

# **[Projet Etable]**

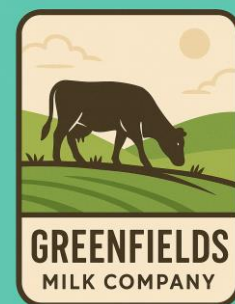
## **[2024 - 2025]**

---

**[LYCEE HENRI BRISSON]**

---

**[BTS CIEL 2 : cybersécurité, informatique  
et réseaux, électronique option A  
informatique et réseaux]  
Auteur : [HEM ARTHUR]**



<b>INTRODUCTION .....</b>	<b>3</b>
<b>CONTEXTE DU PROJET .....</b>	<b>4</b>
<b>REPARTITION DES TACHES.....</b>	<b>5</b>
<b>LES DIAGRAMMES.....</b>	<b>6</b>
DIAGRAMME GANTT.....	6
DIAGRAMME D'EXIGENCES .....	7
DIAGRAMME DE CAS D'UTILISATION.....	8
DIAGRAMME DE SEQUENCE .....	9
DIAGRAMME DE CLASSE .....	10
DIAGRAMME D'ACTIVITE.....	11
<b>MLD - MCD.....</b>	<b>12</b>
<b>TRAVAIL PERSONNEL DETAILLE .....</b>	<b>13</b>
1. BADGE RFID .....	13
2. BASE DE DONNEES .....	15
3. SITE WEB (IHM) .....	16
4. CONCEPTION DE LA MAQUETTE .....	16
<b>OUTILS UTILISES .....</b>	<b>17</b>
<b>LINUX .....</b>	<b>17</b>
<b>RASPBERRY PI 4 .....</b>	<b>17</b>
<b>PHPMYADMIN .....</b>	<b>17</b>
<b>DRAW.IO .....</b>	<b>17</b>
<b>CHROMIUM.....</b>	<b>17</b>
<b>ARDUINO IDE .....</b>	<b>18</b>
<b>CHATGPT.....</b>	<b>18</b>
<b>GITHUB.....</b>	<b>18</b>
<b>MQTT .....</b>	<b>18</b>
<b>PYTHON .....</b>	<b>18</b>
<b>CONCLUSION.....</b>	<b>19</b>

# Introduction

Dans le cadre de l'achèvement de ma formation, il m'a été confié un rôle au sein d'un projet collectif nommé Étable. Ce projet de groupe avait pour mission de centraliser des informations au sein d'un site web unique et fonctionnel. Pour mener à bien cette mission, j'ai été amené à mobiliser l'ensemble des compétences acquises durant mes deux années de formation, notamment en réseaux, en développement et dans divers domaines techniques essentiels. Ce projet a représenté une opportunité concrète de mettre en pratique mes connaissances, tout en collaborant activement au sein d'une équipe pour atteindre un objectif commun.

L'objectif de ce rapport est de détailler les différentes tâches qui m'ont été confiées ainsi que les réalisations accomplies au cours de cette période de projet. Mon rôle s'est articulé principalement autour de la mise en place de l'infrastructure technique. J'ai tout d'abord procédé à la configuration d'un Raspberry Pi 4, sur lequel j'ai installé un serveur LAMP (Linux, Apache, MySQL, PHP). J'ai ensuite conçu la base de données, mis en place un script de sauvegarde automatique, puis développé l'ossature du site web ainsi que son interconnexion avec la base de données. Enfin, j'ai assuré le lien entre le protocole MQTT et la base de données, permettant ainsi l'affichage en temps réel des informations transmises par les capteurs.

## Contexte Du Projet

Le projet Étable est né d'un besoin concret exprimé par un éleveur bovin, confronté à une problématique récurrente dans son activité quotidienne : l'absence de centralisation et de précision dans le suivi de la production laitière de chaque vache, ainsi que le manque d'informations fiables concernant la température du lait — un indicateur essentiel pour détecter d'éventuelles maladies ou anomalies. Ce manque de données en temps réel rendait difficile une gestion optimale de l'élevage, tant sur le plan sanitaire que productif.

Face à cette problématique, le projet a été conçu pour répondre à ces besoins spécifiques par le biais d'un site web centralisant toutes les informations utiles.

**Les grandes étapes du projet ont donc consisté à :**

- Concevoir une infrastructure de collecte et de traitement des données,
- Développer un système de stockage et d'affichage en temps réel,
- Et créer une interface claire et accessible permettant à l'éleveur de consulter rapidement les informations pertinentes pour chaque vache.

# Répartition des tâches

Le projet Étable a été réalisé en groupe, chaque membre de l'équipe se voyant attribuer des responsabilités

