



## Dossier commun du projet

## Sommaire

I.	Cahier des charges.....	3
1.1)	Récapitulatif du cahier des charges.....	3
1.2)	Expression du besoin.....	5
1.3)	Contraintes imposées par le client.....	6
1.4)	Matériel fourni par le client .....	7
1.5)	Logiciels fournis par le client .....	7
1.6)	Synoptique de la réalisation .....	8
1.7)	Répartition des tâches.....	8
II.	Spécifications techniques.....	9
2.1)	Étudiant 1 : Spécification des blocs Raspberry/Anémomètre .....	9
2.2)	Étudiant 2 : Spécification des blocs Web/Solarimètre.....	11
2.3)	Étudiant 3 : Spécification des blocs Base de données/Sonde température .....	13
2.4)	Étudiant 4 : Spécification des blocs Android/Pluviomètre.....	17
III.	Analyse .....	19
3.1)	Étudiant 1 : Analyse des blocs Raspberry/Anémomètre .....	22
3.2)	Étudiant 2 : Analyse des blocs Web/Solarimètre .....	24
3.3)	Étudiant 3 : Analyse des blocs Base de données/Sonde température .....	26
3.4)	Étudiant 4 : Analyse des blocs Android/Pluviomètre.....	28
IV.	Intégration et versionning.....	30
V.	Communication de groupe.....	<b>Erreur ! Signet non défini.</b>
VI.	Conclusion .....	31
VII.	Annexes .....	32
7.1)	Diagramme de Gantt.....	32
7.2)	Procédure d'installation .....	<b>Erreur ! Signet non défini.</b>
VIII.	Glossaire .....	<b>Erreur ! Signet non défini.</b>

## I. Cahier des charges

### 1.1) Récapitulatif du cahier des charges

#### Les bienfaits d'une serre :

Une serre, même petite, ouvre de vastes possibilités de culture, des primeurs aux orchidées. Son équipement peut comprendre des dispositifs automatiques pour assurer les opérations les plus astreignantes : **chauffage, arrosage, aération**.

Une serre libère le jardinier des caprices des intempéries. A l'abri du verre, il peut cultiver des plantes qui, dehors, souffriraient du froid, des pluies et des vents violents.

Les serres peuvent être chauffées ou non :

- Les serres froides

Une serre froide, non chauffée, ne protège pas du gel, mais elle entoure les plantes d'une atmosphère plus tiède en période de végétation, ce qui hâte le mûrissement des fruits et améliore la floraison de nombreuses espèces décoratives. Elle protège les plantes des intempéries, d'une humidité excessive en hiver et des attaques des oiseaux, des insectes et autres animaux nuisibles.

Mais sa fonction la plus précieuse est d'allonger la période de végétation. On peut y mettre les plantes en culture en début de printemps et soit les conserver en serre, soit les planter dans le jardin. En serre, les plantes continuent de prospérer durant l'automne.

Cette prolongation artificielle de la période de végétation est causée, au printemps, par la chaleur du soleil captée par le verre et à l'automne, par la conservation de la chaleur accumulée pendant l'été dans le sol.

- Les serres chaudes

Dans une serre chauffée artificiellement, la gamme des végétaux cultivables augmente de façon considérable, car les plantes molles, qui ne résistent pas aux hivers rigoureux, s'y conservent admirablement.

Des dispositifs automatiques de chauffage, de ventilation, d'ombrage et arrosage rendent presque totale la maîtrise des conditions climatiques intérieures. De plus, toutes les serres sont équipées d'aérateurs qui permettent le renouvellement de l'atmosphère de la serre. Il existe également des ventilateurs électriques. Enfin, par les jours très chauds d'été, l'aération ne suffit pas. Pour prévenir une montée de la température préjudiciable aux plantes, l'ombrage sera assuré par des stores intérieurs que l'on baisse lorsque le soleil est trop fort et que l'on remonte lorsqu'il faiblit. Il existe des stores automatiques extérieurs, mais ils sont coûteux.

Une serre demande donc une **température**, une **hydrométrie**, une **intensité lumineuse** une anémométrie contrôlée.

### Le commanditaire du projet :

Le Groupe Olivier est spécialisé dans la production de tomates et de concombres sous serres verre chauffées. Il regroupe 2 sociétés de production de légumes, la SCA Olivier Frères et la SCA des Courtines.

L'exploitation d'une surface de 18,5 ha de serres est répartie sur 2 sites de production dans la région nantaise à Haute-Goulaine et Saint-Julien-de-Concelles.

Dirigé par Henri, Jean-Luc et Stéphane Olivier, le Groupe emploie environ 170 équivalents temps plein. Les productions de légumes en culture hors sol sont commercialisées par la coopérative OCEANE, créée en 1993 par 9 Maraîchers Nantais, dont la famille Olivier.

Pour plus d'informations sur l'activité du Groupe Olivier : <http://www.groupe-olivier.fr/>

### Aperçu du projet :

La globalité de ce projet aura pour objet la gestion automatique d'une serre maraîchère et se décomposera en deux parties :

1. La **supervision** de l'état de la serre avec récupération et stockage de l'ensemble des données nécessaires ;
2. L'**automatisation** de la régulation de la température, l'hydrométrie et de l'intensité lumineuse de la serre.

Il a été décidé que ce projet sera porté sur deux années. Le projet décrit par la suite se limite donc à la première partie : **La supervision de l'état de la serre.**

Ce projet aura donc pour but de superviser l'état d'une serre :

- L'acquisition et le stockage de l'ensemble des données suivantes :

- L'intensité lumineuse
- La pluviométrie
- La force et la direction du vent
- La température intérieure de la serre et de l'eau des tuyaux de chauffage

- La mise en place de systèmes pour visualiser les différentes données acquises :

- Un système (Raspberry) récupèrera automatiquement les différentes mesures énoncées ci-dessus.
- Un second système (BDD) va permettre la mémorisation de ces mesures sur plusieurs années.
- Possibilité à l'aide d'une application Web de consulter les mesures par des histogrammes et de visualiser en temps réel.
- Information en temps réel de l'état de fonctionnement de l'ensemble du système grâce à une application Android.

### **Finalité du projet :**

Le client souhaite :

- un système situé dans la serre permettant la récupération automatique des différentes mesures nécessaires au projet global ;
- un système situé dans un local proche de la serre permettant la mémorisation des mesures sur une période de plusieurs années ;
- une application web permettant la consultation des mesures par des histogrammes ;
- une application web permettant la visualisation en temps réel des différentes mesures effectuée dans la serre ;
- une application Android permettant d'être informé en temps réel de l'état de fonctionnement de l'ensemble du système.

### **Mesures à effectuer**

Le système devra enregistrer six mesures distinctes.

1. L'intensité lumineuse devra être mesurée par le solarimètre SDE-SOLAR-10000W de la société ARIA. Celui-ci dispose d'une plage allant de 0 à 1000 W/M2. Ce capteur fonctionne en boucle de courant de 4-20 mA.
2. La température intérieure sous serre devra être mesurée par un capteur SDTS ARIA de plage 0 à 45°C. Ce capteur fonctionne également en boucle de courant de 4-20 mA.
3. La température d'eau chaude des tuyaux de chauffage devra être mesurée par un capteur SDTS ARIA de plage 0 à 100°C. Ce capteur fonctionne également en boucle de courant de 4-20mA.
4. L'hydrométrie sera mesurée par le pluviomètre Rain Collector II de la société DAVIS. Ce capteur fonctionne par impulsion.
5. La direction et la force du vent seront mesurée par l'anémomètre-girouette 7911 de la société DAVIS.

### 1.2) Expression du besoin

L'exploitant de la serre est l'acteur principal du système.

L'exploitant doit pouvoir visualiser en temps réel l'état de la serre.

L'exploitant doit pouvoir consulter facilement l'évolution de chaque mesure, en choisissant la période de visualisation (jour, semaine, mois ou année).

L'exploitant doit pouvoir être informé en temps réel sur son smartphone de l'état de fonctionnement du système.

### 1.3) Contraintes imposées par le client

#### **Contraintes financières (budget alloué) :**

Le budget alloué est de 100 euros permettant l'achat éventuel de carte(s) Adaptateur 4-20 mA. Le reste du matériel nécessaire au projet est disponible dans la section.

#### **Contraintes de développement (matériel et/ou logiciel imposé / technologies utilisées) :**

- Carte de gestion : développement sur carte Raspberry en langage python
- Développement de l'application Android avec Android Studio
- Développement de l'application Web sous le patron Modèle-Vue-Contrôleur (éventuellement sous le framework Symfony)

#### **Contraintes qualité :**

- La partie logicielle devra être intégralement testée avant sa présentation au commanditaire en fin de projet. Le client final devra pouvoir utiliser le système sans compétence informatique particulière (hormis éventuellement pour certains paramétrages de configuration).
- Le projet devra être réalisé afin de pouvoir ajouter facilement de nouvelles mesures par la suite. Par exemple, si le client désire ajouter une mesure pH du sol, le travail à réaliser devra être minimal, voire automatique si possible : la mise à jour de l'interface web de visualisation devra être aisée, la visualisation de l'évolution pourra être mise à jour automatiquement.  
**Ceci est une contrainte d'évolutivité forte impliquant une analyse poussée et un travail de développement plus exigeant. Cette contrainte sera notée (\*ContE) et précisera, dans le tableau de répartition des tâches, les tâches pour lesquels cette contrainte devra être prise en compte.**
- Une documentation complète sur le système proposé devra être fournie au client. Elle doit comprendre : un dossier de présentation du système, un mode d'emploi de l'interface, une procédure illustrée d'installation... Les étudiants devront aussi fournir un exemplaire des sources de leur travail, ainsi qu'une nomenclature précise du matériel utilisé pour permettre aux bénéficiaires de donner une suite au projet, par une autre voie que l'équipe d'étudiants.
- Une documentation automatique du code devra être fournie au format html.

#### **Contraintes de fiabilité, sécurité :**

- La carte de gestion devra journaliser en interne tout problème d'acquisition de mesures ou de problème d'accès à la base de données.

#### 1.4) Matériel à disposition des étudiants

Ressources matérielles mises à disposition des étudiants :

- Solarimètre SDE-SOLAR-10000W
- Deux capteurs SDTS ARIA (0 à 45°C et 0 à 100°C)
- Un pluviomètre DAVIS
- Un anémomètre/girouette DAVIS
- Carte de mesure : module boucle 4-20 mA...
- Carte de gestion : carte raspberry Pi 3, arduino...
- 4 PC Windows 7

*Remarque : le smartphone android n'est pas fourni, ce sera donc celui de l'étudiant qui sera utilisé.  
Une tablette pourra être prêtée si besoin.*

#### 1.5) Logiciels à disposition

##### **Ressources logicielles pour le développement**

- Environnement de développement pour Python 2.7
- Environnement de développement Android Studio
- Environnement MySQL pour la base de données
- Environnement Netbeans pour l'application Web, éventuellement le framework Symfony

##### **Autres ressources logicielles disponibles durant le projet**

- Suites bureautiques Microsoft Office 2007 et LibreOffice 5
- MagicDraw 17.0.3 avec plugin SysML
- Microsoft Project
- Gestionnaire de version Git

### 1.6) Synoptique – Fonctions du système et échange d'informations

Le principe de fonctionnement du système sont résumés dans le synoptique de la page suivante.

