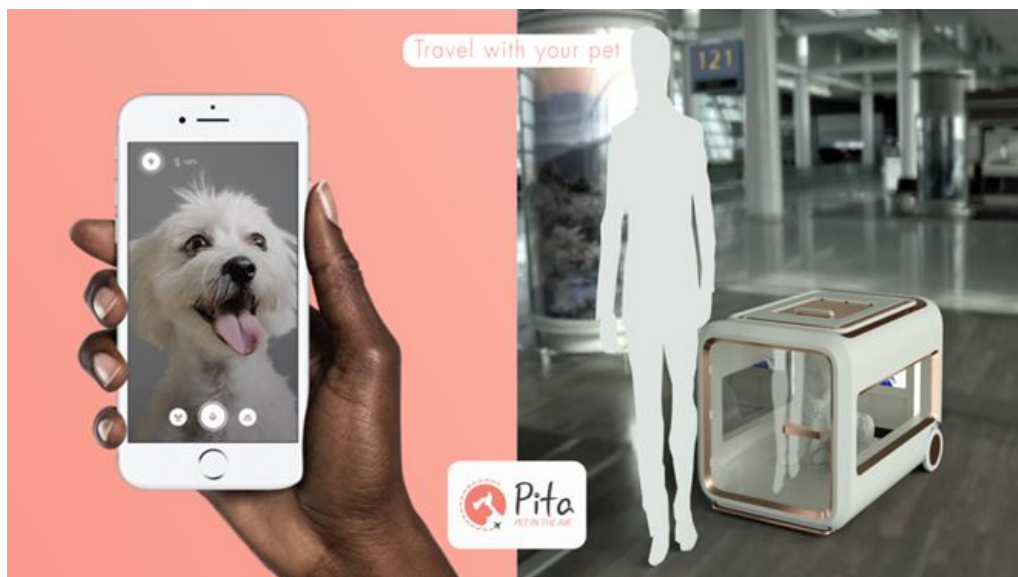
Notre cage connectée est une alternative plus qu'importante à l'abandon d'un animal, elle va permettre sa survie en l'absence de son maître. Ce projet possède trois fonctionnalités principales :

- La distribution automatique de nourriture à un moment et une quantité préalablement sélectionnée
- La détection d'une température inadaptée à l'animal
- La surveillance à tout moment de la cage par l'utilisateur grâce à une caméra.

## Etat de l'art

Depuis plusieurs décennies, les relations entre les humaines et leurs animaux de compagnie ont grandement évolué. Ces maîtres sont prêts à dépenser de plus en plus d'argent pour les nouveaux membres de la famille. Le marché des animaux de compagnie pèse près de 54 milliards d'euros à travers le monde et pourrait atteindre 100 milliards en 2020. En France, l'argent utilisé pour acheter des accessoires représente 560 millions d'euros par an. Quelques entreprises ont vu le jour ces dernières années afin de saisir ce filon d'or, nous allons pour présenter quelques objets qu'elles ont créé :

1- Tout d'abord, le produit de la compagnie PITA (Pet In The Air) qui propose une cage pour le transport des animaux dans les avions. Cette dernière possède certaines fonctionnalités intéressantes comme la possibilité de pouvoir voir et parler à son animal via son smartphone, la cage permet une régulation de la température ainsi qu'un sol intelligent avec une gestion de liquides. Un distributeur d'eau et de nourritures pour les besoins de l'animal est aussi disponible.



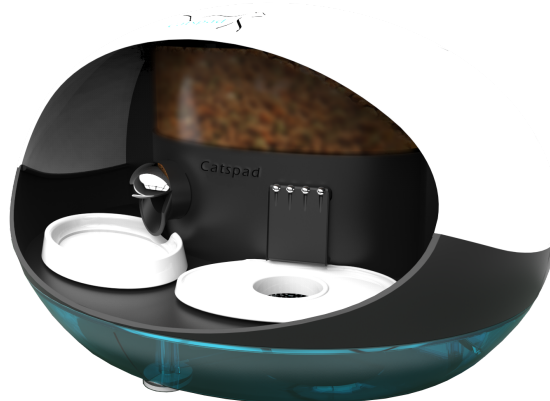
2- Ensuite, il y a des produits qui ne traitent pas le bien-être des animaux domestiques, mais qui utilisent le principe de cage connectée. Ce sont les produits de type pièges qui à l'aide de capteurs vont alerter l'utilisateur via son téléphone que le piège à été enclenché.



3- Un des produits qui se rapproche vraiment de ce que l'on veut faire est le poulailler connecté de chez Eggs iting. C'est un poulailler en bois, qui munie d'une technologie alimenté par un panneau solaire et connecté à Internet via un réseau bas débit, permet d'avertir les utilisateurs que les poules ont pondues. Il permet aussi entre autres d'informer l'utilisateur sur l'état du poulailler tel que la température, le niveau de graine, et l'humidité. Tout cela, sans mettre en danger les poules ou impacter leur confort.



4- Nous avons trouvé un distributeur connecté très intéressant pour les chats. Il distribue les croquettes de manière intelligente, selon la recommandation des vétérinaires, c'est-à-dire plusieurs petits repas tout au long de la journée ou alors vous pouvez le programmer vous-même. La machine possède la capacité de détecter différents chats et de savoir lequel a déjà eu à manger. Au niveau de l'eau, le système la garde fraîche, en mouvement et complètement filtrée. Un suivi de l'alimentation du chat est également proposé afin de détecter un quelconque problème de santé. Le produit possède une réserve équivalente à un mois de nourriture et d'eau. Une batterie de secours est disponible en cas de coupure de courant qui peut aussi tenir un mois. Pour finir, l'entretien est facile et le nettoyage est facile.