En ce qui me concerne, mes responsabilités se sont articulées autour de

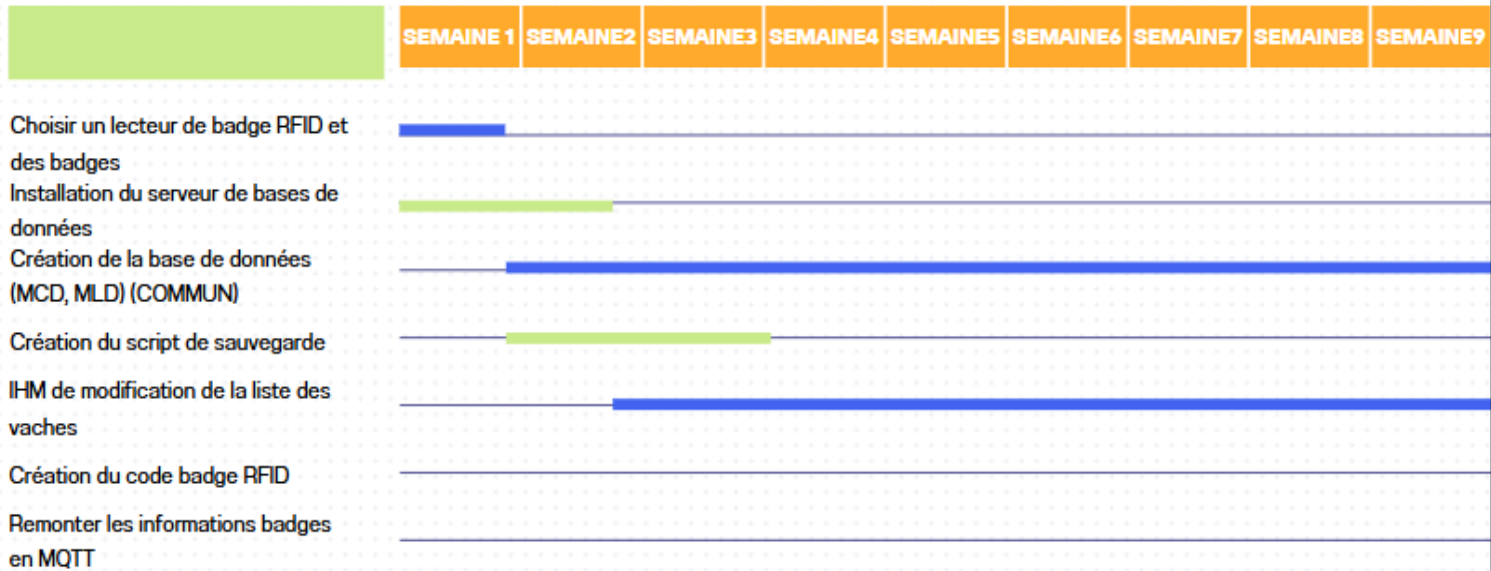
## Quatre axes principaux :

1. **Le choix du badge RFID** : J'ai étudié différentes solutions afin de sélectionner un système de badge RFID adapté à l'environnement de l'étable et compatible avec notre matériel.
2. **Le développement du code pour le lecteur RFID** : J'ai ensuite conçu le programme nécessaire pour assurer l'identification correcte des vaches via leur badge, permettant de les relier individuellement aux données collectées.
3. **L'interface homme-machine (IHM)** : J'ai été chargé de la création de la page web gestion, l'objectif étant de proposer une solution simple à l'éleveur pour consulter toutes les informations utiles.
4. **La mise en place du serveur LAMP et de la base de données** : J'ai installé et configuré le serveur, conçu la base de données pour stocker les données des capteurs, et développé un script de sauvegarde automatique pour en garantir la continuité.
5. **Le flux d'informations en temps réel via MQTT** : Enfin, j'ai mis en place la liaison entre le protocole MQTT et notre base de données, assurant ainsi la transmission et l'affichage instantané des données mesurées par les capteurs (production et température du lait, etc..).

