

SFL6 : SERRE AUTOMATIQUE

SYSTEME DE REGULATION





Sommaire

I.	Présentation du projet	2
A.	Enoncé Général du besoin.....	2
B.	Aspect contractuel du projet.....	2
II.	Expression du besoin.....	2
III.	Identification des équipements.....	4
A.	Synoptique du système de régulation.....	4
B.	Description des équipements.....	8
IV.	Identification des besoins.....	9
A.	Besoins fonctionnels.....	9
B.	Besoins non-fonctionnels	10
V.	Ressources mises à disposition	11
A.	Ressources matérielles.....	11
B.	Ressource Logicielles	11
C.	Librairies utilisées	11
D.	Documentation.....	11
VI.	Analyse	12
A.	Tâches à réaliser	12
B.	Diagramme d'exigence.....	13
C.	Diagramme de cas d'utilisation	14
D.	Diagramme de séquence.....	15
E.	Diagramme de classe.....	22
F.	Diagramme de base de données.....	25



I. Présentation du projet

A. Enoncé Général du besoin

Le groupe Olivier, à travers ce projet souhaite créer un prototype de serre automatisée à moindre coût. L'entreprise est donc passée par l'établissement Saint Félix La Salle et plus particulièrement les professeurs de la section informatique industrielle.

Ce projet, Serre automatisée est lui-même décomposé en deux parties distinctes, « Serre automatisée, système d'acquisition (SFL5) » et « Serre automatisée, système de régulation (SFL6) ». Notre projet sera « Serre automatisée, système de régulation ».

Pour ce prototype, une maquette réduite de serre (*Voir image 1*), ses capteurs et ses actionneurs nous seront mis à disposition.

Deux applications, une application web et l'autre mobile, permettra au client de gérer les différents actionneurs pour un résultat optimal.



Image 1: Maquette de la serre automatisée

B. Aspect contractuel du projet

Commanditaire	Groupe Olivier
Acteurs	Exploitant
Temps de réalisation	Du 8 janvier au ?
Equipe de développement	3 étudiants
Professeur référent	Monsieur Sébastien ANGIBAUD

II. Expression du besoin

Le Groupe Olivier veut automatiser son système de serre avec une solution à moindre coût. Mais avant de le déployer en grande échelle, l'entreprise souhaiterait le tester sous forme de prototype. Par manque de personnel, l'entreprise a donc voulu commanditer sous forme de projet ce prototype à des étudiants en BTS.



Dans un premier temps, le système de régulation gèrera trois différentes grandeurs physiques qui sont :

- L'hygrométrie
- L'intensité lumineuse
- La température

Ces trois valeurs seront chacune gérées par un actionneur, l'hygrométrie par le brumisateur, l'intensité lumineuse par un système de volet roulant et la température par un plateau chauffant.

Le cahier des charges nous impose deux différents types de modes, à savoir le mode profil et le mode manuel.

Le mode manuel permet au client de modifier manuellement l'état de la serre selon les grandeurs physiques rentrées.

Le mode profil permet de préenregistrer des valeurs selon une heure fixée modifiant également l'état de la serre. Il y aura au préalable un profil par défaut qui définit les conditions idéales pour la production végétale. Ce mode peut être défini comme un mode automatique.

Gestion de la base de données

La base de données est commune avec le projet SFL5 « Serre automatisée, système d'acquisition » située sur un serveur à part permet de stocker différentes informations de la serre comme la température extérieure, intérieure, l'hygrométrie, l'intensité lumineuse et enfin la direction et la vitesse du vent.

Elle permet également de stocker les différents profils, les réglages des actionneurs et les capteurs.

Gestion des utilisateurs

Il y a deux modes de régulation différents, le mode manuel et le mode profil. Si le mode profil est choisi, l'utilisateur pourra créer, modifier et/ou supprimer un profil. S'il choisit le mode manuel il pourra entrer des valeurs pour la température, l'humidité et la luminosité.

Ensuite les informations du profil seront stockées sur la base de données.

Gestion de la régulation

Dans un premier temps le système récupère de diverses informations sur la base de données. Ensuite il doit pouvoir savoir si le mode est profil ou manuel, selon le mode il doit pouvoir ensuite activer ou désactiver un ou plusieurs actionneurs.



III. Identification des équipements

A. Synoptique du système de régulation

Plan réseau des équipements du système

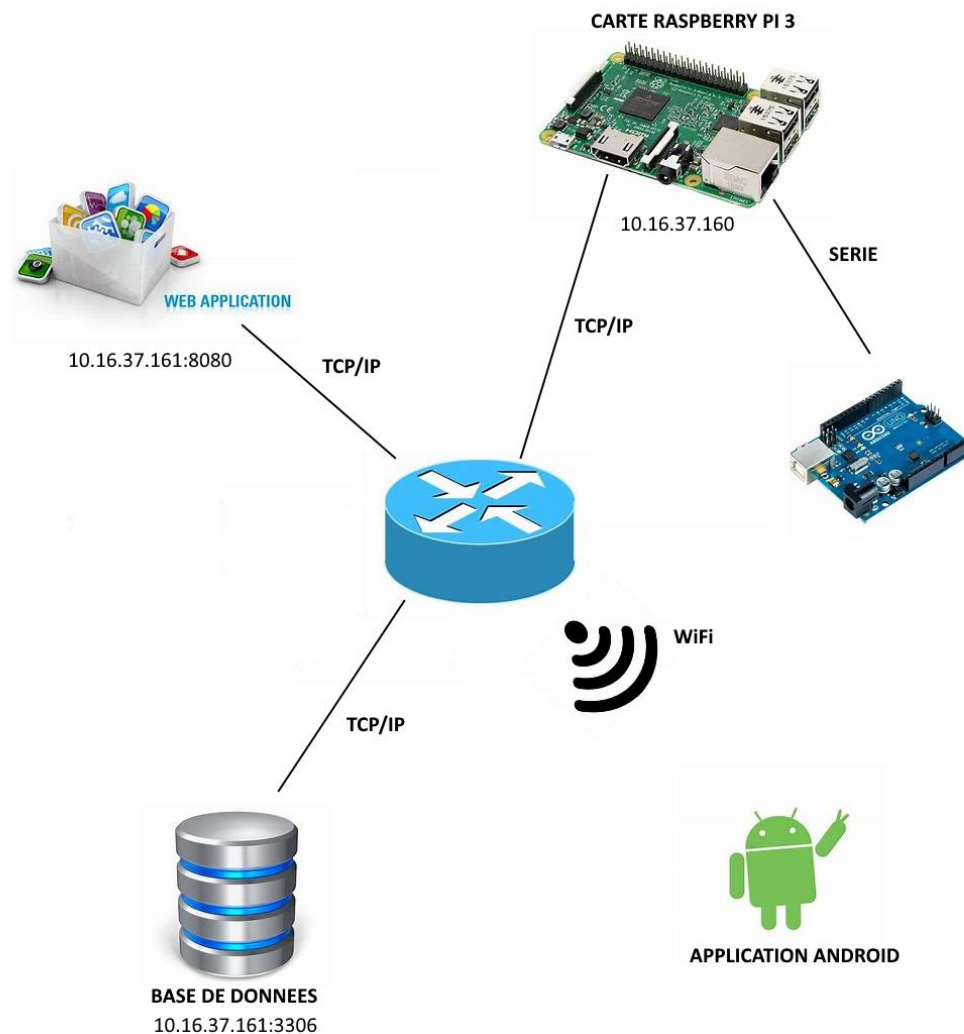


Image 2:

Plan réseau des équipements du système



Schéma de câblage des actionneurs du système

Plateau chauffant :

