

Rapport de projet :



Projet KYUH

Partie personnelle de

Yanis HADJ-BELGACEM

BTS SNIR

Lycée Costebelle



SOMMAIRE

Tables des matières

● Sommaire	2
● Introduction	3
● Carnet de bord.....	4
● Solution technique.....	6
● Diagramme des cas d'utilisation.....	8
● Etude structurelle.....	9
● Diagramme fonctionnelle.....	10
● Maquette du site.....	12
● Description du CSS	14
● Algorithme du php.....	16
● Résultat final	18
● Physique.....	24
● Conclusion.....	25

Introduction

Dans le cadre d'élevage d'abeilles menacées, sans cesse, par l'attaque de frelons, nous allons trouver un moyen de défendre les ruches de ces prédateurs.

Notre objectif est donc de protéger les apiculture et d'améliorer les conditions de travail des apiculteurs.

Voici une estimation de nos dépenses totales : 400 euros.

Afin de mener à bien ce projet, notre groupe « KYUH Frelon » sera composé de 4 personnes :

- Kilian FERREIRA
- Ugo ACQUATI
- Hugo ARBONA
- Yanis HADJ-BELGACEM

Il existe un bouclier de ruche qui permettrait l'entrée, seul, des abeilles et bloquerait les frelons asiatiques hors de la ruche.

Tout apiculteur peut s'en servir.

Pour ma part je dois réaliser un interface homme machine (IHM) pour stocker et visualiser les données récupérées par mes camarades.

Carnet de bord

Dans cette partie, nous avons toutes les tâches effectuées durant chaque semaine de notre projet.

Semaine 1 - 05 au 08 janvier	→ Analyse du dossier. → Répartitions des tâches avec le groupe de projet.
Semaine 2 - 11 au 15 janvier	→ Début de la réalisation des diagrammes de bloc et de séquence. → Listing du matériel pour le groupe. → Début du cahier des charges.
Semaine 3 - 18 au 22 janvier	→ Fin des diagrammes de bloc et de séquence. → Fin du cahier des charges. → Préparations de la revue 1.
Semaine 4 - 25 au 29 janvier	→ Fin de la revue 1. → Début du devis.
Semaine 5 - 01 au 05 février	→ Fin du devis. → Début des recherches sur l'ihm.
Semaine 6 - 08 au 12 février	→ Création de la maquette de l'ihm. → Début de la réalisation de la page web.
Semaine 7 - 15 au 19 février	→ Préparation de la revue 2. → Fin de la revue 2. → Rapport de conception.
Semaine 8 - 08 au 12 mars	→ Connection avec la base de données créée par Hugo Arbona.
Semaine 9 - 15 au 19 mars	→ Création du menu et de mes choix d'onglets. → Création du css. → Développement de l'onglet courbes

	permettant d'analyser la fréquence d'apparitions des frelons.
Semaine 10 - 22 au 26 mars	→ Développement de l'onglet streaming permettant d'observer ce qu'observe la caméra.
Semaine 11 - 29 mars au 02 avril	→ Développement de l'onglet photo permettant d'observer les photos prises par la caméra quand il y a détection de frelons
Semaine 12 - 06 au 09 avril	→ Edition des rapports de projet en groupe et individuel.

Solution technique

Voici ci-dessous la solution technique de la partie IHM du projet KYUH.

Solution	C / C++	PHP	CSS / HTML	Python	JAVASCRIPT
Présentation	C++ est un langage de programmation compilé permettant la programmation sous plusieurs formes : -Programmation orienté objet -Programmation générique	HyperText Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet.	Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web. Ce langage permet d'écrire de l'hypertexte, d'où son nom, de structurer sémantiquement la page, de mettre en forme le contenu.	Python est un langage de programmation interprété, multi paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.	JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web.
Point faible	Dure compréhension du langage.	Nécessite un serveur web.	Pas d'interaction avec le serveur.	Langage interprété. (Lent / pas de compilation)	Interactivité moins évident
Point fort	Langage optimisé	Interaction avec l'utilisateur	Langage de base.	Langage de haut niveau. (Simple d'utilisation et de compréhension)	Possibilité d'interactivité avec la page
Solution	C / C++	PHP	CSS / HTML	Python	JAVASCRIPT
Compatibilité	***	*****	*****	*	*****
Facilité de mise en œuvre	**	*****	*****	***	***

