

Mission Professionnel

Rapport d'Activité professionnel

Master 1 chef de projet informatique

Modelisation d'un systeme de detection de décrochage de la concentration



Nom du candidat:

- **Djaroud Arezki**
-

Sommaire:

Introduction:	4
Présentation de l'entreprise:	5
La demande (le cahier de charge):	6
Ma Mission:	6
La problématique:	7
Les hypothèses de recherche:	8
La méthodologie utilisée :	9
Analyse Fonctionnel du système:	9
Contrainte du système:	10
Limite du système:	11
Exigences Fonctionnelles:	11
Exigence non fonctionnelles:	13
Besoins Matériels du système:	15
Besoins logiciels du système:	18
Analyse et Spécification des fonctionnalités du système:	22
Les acteurs du système:	23
Les acteurs primaires:	23
Les acteurs secondaire:	24
Les opération du système:	25
Diagramme de cas d'utilisation:	25
Diagramme de séquence et Scénario :	26
scénario avec les acteurs principaux:	26
Scénario avec les acteurs secondaires:	29
Diagramme des classes :	31
Définition des classes et leur description	32
Les contraintes en temps réels:	33
• Contraintes temporelles :	34
Modélisation dynamique des réseaux de capteurs :	34
Langage de modélisation:	34
Concepte de modélisation SysML :	35

Vision dynamique du système:	35
Comportement du système :	35
Contrôles de l'application:	35
Diagramme d'état de transition:	36
Diagramme d'activité:	37
Diagramme de packages :	38
• Les quatres de packages de notre application:	38
• Les relations de contenance entre les packages et les classes:	38
• La relation entre classe implique relation entre packages:	39
• L'ensemble des classes dans leurs packages:	39
Les diagrammes des noeuds et composants de notre systemes:	40
Le diagramme de déploiement:	41
motivation:	42
PAHO et Mosquitto:	42
Le principe de communication du MQTT:	43
Langage ,Librairies et Environnement de travail:	44
Gestion des dépendances dépendance : MAVEN	44
Les commandes de mosquitto:	45
Pour la base de donnée:	46
Les autres apis pour les besoins matériels et leur configuration :	46
Confrontation avec la réalité:	47
Dans le future:	47
Apports et difficultés du projet:	48
Conclusion :	49
Annexe:	50
Bibliographie:	62
Source internet:	62
Livres consultés:	63

Introduction:

L'une des plus grandes problématiques auquel est confronté à ce jour le monde de l'éducation et plus particulièrement les enseignants et enseignantes est de savoir combien de temps leurs étudiants restent concentrés durant leurs cours ? et quels sont les facteurs qui mènent au décrochage ?

C'est d'ailleurs les questions que s'est posé l'entreprise zocus et qui a décidé de créer un système qui peut détecter le décrochage à l'aide d'objets connectés comme des capteurs, mais pour cela il va lui falloir concevoir et imaginer tous les systèmes en créant une équipe IT qui va concrétiser le projet.

Durant la période passée avec l'entreprise ma mission a été d'aider à la conception et la modélisation du projet et cela en mettant à disposition mes connaissances en matière de génie logiciel que j'ai acquis durant ma formation en tant que chef de projet.

Ce rapport contient quatre grands chapitres:

Chapitre 1: Brève présentation de l'entreprise Zocus, ma mission, la problématique, l'idée de départ.

Chapitre 2: Contient les exigences et besoins du système ainsi qu'une analyse des fonctionnalités.

Chapitre 3: Conception du projet et solution préconisée.

Chapitre 4: Conclusion et clôture du projet.

Chapitre 1: l'entreprise et ma Mission

Présentation de l'entreprise:

Zocus est une entreprise de conseil en science cognitif, le coeur des activités de cette dernière consiste à donner à d'autres entreprise ou établissements scolaires des conseils pour améliorer les conditions d'apprentissages et de travail de leurs employés (élèves) et à aider les jeunes sociétés à construire un système et une organisation qui augmente la détermination, la concentration et la collaboration des employés dans l'entreprise et cela en utilisant les sciences cognitives.