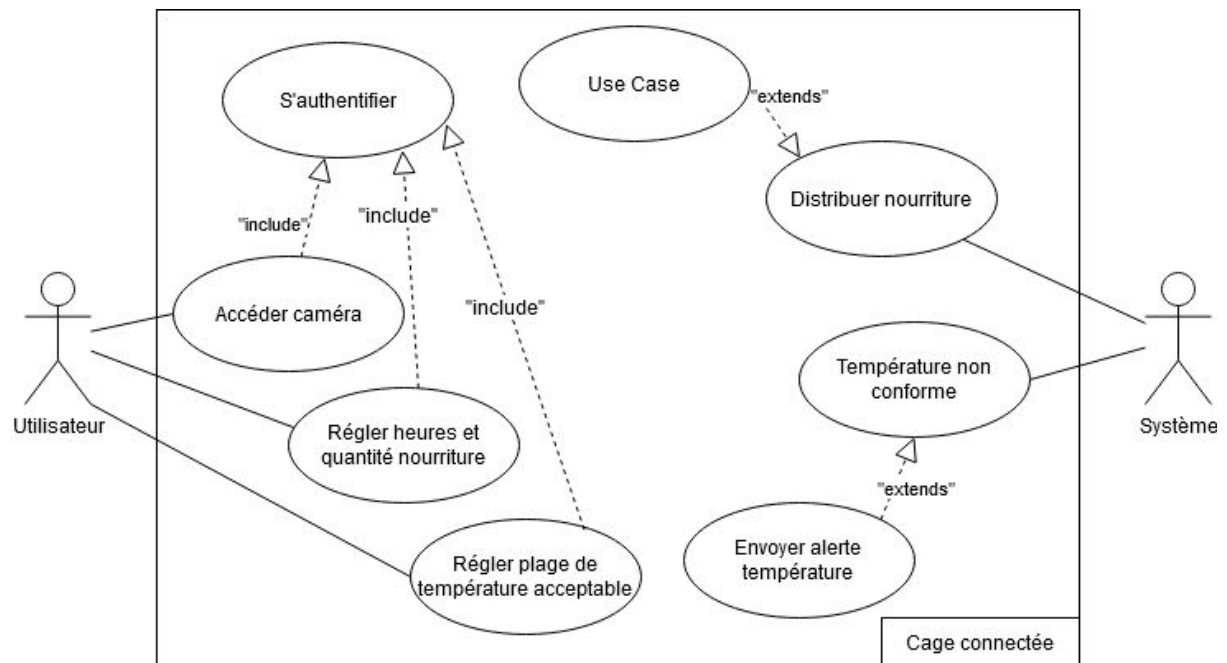


5- Enfin, nous avons aussi vu qu'il y avait de nombreux moniteurs d'aquariums permettant de contrôler l'état de son aquarium tout en ayant la possibilité de contrôler celui-ci en temps réel via son smartphone. Fishbit est un exemple parmi tant d'autres de moniteurs d'aquarium.