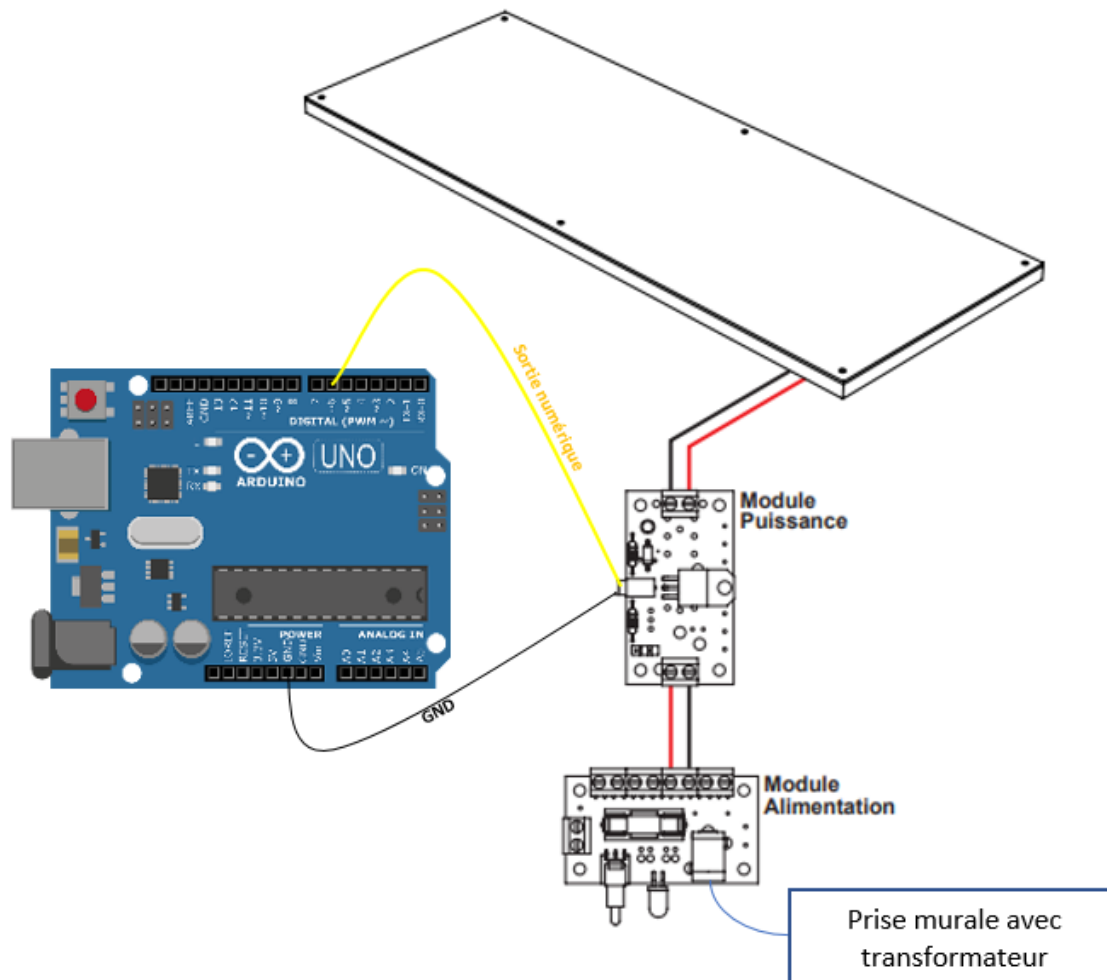


Image 3: Schéma de câblage du plateau chauffant



Servomoteur :

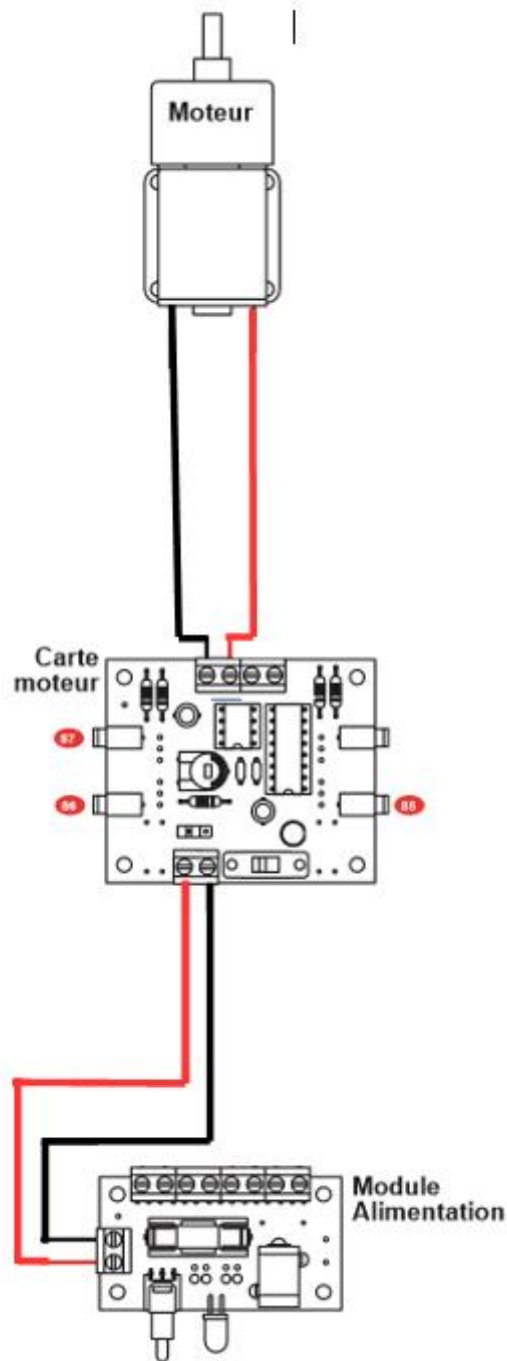


Image 4: Schéma de câblage du servo-moteur



Brumisateur :

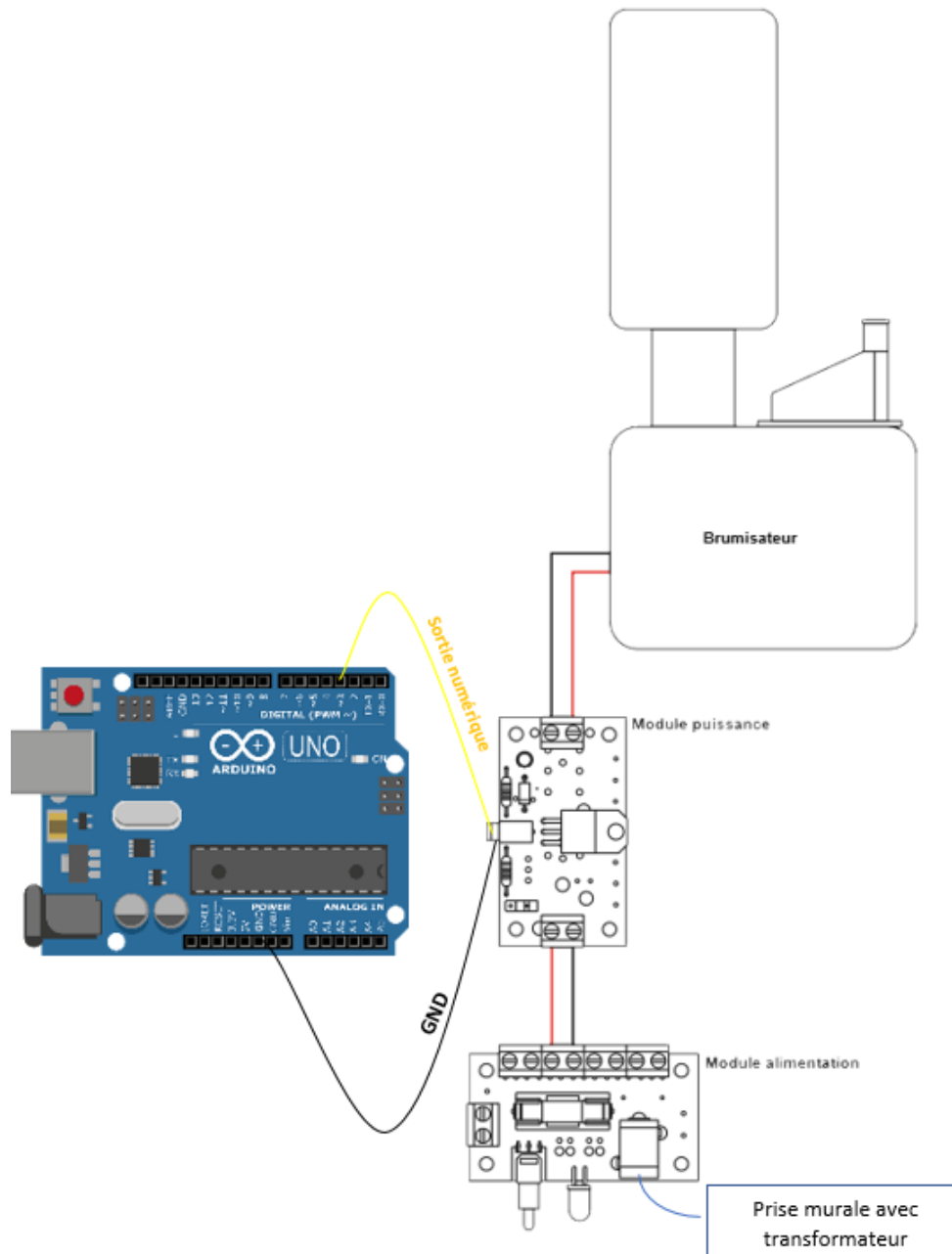


Image 5: Schéma de câblage du brumisateur



B. Description des équipements

Carte de gestion : Selon le cahier des charges, il nous est imposé d'exploiter une carte Raspberry Pi 3. Celle-ci permettra de récupérer les informations et d'envoyer des ordres aux actionneurs.

Carte Arduino : La carte Arduino (Uno/Mega) reçoit les données envoyées par la carte Raspberry et commande les actionneurs.

Actionneurs : Il y en a donc trois :

- Brumisateur : Permet de gérer l'hygrométrie de la serre
- Moteur : Permet de gérer les volets de la serre
- Plateau chauffant : permet de gérer la température de la serre

Ces trois actionneurs sont commandés par la carte Arduino.

Application Android : Elle permet de choisir le mode entre profil et manuel, et d'entrer manuellement des valeurs.

Application Web : Elle permet de créer, modifier ou supprimer un profil.



IV. Identification des besoins

A. Besoins fonctionnels

Le principe du système est la gestion de l'état de la serre via différents moyens informatiques.

Récupération des données

Lors du lancement du système la première action est de se connecter à la base de données, le système envoie une requête pour récupérer plusieurs informations comme :

- Les actionneurs avec l'ID, le nom, le port, l'unité, l'heure de la dernière utilisation
- Le mode, soit profil ou manuel

Traitement des données

Après la récupération des données il faut bien sûr traiter les informations. Selon l'actionneur le système récupérera son instance et activera ou désactivera finalement l'actionneur.

Carte de gestion

Communiquer avec la carte Arduino :

La carte Raspberry enverra dans un premier temps une donnée via le protocole RS232 pour pouvoir contrôler les actionneurs après la récupération et le traitement des données.

Communiquer avec les actionneurs :

Après avoir reçu les données sur la carte Arduino, celle-ci va traiter les données et commander l'activation ou la désactivation des broches de la carte. Elle fonctionne donc comme un interrupteur ON/OFF.

Les différentes fonctions présentées ci-dessus sont représentées par des diagrammes de séquences, de classes et de cas d'utilisation (page ?).



