

Partie personnelle

de

Killian LABATTUT

**Serre Automatique**

Système d’acquisition

# Dossier Technique - Partie Personnelle

Table des matières

[Dossier Technique - Partie Personnelle 1](#_Toc8992179)

[Situation dans le projet 1](#_Toc8992180)

[La tâche de l’étudiant dans le contexte du projet 2](#_Toc8992182)

[Diagrammes Sys ML 2](#_Toc8992183)

[Diagrammes SysML 3](#_Toc8992184)

[Présentation des logiciels utilisées 6](#_Toc8992185)

[Diagramme de déploiement 6](#_Toc8992186)

## Situation dans le projet

## 

## Démarrer l’acquisition,

## Au sein du projet, j’ai eu pour mission de faire échanger les cartes de gestion et d’acquisition afin de récupérer les valeurs des sept capteurs présents dans la serre. Pour se faire un programme est codé en langage python et interprété par la Debian 8.7 installée.

Acquérir les mesures direction et force du vent,

Le station météo La Crosse « Weather Station La Crosse WS 3650 » m’a été confié pour réaliser les captures de direction et de force du vent. Cette station est un module qui combine les deux capteurs, ainsi les données sont transmises ensemble par un BUS RS232 et un câble RJ-11.

Définir la périodicité des mesures,

La périodicité est récupérée par le programme d’acquisition depuis la base de données et est directement modifiable via l’application.

Enregistrer toutes les mesures dans la BDD,

Les mesures sont « effectuées » ou non par la carte d’acquisition Arduino Uno, d’après le calcul de périodicité.

Journaliser les évènements,

Ils sont écrits dans un fichier dit « fichier log » ainsi toutes traces d’envoi de données dans la base d’erreurs d’envoi est justifié.

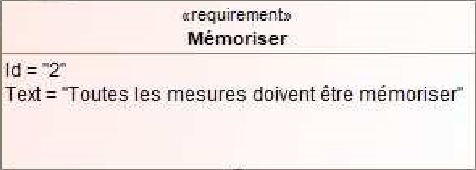
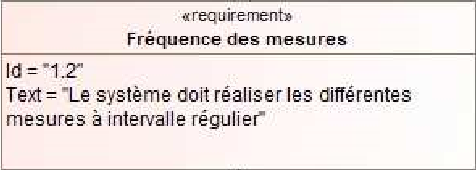
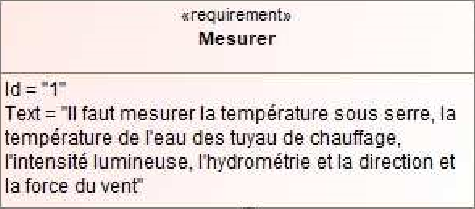
## La tâche de l’étudiant dans le contexte du projet

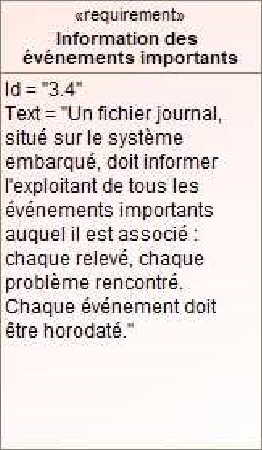
## Carte de Gestion Carte d’acquisition

## 

**Capteurs**

**Capteurs**

À l’aide du diagramme d’exigences formulées par le Groupe Olivier, j’ai extrait les différentes exigences qui me concernent affichés ci-dessous.



J’ai réfléchi aux à la construction des programmes, et je me suis décidé de programmer la Raspberry à demander constamment les valeurs des sept capteurs. Le message contient le numéro du capteur ainsi que le temps en secondes écoulé depuis 1970 et la période d’acquisition du capteur.