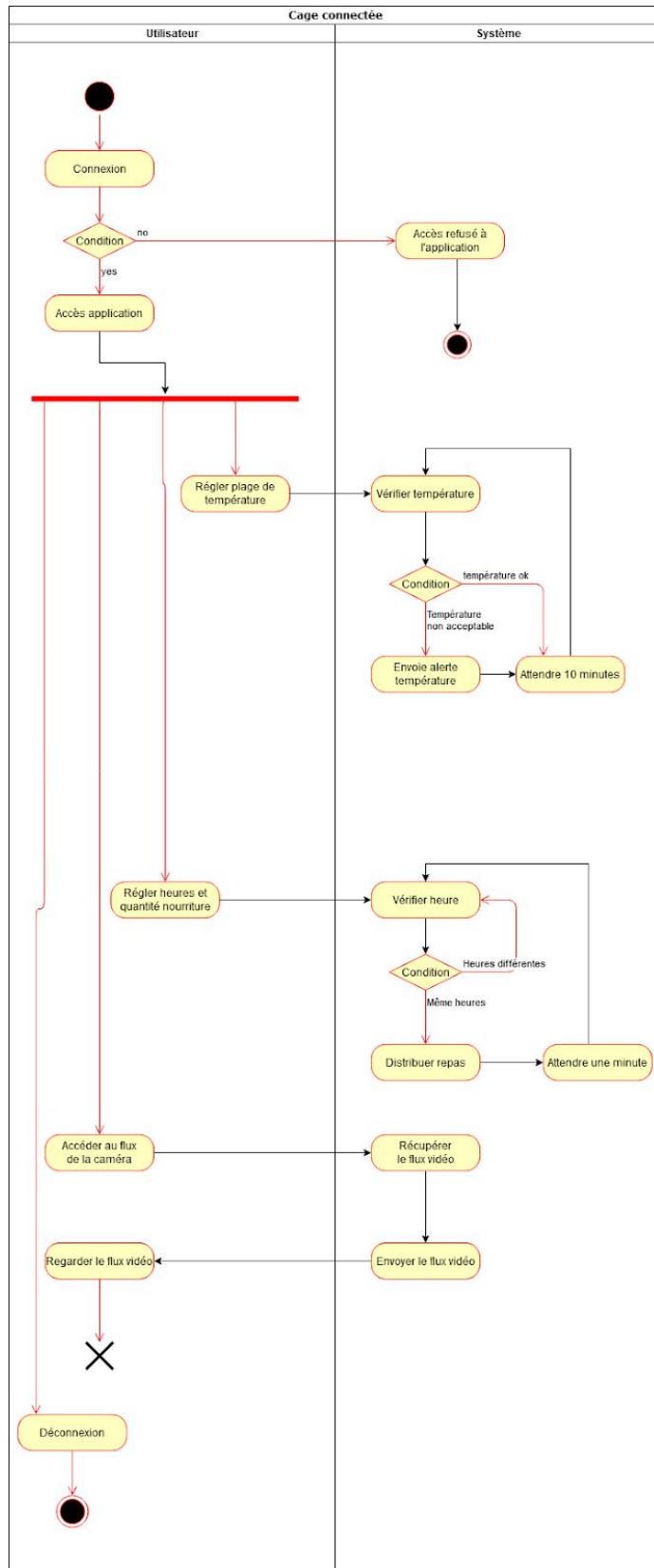
# Modélisation

## Diagramme des cas d'utilisations





# Diagramme d'activité

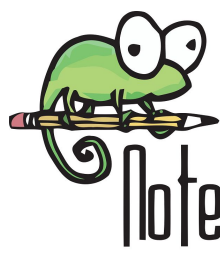


## Logiciels utilisés



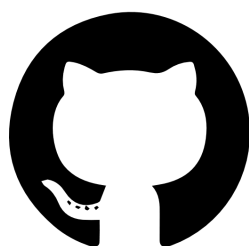
**Wamp Server**, (**W**indows **A**pache **M**ySQL **P**HPMyAdmin) est une plateforme de développement Web permettant de faire fonctionner localement des scripts PHP. C'est un environnement comprenant les trois serveurs Apache, MySQL et MariaDB.

Utilisé dans l'application web afin de créer la base de données et de traiter les scripts PHP.



**Notepad++**, est un éditeur de texte libre, créé par des Français en 2003. Il intègre la coloration syntaxique de nombreux langages.

Utilisé pour développer l'application web.



**Github**, est un service web d'hébergement gratuit et de gestion de développement de logiciels. Lancé en 2008, il permet le partage de données.

Utilisé pour partager le code et les différentes étapes du projet.



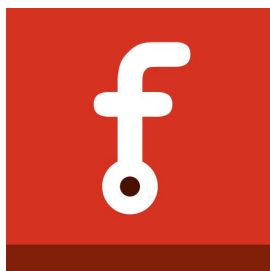
**Discord**, est né en 2015 de Jason Citron avec pour but de rassembler tous les logiciels de VoIP existants (Skype, TeamSpeak, Mumble, etc.) dans un seul logiciel. Le logiciel est d'abord utilisé par les joueurs, mais d'autres communautés se mettent rapidement à l'utiliser, comme les développeurs.

Utilisé pour communiquer entre les différents membres de l'équipe.



**Arduino IDE**, est une application multi-plateforme et est utilisé pour écrire et télécharger des programmes sur des cartes compatibles Arduino.

Utilisé pour écrire les programmes sur la carte ESP 32.



**Fritzing**, est un logiciel libre de conception de circuit imprimé qui permet de concevoir de façon entièrement graphique le circuit.

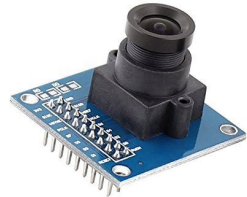



Utilisé pour réaliser les schémas du câblage arduino.

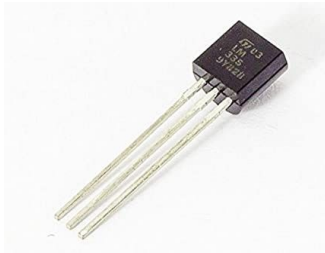
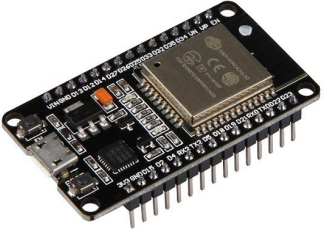



**Shotcut**, est un logiciel libre de montage vidéo multi-plateforme démarré en 2011.

Utilisé pour monter la vidéo de présentation du projet.

## Matériels utilisés

Matériel	Prix	Image
Module de caméra Haljia VGA Ov7670 300 KP	8.99 €	
40 fils jumpers mâles-femelles pour arduino	3.99 €	
Lot de jumper wires longs	Déjà fournis	
Moteur pas à pas	Déjà fourni	

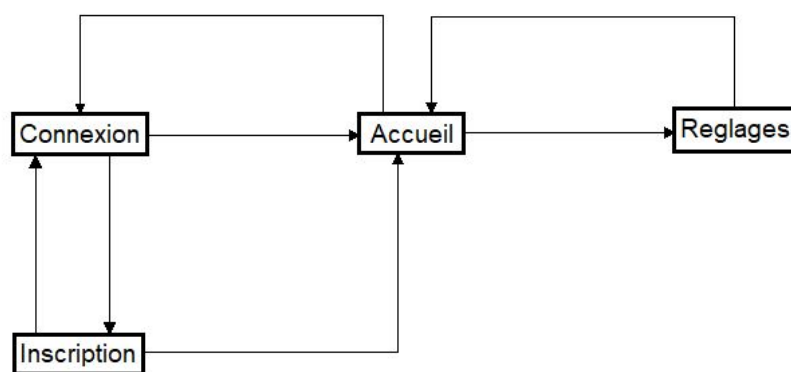
Thermomètre	Déjà fourni	
ESP 32	Déjà fourni	
Carton	Stock personnel	

# L'application web

Nous avons réalisé une application web sera principalement utilisé par les clients. Ils ont leur propre compte et peuvent accéder aux données récoltées par les différents récepteurs placées sur la cage.