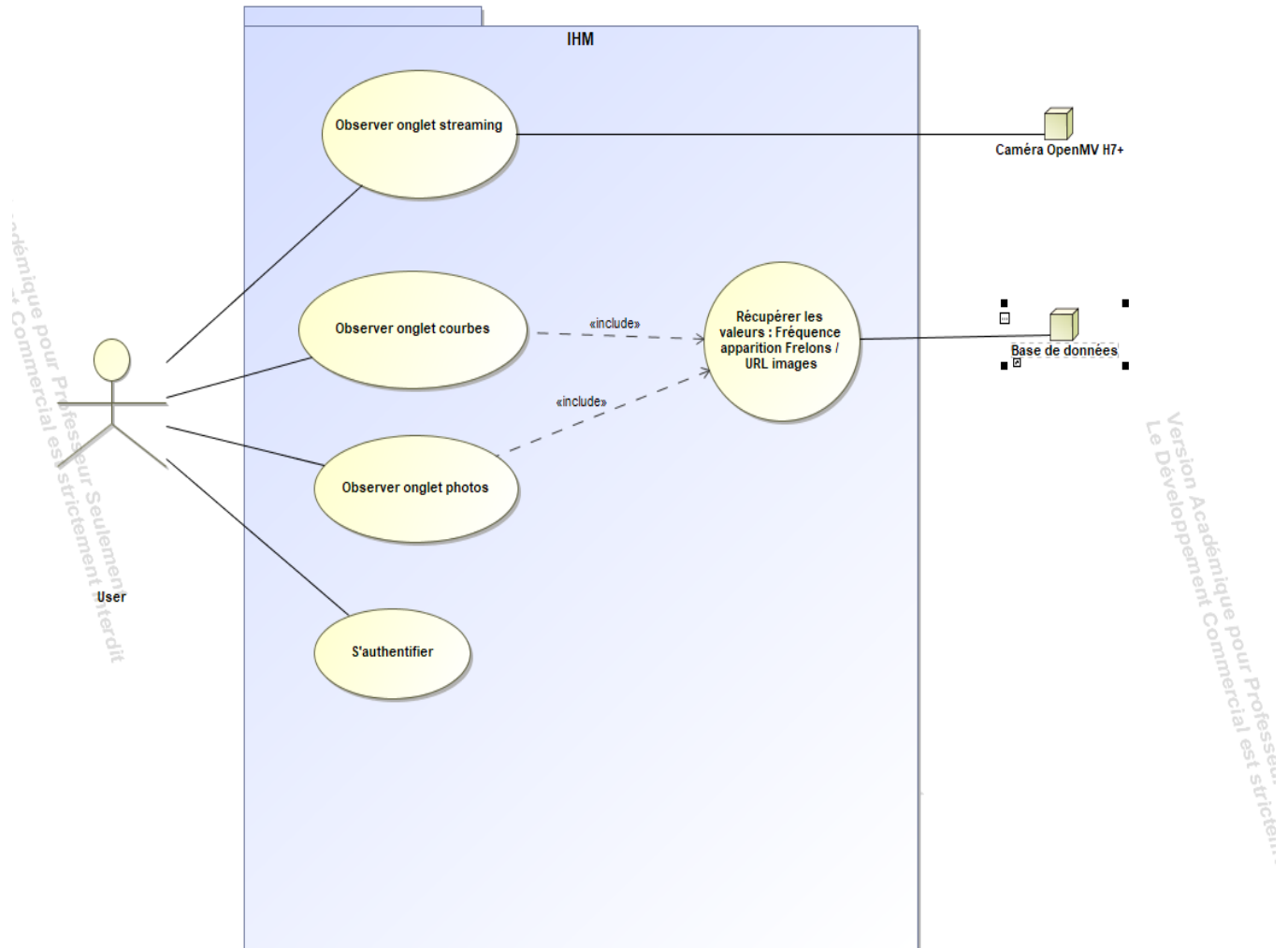
Langages utilisés:

Suite à ceci mon choix se portera sur le langage PHP et HTML/CSS, car les autres langages de programmation ne sont pas adaptés à mes choix d'utilisations et à la facilité à mettre en œuvre.



Diagramme des cas d'utilisation

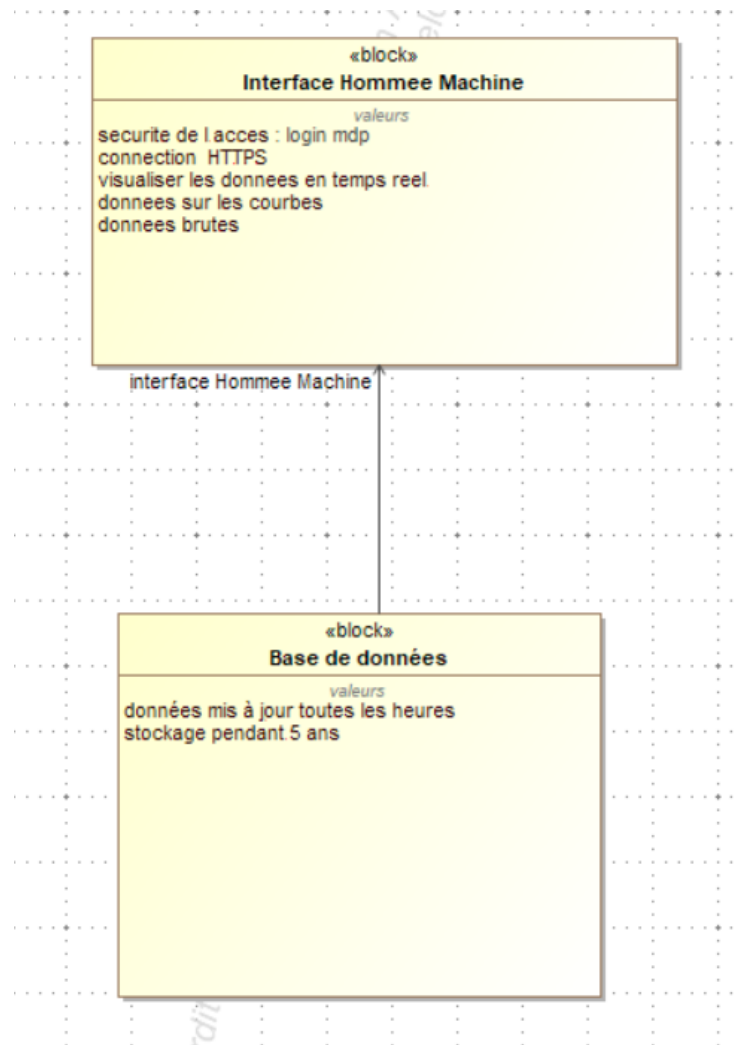
Voici mon diagramme de cas d'utilisation:



Il possède un acteur “ User” qui peut utiliser l’IHM (Interface Homme Machine) en allant dans l’onglet de son choix. Toutes les données sont récupérées de la base de données pour pouvoir les afficher sur l’IHM pour permettre à l’User d’observer.