Dans la serre, un système effectue une prise des mesures à intervalle régulier. Ces informations sont enregistrées dans une base de données distante. Le type de liaison sera à déterminer par les étudiants et devra correspondre aux spécificités du site.

Dans le local proche de la serre, se trouvent le PC d'archivage. La visualisation des informations récoltées est effectuée sur le PC à l'aide d'une application web qui permet d'afficher les courbes d'évolution sur une période définie, pouvant aller jusqu'à un an. Il est également possible de visualiser l'état actuel de la serre depuis l'application web.

L'utilisateur dispose aussi d'un moyen de contrôle du système, une application Android. Celle-ci lui permet de visualiser le bon fonctionnement de chaque entité du système.



### 1.7) Répartition des tâches

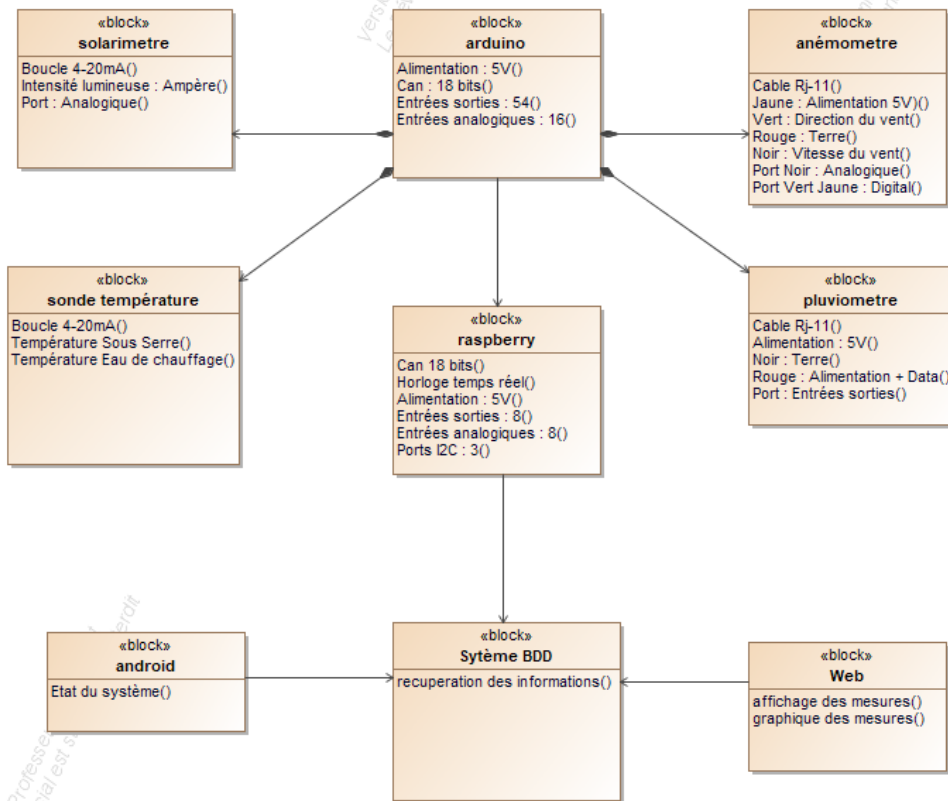
Pour la réalisation du projet, les tâches ont été réparties de la façon suivante :

- **Etudiant 1 : Steven NGUENE**, partie Raspberry/Anémomètre
- **Etudiant 2 : Willy RINEAU**, partie Web/Réseau/Solarimètre
- **Etudiant 3 : Samuel GERARD**, partie BDD/Web/Sonde température
- **Etudiant 4 : Dylan CHESNOUARD**, partie Android/Pluviomètre



## II. Spécifications techniques

### 2.1) Étudiant 1 : Spécification des blocs Raspberry/Anémomètre



Le bloc Raspberry permettra la connexion à la base de données du système, la récupération et l'entrée des mesures reçues par tous les capteurs. Les capteurs (Anémomètre et Pluviomètre) seront connectés sur une carte Arduino et elle sera connectée à la carte Raspberry en USB.



## Dossier technique du projet – Partie commune

L'utilisateur pourra ainsi récupérer les informations de la carte Arduino via la carte Raspberry. Un programme python sera ensuite exécuter sur la Raspberry qui enverra automatiquement les mesures récupérer par la Raspberry sur la base de données.

Ces mesures seront donc envoyées selon une période que l'utilisateur aura définie auparavant.

Cette partie fonctionnera automatiquement et récupèrera continuellement les informations.

### 2.1.1) Spécification du bloc Anémomètre

Pour l'Anémomètre la récupération des mesures se fera en deux parties, la vitesse et la direction du vent.

Pour la vitesse le capteur sera connecté sur le port analogique de l'Arduino. Pour récupérer les mesures, il faut calculer le nombre de pulsations suivant une période qui sera converti en mph puis en kmh. (1mph = 1.61kmh).

Pour la direction le capteur sera connecté sur le port digital de l'Arduino. Pour récupérer les mesures le capteur envoie une valeur entre 0 et 1023 qui sera converti par le Can (Convertisseur Analogique Numérique) de 10 bits de 0 à 360°.

Mesures :

La vitesse du vent (1 à 322km/h) avec une précision de 3km/h

La direction du vent (0° à 360°) avec une précision de 7°

Le parcours du vent (0 à 1999.9km/h) avec une précision de 5%

Ces mesures seront ensuite traitées puis envoyées à la Raspberry et enfin la base de données qui les utilisera pour le site web.

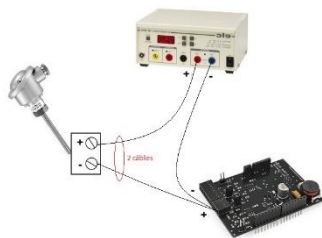
### 2.2) Étudiant 2 : Spécification des blocs Web/Solarimètre

Objectif : Récupérer les données concernant l'intensité lumineuse

Ce système permettra d'acquérir l'intensité lumineuse, afin de pouvoir superviser la serre et dans un second temps, d'activer des stores intérieurs, ce qui permettra aux plantes de pousser dans de meilleures conditions. De plus, l'exploitant aura un rendu visuel sur les valeurs données par le solarimètre.

Pour le système, nous étai imposé la boucle de courant 4-20 Ma. Nous nous sommes ensuite dit qu'il était préférable de diviser les capteurs en deux parties, ceux qui appartiennent à la boucle de courant, et ceux qui n'en ont pas besoin. Ainsi, la boucle de courant sera reliée à une carte Arduino qui traitera les données et les enverras sur la carte Raspberry qui est la carte de gestion.

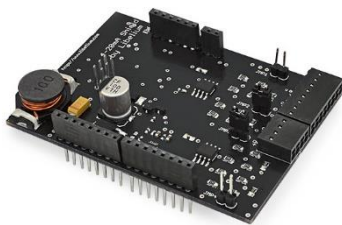
#### 2.2.1) Boucle 4-20 Ma



*Schéma de câblage*

La boucle 4-20 Ma est un standard dans l'industrie, elle permet de transmettre un signal analogique sur une grande distance, sans perte ou modification du signal. Ainsi, pour mettre en place cette boucle de courant, nous devons avoir le capteur, une alimentation externe, un récepteur et un émetteur.

#### 2.2.2) Choix de l'adaptateur



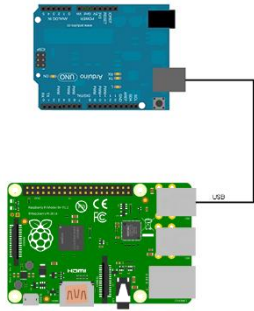
*Adaptateur 4-20 mA*

Nous avons le choix entre deux adaptateurs, avec la contrainte de prix, nous ne pouvons éliminer la seconde, car elle ne gèrait que 2 canaux, alors que nous avons 3 capteurs dans la boucle 4-20 Ma. Ainsi, nous avons choisi la carte Adaptateur 4-20Ma Current Loop Sensor Board.

Celle-ci possède 4 canaux, pour un prix de 78.00€.

## Dossier technique du projet – Partie commune

### 2.2.3) Arduino à la Raspberry



Connexion USB

La connexion à la Raspberry se fait pas USB, une autre solution était possible, il s'agissait du TX-RX, qui est plus rapide que la connexion USB, car elle ne possède pas d'encapsulation. Or, pour notre projet, l'envoi des données ne se fera pas toutes les minutes, donc un traitement plus long ne nous gênera pas.

### 2.2.4) Spécification du site Web

Objectif : Visualiser l'évolution de chaque mesure en choisissant la période

### 2.3) Étudiant 3 : Spécification des blocs Base de données/Sonde température

#### **Bloc Base de données :**

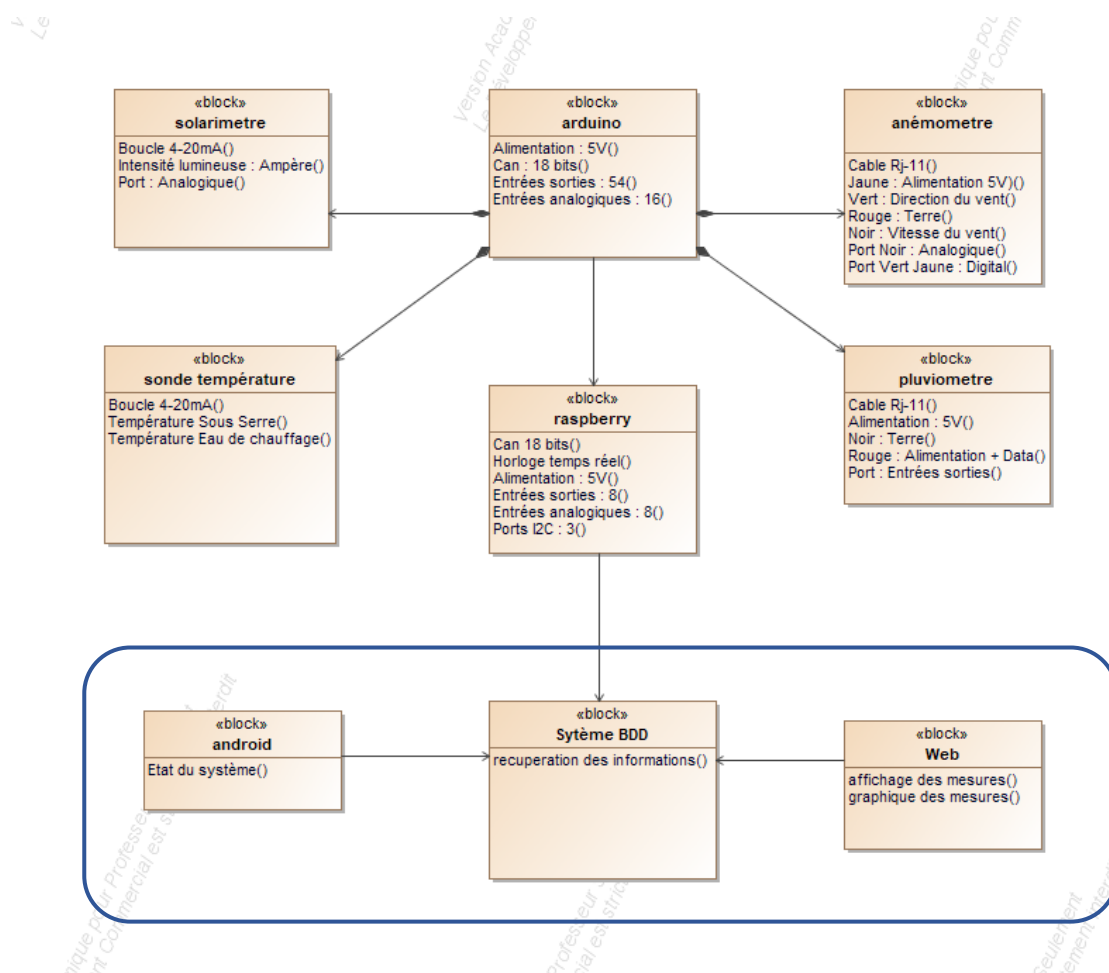
**Objectif :** Stocker les différentes mesures et informations

Le bloc Base de données permettra le stockage des mesures acquises ainsi que des informations requises pour le site web et les applications.