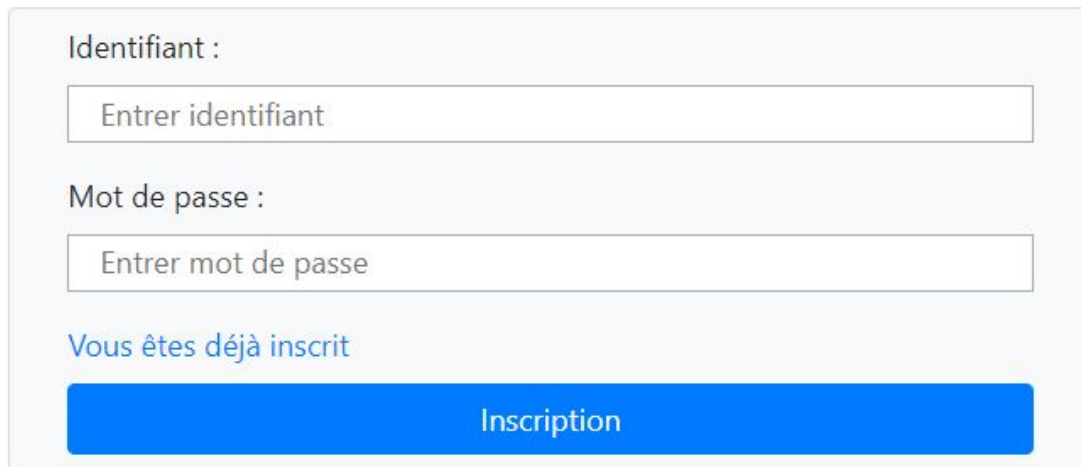
## Diagramme de navigation

Voici le diagramme de navigation de l'application web, l'utilisateur va d'abord se connecter à son compte sur la page connexion. S'il n'a pas de compte, il doit en créer un sur la page inscription. À la suite des deux pages précédentes, il sera redirigé vers la page accueil où il pourra accéder aux données, puis ensuite sur la page réglages où comme son nom l'indique réglera les différentes données pour son animal.



## Maquette de l'application WEB

Nous avons tout d'abord la page de connexion, c'est la page par défaut si l'utilisateur n'est pas connecté. Elle demande seulement un identifiant et un mot de passe. Si vous n'êtes pas enregistré dans la base de donnée, vous ne pourrez pas vous connecter. Il y a un lien pour s'inscrire.

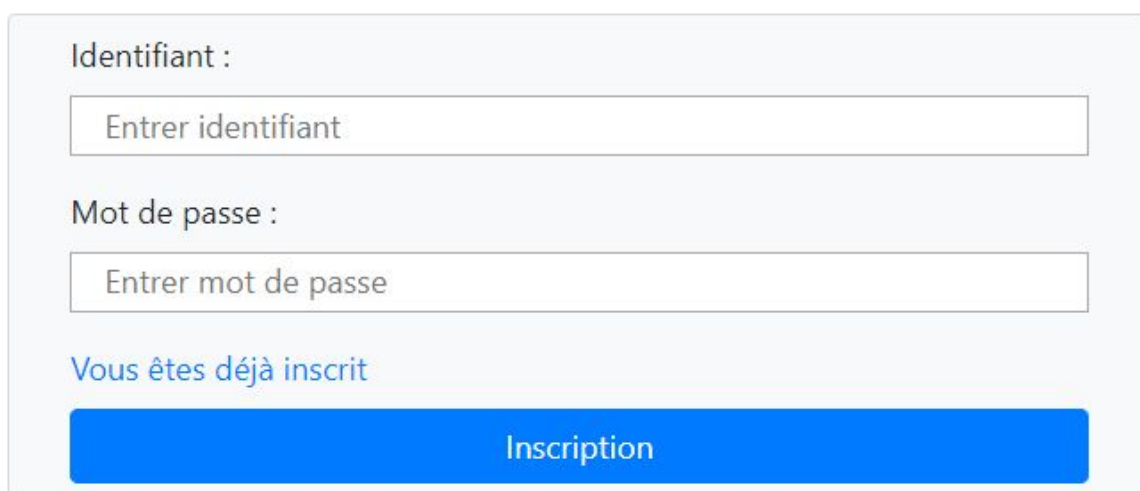


Identifiant :

Mot de passe :

[Vous êtes déjà inscrit](#)

Ce lien nous renvoie vers la page inscription, qui ne demande elle aussi qu'un identifiant et un mot de passe. Elle permet l'enregistrement de l'utilisateur dans la base de données et renvoie vers l'accueil. Si l'utilisateur se rend compte qu'il possède déjà un compte, il peut retourner sur la page précédente via le lien sous le mot de passe.



Identifiant :

Mot de passe :

[Vous êtes déjà inscrit](#)



Ensuite, voici la page principale de l'application, il faut être connecté pour y accéder. Cette page montre qui est connecté, elle propose un lien direct avec la caméra installer dans la cage de l'animal. Pour les informations de température et de nourriture, si l'utilisateur vient de s'inscrire, il n'y en aura pas. Sinon, il pourra voir la température minimale et maximale autorisée dans la cage, ainsi que la quantité de nourriture distribuée et à quel(s) horaire(s).

Il y a un bouton déconnexion qui renvoie vers la page de connexion et un bouton réglage qui renvoie vers la page des réglages.

On peut voir un screen sans les réglages initialisés et un autre avec.

Connecté en tant que mathieu

Déconnexion

CAMERA

Température

Minimum acceptable :

--°C

Maximum acceptable :

--°C

Nourriture

Quantité :

-- grammes

Horaires(s) :

--:--:-- H

--:--:-- H

--:--:-- H

--:--:-- H

Réglages

Connecté en tant que a

Déconnexion

CAMERA

Température

Minimum acceptable :

1°C

Maximum acceptable :

1°C

Nourriture

Quantité :

25 grammes

Horaires(s) :

02:01:00 H

01:01:00 H

01:01:00 H

01:01:00 H

Réglages

Enfin, c'est la page des réglages, ici l'utilisateur va pouvoir initialiser ou changer les valeurs pour la cage de son animal. Il devra rentrer les températures minimales et maximales ainsi que choisir la quantité de nourriture à fournir. Quatre horaires de distribution sont disponibles, mais seulement un est obligatoire. Les trois autres sont au bon vouloir de l'utilisateur.

Le bouton "appliquer" mettra à jour la base de données et renverra l'utilisateur vers la page principale. Le bouton "retour" quant à lui renvoie aussi vers la page principale, mais sans toucher à la base de données.