B. Besoins non-fonctionnels

Développement

La carte de gestion sera commandée par un Raspberry Pi3, sous un OS Raspbian, le programme sera codé sous python avec le logiciel PyCharm. Les librairies utilisées par ce système sont :

- Pip : qui est un installateur de librairie pour python.
- Pyserial : qui est une librairie pour la communication série.
- Mysql : mysql-python, mysql-connector-python : permet la connexion à la base de données.

La carte Arduino sera connectée en série sur la carte Raspberry, on utilisera un protocole particulier de communication qui est le SciPy. C'est un protocole permettant de communiquer avec des instruments de mesure. Ici nous utiliserons plutôt un schéma d'envoi de données en série permettant de mieux structurer celles-ci.

L'application web sera développée avec un Framework nommé Symfony. C'est un logiciel très répandu dans le milieu professionnel. De plus il permet un modèle MVC (Model View Controller). Il sera installé sur un serveur à part avec la base de données sous le format MySQL.

L'application Android sera développée sous Java avec Android Studio. Le choix de ce langage est dû au développement d'une interface graphique sous un appareil Android.

Le langage UML utilisé pour les différents diagrammes, nous permet de façonner le projet de façon structurée et de mieux en comprendre les limites. Il nous permet d'éclaircir nos objectifs.

Contraintes financières

La réalisation du projet implique l'achat de plusieurs matériels :

- La maquette de la serre automatisée : 607,78 €
- Les différents actionneurs (Plateau chauffant : 128.30 €, Brumisateurs : 215.15 €, et 3 Servomoteurs : 80 €)
- Une carte Raspberry : 35.36 (Prix sur le site officiel)
- Une carte Arduino Uno : 20 € (Prix sur le site officiel)

Le coût total de notre projet est de 1086,59 €

Contraintes qualité

L'application web sera accessible via une adresse IP définie par le constructeur, il faut que l'utilisateur se connecte au réseau du serveur d'accès de la base de données et de l'application web pour pouvoir y accéder.



V. Ressources mises à disposition

A. Ressources matérielles

- Une carte de gestion Raspberry sous Raspbian
- Un serveur sous Debian pour héberger la base de données et l'application Web
- Un smartphone pour l'application Android
- Une carte Arduino Uno
- Un plateau chauffant
- Plusieurs servomoteurs
- Un brumisateur
- Une maquette de serre automatisé

B. Ressource Logicielles

- PyCharm pour le développement en python
- Android studio pour le développement de l'application
- MySQL pour la base de données
- MagicDraw pour les diagrammes UML
- Microsoft Projet pour le diagramme de GANTT
- NetBeans pour l'application Web

C. Librairies utilisées

- Arduino :
 - SerialSoftware
- Python :
 - pyserial
 - mysql-connector-python
 - pip
 - PyQt5
- Android :
 - Volley

D. Documentation

- Documentation de la serre automatisée en format PDF
- Documentation technique du module de puissance en format PDF



VI. Analyse

A. Tâches à réaliser

Le projet est décomposé en 3 tâches distinctes. Ci-dessous vous trouverez comment a été réparti notre travail.

Etudiant 1 : Carte de Gestion

- Charger un profil depuis la base de données
- Réguler en mode profil / manuel
- Chauffer la serre
- Journaliser les évènements
- Couleur : Rouge

Etudiant 2 : Application Web / Base de données / Carte de gestion

- Création / suppression / modification d'un profil
- Mise en place de la base de données (Commun avec un étudiant de l'autre projet)
- Stockage des profils et des informations pour le mode manuel dans la base de données
- Ouvrir / fermer les volets et journaliser les événements sur la carte de gestion
- Couleur : Vert

Etudiant 3 : Application Android / Carte de gestion

- Choisir le mode de régulation
- Définir les valeurs pour le mode manuel
- Visualisation en temps réel de l'état de la serre pour les trois grandeurs physiques régulées
- Activer / Désactiver le brumisateur et journaliser les événements sur la carte de gestion
- Couleur : Bleue



B. Diagramme d'exigence

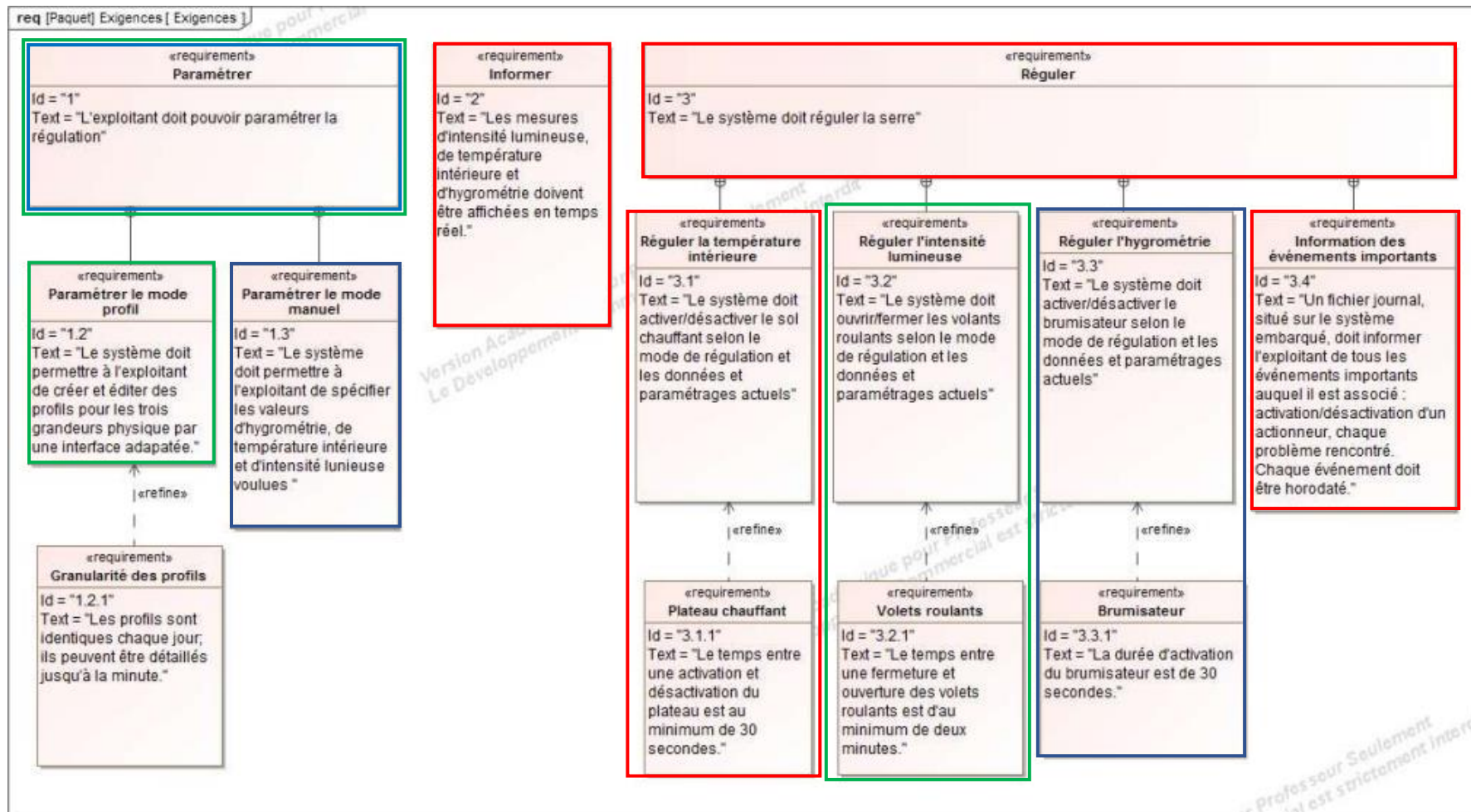


Image 6:

Diagramme d'exigences



C. Diagramme de cas d'utilisation

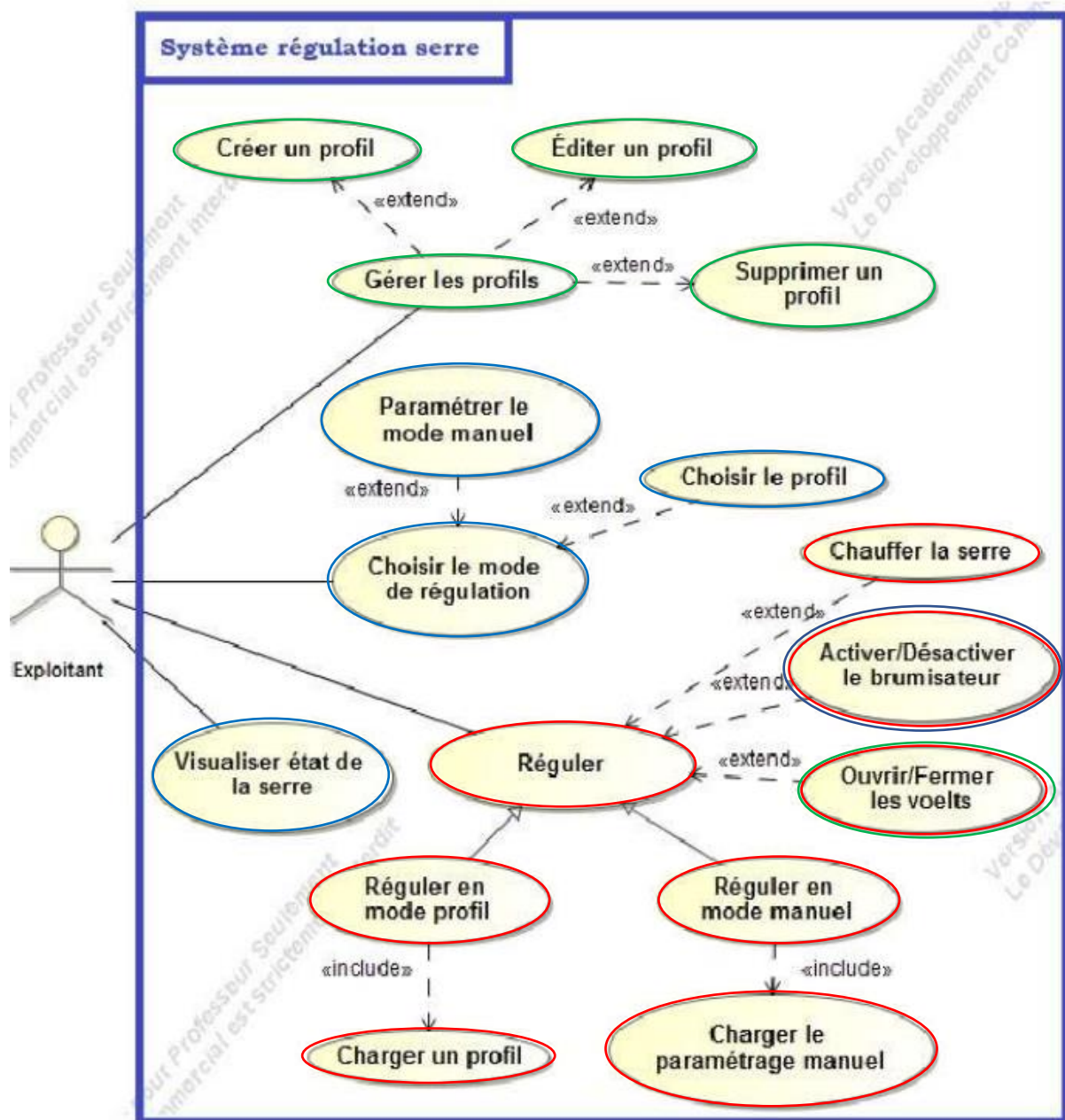


Image 7: Diagramme des cas d'utilisation



D. Diagramme de séquence

Etudiant 1 :

Mise en situation 1 : Mise en marche de la régulation

Lors de l'allumage de la carte Raspberry, le diagramme ci-dessous représente les différentes étapes du programme de régulation.

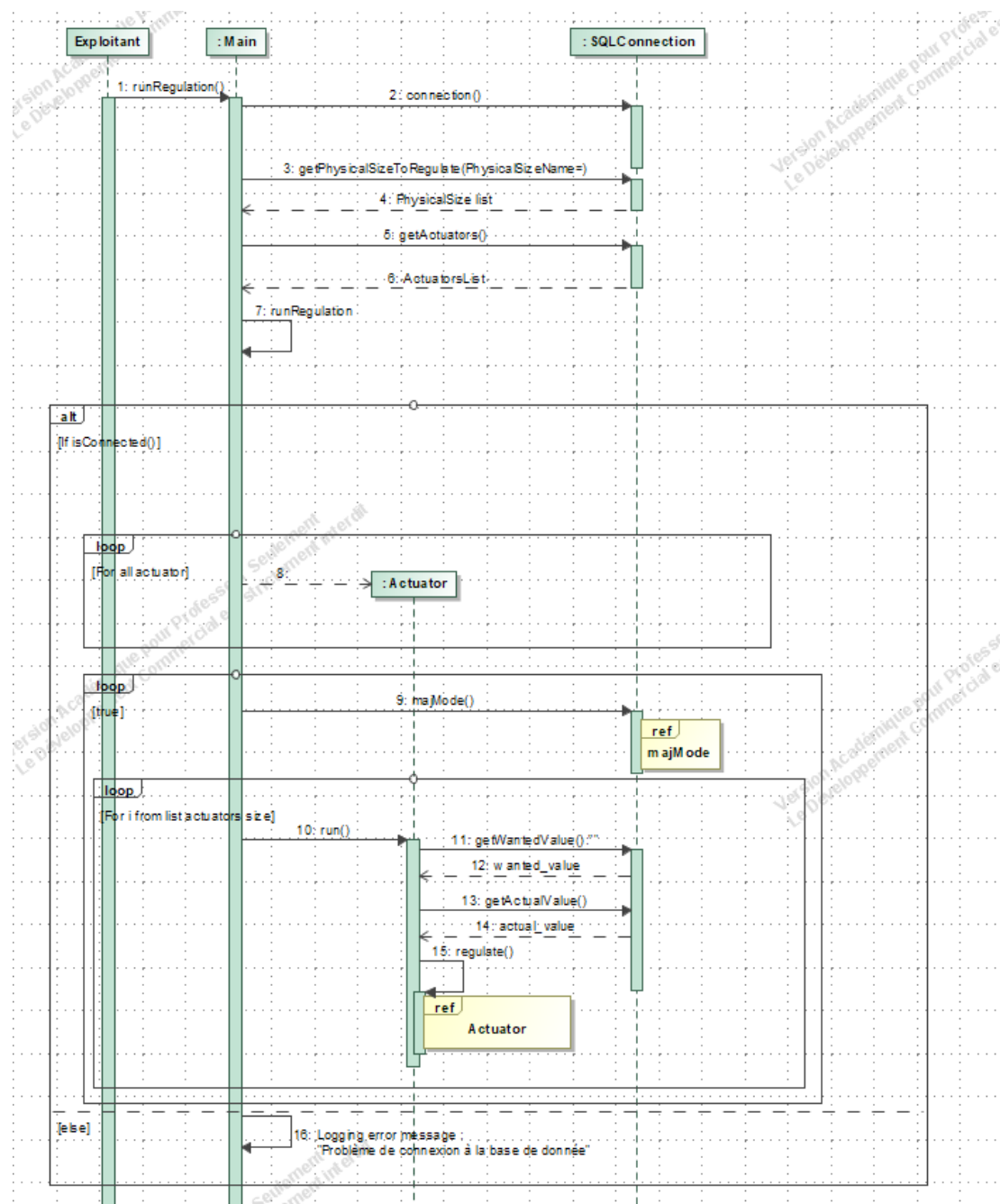


Image 8: Diagramme de séquence de la mise en marche de l'application



Mise en situation 2 : Référence majMode

Ce diagramme représente la façon dont on récupère le mode actuel de la serre, et si besoin est, de changer l'état de ce mode.

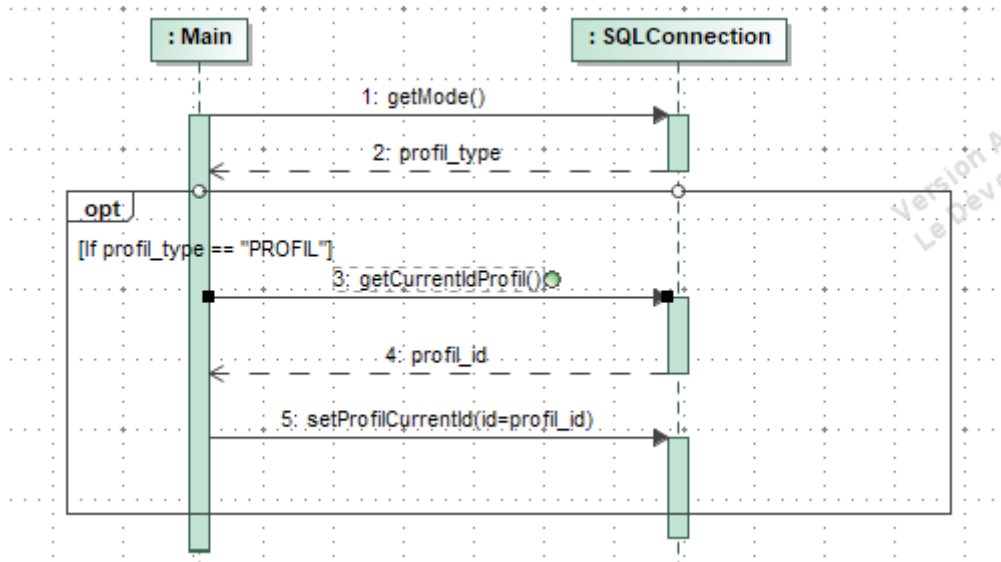


Image 9: Diagramme de séquence de la récupération d'un profil

Mise en situation 3 : Référence Actuator

Avant d'activer les actionneurs, on vérifie que si le temps actuel est inférieur au temps voulu. Le cahier des charges nous impose un temps pendant lequel nous ne pouvons pas agir sur les actionneurs.

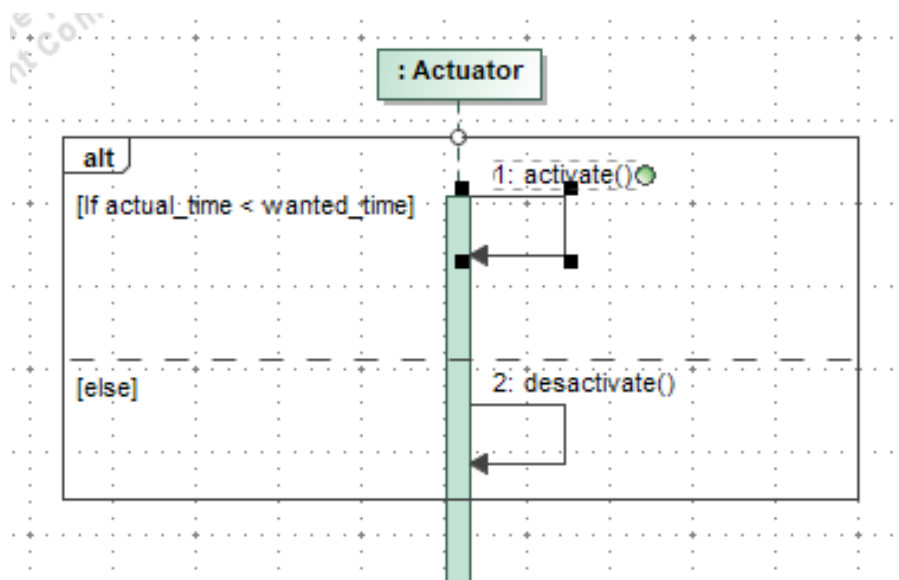


Image 10: Diagramme de séquence des actions des actionneurs



Mise en situation 4 : Régulation plateau chauffant

Ce diagramme représente comment actionner le plateau chauffant.