L'exploitant pourra ainsi :

- Visualiser les mesures et informations à partir de l'interface de la base de données.
- Visualiser ces mêmes mesures et informations à partir du site web.
- Visualiser l'état de fonctionnement des différents capteurs à partir de l'application Android.

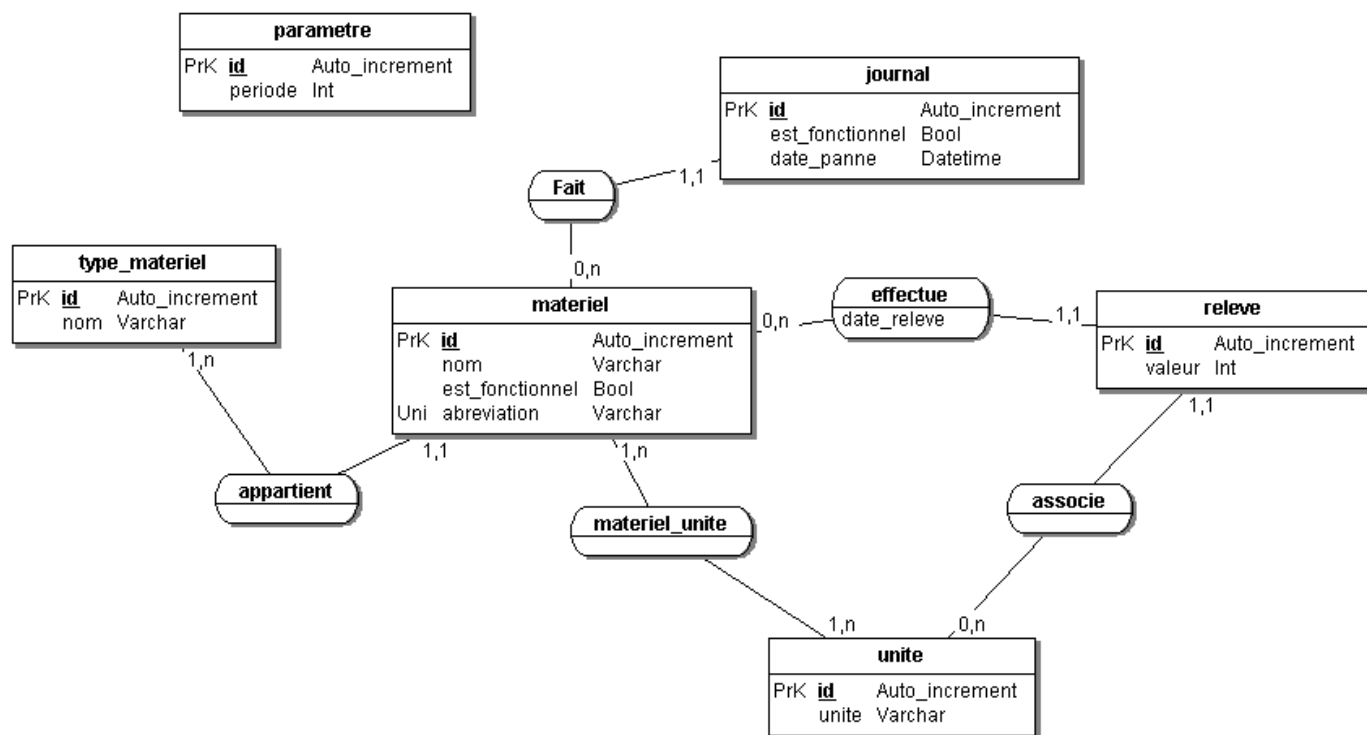
Le bloc base de données est donc essentiel au bon fonctionnement de l'ensemble des blocs Web et Android.



## Dossier technique du projet – Partie commune

La base de données est générée grâce à un schéma entité-relation fait sur JMerise. Avant la base de données finale, il y a eu 4 versions de celle-ci. N'ayant pas terminé le projet, la base de données pourra connaître des changements ultérieurement à ce dossier.

JMerise actuel de la base de données :



**Bloc Sonde température :**

Objectif : Mesurer la température sous serre et la température des tuyaux de chauffage

Ce bloc permettra d'acquérir les températures sous serre et des tuyaux de chauffage afin de pouvoir les stocker dans la base de données.

L'exploitant pourra ainsi surveiller les températures.

Ce bloc est composé de deux sondes de températures pt100 classe A :

Sonde température Pt100 pour l'eau des tuyaux de chauffage :

Capteur : Pt100 classe A  
Plage : 0-100°C  
Précision : +/- 0,3°C  
Convertisseur : intégré 4-20mA 2 fils  
Alimentation : 20 à 30 volts régulés  
Charge typique 500 Ohm  
Connexion par bornes à visser



Sonde température Pt100 pour l'air dans la serre :

Capteur : Pt100 classe A  
Plage : 0-45°C  
Précision : +/- 0,3°C  
Convertisseur : intégré 4-20mA 2 fils  
Alimentation : 20 à 30 volts régulés  
Charge typique 500 Ohm  
Connexion par bornes à visser



Afin de transmettre les mesures à la base de données, les deux sondes sont branchées à une carte Arduino elle-même reliée à une carte Raspberry.

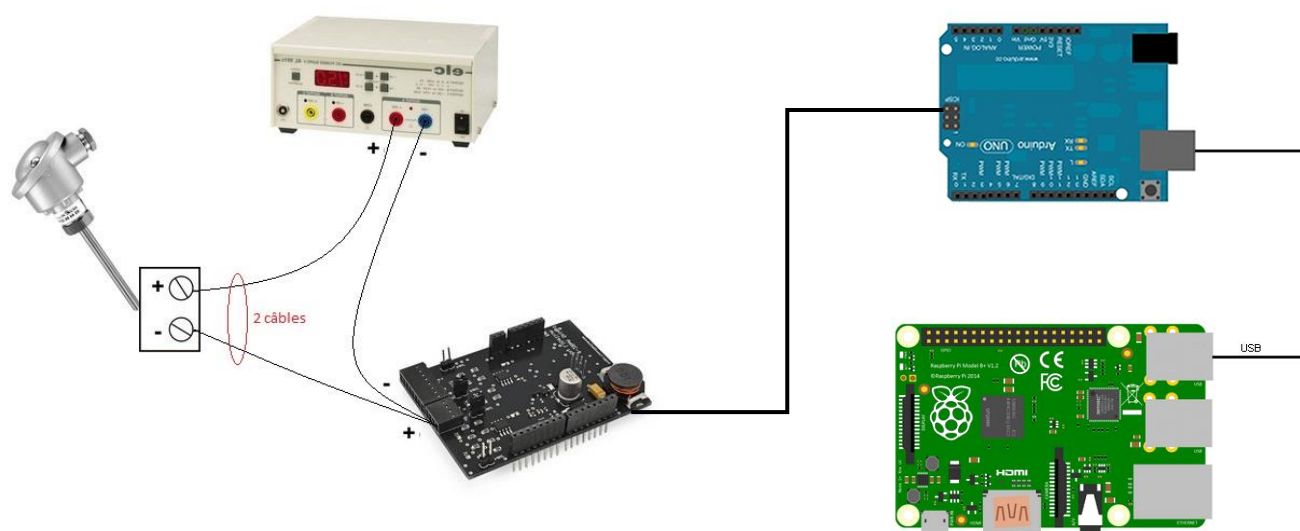
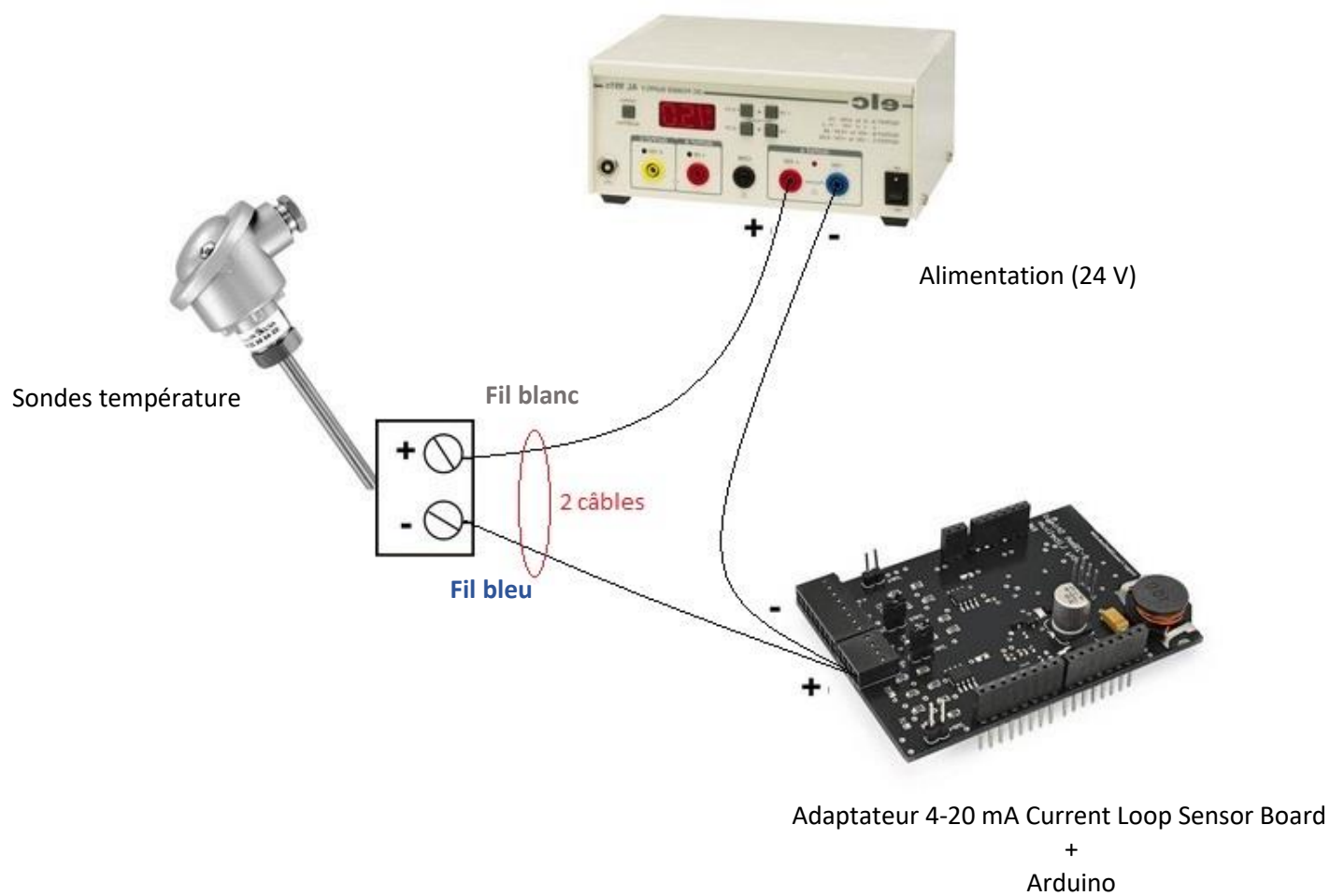
Pour transmettre les données à l'Arduino, une boucle 4-20 ma est mise en place à l'aide d'un adaptateur, l'adaptateur 4-20 mA Current Loop Sensor Board.