Etude structurelle

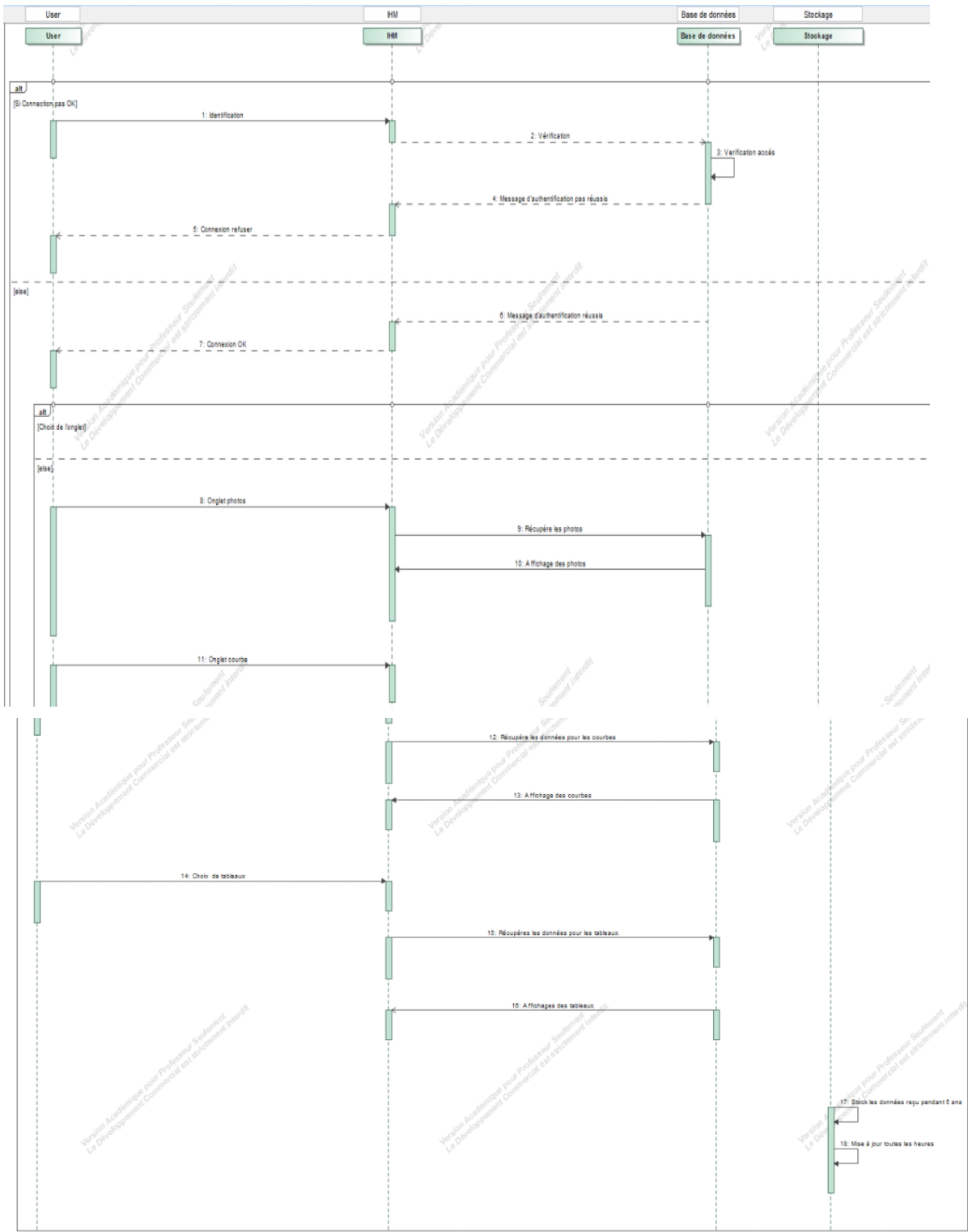
Voici mon diagramme de bloc :



Il permet d'observer les fonctions que l'IHM comme avoir une sécurité à l'accès avec login et mot de passe, une connexion HTTPS, regarder les données en temps réel et observer les données sur des courbes.

Diagramme de séquence

Voici mon diagramme de séquence:



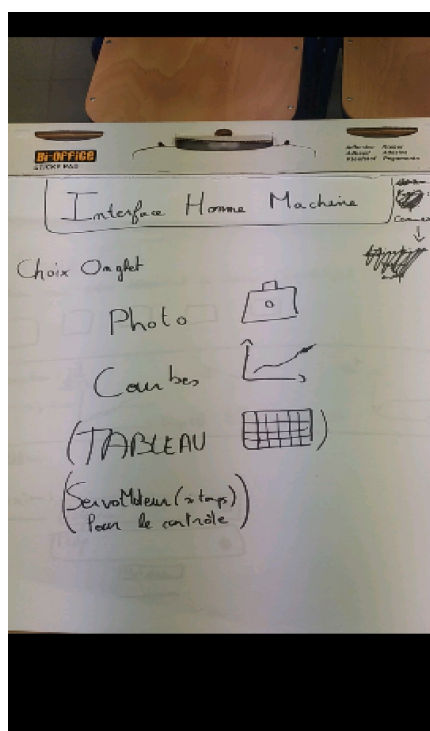
Mon diagramme de séquence possède quatre lignes de vie qui montre la connexion entre le user, l'IHM (interface homme machine), la base de donnée et le stockage.

Le user est la personne pouvant accéder à l'ihm. L'IHM envoie les données à la base de données pour permettre la connexion au user sur l'IHM toutes les données y sont stockées.

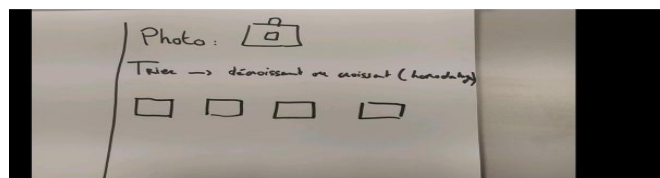
Maquette de site

Voici la maquette de l'ihm du projet KYUH:

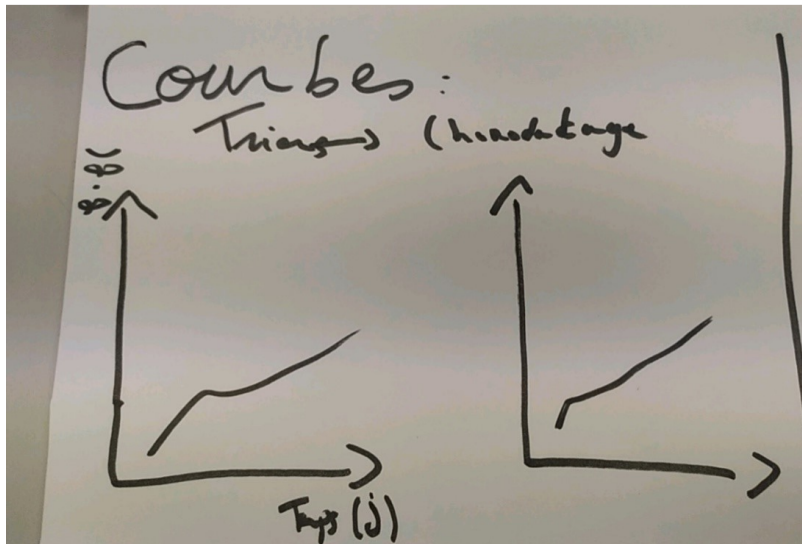
-Tout d'abord j'ai commencé par élaborer le menu.



-Ensuite j'ai dessiné l'onglet "photo".



-Puis la conception de l'onglet "courbes"



-Et pour finir j'ai créé la page de connexion qui est la première page lorsque l'on arrive sur le site.

Connexion

Nom d'utilisateur

Mot de passe

Connexion

Description de CSS

Cette ligne est utilisée pour mettre un fond à l'IHM.

```
background: url(background.png);
```

Voici le bouton de connexion.

Connexion

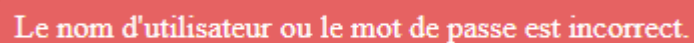
Il s'affiche grâce à box-button ou nous avons aligner le texte, la couleur du fond, la couleur de la police, la taille de la police.

```
.box-button {  
  border-radius: 5px;  
  background: #D59531;  
  text-align: center;  
  cursor: pointer;  
  font-size: 19px;  
  width: 100%;  
  height: 51px;  
  padding: 0;  
  color: #fff;  
  border: 0;  
  outline: 0;  
}
```

Si mot de passe ou le nom d'utilisateur faux alors affiche message d'erreur.

Avec la taille de la bordure, la couleur du fond, la couleur de la police, et la l'espace des bords.

```
p.errorMessage {  
  background-color: #e66262;  
  border: #AA4502 1px solid;  
  padding: 5px 10px;  
  color: #FFFFFF;  
  border-radius: 3px;  
}
```

Le nom d'utilisateur ou le mot de passe est incorrect.

```
#header{
color: #D5CFCF;
text-decoration: none;
background-color: rgb(238, 160, 36);
position: fixed;
top: 0px; left: 0px;
width: 100%;
height: 70px;
}
```

Ceci est la barre de navigation. Sur celle-ci nous pouvons accéder à tous nos onglets avec leur taille, leur position et leur couleur.

Il s'agit de la zone permettant de naviguer entre nos onglets avec la couleur, l'espace des bordures, la taille de la police.

```
#raccourci{
color: black;
padding-top: 20px;
float:right;
width: 25%;
font-size: 150%;
}
#lien1,#lien2,#lien3,#lien4{
margin-right: 1%;
width:15%;
```

```
#lien1{
transition: 1.5s;
position: relative;
}
#lien1:hover{
transition: 1.5s;
top: -5px;
```

Permet de faire une animation lorsque le curseur est sur le lien grâce au hover. Ceci a été fait sur tous les liens.

Algorithme du PHP

Pour commencer nous devons nous connecter à la base de données.

```
// Informations d'identification
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'frelons');

// Connexion à la base de données MySQL
$conn = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);
```

Puis nous vérifions la connexion avec une dernière ligne.

```
// Vérifier la connexion
if($conn === false){
    die("ERREUR : Impossible de se connecter. " . mysqli_connect_error());
}
```

Suite à cela nous devons nous connecter en entrant dans le formulaire les résultats attendus de la base de données en utilisant la méthode POST qui sont le nom d'utilisateur et le mot de passe.

```
<?php

require('config.php');
session_start();
if (isset($_POST['username'])){
    $username = stripslashes($_REQUEST['username']);
    $username = mysqli_real_escape_string($conn, $username);
    $password = stripslashes($_REQUEST['password']);
    $password = mysqli_real_escape_string($conn, $password);
    // $query = "SELECT * FROM users WHERE username='$username' and password='".hash('sha256', $password)."'";
    $query = "SELECT * FROM users WHERE username='$username' and password='$password'";
    $result = mysqli_query($conn,$query) or die(mysql_error());
    $rows = mysqli_num_rows($result);
    if($rows==1){
        $_SESSION['username'] = $username;
        header("Location: login.php");
    }else{
        $message = "Le nom d'utilisateur ou le mot de passe est incorrect.";
    }
}
?>
```