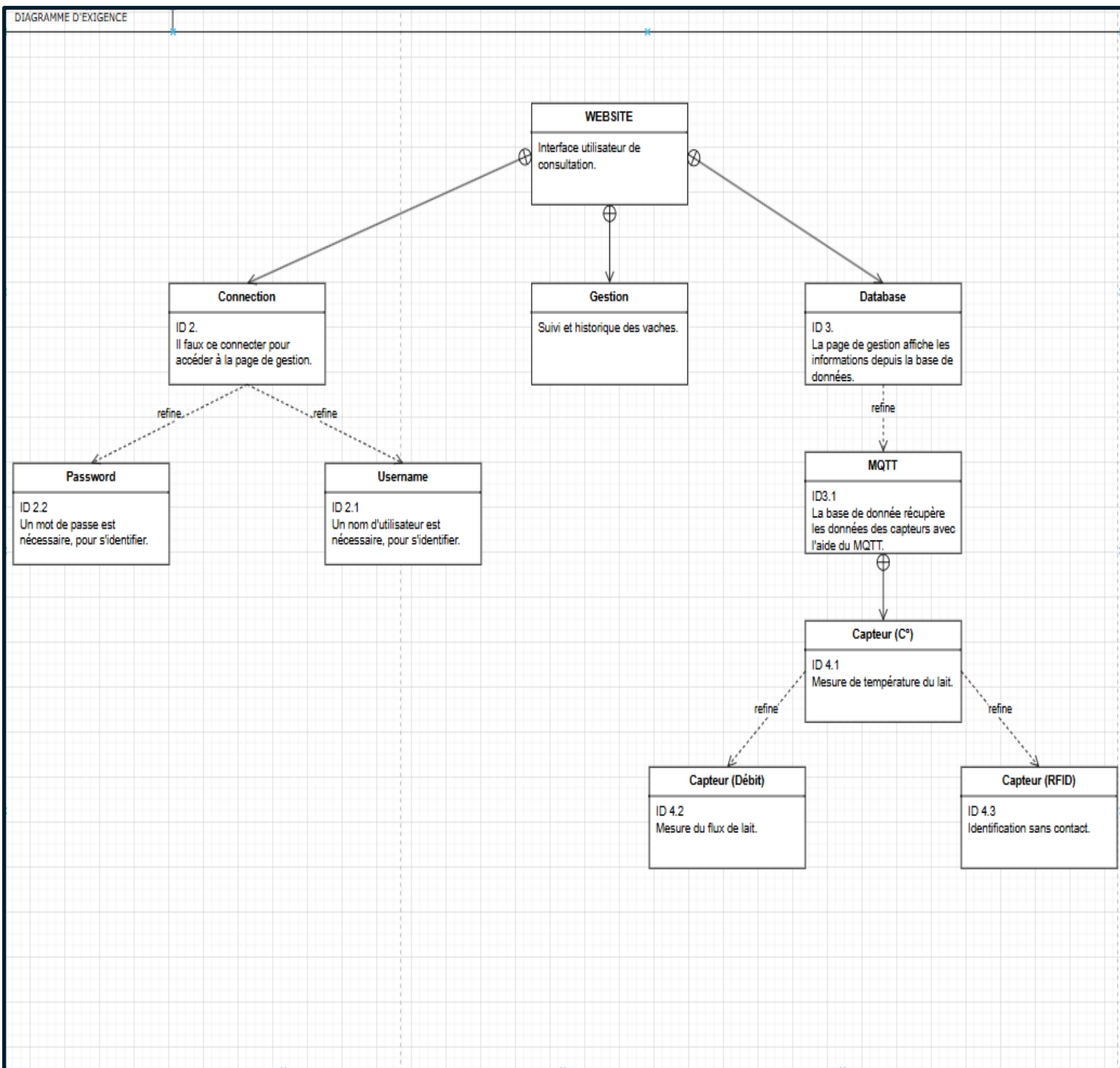
# Les Diagrammes

## Diagramme Gantt