**Adaptateur 4-20 mA Current Loop Sensor Board**

Nombre de canaux : 4  
Livré avec câbles en paire torsadée.  
Température de service : 0°C à +85°C  
Dimensions : 73 ;5 x 51 x 13 mm.  
Module prêt à l'emploi.  
Prix : 78€00

Schémas de câblage :





## 2.4) Étudiant 4 : Spécification des blocs Android/Pluviomètre

### 2.4.1) Pluviomètre DAVIS

**Objectif :** Mesurer les précipitations.

Le pluviomètre aura pour objectif de mesurer la quantité de précipitations tombée durant une certaine période.

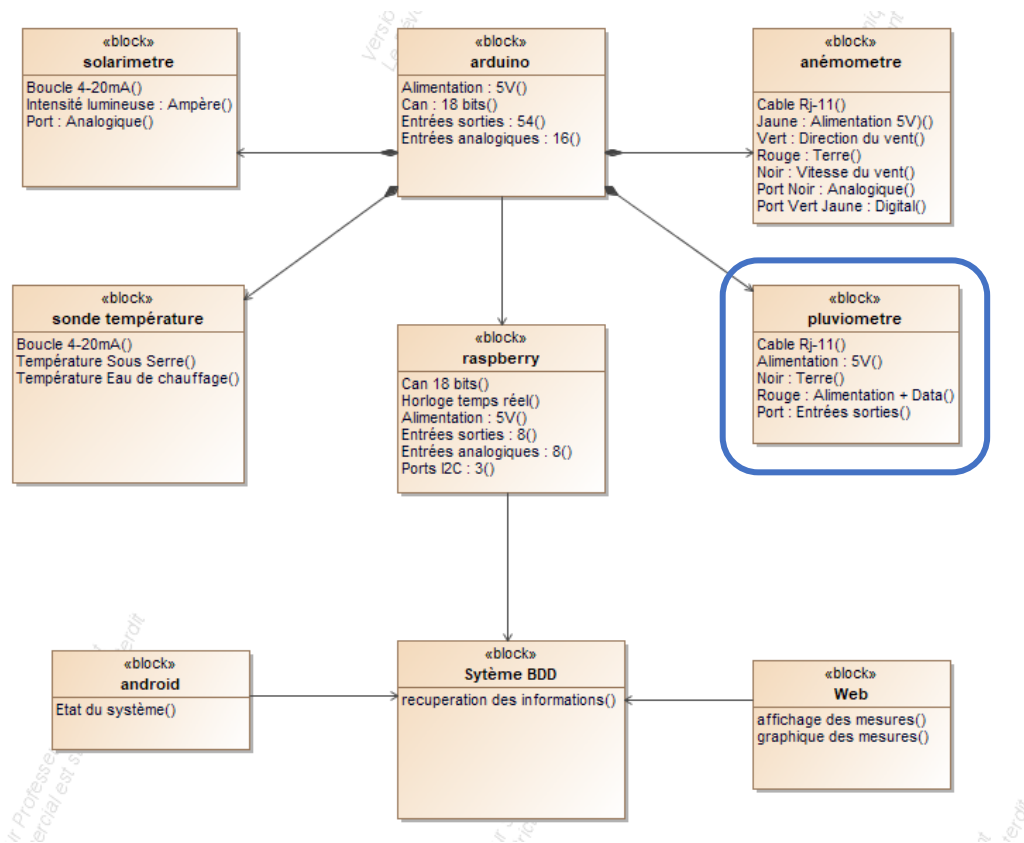


Figure 1 : Diagramme de blocs

Le pluviomètre Davis enverra une impulsion (état logique bas) à la carte Arduino pour chaque basculement de l'auget basculeur :

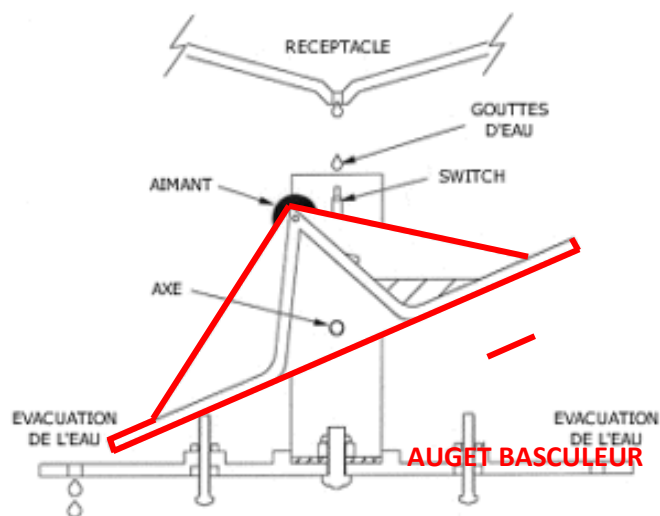


Figure 2 : Schéma illustrant le fonctionnement du pluviomètre DAVIS

## Dossier technique du projet – Partie commune

Lorsque la carte Arduino recevra une impulsion provenant du pluviomètre, elle enverra un message (une trame) à la carte Raspberry via USB. Lors de la réception d'une trame correspondant au basculement du pluviomètre, la carte Raspberry incrémentera une variable qui permettra de suivre le nombre de basculement durant chaque période de mesure. La période de mesure sera une variable définie par l'utilisateur, et stocker dans la base de données.

### 2.4.2) Application Android

**Objectif :** Afficher les états de fonctionnements des différents matériels.

L'application Android aura pour but de permettre à l'utilisateur de visualiser les états de fonctionnements de chaque matériels (capteurs, microcontrôleurs, base de données) faisant partis du système de supervision de serre.

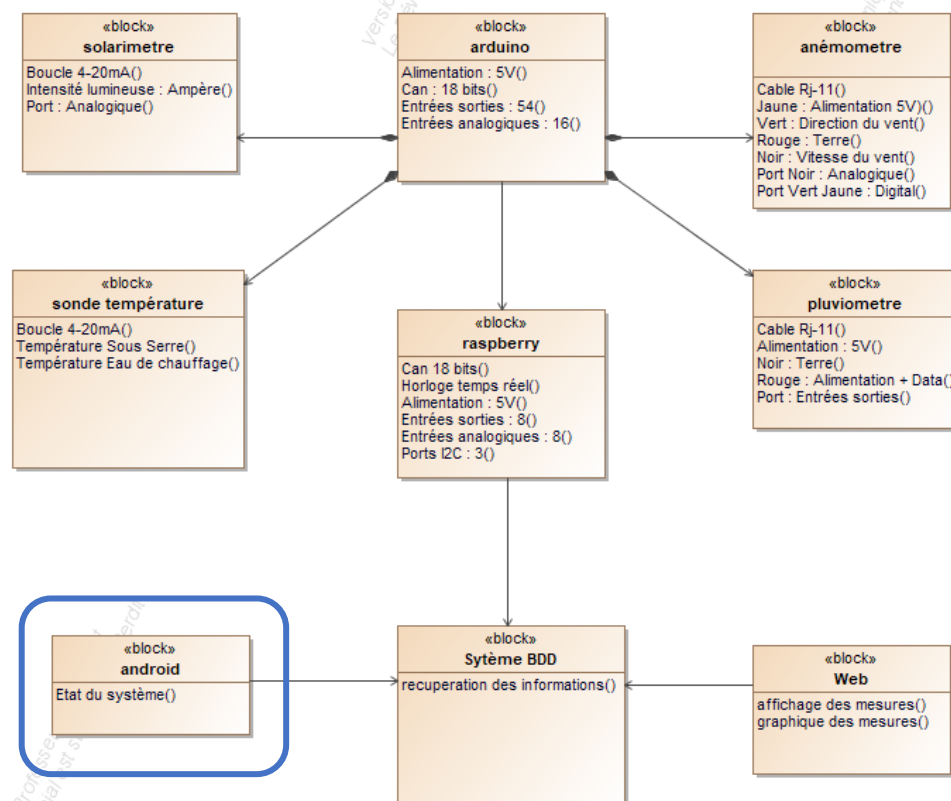


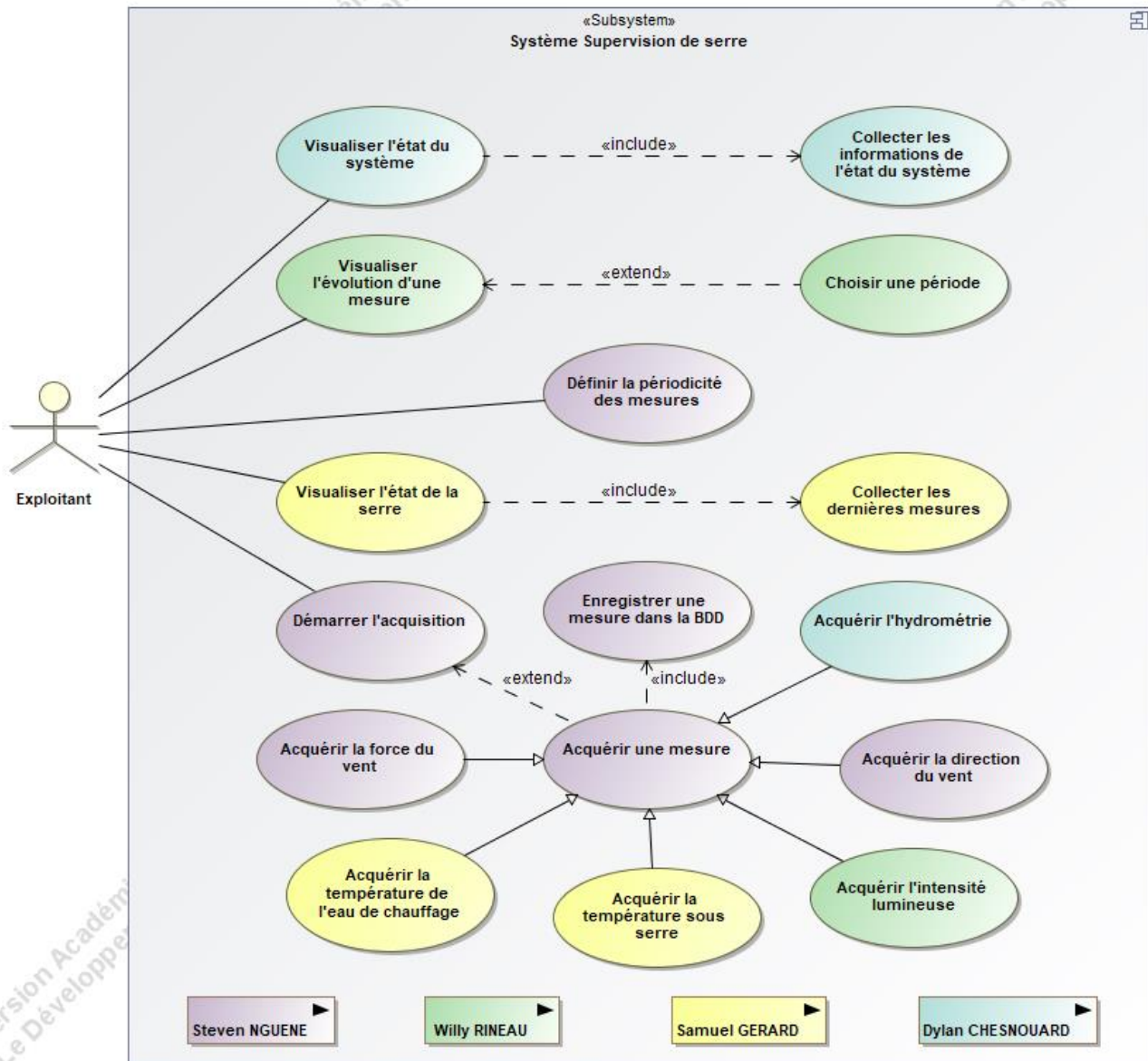
Figure 3 : Diagramme de blocs

Au lancement de l'application, une tentative de connexion avec la base de données est effectuée. Si la connexion fonctionne, l'utilisateur aura alors accès à la liste des capteurs, des microcontrôleurs, ainsi que l'historique de fonctionnement (les logs) de chaque matériel.