Connecté en tant que mathieu

Retour

### Température

Température minimale

Température maximale

### Nourriture

**Quantité**

☐ 5 grammes ☐ 10 grammes

☒ 15 grammes ☐ 20 grammes

☐ 25 grammes ☐ 30 grammes

**Horaires**

Horaire 1 :

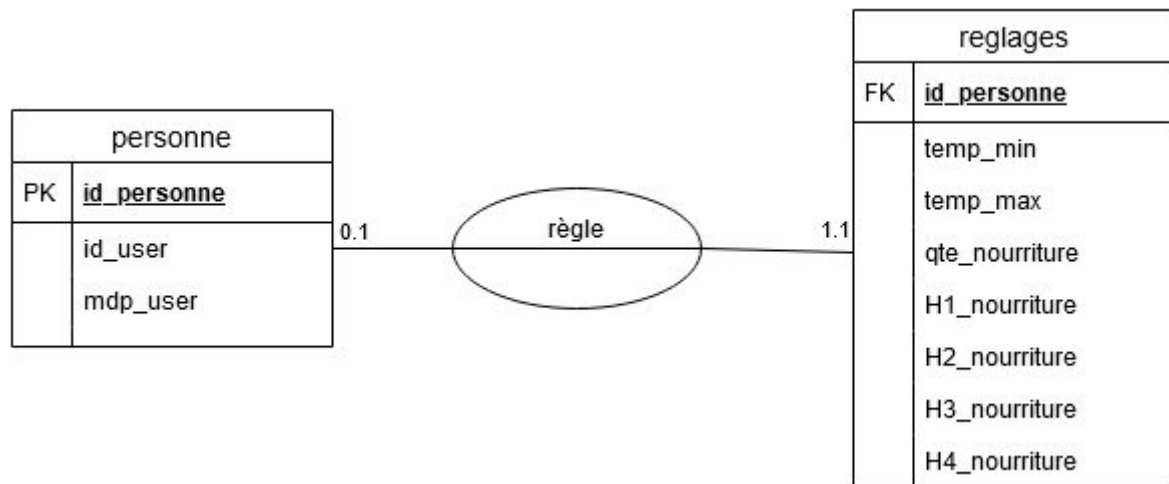
Horaire 2 :

Horaire 3 :

Horaire 4 :

Appliquer

## Base de données



La base de données est composé de deux tables.

La table personne est composé de l'identifiant et du mot de passe, la clé primaire est un entier auto incrémenté.

La table réglages comporte tous les paramètres que l'utilisateur peut modifier; la température minimum et maximum acceptable, la quantité de nourriture distribuée ainsi que les horaires de distribution.

## Partie fonctionnelle

Voici l'interface de l'application web provisoire, on peut y voir la température actuelle captée par le thermomètre. Il y a un bouton feed qui distribue la nourriture grâce au moteur pas à pas. Enfin, nous avons les images retranscrites par la caméra (ne tenez pas compte de la qualité, elle dépend de la connexion wifi, la personne chez qui se trouve le matériel en a une mauvaise).