Zocus s'est doté il y a peu d'un service informatique dans l'optique d'utiliser les nouvelles technologies IOT pour renforcer sa compréhension du cerveau et des comportements humains, le service est composé de trois équipes qui ont des tâches différentes et d'un responsable informatique, ce dernier s'occupe de manager les projets informatiques de l'entreprise ainsi que de vérifier l'état d'avancements des tâches chez chaque équipe et puis de l'avancement du projet dans sa globalité, il s'occupe aussi de rendre compte du travail effectué aux différents autres services de l'entreprise quand cela est nécessaire mais aussi de rendre des rapports réguliers au CEO sur le travail déjà effectué, ce qui reste à faire, combien de temps cela prendra ainsi que du budget nécessaire au bon maintien du projet; quand aux équipes qui forment le service informatique de l'entreprise sont:

- **Team conception et modélisation** qui va s'occuper de proposer une solution aux problèmes posés et de la modéliser.
- **Team réalisation** qui va s'occuper d'implémenter la solution choisie par la team conception, elle s'occupera aussi des tests unitaires à la fin de chaque projet.
- **Team réseaux** qui va s'occuper de la gestion des réseaux de l'entreprise et aussi du maintien des solutions IT développées.

L'entreprise détient aussi un parc informatique avec un réseau utilisateur qui contient toutes les machines dont a besoin les employées et d'un réseau administratif qui contient les serveurs et bases de données mais aussi d'accès étendu sur le réseau.

La demande (le cahier de charge):

Zocus cherche à développer grâce à son service informatique un système capable de détecter le décrochage des élèves pendant les cours et cela pour consolider son offre de conseil cognitif aux établissements scolaire qu'elle accompagne, voici donc ce qui a été demandé spécifiquement:

- On cherche à identifier les facteurs stimulant l'attention et plus précisément à maîtriser les variables qui produisent un décrochage, que l'on cherche à expliquer.

- Pour cela, on souhaite concevoir une application, qui génère des mesures en temps réel concernant la température ambiante, le bruit, la luminosité, et indiquant le moment de la journée, grâce à des capteurs existants, cette même application enregistre de façon parallèle et/ou simultanée des données qui nous renseignent sur les facteurs internes de l'apprenant : son intérêt, sa motivation, sa condition physique et mentale. Le formateur ou l'enseignant, est renseigné en temps réel sur l'évolution de ces paramètres à travers des données graphiques. Il peut être ainsi averti par un bip ou un signal clignotant lorsque des variables à forte corrélation coexistent, et augmentent les risques de décrochage.

Ma Mission:

Dans le cadre de mon stage chez ZOCUS j'ai du aider à la modélisation et à la conception du système de détection de décrochage, j'ai fait partie de l'équipe de modélisation et j'ai participé au projet jusqu'à la partie implementation

L'essentiel du travail consiste à recevoir des petite mission pour aider l'équipe comme proposer des fonctionnalités aux systèmes, élaborer certains diagrammes de séquence ou d'activités...etc

La problématique:

Afin de cadrer au mieux les dimensions du notre problématique, et obtenir toute les informations suffisantes pour identifier les aspects essentiels, on a opté pour l'utilisation de l'outil QQQCP qui est utilisé avec une démarche d'analyse constructive basée sur le questionnement systématique ci-dessous :

-
- **Qui ?** l'entreprise Zocus et plus particulièrement le service informatique de l'entreprise.
 - **Quoi ?** modéliser et concevoir un système qui lui permettra de détecter le décrochage chez les étudiants et maîtriser les variables qui déclenche ce phénomène.
 - **Où ?** au sein des locaux de zocus et dans des salles de cours des partenaires de l'entreprise.
 - **Quand ?** depuis Mars 2020.
 - **Comment ?**
 - **Comment apparaît le problème ?** Pas de système de détection de décrochage.
 - **Comment résoudre le problème ?** En mettant en place une étude et une modélisation qui respecte le cahier de charge pour concevoir le système.
 - **Comment mesurer le problème ?** Définir les fonctionnalités, besoin et limite du système.
 - **Pourquoi ?**
 - Pour permettre à Zocus d'élargir son offre auprès de ses clients
 - Pour collecter un maximum de données sur la concentration, qu'elle soit interne ou externe
 - Pour maîtriser les variables qui déclenche le décrochage et le signaler si ça se produit.