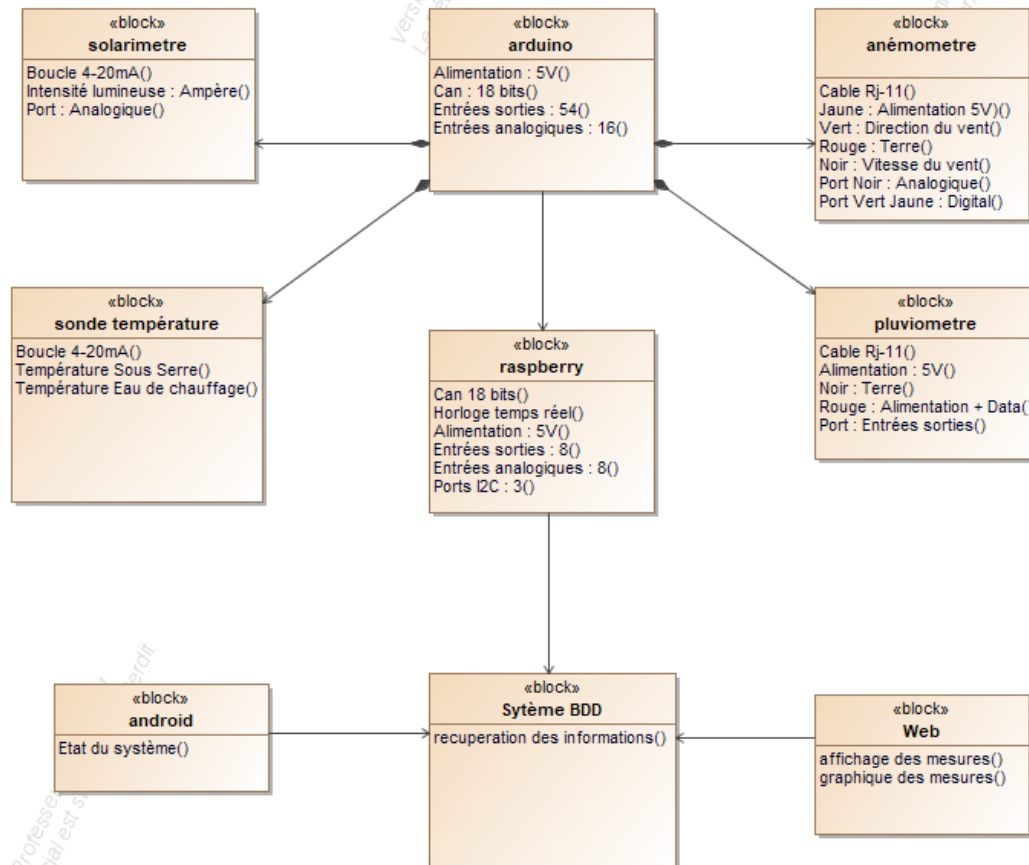
### III. Analyse

Pour l'analyse, nous nous intéresserons dans un premier temps aux différents cas d'utilisations de chaque bloc à créer. Voici un diagramme d'utilisation imposé par le client dans le cahier des charges. Etant très complet, nous nous en sommes inspirés pour s'en servir de base.

#### Diagramme d'utilisation :



**Diagramme de définition de blocs :**

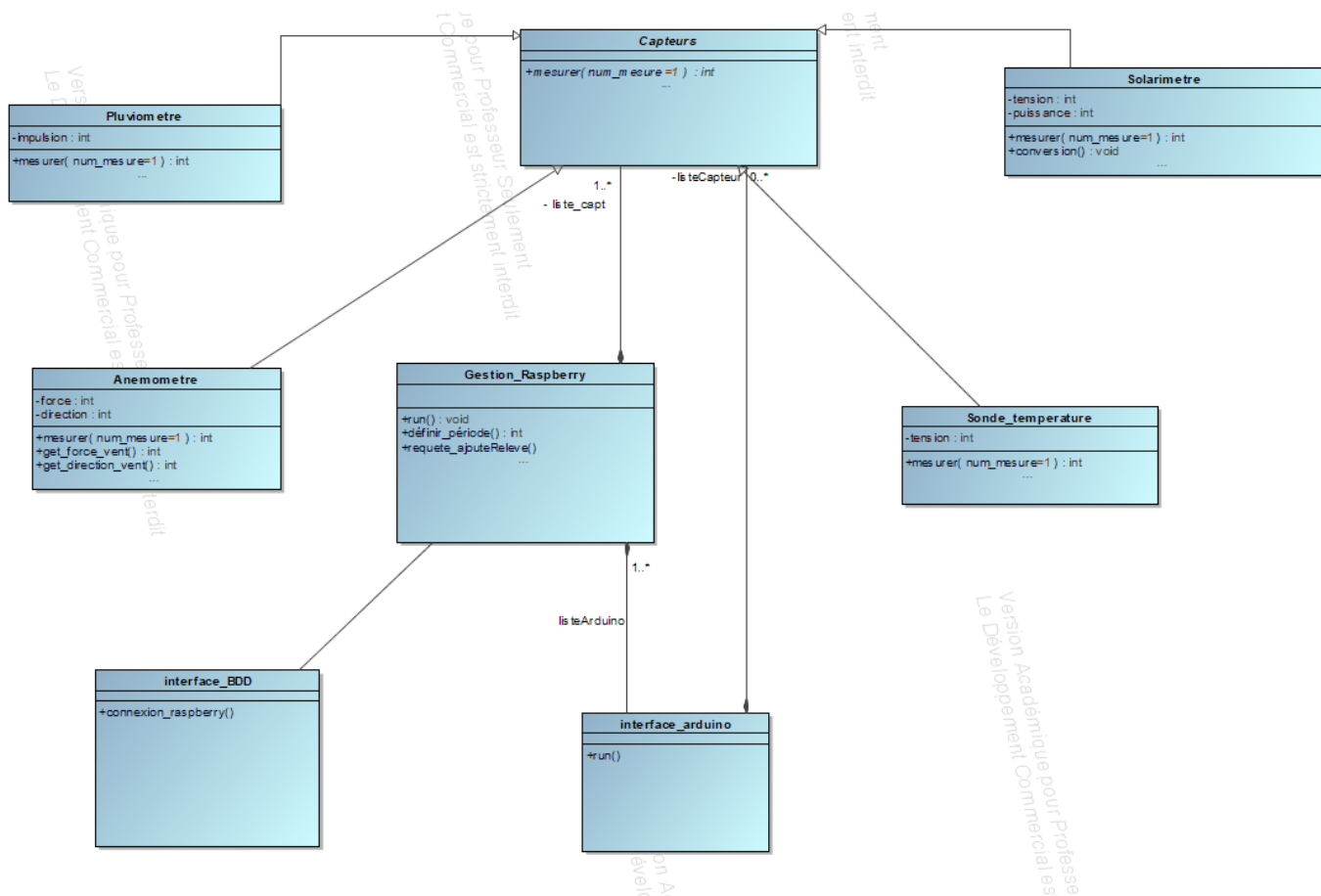


Ce diagramme permet de visualiser l'architecture du système et représente les éléments matériels.

Les liens sont également représentés par :

- Des compositions entre l'Arduino et les capteurs ce qui signifie que **l'Arduino** contient tous les **capteurs** et ceux-ci seront détruits avec elle.
- Des associations qui montrent une connexion entre les différents systèmes.

**Diagramme de classe :**



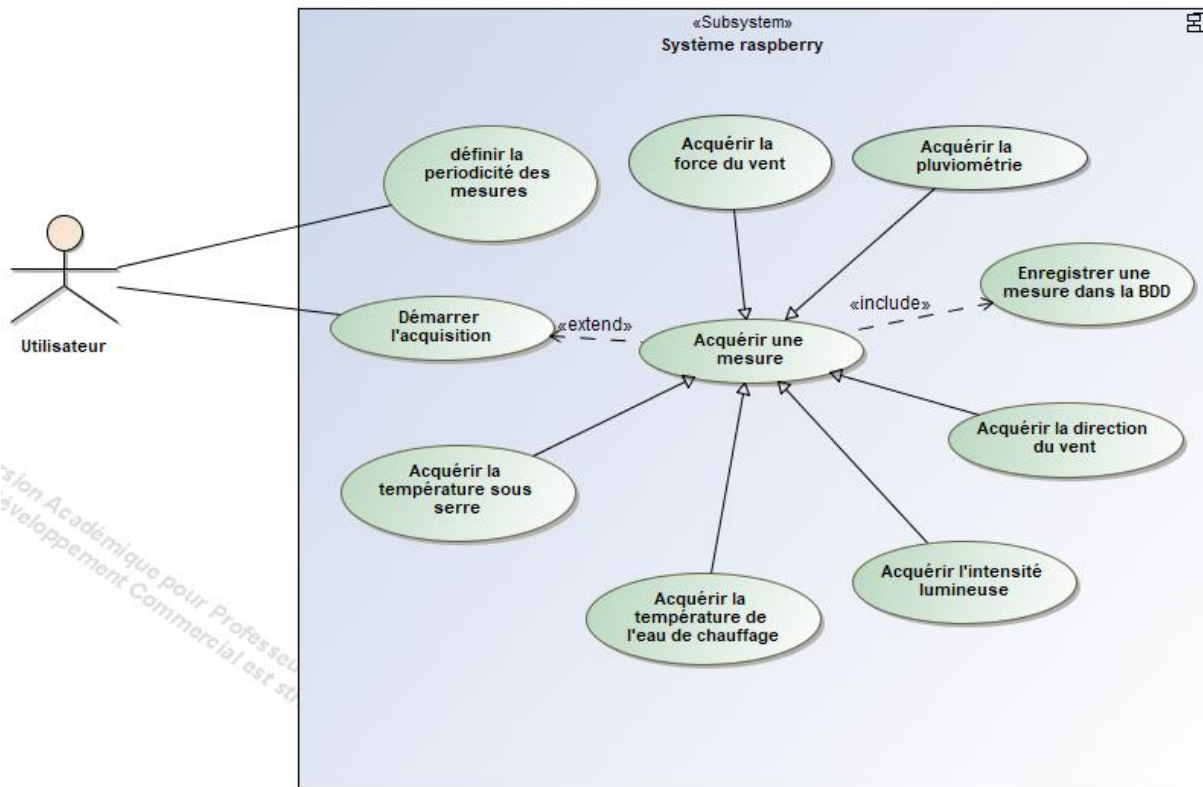
Ce diagramme de classe nous présente les différents liens entre les classes.