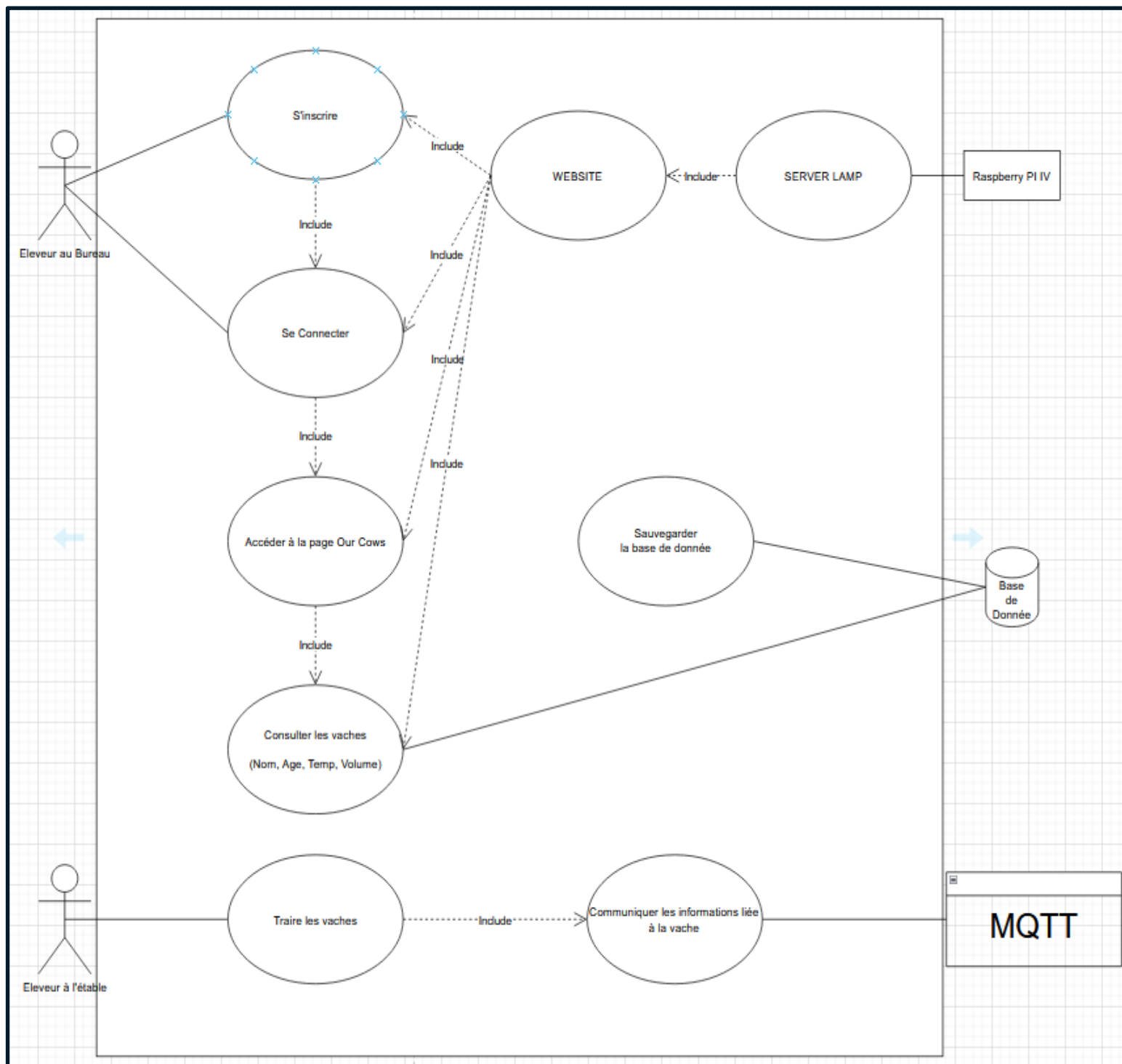
### Gantt Chart PROJETS ETABLE



# Diagramme d'exigences