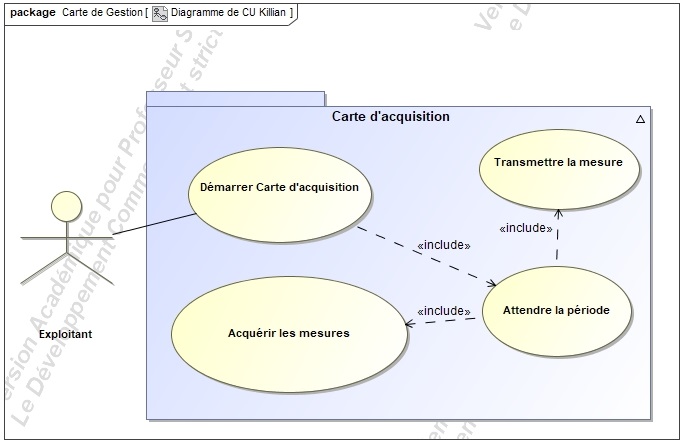
Pendant ce temps la carte d’acquisition reçoit le message et traite les données reçus elle fait la différence entre le temps du message et le dernier temps reçu. S’il est égale à la période d’acquisition transmise, la valeur du capteur et son id sont envoyé à la Raspberry qui enregistre envoi la valeur dans la base de données.

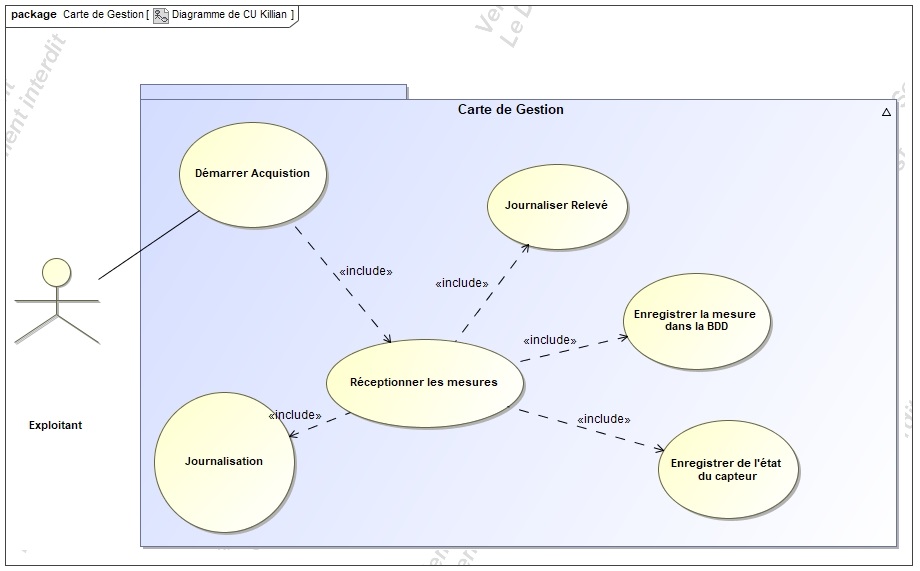
Pour pouvoir répondre aux exigences j’ai réalisé différents diagrammes UML afin de détailler ma partie.