Les capteurs héritent de la classe **Capteurs** où ils récupèrent la méthode virtuelle « *mesurer ()* ».

Les capteurs étant connectés directement à l'Arduino, il y a donc une composition, de même entre l'Arduino et la Raspberry. Le lien entre la Raspberry et l'interface BDD montre la connexion distante entre ces derniers.

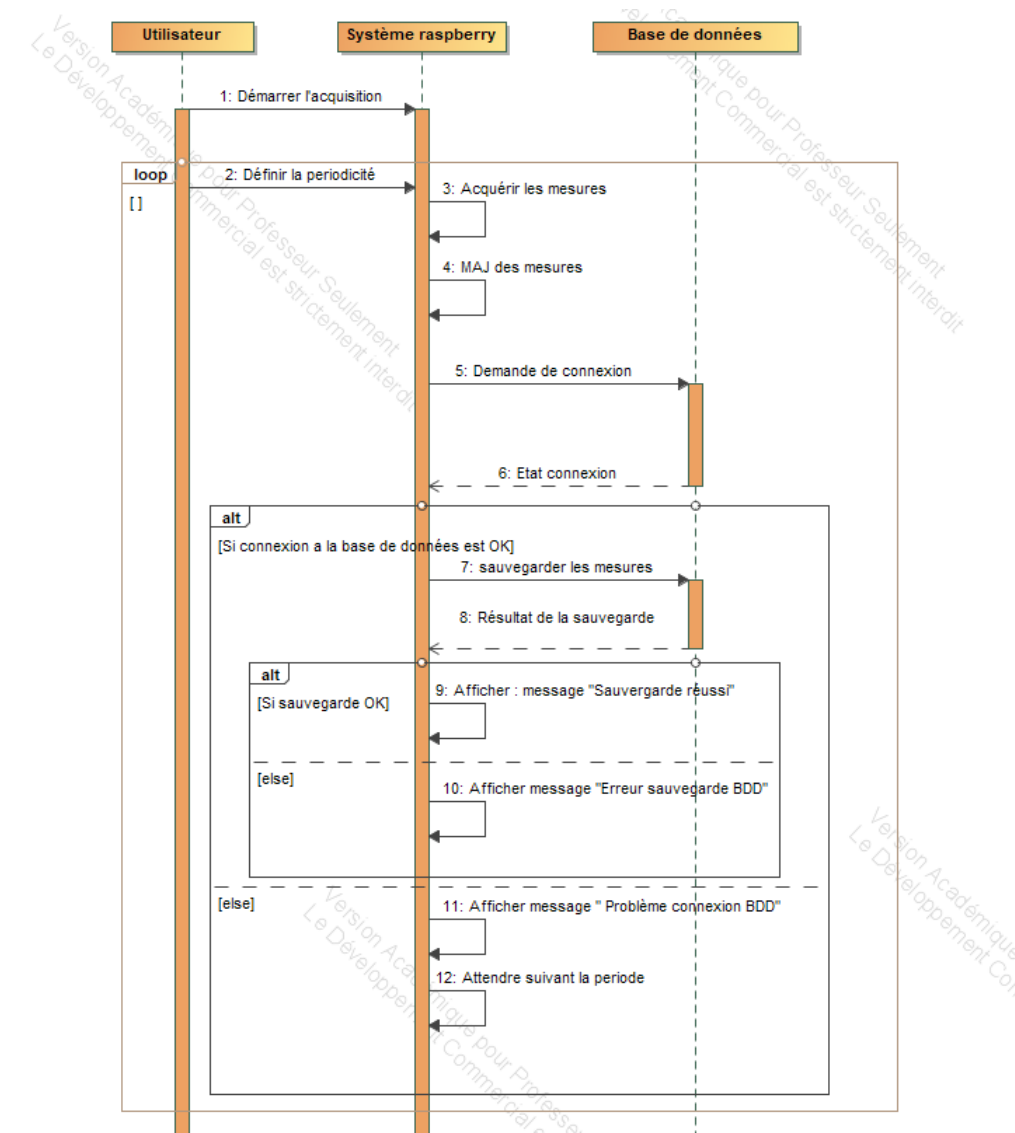
### 3.1) Étudiant 1 : Analyse des blocs Raspberry/Anémomètre

#### Diagramme de cas d'utilisation :



Pour le bloc Raspberry, l'utilisateur pourra définir la périodicité des mesures, démarrer l'acquisition et ainsi acquérir les mesures pour chaque capteur. Les mesures seront ensuite automatiquement enregistrées dans la base de données.

**Diagramme de séquence :**



L'utilisateur pourra acquérir les mesures puis si la connexion à la base de données est effective, il pourra les sauvegarder dans celle-ci.



### 3.2) Étudiant 2 : Analyse des blocs Web/Solarimètre

**Diagramme de cas d'utilisation :**

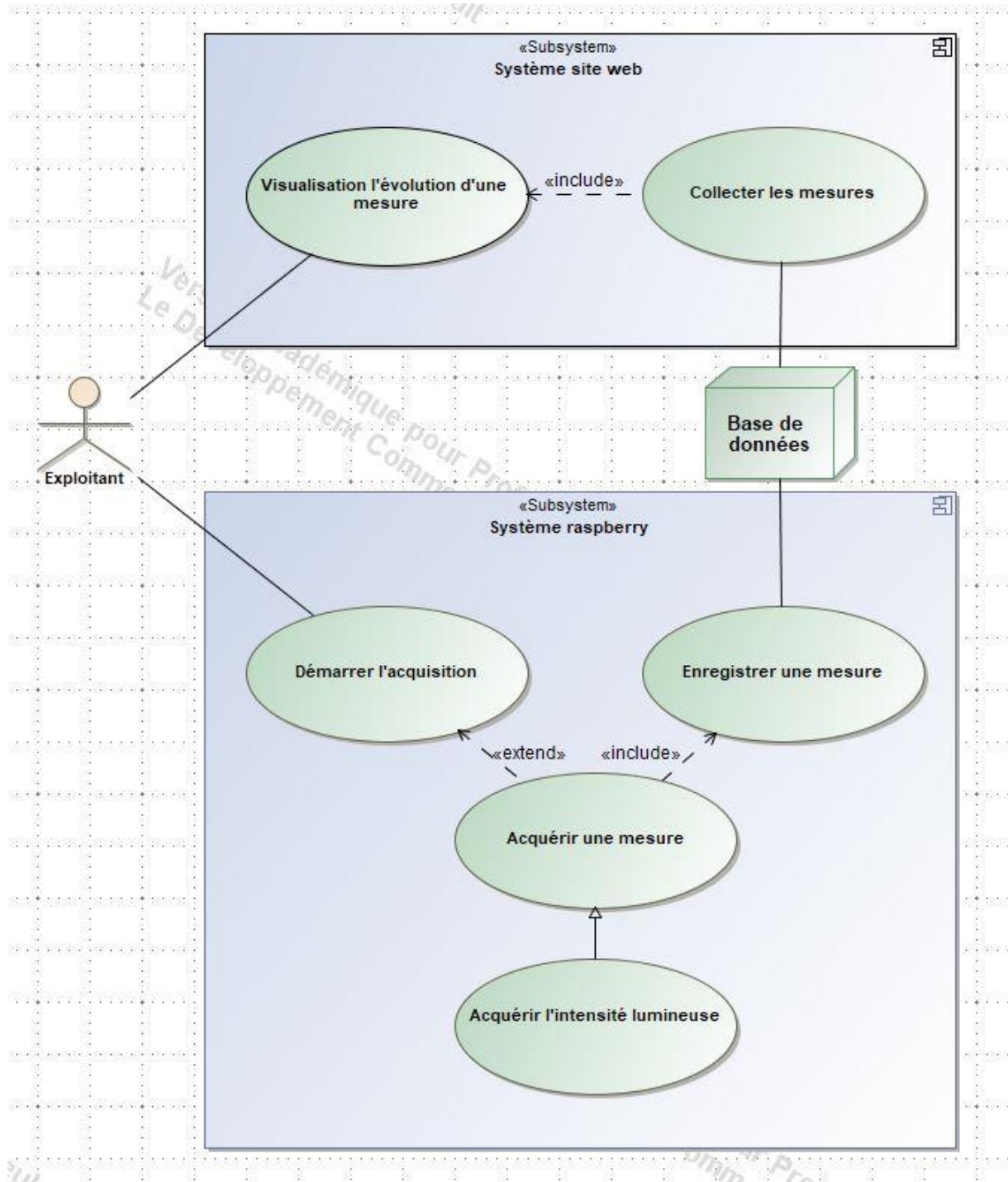
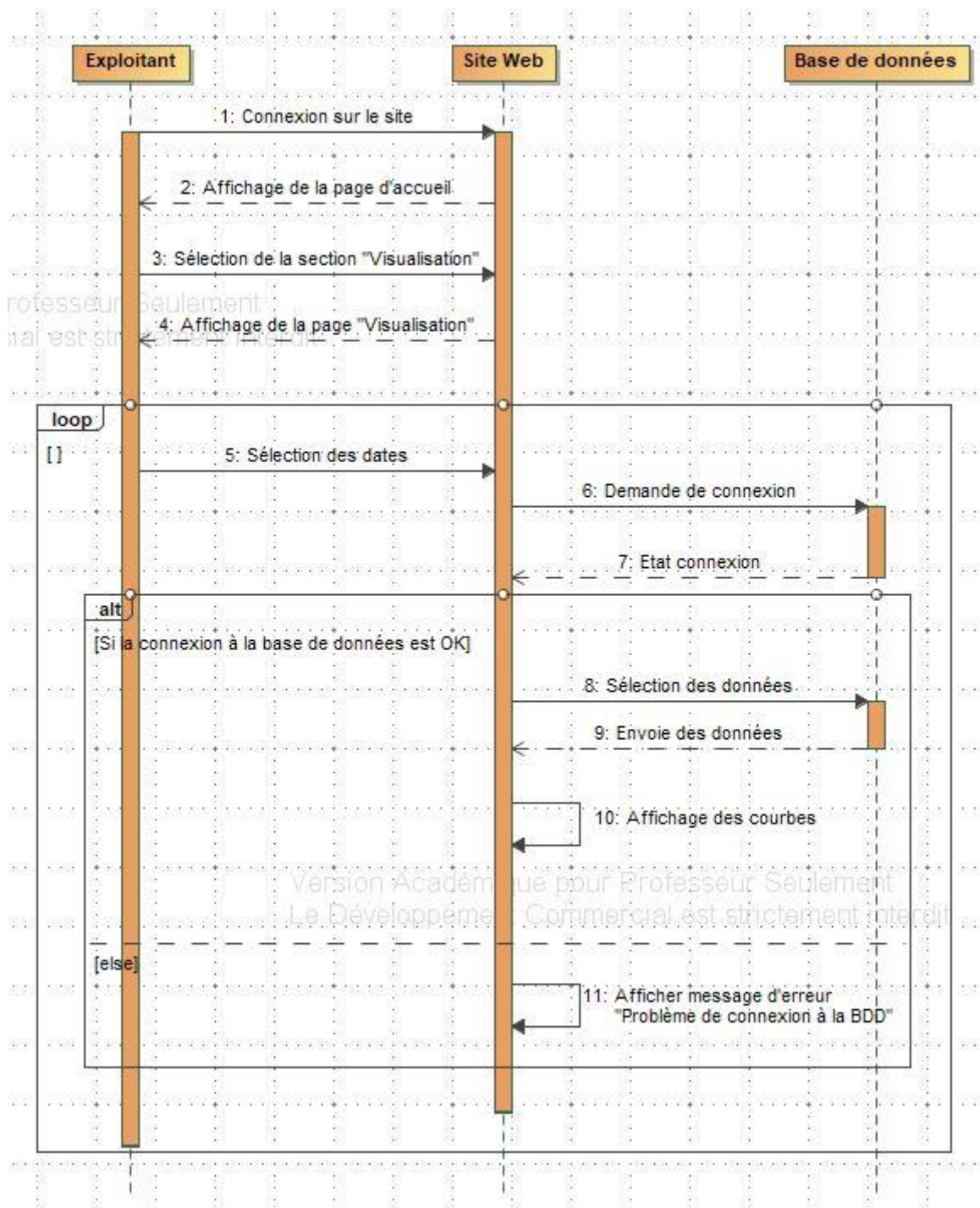