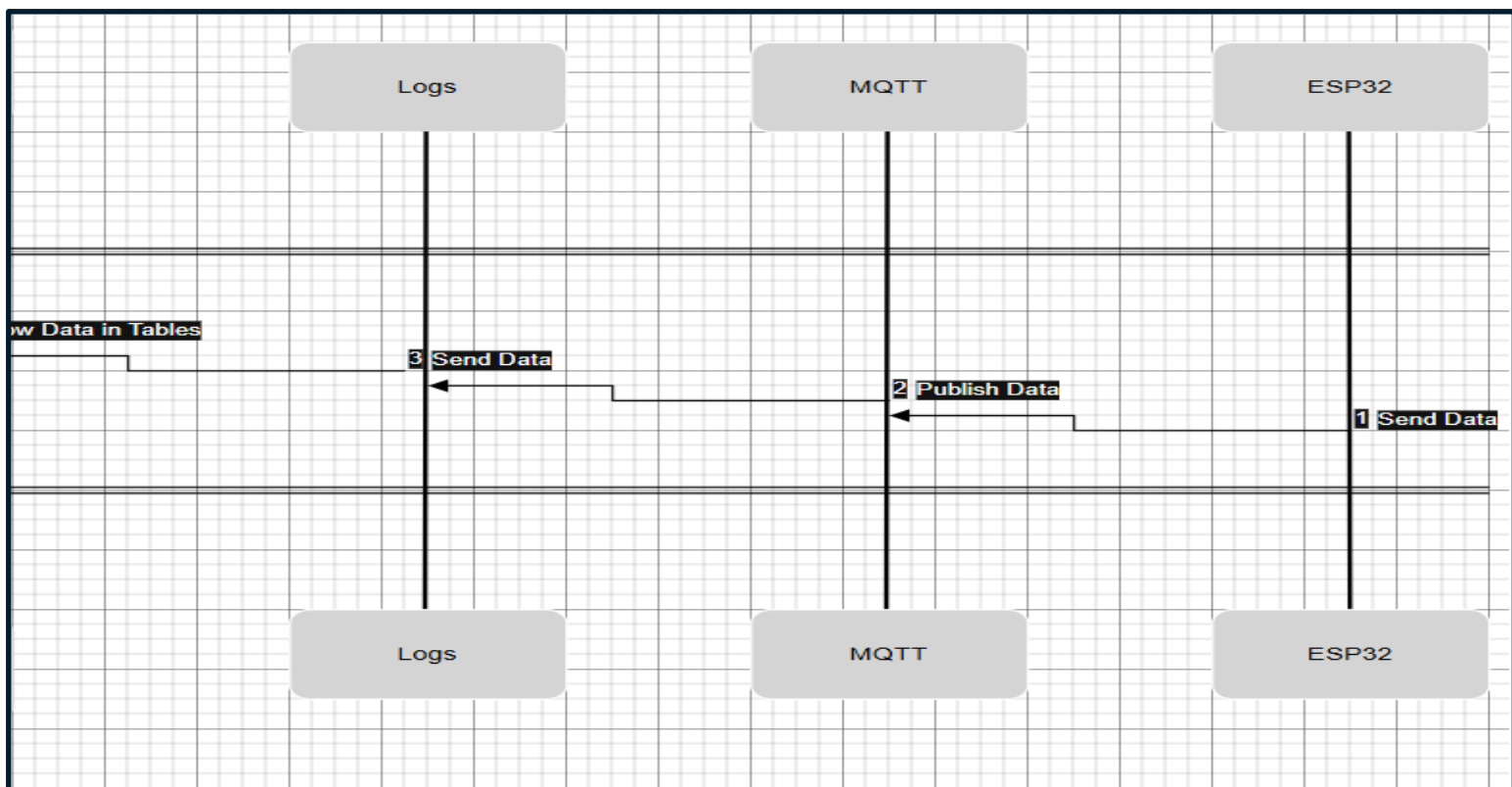
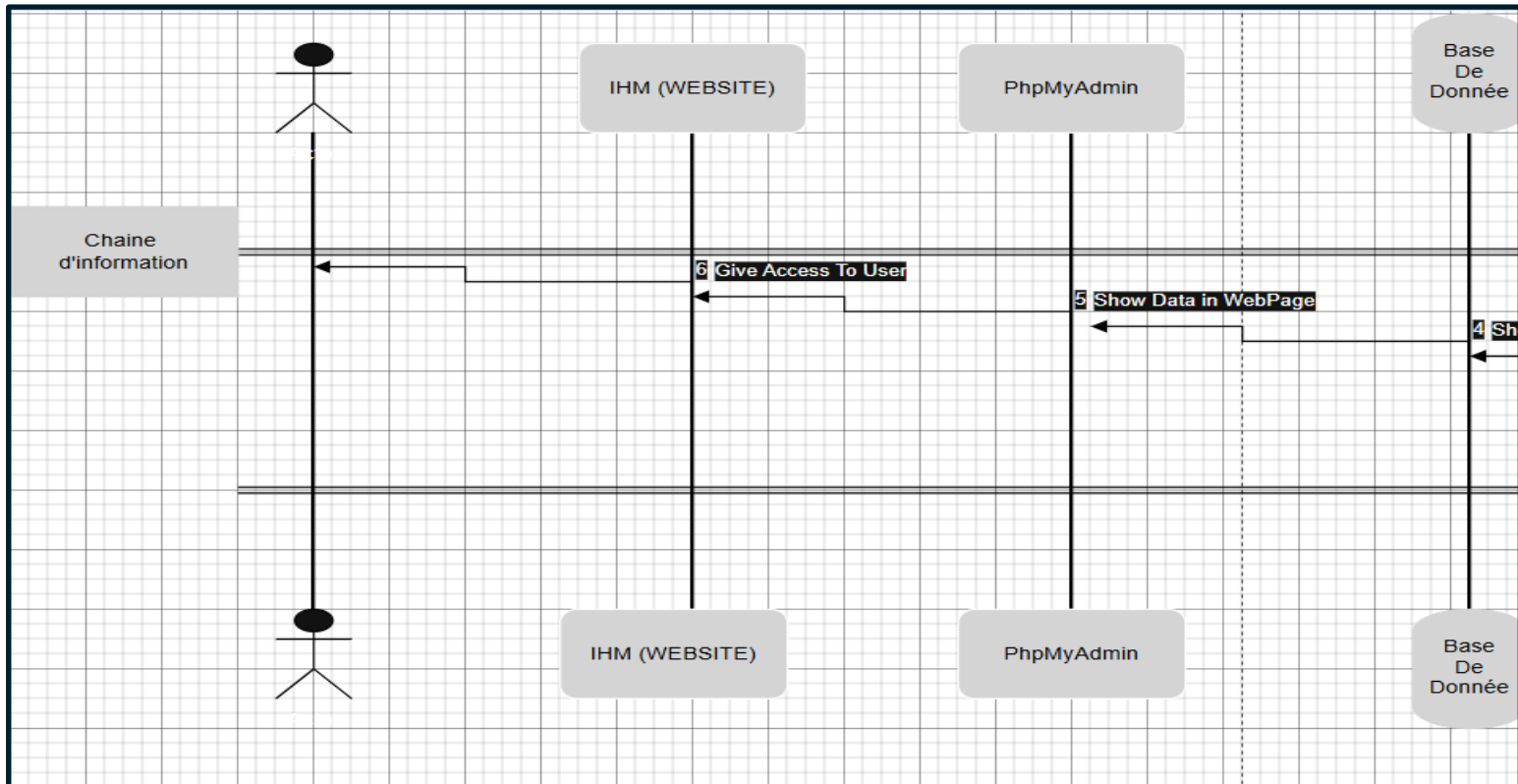