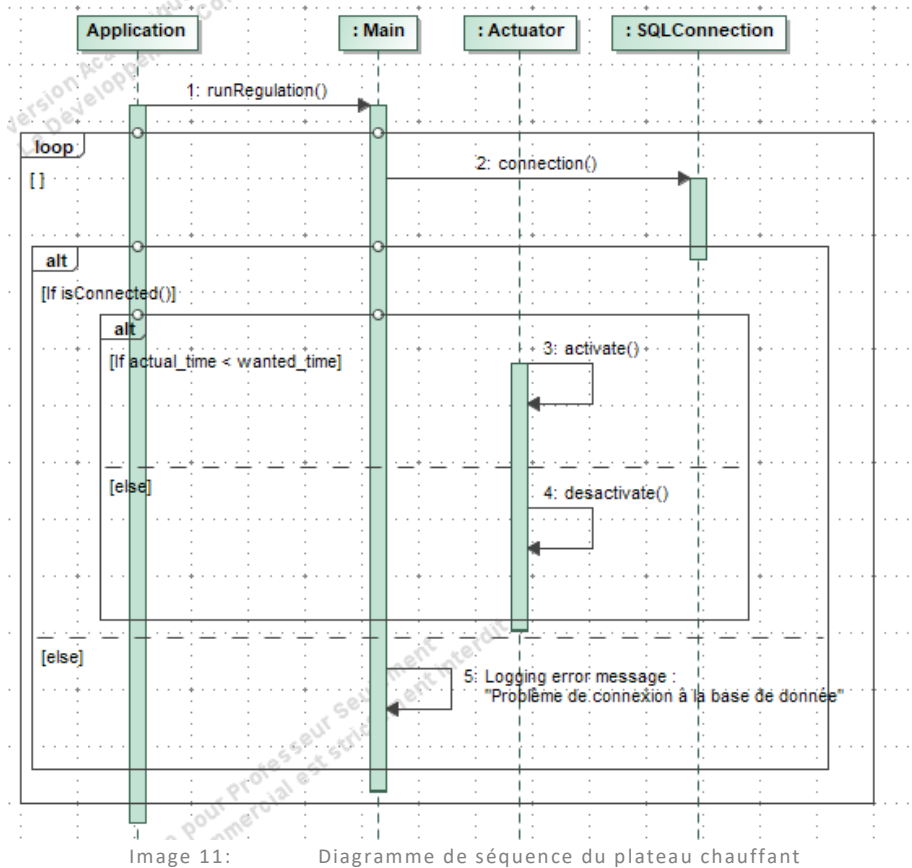


Image 11: Diagramme de séquence du plateau chauffant

Mise en situation 5 : Mise en place du protocole de communication

Ce diagramme visualise les méthodes importantes que le système utilise pour communiquer sur le port série.

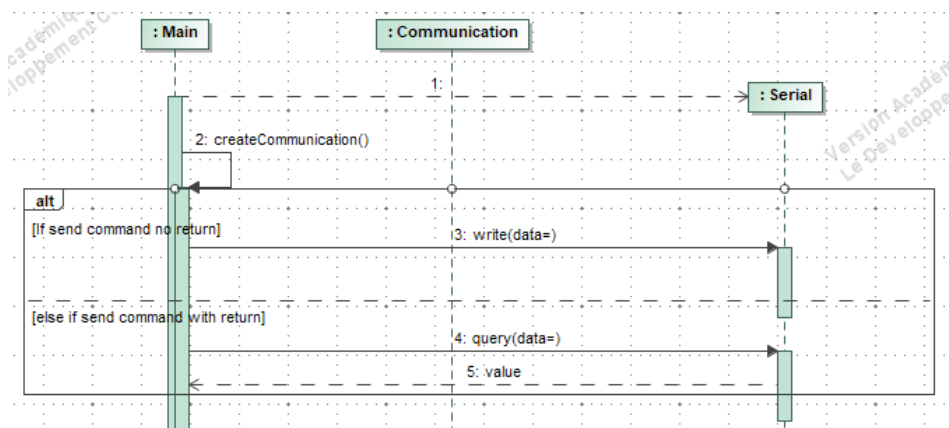


Image 12: Diagramme de séquence de la mise en place du protocole de communication



Etudiant 2 :

Mise en situation 6 : Gestion de profil

Ce diagramme représente la relation entre l'utilisateur, le site WEB, et la base de données.

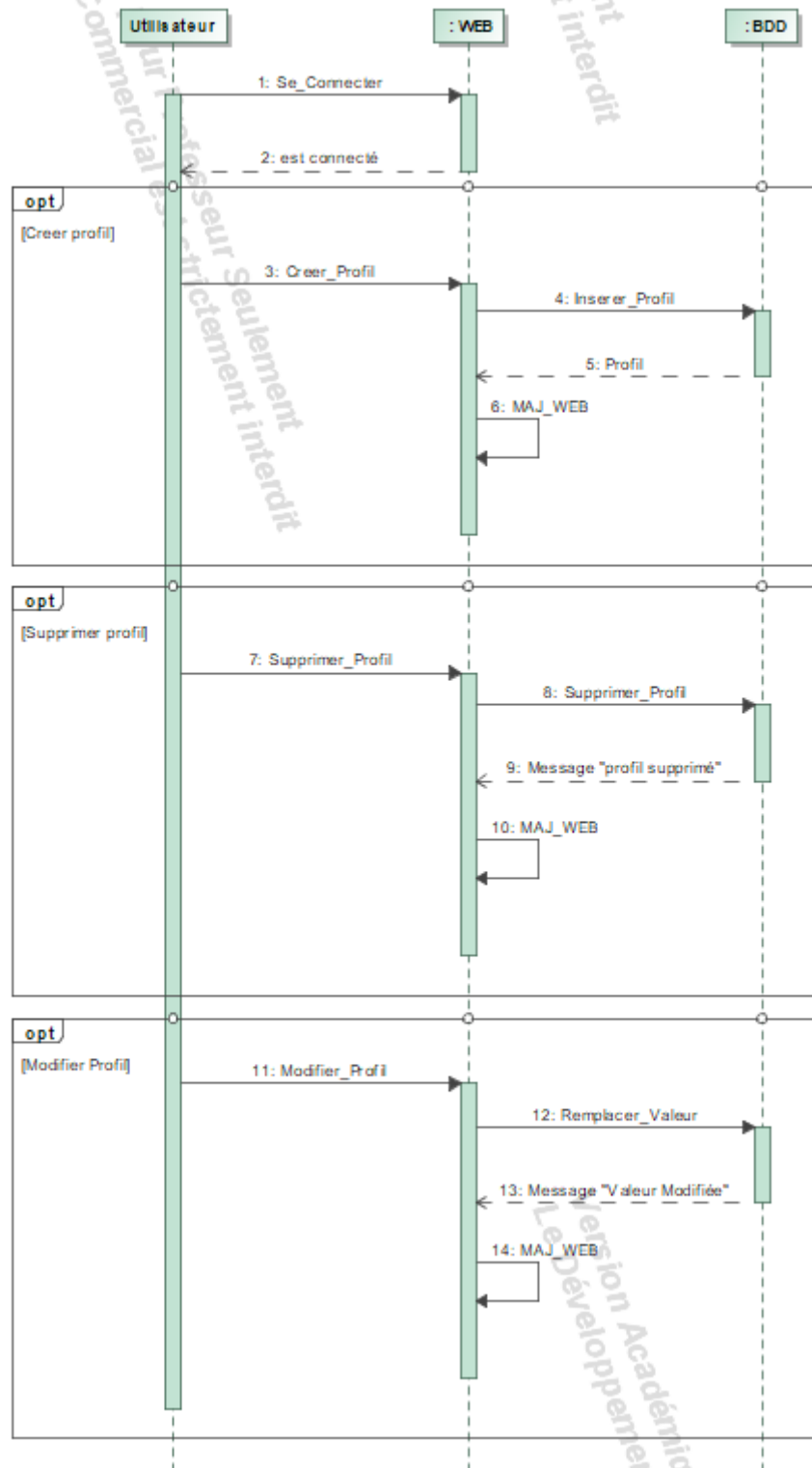


Image 13: Diagramme de séquence gestion de profil



Mise en situation 7 : Connexion base de données

Ce diagramme représente l'interaction de la carte de gestion avec la base de données.