### ESP32 Food App



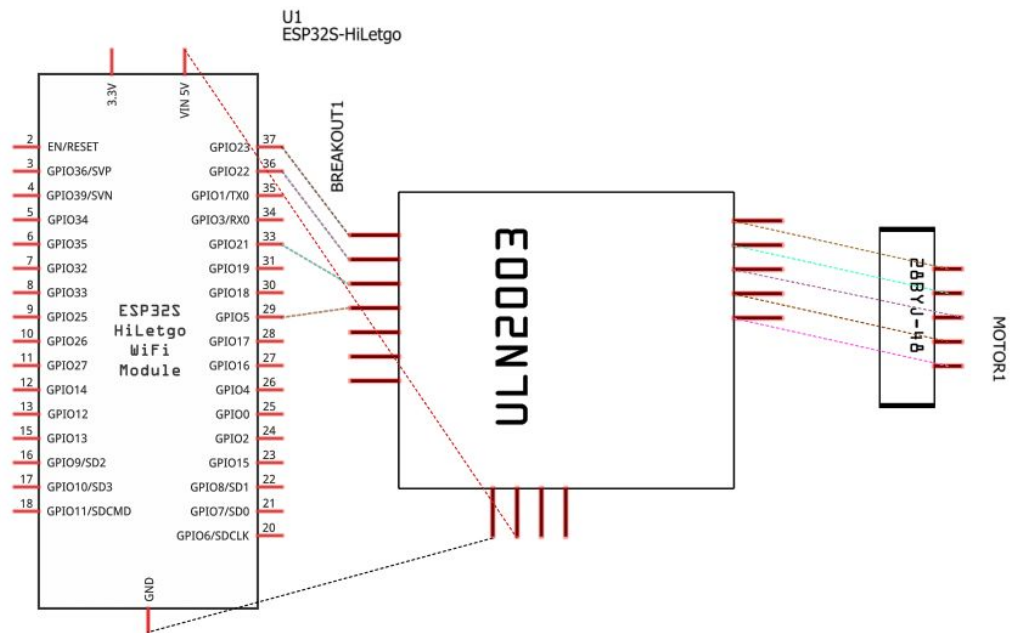
**Current Temperature :**

7.13 ° C

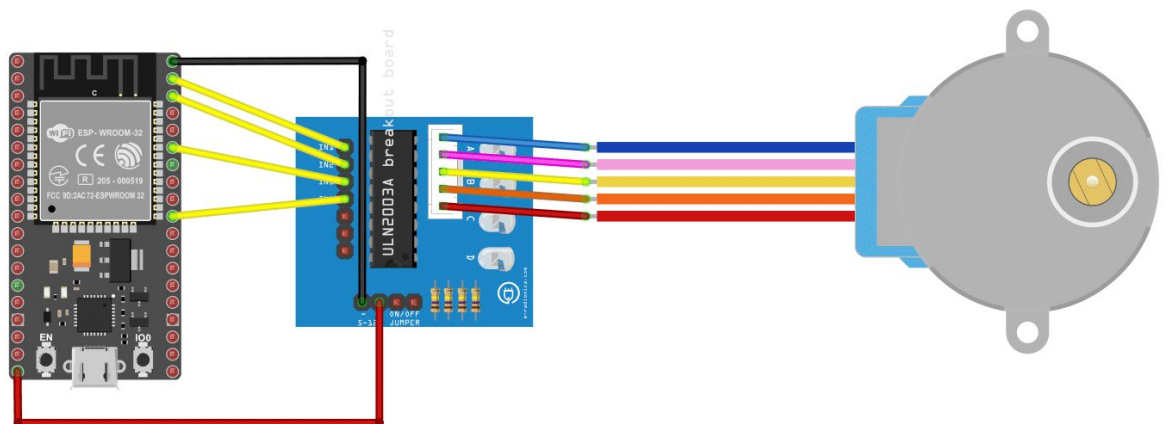
**Camera :**



Commençons par le distributeur automatique de nourriture. C'est donc un moteur pas à pas qui en est le principal acteur. Il se déclenche ici lorsque l'utilisateur appuie sur le bouton feed, mais à l'avenir, il se déclenchera lorsque ce sera le bon horaire.



fritzing



fritzing

Pour contrôler le moteur pas à pas qui enclenche la distribution de la nourriture, on utilise deux fonctions :

D'abord la fonction OneStep permet d'effectuer un pas dans le sens du moteur ou dans le sens inverse du moteur en fonction du booléen qu'elle reçoit en paramètre :

```

void OneStep(bool dir){
    if(dir){
        //one step in a direction
    switch(step_number){
        case 0:
            digitalWrite(STEPPER_PIN_1, HIGH);
            digitalWrite(STEPPER_PIN_2, LOW);
            digitalWrite(STEPPER_PIN_3, LOW);
            digitalWrite(STEPPER_PIN_4, LOW);
            break;
        case 1:
            digitalWrite(STEPPER_PIN_1, LOW);
            digitalWrite(STEPPER_PIN_2, HIGH);
            digitalWrite(STEPPER_PIN_3, LOW);
            digitalWrite(STEPPER_PIN_4, LOW);
            break;
        case 2:
            digitalWrite(STEPPER_PIN_1, LOW);
            digitalWrite(STEPPER_PIN_2, LOW);
            digitalWrite(STEPPER_PIN_3, HIGH);
            digitalWrite(STEPPER_PIN_4, LOW);
            break;
        case 3:
            digitalWrite(STEPPER_PIN_1, LOW);
            digitalWrite(STEPPER_PIN_2, LOW);
            digitalWrite(STEPPER_PIN_3, LOW);
            digitalWrite(STEPPER_PIN_4, HIGH);
            break;
    }
    }else{
        //step in opposite direction
        switch(step_number){
            case 0:
                digitalWrite(STEPPER_PIN_1, LOW);
                digitalWrite(STEPPER_PIN_2, LOW);
                digitalWrite(STEPPER_PIN_3, LOW);
                digitalWrite(STEPPER_PIN_4, HIGH);
                break;
            case 1:
                digitalWrite(STEPPER_PIN_1, LOW);
                digitalWrite(STEPPER_PIN_2, LOW);
                digitalWrite(STEPPER_PIN_3, HIGH);
                digitalWrite(STEPPER_PIN_4, LOW);
                break;
            case 2:
                digitalWrite(STEPPER_PIN_1, LOW);
                digitalWrite(STEPPER_PIN_2, HIGH);
                digitalWrite(STEPPER_PIN_3, LOW);
                digitalWrite(STEPPER_PIN_4, LOW);
                break;
            case 3:
                digitalWrite(STEPPER_PIN_1, HIGH);
                digitalWrite(STEPPER_PIN_2, LOW);
                digitalWrite(STEPPER_PIN_3, LOW);
                digitalWrite(STEPPER_PIN_4, LOW);
                break;
        }
    }
    step_number++;
    if(step_number > 3){
        step_number = 0;
    }
}

```

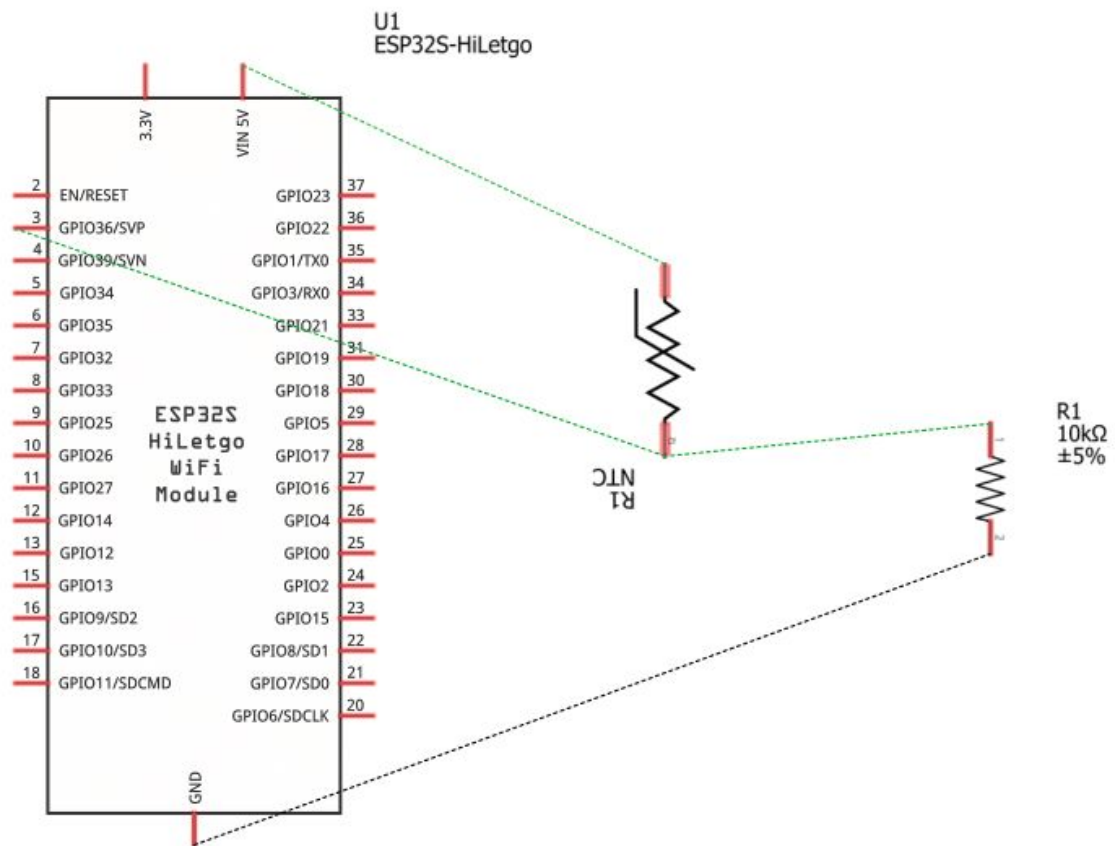
On peut voir que pour faire un pas, il faut délivrer un courant dans les bobines du moteur (4 pins en tout) dans un ordre précis.