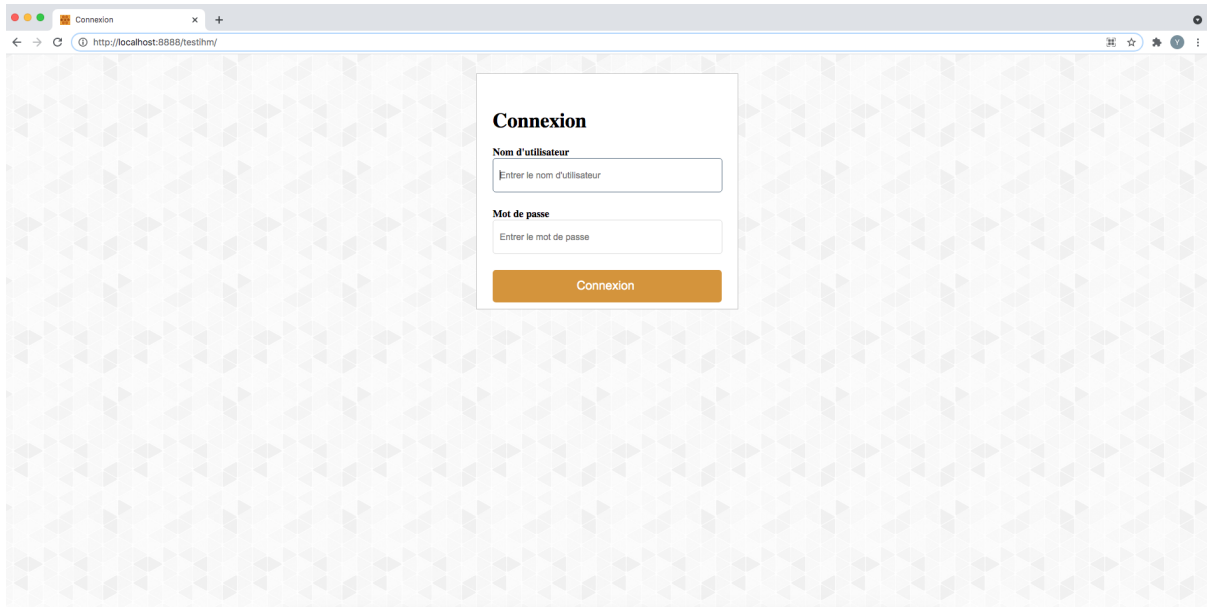

La méthode POST envoie un en-tête et un corps de message au serveur. Le corps est généralement constitué des données entrées dans le champ de formulaire par l'utilisateur. Les données du formulaire n'apparaissent pas dans l'URL.

Dans mon code login.php nous utilisons un GET pour récupérer la valeur de l'URL.

```
if ($_GET["page"]==courbes){  
    include ("courbes.php");  
    include ("menu.php");  
}  
  
else if ($_GET["page"]==photo){  
    include ("photo.php");  
    include ("menu.php");  
}  
else if ($_GET["page"]==streaming){  
    include ("streaming.php");  
    include ("menu.php");  
}  
else{  
    include ("menu.php");  
    include("accueil.php");  
}  
if(!isset($_SESSION["username"])){  
    exit();  
}
```

Résultat final

Voici ma page de connexion lors du lancement du site (“index.php”) et nous pouvons nous connecter grâce à (“connexion.php”).



The screenshot shows a web browser window with the address bar displaying 'http://localhost:8888/testhtml/'. The page has a light gray background with a subtle geometric pattern. In the center, there is a white box titled 'Connexion'. Inside this box, there are two input fields: 'Nom d'utilisateur' with the placeholder text 'Entrez le nom d'utilisateur' and 'Mot de passe' with the placeholder text 'Entrez le mot de passe'. Below these fields is an orange button labeled 'Connexion'.

Une fois connecté nous avons notre page d'accueil (“login.php”).



The screenshot shows a web browser window with the address bar displaying 'http://localhost:8888/projet/login.php'. The page has a light gray background with a subtle geometric pattern. At the top, there is an orange header bar. On the left side of the header is the KYUH logo, and on the right side are the links 'Courbes', 'Photos', 'Streaming', and 'Déconnexion'. Below the header, there is a white box with a black border containing the following text:

Dans le cadre d'élevage d'abeilles à miel menacées, sans cesse, par l'attaque de frelons, nous allons trouver un moyen de défendre les ruches de ces prédateurs.

Notre objectif est donc de protéger les apiculture et d'améliorer les conditions de travail des apiculteurs.

Voici une estimation de nos dépenses totales : 400 euros.

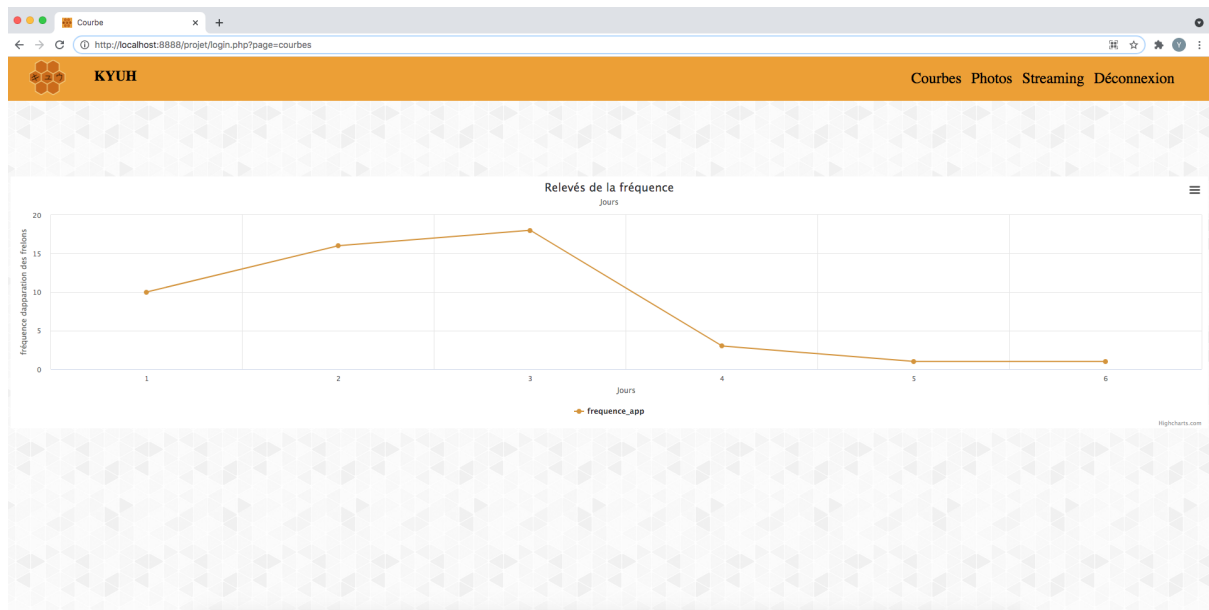
Afin de mener à bien ce projet, notre groupe « KYUH Frelon » sera composé de 4 personnes :