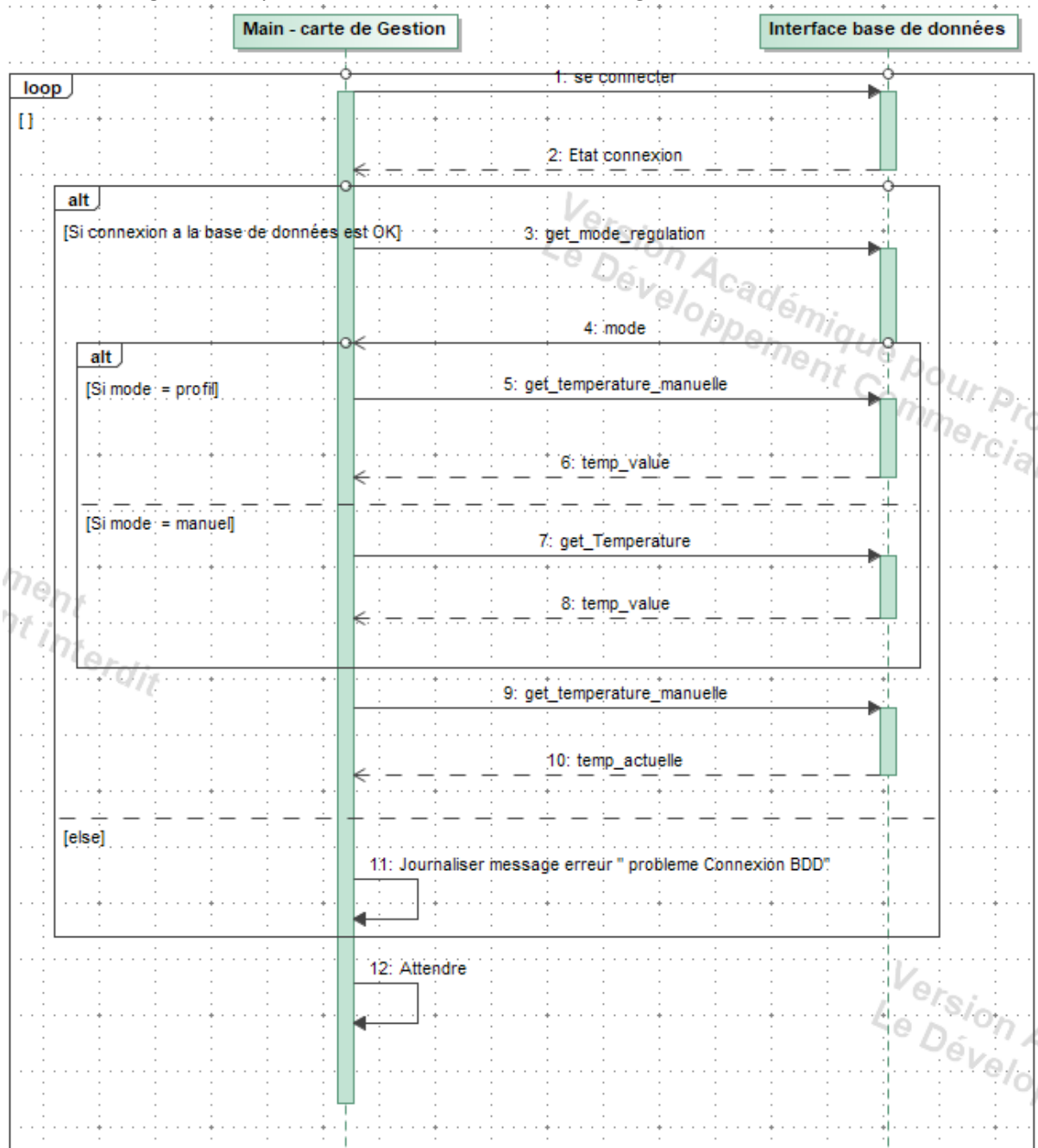


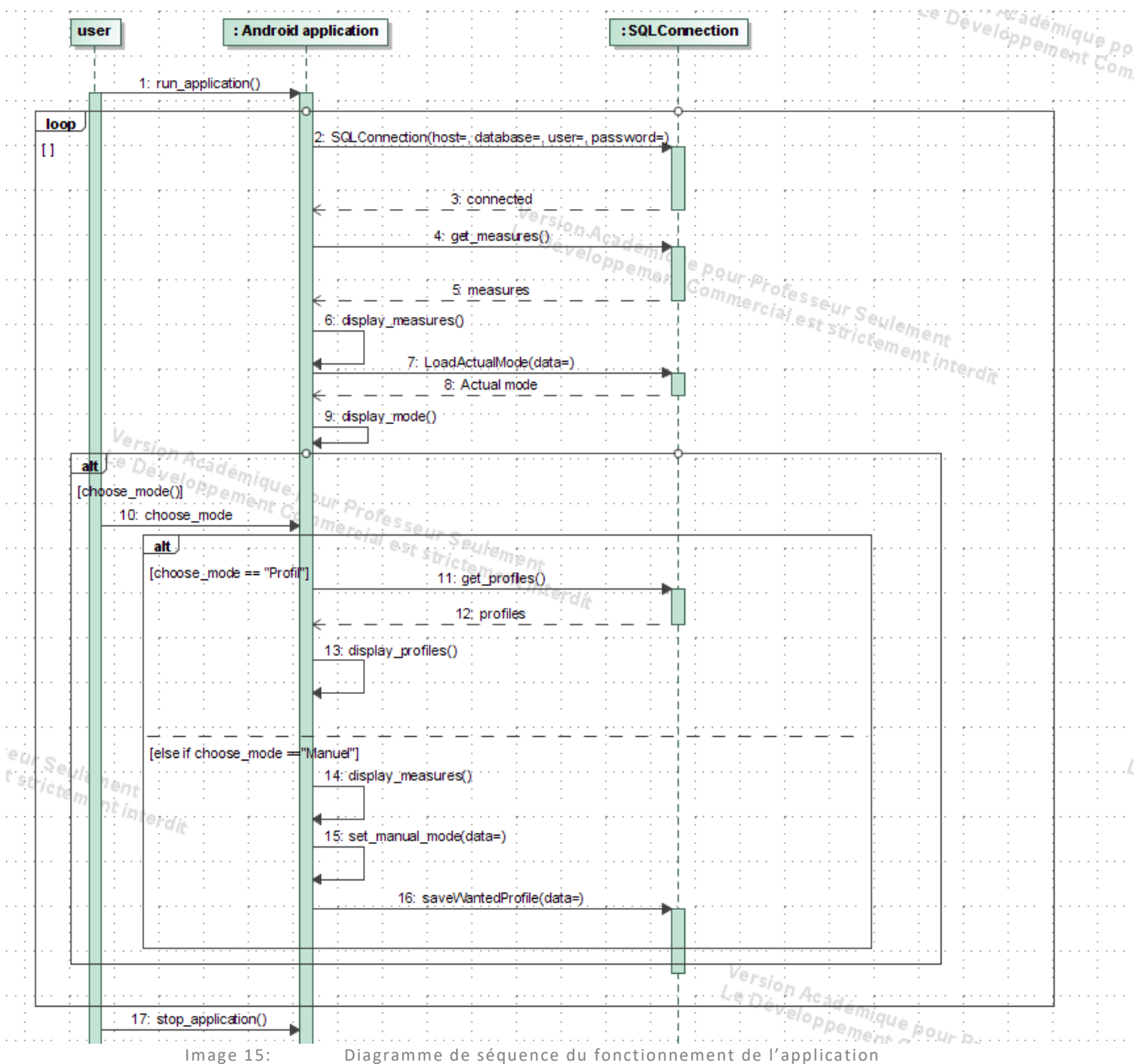
Image 14: Diagramme de séquence connexion à la base de données



Etudiant 3 :

Mise en situation 8 : Fonctionnement de l'application Android

Ce diagramme représente le fonctionnement de l'application, de son lancement à sa fermeture, avec toutes les fonctionnalités. L'application récupère d'abord les mesures, les affiche et charge le mode actuel. On imagine ensuite qu'on veut changer de mode. Si le mode choisi est « Profil » on obtient via la base de données les profils et on les affiche, ce qui permet de les choisir, et si le mode est « manuel » on affiche les mesures et on permet de rentrer les valeurs à atteindre pour les mesures. Avant la fermeture, on sauvegarde les valeurs sous forme de profil.





Mise en situation 9 : Régulation brumisateur

Ce diagramme représente la façon dont l'actionneur va s'activer et se désactiver. La méthode « runRegulation() » de l'application du système de régulation va se connecter au serveur mysql et activer ou désactiver l'actionneur selon le temps de refroidissement de l'actionneur.