Ensuite, voici la fonction RotationOccur le nombre de pas et de révolution que notre moteur doit effectuer (Ici, j'ai codé la fonction pour qu'elle effectue un rotation dans un sens puis dans l'autre, conformément au fonctionnement de notre distributeur automatique):

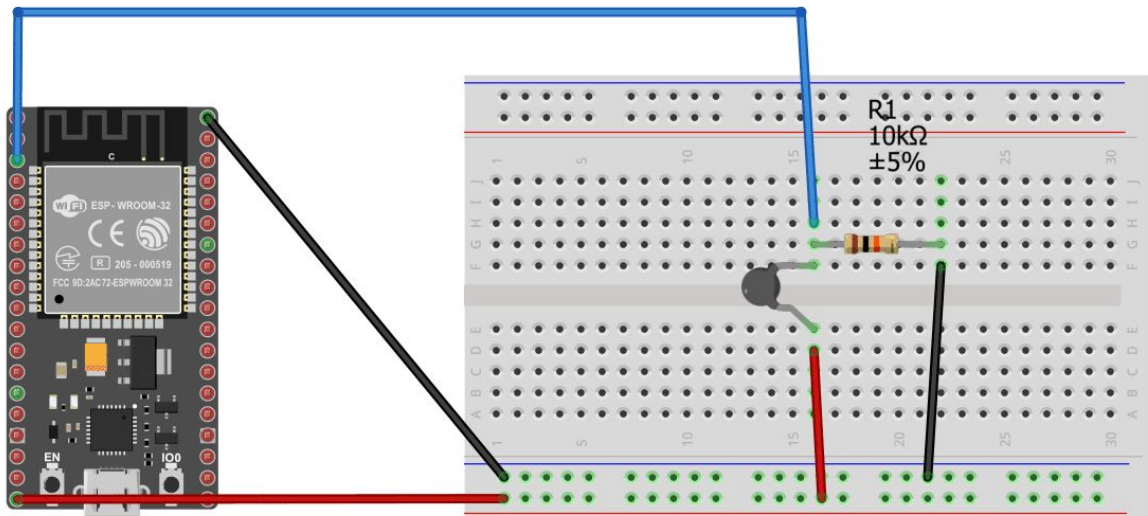
```
void RotationOccur(int o,int stepp){
  for(int j=0;j<o;j++){
    //max step =2048
    for(int a=0;a<stepp;a++){
      OneStep(false);
      //Minimum time between steps
      delay(2);
    }
    for(int a=0;a<stepp;a++){
      OneStep(true);
      //Minimum time between steps
      delay(2);
    }
  }
}
```

Notre moteur effectue 2048 pas par révolution. Le temps minimum à utiliser entre chaque pas pour notre moteur est de 2ms. Sa vitesse maximal est donc de 500 pas par secondes.

Ensuite, nous avons le capteur de température qui nous servira à déterminer si la température est trop basse ou trop élevée.



fritzing



fritzing

Pour calculer la température, il faut d'abord connaître la résistance du thermistor. Pour cela, on va récupérer la tension entre une résistance(R1) et notre thermistor pour ensuite calculer la valeur de la résistance du thermistor en utilisant l'équation d'un diviseur de tension :



$$V_{out} = V_{cc} * \frac{R_2}{R_1 + R_2} \text{ qui peut se réécrire } R_2 = R_1 * \left( \frac{V_{cc}}{V_{out}} - 1 \right)$$

$V_{out}$  : Tension mesurée entre  $R_1$  et le thermistor

$V_{cc}$  : Tension délivrée par le générateur (5V)

$R_1$  : Résistance connue (10 k $\Omega$ )

$R_2$  : Résistance du thermistor à calculer

Enfin, on utilise la relation de Steinhart-Hart pour convertir  $R_2$  en température :