- Kilian FERREIRA
- Ugo ACQUATI
- Hugo ARBONA
- Yanis HADJ-BELGACEM

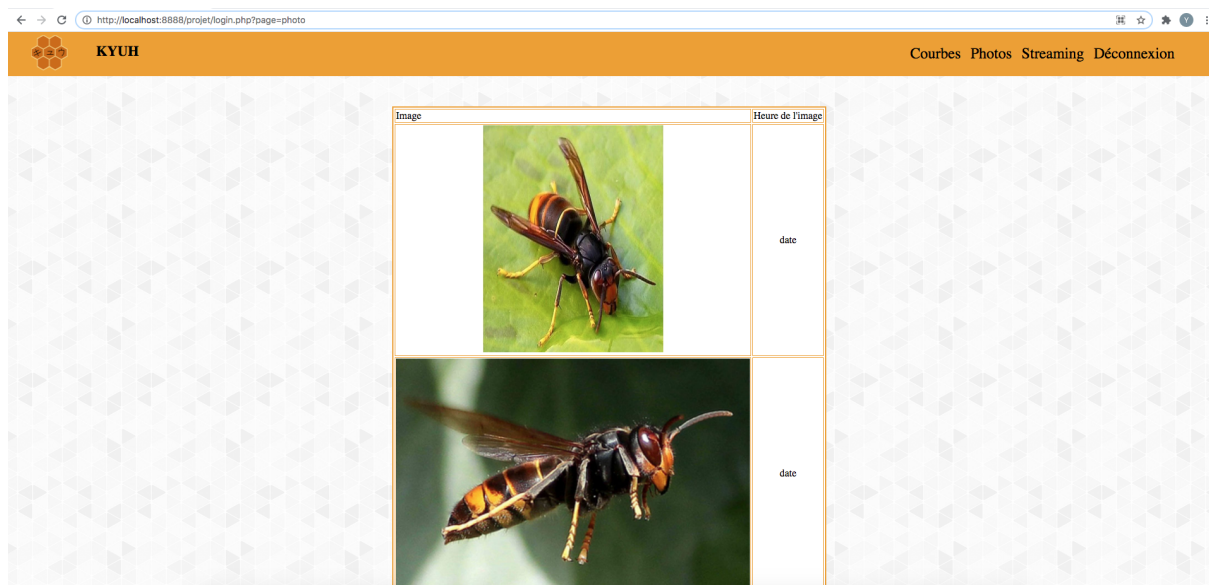
Il existe un bouclier de ruche qui permettrait l'entrée, seul, des abeilles et bloquerait les frelons asiatiques hors de la ruche.

Tout apiculteur peut s'en servir.

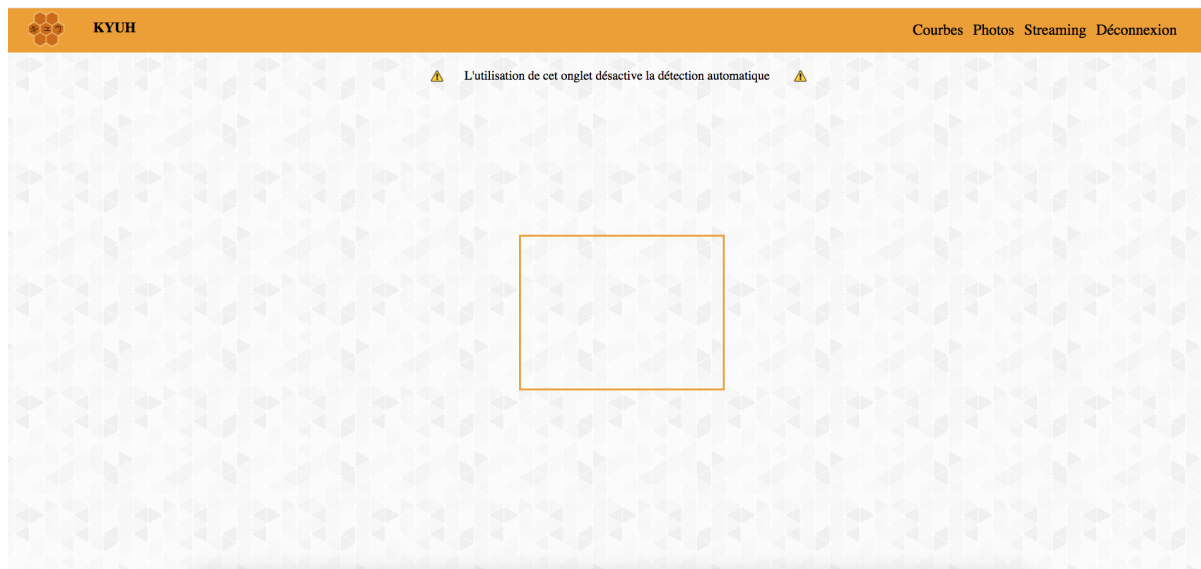
Nous allons observer l'onglet courbes ("courbes.php") qui nous permet d'observer la fréquence d'apparitions des frelons par jours.



Puis sur l'onglet photo ou nous allons mettre des exemples de photos ("photo.php"). Qui permet d'observer lorsque la caméra produit une alerte envoie une photo avec sa date et l'heure pour l'afficher.



Puis notre dernier onglet streaming qui nous permet de voir en temps réel ce que la caméra observe (“streaming.php”).



Et notre onglet déconnexion (“logout.php”) qui nous permet de nous ramener à notre “index.php”.

Physique

Mesure physique : Mesures autour de la modulation de l'information sur une transmission IP

Pour ceci nous avons eu besoin de matériel :

Nous utilisons un analyseur de spectre portable, spectran HE-6065.



Ainsi que notre émetteur WI-FI le Raspberry PI 3.

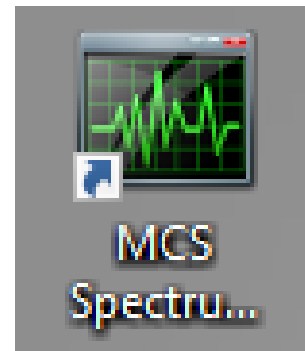


Une fois le matériel branché nous utilisons deux logiciels pour observer la modulation de l'adresse IP :

- **Wireshark** est un analyseur de paquets libre et gratuit. Il est utilisé dans le dépannage et l'analyse de réseaux informatiques, le développement de protocoles, l'éducation et la rétro-ingénierie.