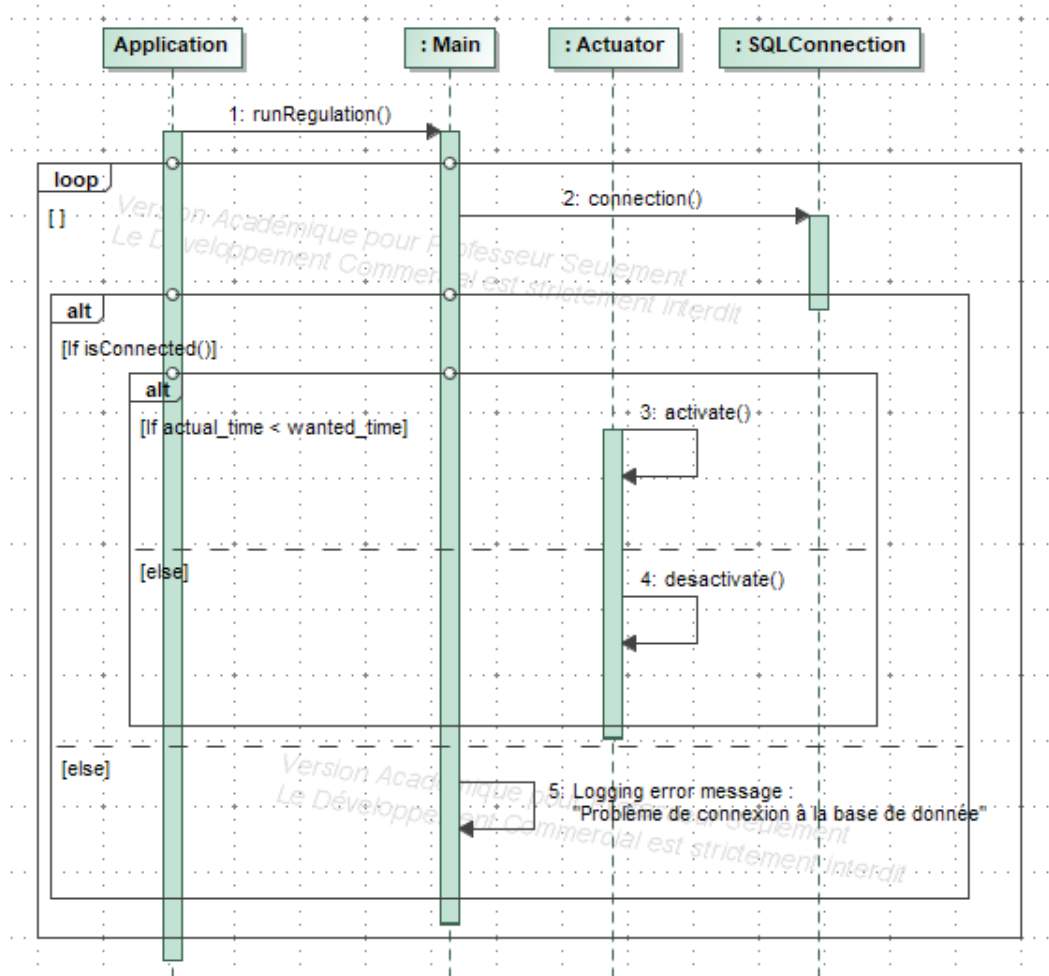


Image 16: Diagramme de séquence du fonctionnement du brumisateur



E. Diagramme de classe

Carte de régulation

Voici le diagramme de classe du programme de régulation. Il possède cinq classes. Nous avons la classe :

- Main : Classe principale, c'est la classe lancée lors du démarrage de la Raspberry.
- SQLConnection : Permet de faire le lien entre le programme et la base de données.
- Actuator : La classe « Actuator » permet de renseigner les différentes informations des actionneurs récupérés dans la base de données.
- Communication : Classe mère avec deux méthodes de visibilité protégées (write et query)
- Serial : Classe fille héritant de la classe mère « Communication ». Elle permet de créer une communication entre la carte Raspberry et le carte Arduino.

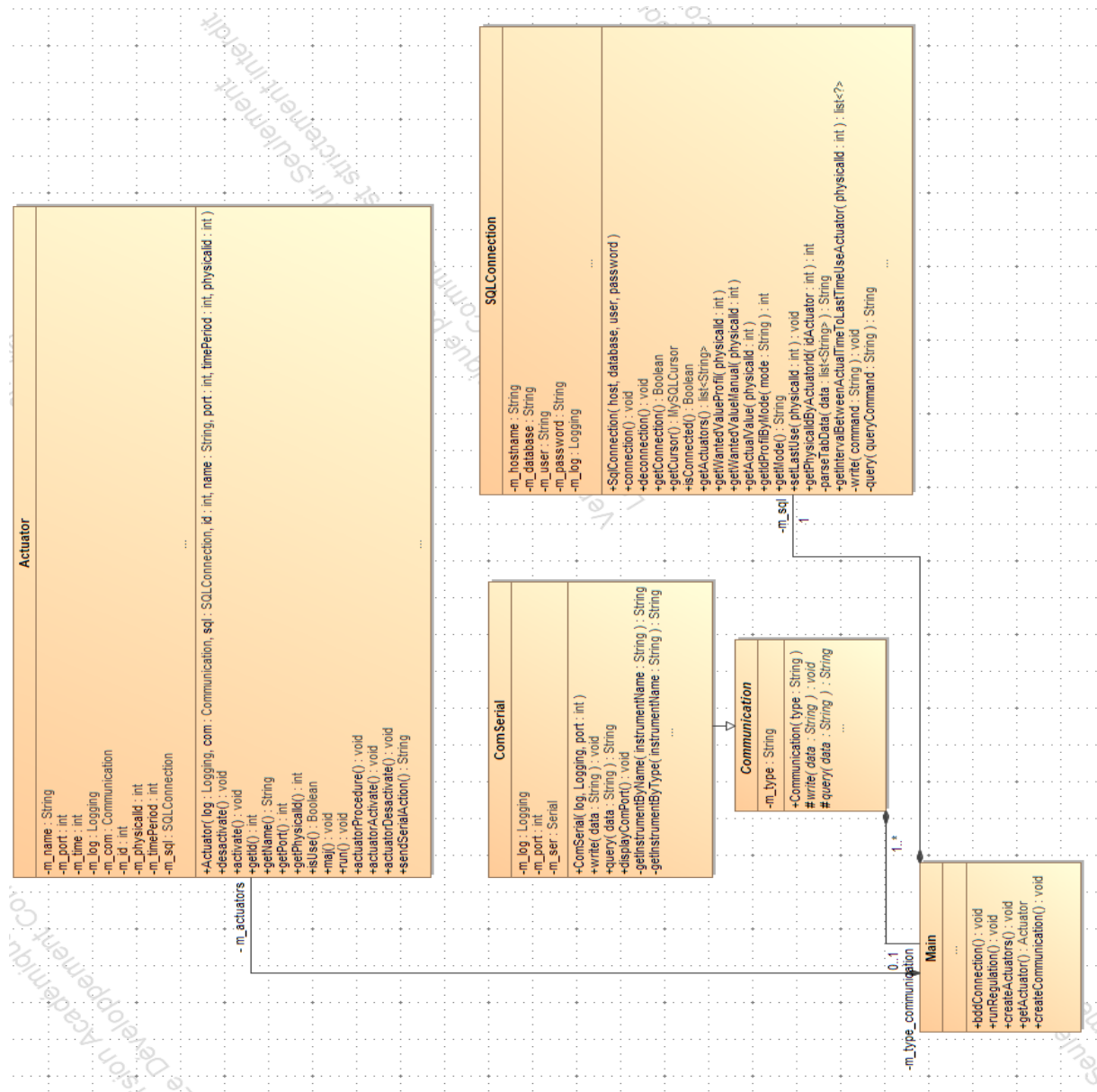


Image 17:

Diagramme de classe de la partie régulation

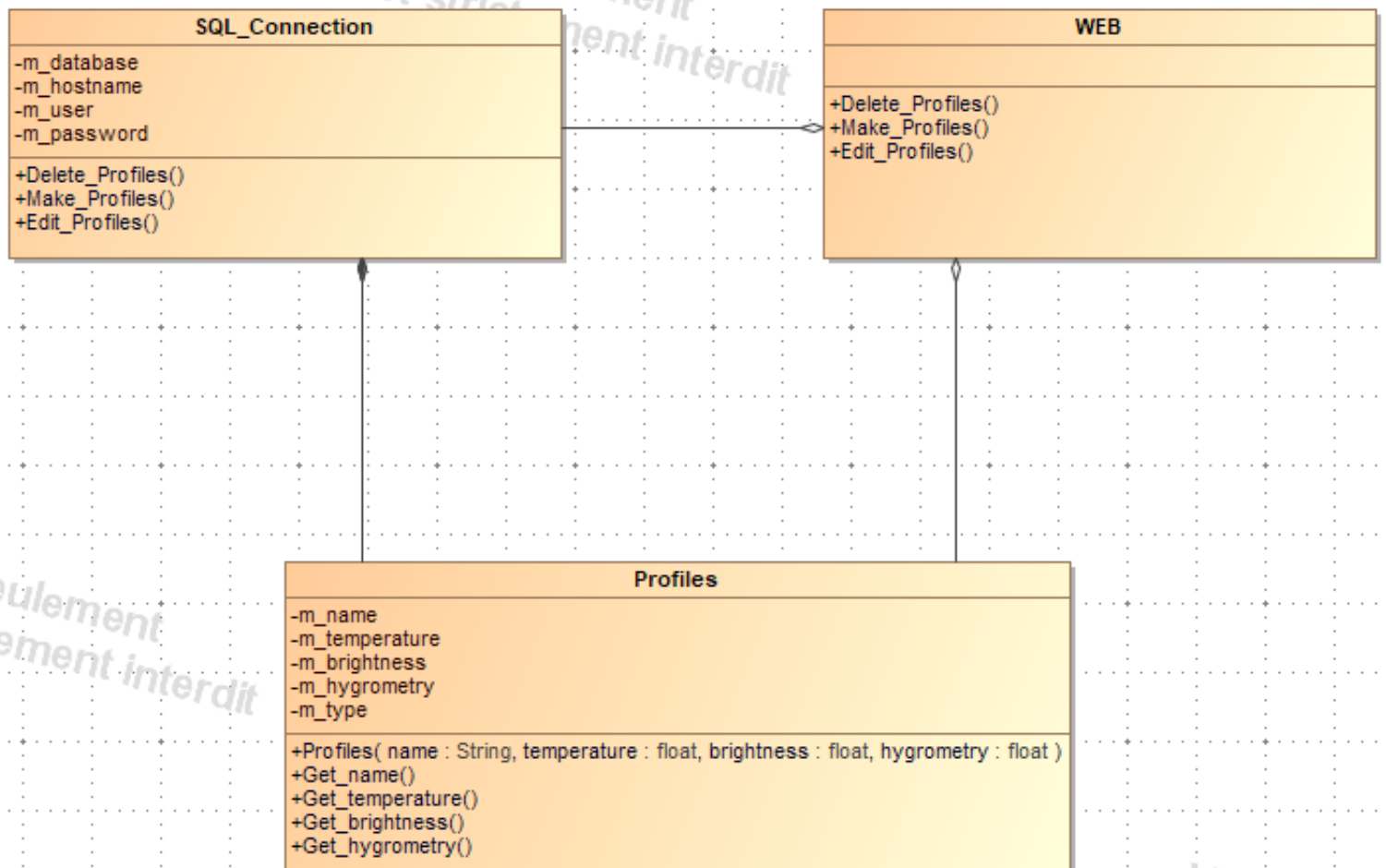


Application Web

Voici donc le diagramme de classe de l'application WEB. Il possède trois classes. Nous avons la classe :