## Diagramme de cas d'utilisation

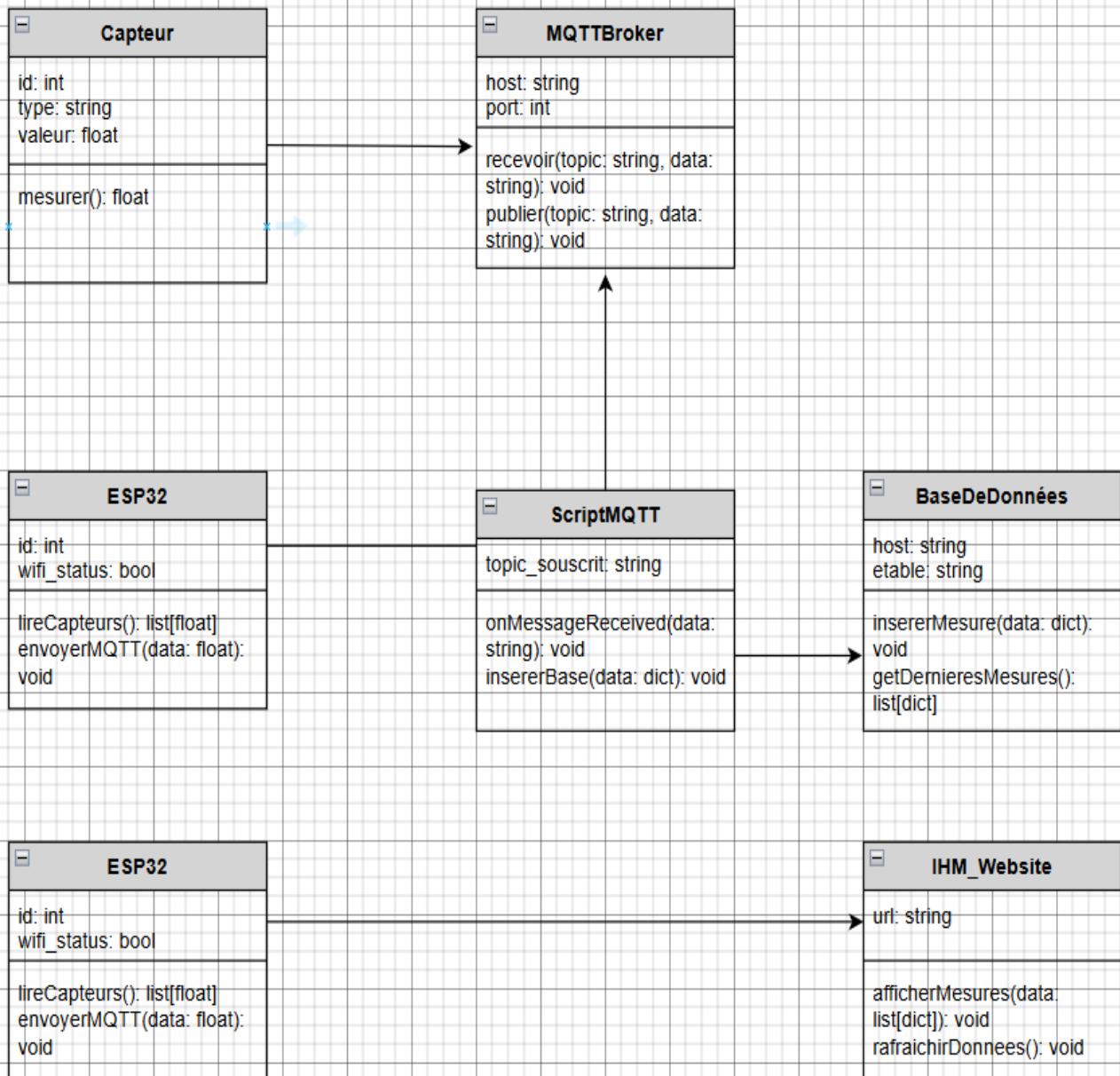




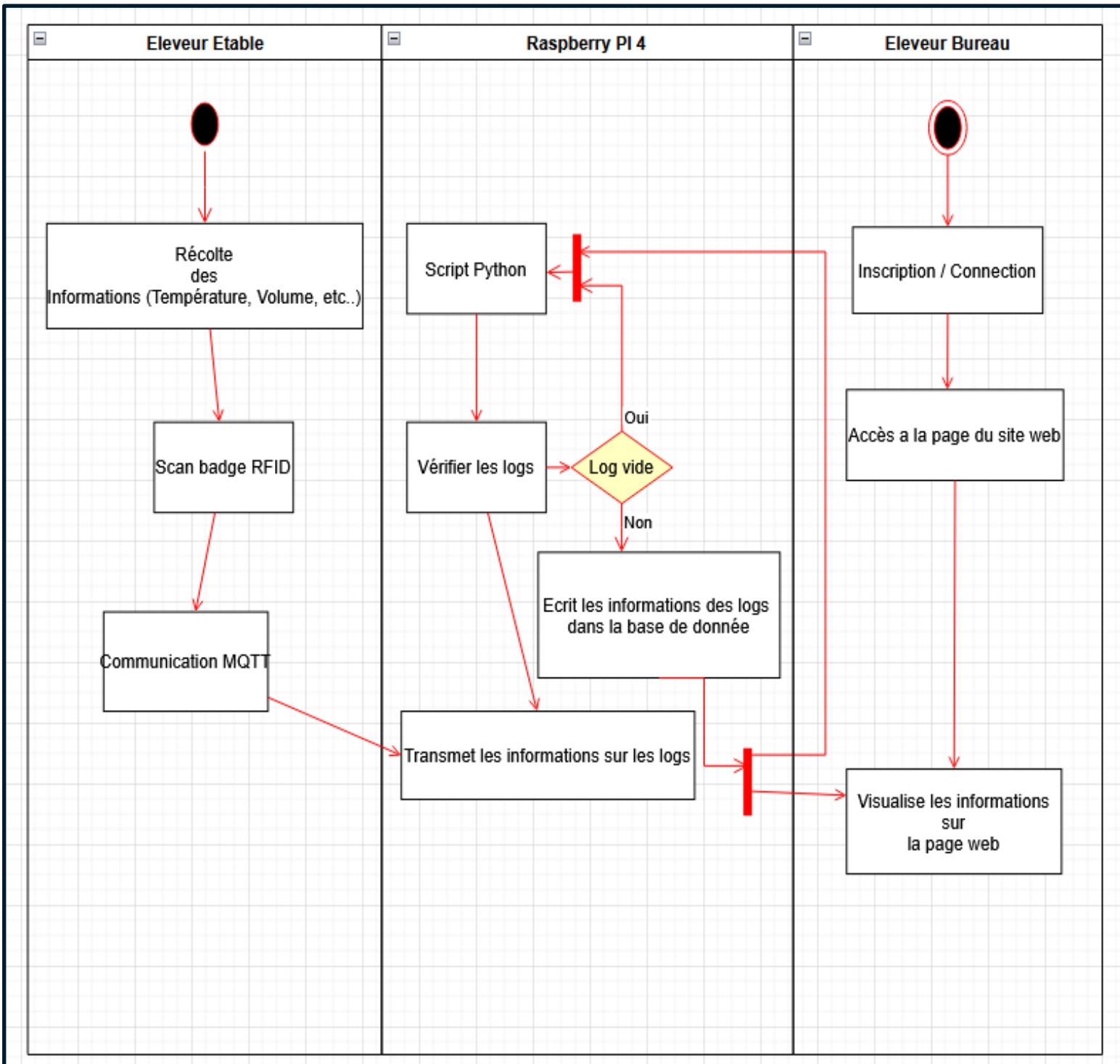
## Diagramme de séquence



## Diagramme de classe



## Diagramme d'activité



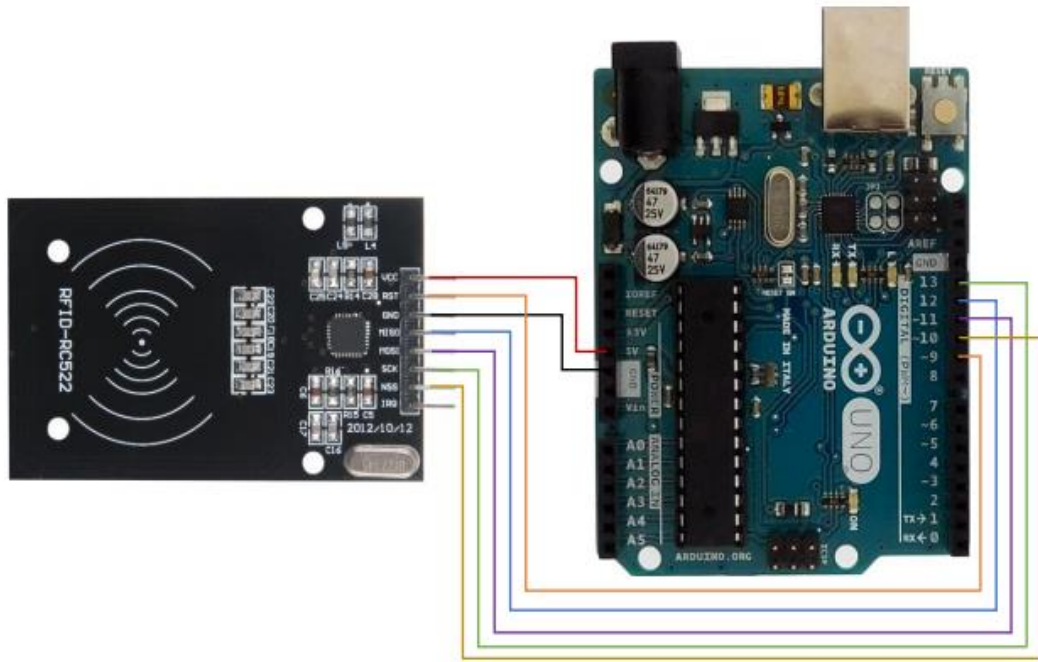
## MLD - MCD



# Travail personnel détaillé

## 1. Badge RFID

- Choix du modèle de badge RFID adapté au projet.



- Écriture du code de lecture et de traitement du badge via Arduino.

```
void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    // Vérifier la présence d'un tag RFID
    if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
        Serial.print("RFID Tag Detecté: ");

        // Convertir l'UID RFID en String
        String tagID = "";
        for (byte i = 0; i < rfid.uid.size; i++) {
            tagID += String(rfid.uid.uidByte[i], HEX);
        }
        Serial.println(tagID);

        // Convertir String en char array
        char tagCharArray[30];
        tagID.toCharArray(tagCharArray, sizeof(tagCharArray));

        // Publier les données RFID via MQTT
        if (client.publish(mqtt_topic_rfid, tagCharArray)) {
            Serial.println("RFID publié avec succès !");
        } else {
            Serial.println("Échec de la publication MQTT");
        }
    }
}
```

```
delay(100);

// Mettre à jour cowID dans la base de données
if (conn.connected()) {
    char query[200];
    sprintf(query, "UPDATE cows SET rfid_tag='%s' WHERE cowID='%d'", tagCharArray, cowID);

    // Création d'un curseur MySQL pour exécuter la requête