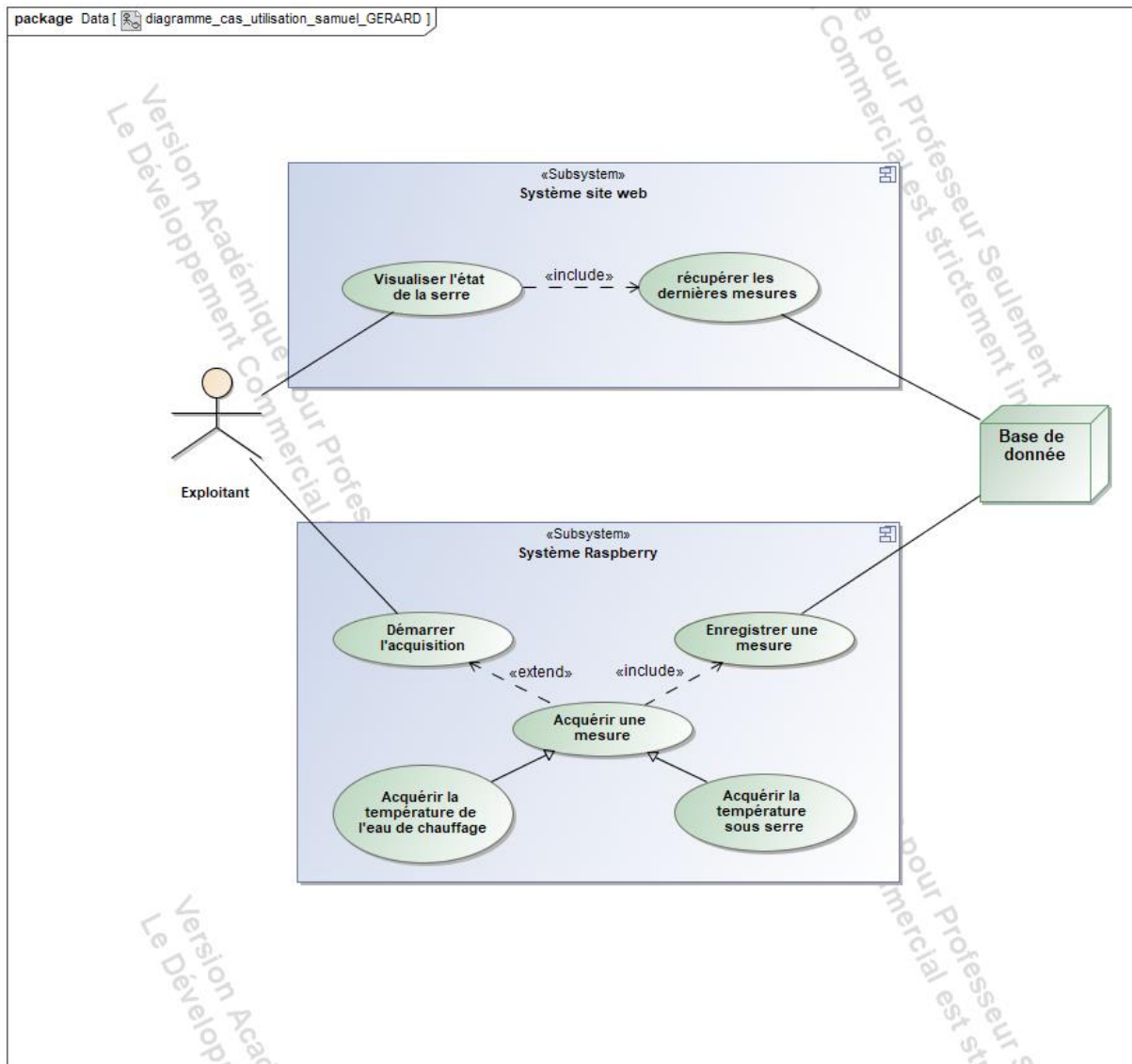


Diagramme de séquence :