- WEB : Classe représentant le site WEB et ces actions.
- SQLConnection : Permet de faire le lien entre le programme et la base de données.
- Profiles : La classe « Profiles » est une classe représentant les profils présent dans la base de données. Elle servira donc à envoyer les valeur des capteurs sur le site WEB

Image 18: Diagramme des classes de l'application web





Application Android

Voici donc le diagramme de classe du programme Android. Il possède quatre classes. Nous avons la classe :

- Main : Classe principale, c'est la classe lancée lors du démarrage de la Raspberry.
- SQLConnection : Permet de faire le lien entre le programme et la base de données.
- Actuator : La classe « Actuator » permet de renseigner les différentes informations des actionneurs récupérés dans la base de données.
- Android application : Cette classe est la classe centrale de la partie de l'étudiant 3. Elle comprend toutes les méthodes permettant de gérer l'application Android et son bon fonctionnement.

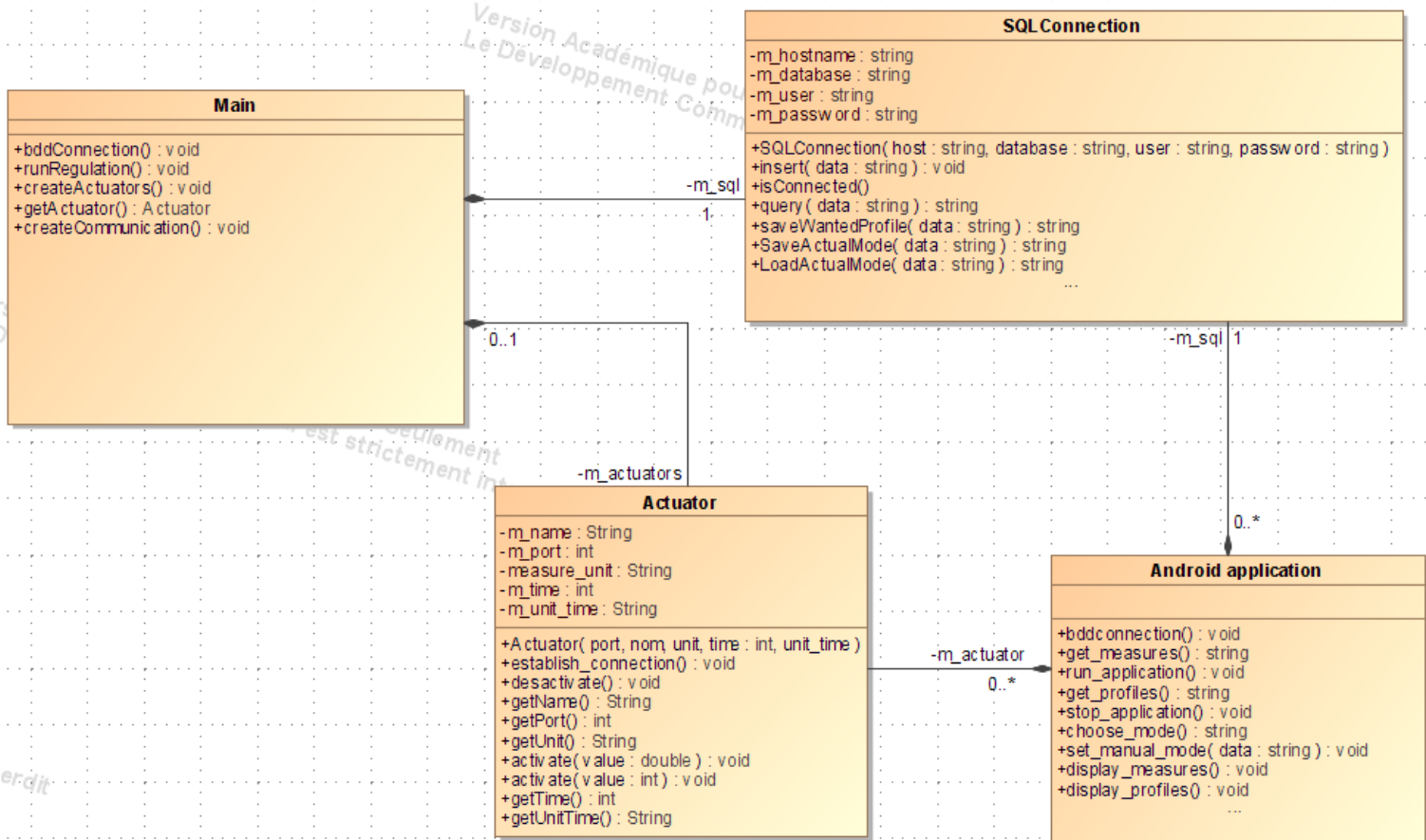


Image 19: Diagramme des classes de l'application Android



F. Diagramme de base de données

Ces différents diagrammes représentent la structure du serveur commun aux deux groupes de projet SFL5 et SFL6 sur lequel nous stockons et exploitons nos données pour le bon fonctionnement du projet.

Modèle entité-association

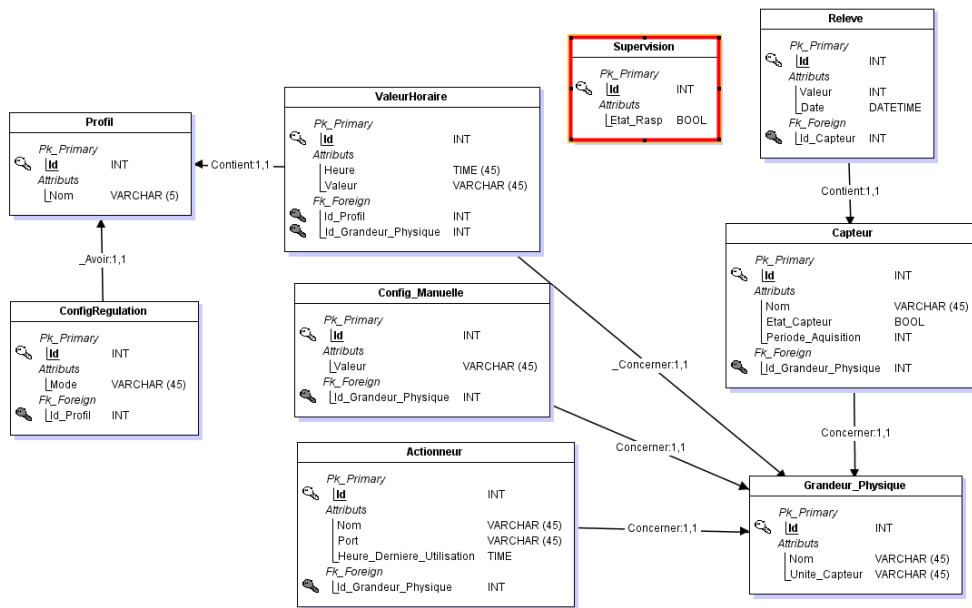


Image 20: Schéma entité-association de la base de données

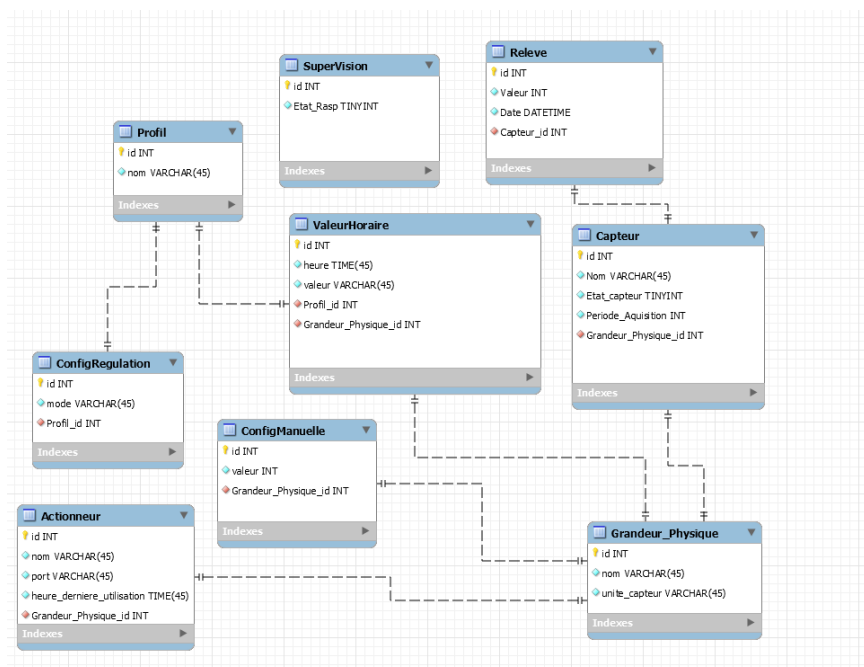


Image 21: Schéma relationnel de la base de données

Modèle
relationnel