    MySQL_Cursor *cur = new MySQL_Cursor(&conn);
    cur->execute(query);
    delete cur; // Libération de la mémoire
    Serial.println("RFID assigné à la vache avec succès");
} else {
    Serial.println("Connexion à MariaDB perdue !");
}

Serial.print("Cow ID Assigné: ");
Serial.println(cowID);

cowID++;

rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
}
```

- Liaison du lecteur RFID avec le Raspberry Pi pour centralisation des identifiants.

## 2. Base de Données

- Installation du serveur LAMP sur le Raspberry Pi 4.

```
# Mise à jour du système
sudo apt update && sudo apt upgrade -y

# Installation d'Apache
sudo apt install apache2 -y
sudo systemctl enable apache2
sudo systemctl start apache2

# Installation de MariaDB
sudo apt install mariadb-server -y
sudo systemctl enable mariadb
sudo systemctl start mariadb

# Sécurisation de l'installation MariaDB
sudo mysql_secure_installation

# Installation de PHP et modules nécessaires
sudo apt install php libapache2-mod-php php-mysql -y

# Redémarrage d'Apache pour prise en compte de PHP
sudo systemctl restart apache2

# Test PHP (phpinfo)
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

# Installation de phpMyAdmin (interactif : répondre "apache2", puis "oui" à la configuration DB)
sudo apt install phpmyadmin -y

# Lier phpMyAdmin à Apache si ce n'est pas automatique
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin

# Redémarrage final d'Apache
sudo systemctl restart apache2
```

- Création de la base de données et des différentes tables (Etable, vaches...).