- **MCS Spectrum Analyzer** est un logiciel qui permet d'analyser les spectres de champ électromagnétique.



*Ethernet 2

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless

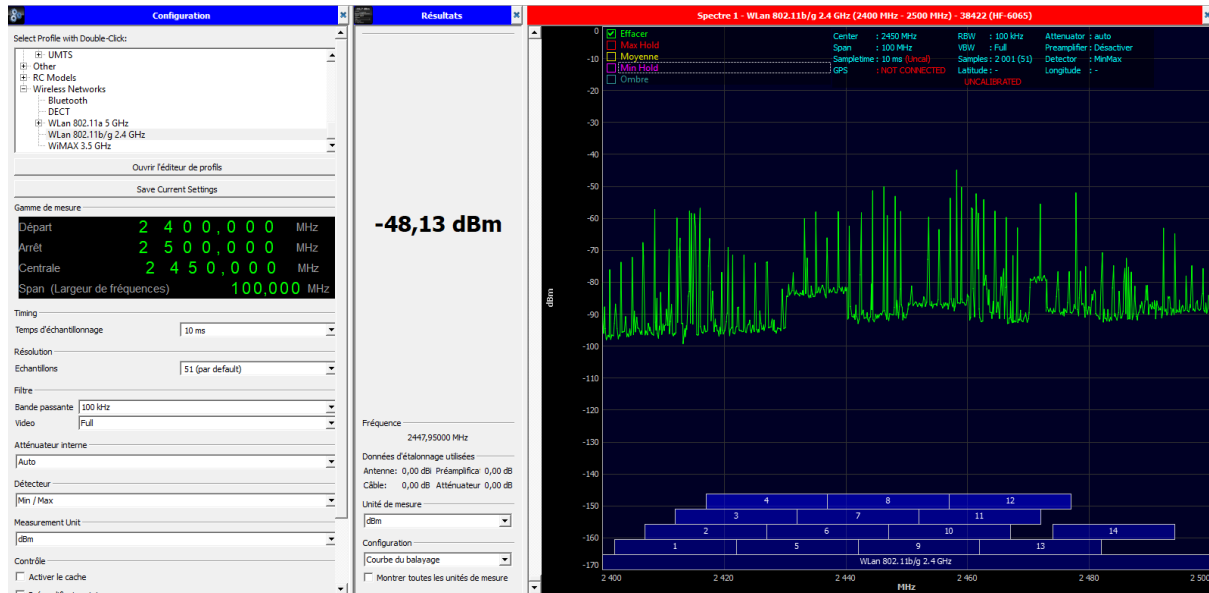
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol
1	0.000000	fa:45:ce:4f:59:c3	da:b0:85:ac:e7:f6	ARP
2	0.000013	da:b0:85:ac:e7:f6	fa:45:ce:4f:59:c3	ARP
3	0.097225	192.168.42.24	64.233.166.188	TCP
4	0.139634	64.233.166.188	192.168.42.24	TCP
5	12.668462	192.168.42.24	172.217.18.238	UDP
6	12.704402	172.217.18.238	192.168.42.24	UDP
7	15.229983	192.168.42.24	192.168.42.129	DNS
8	15.386517	192.168.42.24	192.168.42.129	DNS
9	15.412633	192.168.42.129	192.168.42.24	DNS
10	15.413158	192.168.42.24	192.168.42.129	DNS
11	15.569575	192.168.42.24	192.168.42.129	DNS
12	15.619721	192.168.42.129	192.168.42.24	DNS
13	15.620211	192.168.42.24	192.168.42.129	DNS
14	15.776870	192.168.42.24	192.168.42.129	DNS
15	15.827809	192.168.42.129	192.168.42.24	DNS
16	15.829396	192.168.42.24	192.168.42.129	DNS
17	15.869730	192.168.42.129	192.168.42.24	DNS
18	15.873009	192.168.42.24	192.168.42.129	DNS
19	16.028496	192.168.42.24	192.168.42.129	DNS
20	16.044253	192.168.42.129	192.168.42.24	DNS
21	16.044758	192.168.42.24	192.168.42.129	DNS
22	16.201308	192.168.42.24	192.168.42.129	DNS

> Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
 > Ethernet II, Src: fa:45:ce:4f:59:c3 (fa:45:ce:4f:59:c3), Dst: da:b0:85:ac:e7:f6 (da:b0:85:ac:e7:f6)
 > Address Resolution Protocol (request)

Nous allons dans un premier temps observer le trafic réseau résiduel lorsqu'il n'y a pas de sollicitation particulière du point d'accès. Pour cela nous utilisons Wireshark. On peut observer que l'adresse source envoie une vingtaine de trame ethernet en 16 secondes à l'adresse de destination.

Suite à ça nous observons le spectre du champ électromagnétique WI-FI grâce à l'analyseur spectrale et à son application.



On observe une puissance de -48.13 dBm lorsqu'il envoie peu d'informations.

Ensuite nous allons observer sur wireshark et mcs spectrum analyzer en envoyant beaucoup d'information grâce à un speed test.