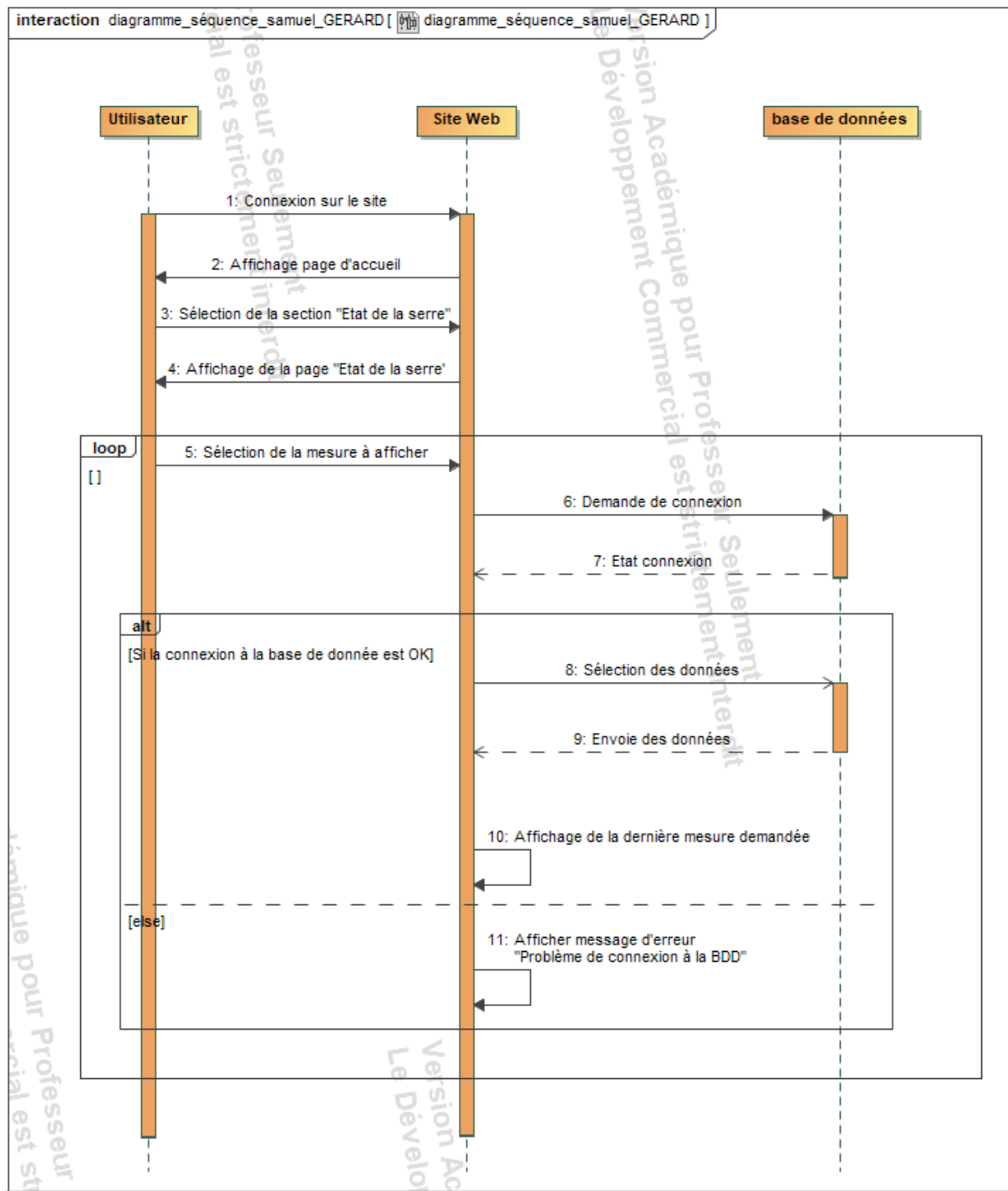


### 3.3) Étudiant 3 : Analyse des blocs Base de données/Sonde température

#### Diagramme de cas d'utilisation :

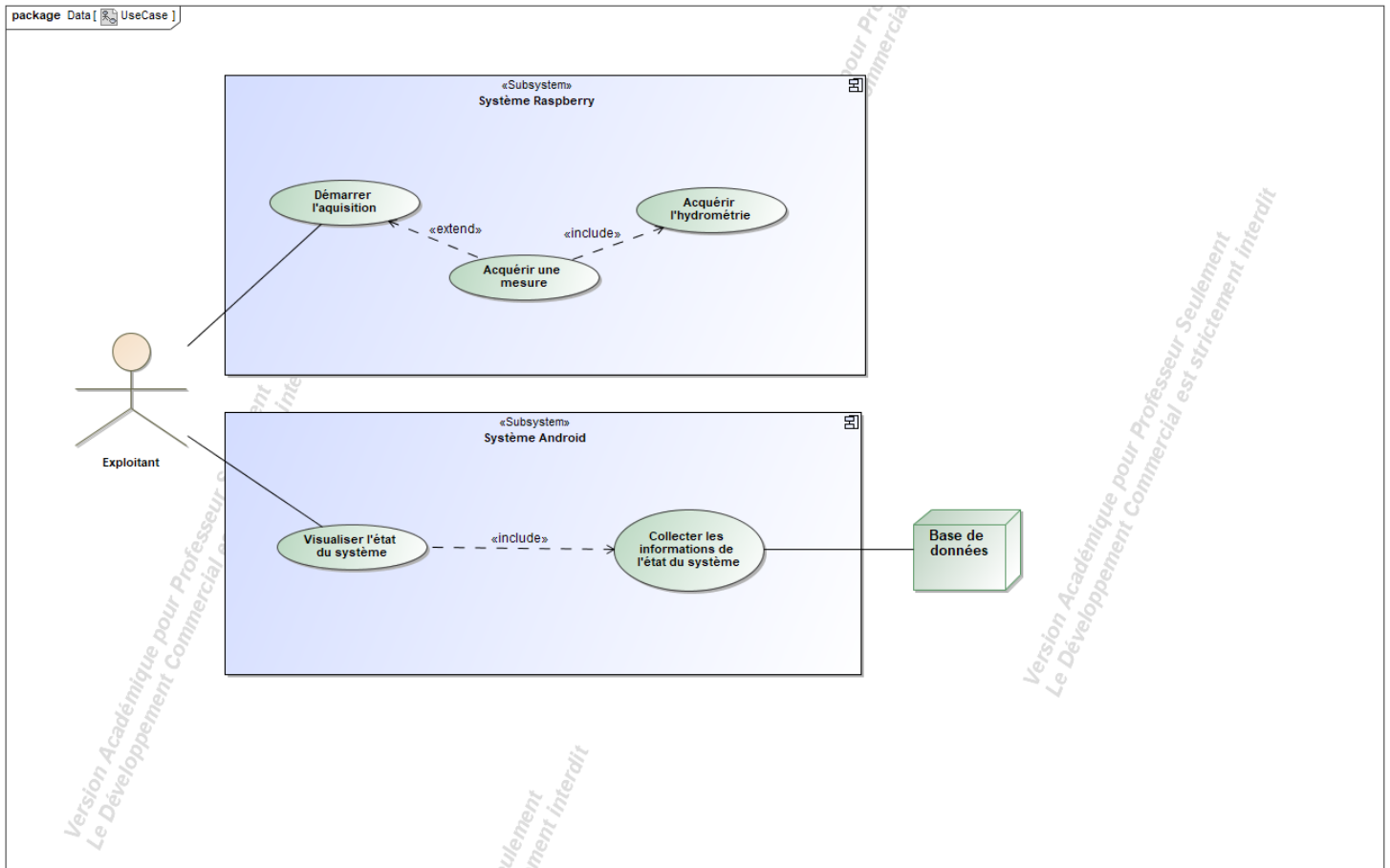


**Diagramme de séquence :**

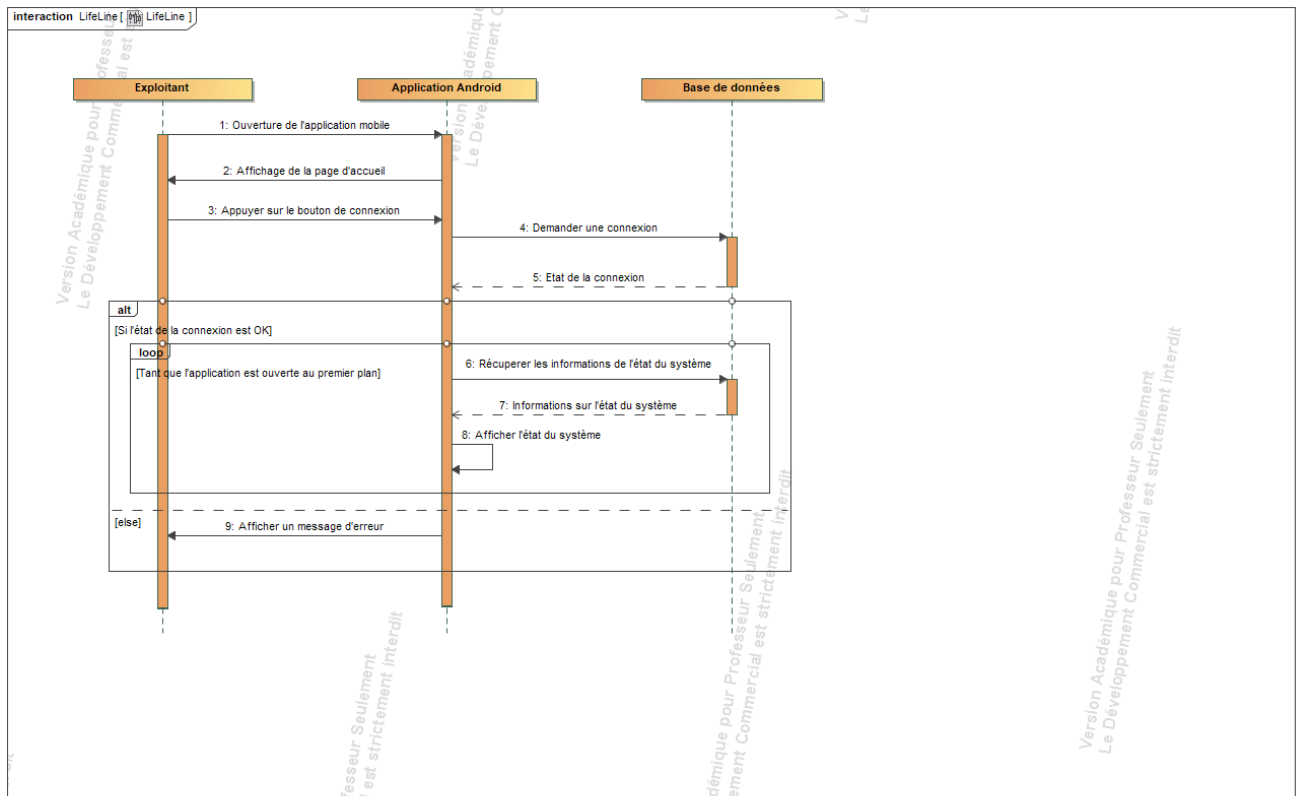


### 3.4) Étudiant 4 : Analyse des blocs Android/Pluviomètre

#### Diagramme de cas d'utilisation :



**Diagramme de séquence :**



### IV. Intégration et versionning

Comme nous travaillions tous les trois sur un seul et même logiciel, nous avons décidé de mettre en place un dépôt privé avec l'utilitaire GitHub afin de nous simplifier grandement la tâche d'intégration de chaque bloc. Après avoir exporté le projet initial vers le dépôt, il nous a été possible de le cloner sur chacun de nos ordinateurs.

Durant la réalisation du projet, après chaque phase de travail, chacun pouvait enregistrer ses modifications puis les exporter vers le projet global en ligne. Ainsi, nous avions tous en permanence les modifications des uns et des autres ce qui permettait d'avoir une intégration continue. Cela nous a permis de gagner du temps de conception et de réalisation.

Pour cela, GitHub propose un utilitaire Windows nommé GitHub Desktop qui permet d'effectuer toutes ces opérations très simplement. Chaque modification du code ou des fichiers sources sont réversibles et on peut explorer les versions plus anciennes du projet. Les lignes de code ajoutées et supprimées apparaissent de manière très lisible.

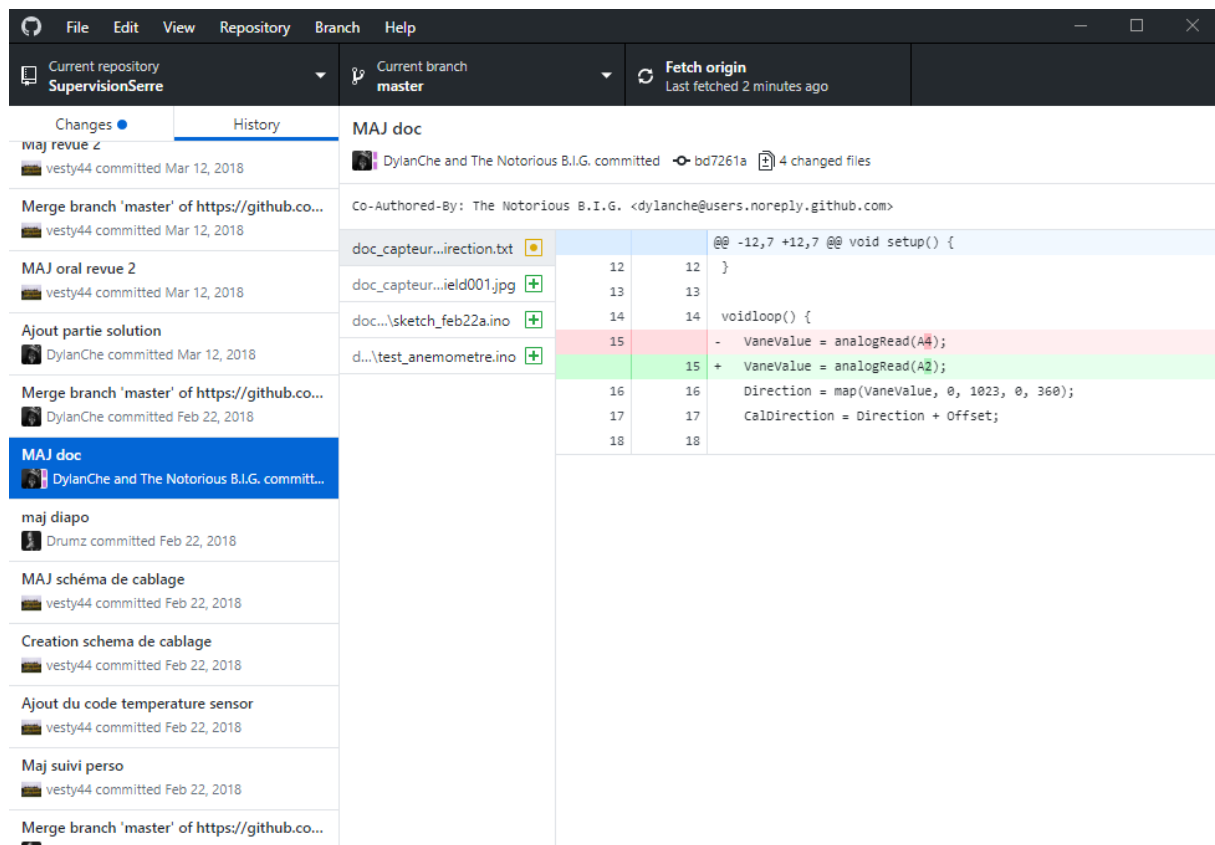


Figure 1 : Fenêtre principale de l'utilitaire GitHub Desktop

## V. Conclusion

Durant les phases de spécifications et d'analyse, nous avons mis en place un diagramme de Gantt avec le logiciel Microsoft Project afin d'ordonner judicieusement les tâches à réaliser. Ce diagramme est disponible en annexe 1 et a été mis à jour tout au long du déroulement du projet. Les tâches détaillées des étudiants sont consultables dans les parties de rédaction personnelles.

Cette phase de rédaction des spécifications nous a permis de mettre en évidence les différents besoins exprimés au travers du cahier des charges. Nous avons ainsi pu reformuler ce besoin pour chacune de nos parties pour les modéliser sous forme de diagrammes inspirés de ceux proposés par le client. Nous avons donc pu mettre en évidence les liens entre nos parties.

Pour la réalisation de ce projet, nous n'avons pas eu de phase d'intégration à proprement parler. En effet, le fait de travailler sur une seule et même base logicielle et d'utiliser GitHub nous a permis d'avoir une intégration continue.

A la fin de la réalisation, nous avons mis en œuvre des manuels utilisateurs pour les différentes parties. Les différents manuels sont disponibles à la fin du classeur.

Enfin, les rapports personnels des phases de conception et de réalisation détaillées pour chacune des parties sont disponibles à la suite de cette partie.



## VI. Annexes

### 7.1) Diagramme de Gantt