- Développement d'un script de sauvegarde automatique des données.

```
etable@raspberrypi:~ $ cd /home/pi/db_backups/
etable@raspberrypi:/home/pi/db_backups $ ls
backup_mariadb.sh  etable.sql
etable@raspberrypi:/home/pi/db_backups $ nano backup_mariadb.sh
etable@raspberrypi:/home/pi/db_backups $ crontab -e
No modification made
etable@raspberrypi:/home/pi/db_backups $ ls -l /home/pi/db_backups/etable.sql
-rw-r--r-- 1 etable etable 1292  9 janv. 17:28 /home/pi/db_backups/etable.sql
etable@raspberrypi:/home/pi/db_backups $ ls
backup_mariadb.sh  etable.sql
etable@raspberrypi:/home/pi/db_backups $ ./backup_mariadb.sh
mysqldump: Got error: 1049: "Unknown database 'etable'" when selecting the database
etable@raspberrypi:/home/pi/db_backups $ nano backup_mariadb.sh
etable@raspberrypi:/home/pi/db_backups $ nano backup_mariadb.sh
etable@raspberrypi:/home/pi/db_backups $ ./backup_mariadb.sh
etable@raspberrypi:/home/pi/db_backups $ ls -l /home/pi/db_backups/etable.sql
-rw-r--r-- 1 etable etable 5855  6 mai 16:17 /home/pi/db_backups/etable.sql
etable@raspberrypi:/home/pi/db_backups $
```

- Mise en place d'un script de transmission pour intégrer les données reçues via MQTT.

### 3. Site Web (IHM)

- Création de la structure du site dans le répertoire www/.
- Mise en place des fichiers HTML/CSS nécessaires à l'interface.
- Développement de la page de gestion accessible uniquement après authentification.
- Intégration à la base de données pour affichage dynamique.
- Implémentation d'un formulaire permettant l'ajout de vaches avec caractéristiques (nom, RFID, etc.).

### 4. Conception de la maquette

- Croquis du design initial de la maquette.
- Recherche et sélection du matériels nécessaires.
- Conception initial de la maquette.
- Révision du croquis en fonction des contraintes techniques.
- Conception final.



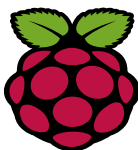
# Outils Utilisés

## Linux



Système d'exploitation libre, Linux a servi de socle à l'ensemble de l'infrastructure du projet. Installé sur le Raspberry Pi 4, il a permis l'exécution fluide des différents services du serveur LAMP (Apache, MySQL, PHP) ainsi que des scripts nécessaires au fonctionnement du système.

## Raspberry Pi 4



Le Raspberry Pi 4 est le cœur physique du projet : un micro-ordinateur compact, choisi pour héberger le serveur web, gérer la base de données, et exécuter les scripts de communication (comme ceux liés au MQTT et au lecteur RFID). Il a permis de centraliser toutes les fonctions du système sur un seul point de contrôle.

## phpMyAdmin



phpMyAdmin est un outil web permettant de gérer visuellement les bases de données MySQL. Il a été utilisé pour créer, modifier et surveiller les tables du projet, facilitant le développement, la vérification et la maintenance de la structure de données.

## Draw.io



Outil de modélisation en ligne, Draw.io a été exclusivement utilisé pour la création des diagrammes UML du projet. Ces schémas, tels que les diagrammes de cas d'utilisation, diagrammes de classes et diagrammes de séquence, ont permis de représenter visuellement la structure logique et les interactions du système, facilitant ainsi la planification et la documentation.

## Chromium



Navigateur web open source, Chromium a servi à tester l'affichage et le comportement du site développé. Il a également été utilisé pour explorer le code côté client à l'aide de la fonction "Inspector", me permettant de diagnostiquer les erreurs, et d'optimiser l'affichage.

## Arduino IDE



L'Arduino IDE a été utilisé pour développer, tester et déployer le code nécessaire à la communication avec certains composants matériels, notamment pour les premiers tests du lecteur RFID et autres interactions physiques possibles avec le système.

## ChatGPT



ChatGPT, a été un assistant précieux tout au long du projet, offrant des réponses techniques, des suggestions de code, des explications et une aide à la rédaction, accélérant le développement et renforçant la compréhension des concepts complexes.



## GitHub

GitHub a permis la gestion du code source du projet, en facilitant la sauvegarde, le suivi des modifications et la collaboration entre membres de l'équipe. Cet outil a aussi servi à documenter le projet dans un environnement partagé et accessible.



## MQTT

Protocole de messagerie léger, MQTT a été au centre de la transmission en temps réel des données provenant des capteurs. Il a permis aux données de production laitière et de température d'être envoyées instantanément au serveur via des topics spécifiques.



## Python

Python a été utilisé pour développer les scripts de communication avec le MQTT, la lecture RFID, ainsi que pour l'insertion automatisée des données dans la base. Sa simplicité et ses nombreuses bibliothèques ont permis un développement rapide et efficace.

# Conclusion

Ce projet a été une belle étape, mais il peut encore être amélioré. Par exemple, on pourrait créer une application mobile pour faciliter l'accès au système. Il serait aussi intéressant d'ajouter de nouvelles pages sur le site, comme une boutique en ligne, un espace pour créer des comptes utilisateurs, et des formulaires pour mieux interagir avec les visiteurs.

Je remercie aussi mon équipe pour leur aide tout au long du projet. Grâce à eux, j'ai pu avancer et aussi mieux comprendre mes difficultés, surtout avec l'Arduino. Leur soutien m'a beaucoup aidé à progresser.