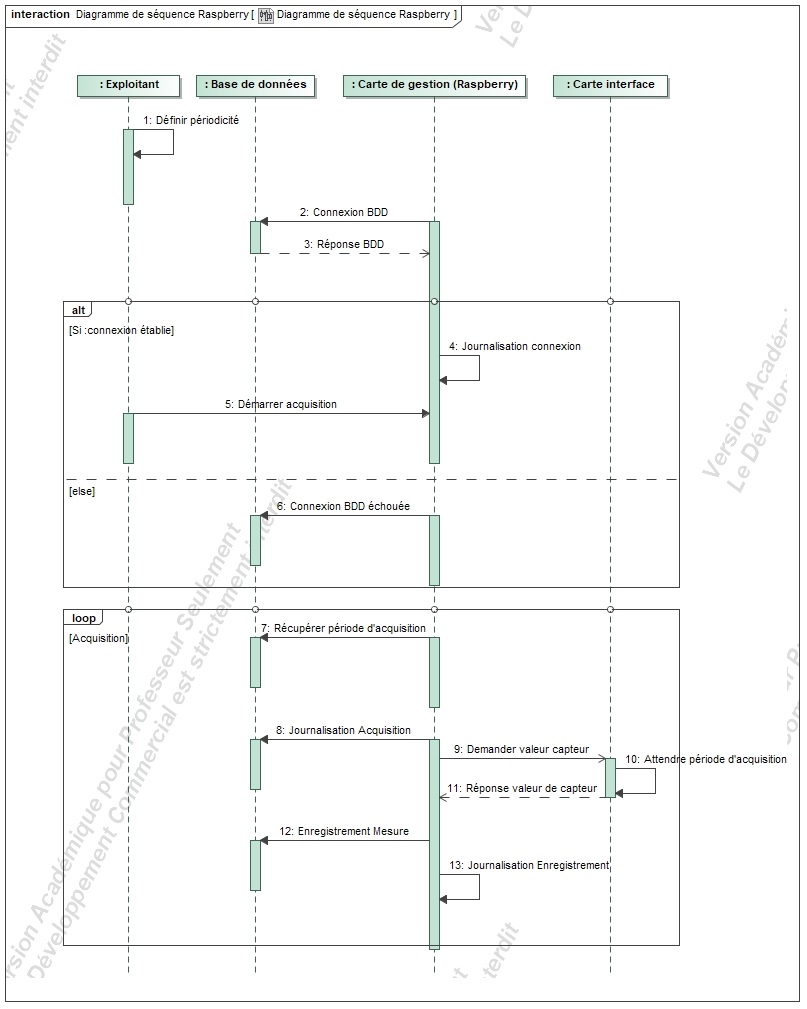
## Diagrammes SysML

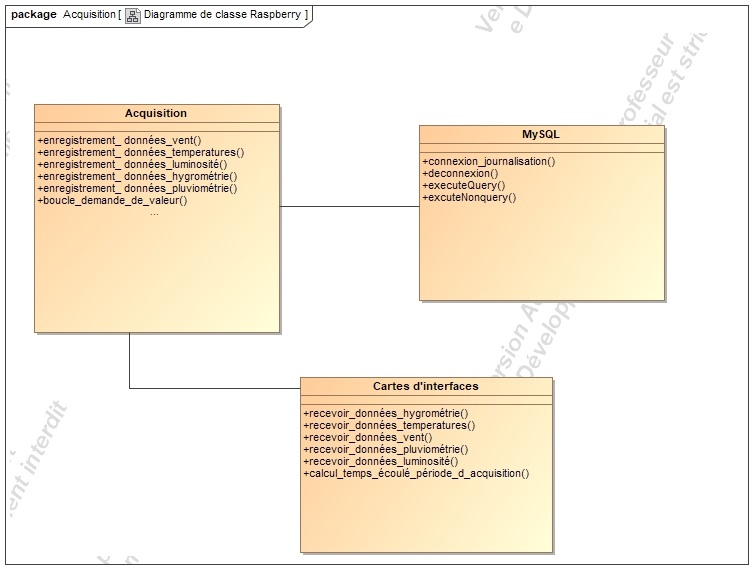
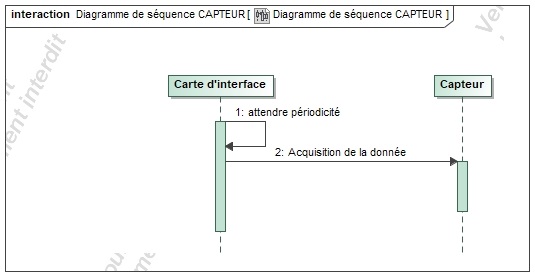
Voici les diagrammes de cas d’utilisation de la carte de Gestion et de la carte d’Acquisition.

Les diagrammes représentent les deux systèmes et les actions qu’ils effectuent.





Voici le diagramme de séquence il affiche l’ordre chronologique



## Présentation des logiciels utilisées

## Diagramme de déploiement