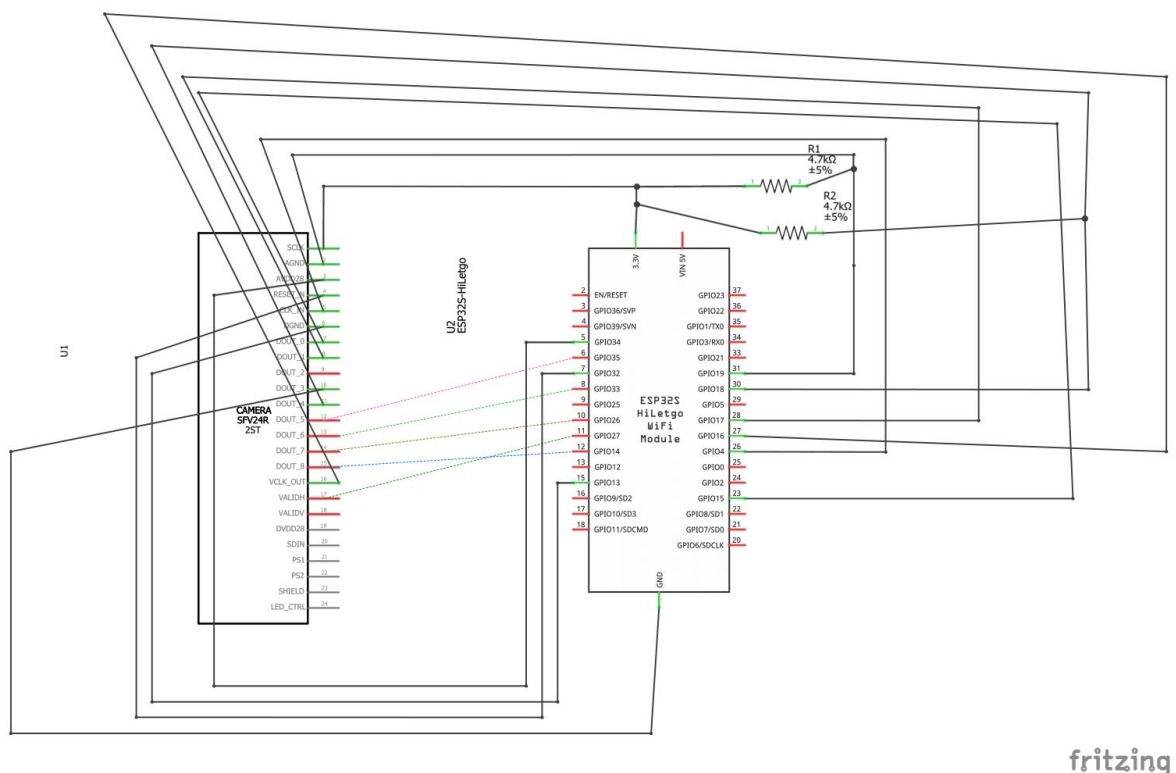
$$T = \frac{1}{A + (B * \log R_2) * C (\log R_2)^3}$$

T : Température en Kelvins

$R_2$  : Résistance en Ohms

A, B, C les coefficients de Steinhart qui caractérisent chaque thermistance

Ensuite la caméra, placée au dessus de la cage, elle retransmet les images en direct.



Enfin le schéma général avec les trois modules connectés à l'esp32.

fritzing



## Problèmes rencontrés

Nous avons détecté un problème assez tard, c'est à propos du thermomètre. En effet, nous nous sommes rendu compte pendant le confinement, que ce module ne marchait pas. Heureusement, la personne qui possédait le matériel chez lui avait dans son stock personnel un module de rechange. Grâce à cela, il a pu finir cette partie du projet.

À notre connaissance nous sommes un des deux seuls groupes qui ai reçu un esp32 au lieu d'une carte arduino. La documentation de notre carte est beaucoup moins importante que celle de l'arduino, il nous a fallu redoubler d'efforts afin de faire fonctionner nos modules.

Nous avons commandé la caméra compatible arduino, avant de recevoir le matériel de l'université, nous pensions donc recevoir une carte arduino. Cependant vous le savez nous avons eu un ESP 32 qui a proposé certains problèmes de comptabilité.

Bien entendu, comme tous les groupes de la classe, nous avons été fortement gênés dû au confinement. En effet seulement une personne possédait le matériel arduino et a dû se débrouiller seulement avec ses yeux, car nous n'avions pas la vision. La communication est aussi plus difficile, malgré le logiciel discord, il y a toujours des limites.

# Répartition des tâches

Tâches	Rôles	Temps passé ( approximativement )
Etat de l'art	Toutle l'équipe	Cinq jours en mars
Mise en place du projet	Toutle l'équipe	Un cours d'IOT en classe
Prise en main sur la caméra et le moteur	Toutle l'équipe	Deux cours d'IOT en classe
Calculer la température avec le thermistor	Hasseeb JAVAID + Clément ZHOU	Deux jours
Réalisation du distributeur de nourriture avec le moteur pas à pas	Hasseeb JAVAID + Clément ZHOU	Cinq jours ( ~1h30/jour )
Réalisation du serveur web	Hasseeb JAVAID	Un jour
Ajout de la caméra au serveur web	Hasseeb JAVAID	Cinq jours ( ~1h30/jour )
Schémas électriques sur Fritzing	Hasseeb JAVAID	Deux heures
Plan de la maquette	Clément ZHOU	Une heures
Réalisation de la maquette en carton	Mathieu JUCI	Deux jours
Aplication WEB, Front-end	Mathieu JUCI + Clément ZHOU	Deux semaines ( ~1h/jour )
Aplication WEB, Back-end	Mathieu JUCI	Cinq jours ( ~5h-6h/jour )
Tournage et montage de la vidéo de présentation	Mathieu JUCI	Trois heures
Ecriture du Resumé	Clément ZHOU	Un jour
Rapport final	Toutle l'équipe	Trois jours

# Conclusion

Notre produit pourrait apporter beaucoup de bienveillance envers les animaux de compagnie, les maîtres qui les abandonnent n'auront plus besoin avec notre projet. Certes, notre cage n'est pas prête pour la vente, mais c'est un bon prototype dont pourrait s'inspirer d'autres personnes.

Nous avons beaucoup appris, du côté électronique notamment. Ce sont des connaissances qui pourront nous servir à nouveau non pas seulement en informatique, mais aussi dans la vie de tous les jours.

Pour conclure ce projet, nous pouvons affirmer qu'il n'était pas comme les autres, c'est la première que nous avons une vraie maquette à construire. L'internet des objets apporte un vrai nouvel aspect et de nouvelles possibilités pour le futur.