Ethernet 2

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.42.24	172.217.171.202	TCP	54	51235 → 443 [ACK] Seq=1 Ack=1 Win=513 Len=0
2	0.268648	172.217.171.202	192.168.42.24	TLSv1.2	440	Application Data
3	0.268648	172.217.171.202	192.168.42.24	TLSv1.2	389	Application Data, Application Data
4	0.268648	172.217.171.202	192.168.42.24	TLSv1.2	93	Application Data
5	0.268691	192.168.42.24	172.217.171.202	TCP	54	51235 → 443 [ACK] Seq=1 Ack=761 Win=510 Len=0
6	0.269871	192.168.42.24	172.217.171.202	TLSv1.2	93	Application Data
7	0.279802	172.217.171.202	192.168.42.24	TCP	54	443 → 51235 [ACK] Seq=761 Ack=40 Win=742 Len=0
8	3.728307	192.168.42.24	10.40.42.2	TCP	55	51282 → 443 [ACK] Seq=1 Ack=1 Win=513 Len=1 [TCP segment of a reassembled PDU]
9	4.639699	192.168.42.24	172.217.18.46	TLSv1.2	294	Application Data
10	4.639785	192.168.42.24	172.217.18.46	TLSv1.2	93	Application Data
11	4.639837	192.168.42.24	172.217.18.46	TCP	1484	51180 → 443 [ACK] Seq=280 Ack=1 Win=510 Len=1430 [TCP segment of a reassembled PDU]
12	4.639837	192.168.42.24	172.217.18.46	TCP	1484	51180 → 443 [ACK] Seq=1710 Ack=1 Win=510 Len=1430 [TCP segment of a reassembled PDU]
13	4.639837	192.168.42.24	172.217.18.46	TLSv1.2	77	Application Data
14	4.639901	192.168.42.24	172.217.18.46	TCP	1484	51180 → 443 [ACK] Seq=3163 Ack=1 Win=510 Len=1430 [TCP segment of a reassembled PDU]
15	4.639901	192.168.42.24	172.217.18.46	TLSv1.2	825	Application Data
16	4.641644	192.168.42.24	172.217.18.238	UDP	1392	62941 → 443 Len=1350
17	4.641681	192.168.42.24	172.217.18.238	UDP	1392	62941 → 443 Len=1350
18	4.641700	192.168.42.24	172.217.18.238	UDP	1392	62941 → 443 Len=1350
19	4.641720	192.168.42.24	172.217.18.238	UDP	1392	62941 → 443 Len=1350
20	4.641744	192.168.42.24	172.217.18.238	UDP	1392	62941 → 443 Len=1350
21	4.641761	192.168.42.24	172.217.18.238	UDP	650	62941 → 443 Len=608
22	4.667521	172.217.18.46	192.168.42.24	TCP	54	443 → 51180 [ACK] Seq=1 Ack=280 Win=7747 Len=0

> Frame 24: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{1D06327D-FF23-4171-88EC-636098FEE86D}, id 0

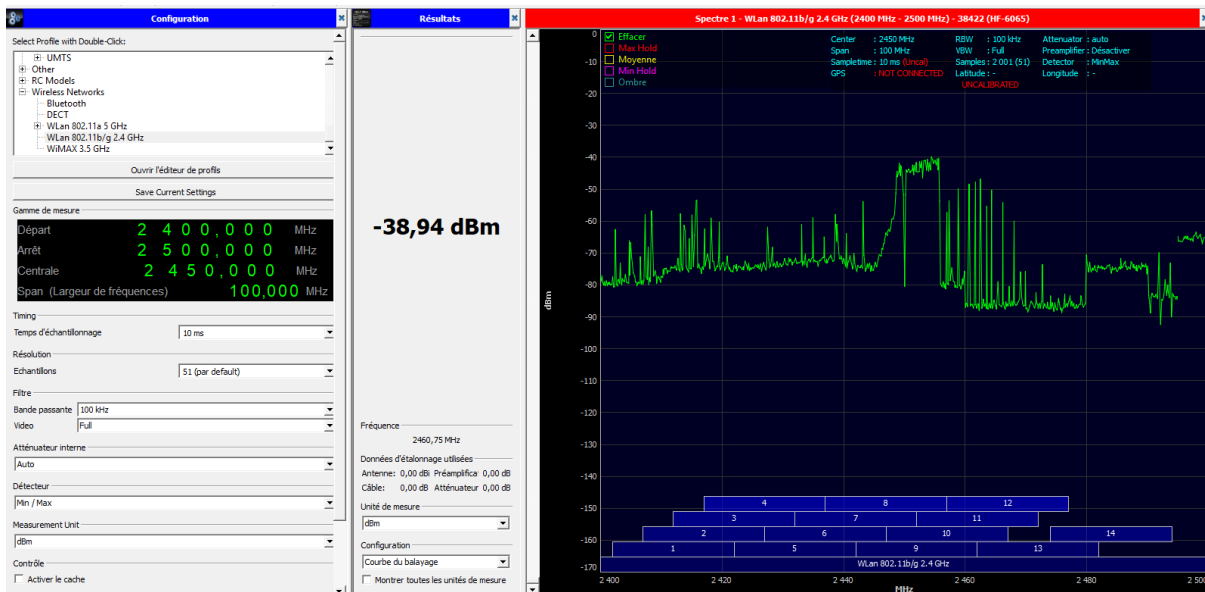
> Ethernet II, Src: fa:45:ce:4f:59:c3 (fa:45:ce:4f:59:c3), Dst: da:b0:85:ac:e7:f6 (da:b0:85:ac:e7:f6)

> Internet Protocol Version 4, Src: 172.217.18.46, Dst: 192.168.42.24

> Transmission Control Protocol, Src Port: 443, Dst Port: 51180, Seq: 40, Ack: 3140, Len: 0

On a pu observer une trame qui est très rapide durant l'acquisition de la trame avec Wireshark.

Ensuite nous avons observé le spectre lors de la sollicitation particulière et on a pu observer un gain de +10 dBm.



Conclusion

Ce projet nous a appris comment travailler et s'entraider dans le cadre d'une équipe avec un objectif commun. Cela m'a permis de mettre en œuvre ce que j'ai pu apprendre durant mes deux ans de BTS SNIR.

Ce fut l'occasion d'être confrontés à différentes difficultés dues à la crise sanitaire. Nous avons dû travailler sur nos affaires personnelles, n'ayant pas le même système d'exploitation qu'en cours j'ai dû m'adapter à travailler en local ce qui m'a poussé à avoir plusieurs versions de code avec deux bases de données différentes.

Ce projet m'a permis d'acquérir de l'expérience pour mon futur projet professionnel dans le développement.