Les hypothèses de recherche:

Pour concevoir aux mieux le système qui nous a été demandé de faire et pour qu'il soit le plus fidèle à la commande on a décidé de faire quelque recherche sur comment collecter des données externe comme le bruit la luminosité et la température et interne comme la motivation et l'intérêt, ensuite on a décidé de faire des réunion pour que chacun expose une idée et que ensuite on essaye de choisir la meilleur, cela en apportant des amélioration et modification nécessaire, et les hypothèses qui sont sortie de cette entrevue sont:

- Pour collecter les données des bruits ambiants et de la luminosité on pourrait utiliser les capteurs de nos smartphone et pour la température on la prendrait directement sur internet.

-
- Un autre moyen pour collecter ses données serait d'utiliser des capteurs spécialisés qu'on placera de manière adéquate dans la salle de cours pour avoir des prises de données homogène et non biaisées.
 - Une dernière idée sur la collecte des donnée qui a été proposé c'était de développer un produit qui engloberait les capteurs nécessaire au projet et qu' on pourrait déplacer partout pour une analyse des donnée a toute endroit mais cela n'est pas vraiment l'idéale car on perdrait de la précision sur les données collecté et ce que veut zocus en premier lieux c'est un système très précis et le coup pour développer des capteur puissant et petit serait trop importante.
 - Pour le temps réel on va devoir envoyer les données pris par les capteurs à intervalle régulier c'est pour cela que j'ai proposé d'utiliser des capteur doté de microcontrôleur pour justement assurer le temps réel d'envoi de donnée.
 - pour gérer les données qui sont envoyée au système et aussi pour gérer tous l'environnement IT il a été proposé d'utiliser un middleware qui nous facilitera la tâche
 - Pour la collecte des données interne des étudiants il a été proposé de soit utiliser un formulaire qui doit être rempli par ces derniers afin d'avoir leurs ressenti avant le début du cours ou bien mettre en place une marge d'erreur due à la complexité d'interprétation de données comme des sentiment.
 - Il est paru clair qu'il allait falloir mettre des règles d'accessibilité et cela en mettant en place un système d'authentification à l'application et selon le statut de l'utilisateur il aura des droits différents (droit qu'on détaillera dans l'analyse du système).

En fin de compte et après avoir exposés et étudié toutes les hypothèses mentionnées ci-dessus on a choisi de faire un système avec des capteurs doté de microcontrôleur pour l'aspect temps réel système et l'interprétation de donnée brute en données numérique, capteurs qui seront déposés de manière stratégique dans la salle de classe pour avoir des données homogène, on se fera aider d'un middleware pour pouvoir facilite la gestion de l'environnement IT, bien sur on utilisera une base de donnée à temps réel pour stocker les données qui arrive à intervalle régulier et enfin l'accès au système (l'application) passera d'abord par une procédures d'authentification.

La méthodologie utilisée :

Pour réussir à concevoir notre système, on a dû d'abord choisir une méthode de travail et après mûre réflexion on a choisi le cycle en V. le cycle en v est un modèle d'organisation des activités d'un projet qui se caractérise par un flux d'activité descendant qui détaille le produit jusqu'à sa réalisation, et elle contient une phase de conception puis de réalisation et enfin de validation (test unitaire...)

Cette méthode nous convient parfaitement car elle nous permette d'avoir une bonne visibilité sur le projet et donne la possibilité de revenir aux étapes précédentes, en plus du faite qu'on réalise pas le projet pour une entreprise externe mais au contraire c'est un projet interne à l'entreprise, qui est très spécifique et particulier, ce qui est parfait pour utiliser le cycle en V.

Chapitre 2: Analyse du système

Analyse Fonctionnel du système:

Dans cette partie nous allons d'abord voir les différentes contraintes (qu'on a relevées) liées à notre système et par la suite établir la liste des exigences fonctionnelles et non fonctionnelles de notre système et au final voir les différents besoin matériel et logiciel dont notre système aura besoin pour assurer ses fonctionnalités.

Contrainte du système:

Notre système devra bien sûr répondre à certaines contraintes très importante pour assurer son bon fonctionnement comme:

faire un système qui marchera en temps réel c'est à dire qu'il devra générer des mesure en temps réel grâce à ses capteurs et les envoyer au système de calcul qui les traitera instantanément pour les afficher sur l'afficheur et être consulté par le professeur a

chaque moment du cours, et aussi activé le bipeur dès que le système de calcul remarque un décrochage de la part d'un étudiant.

Il faut aussi mettre des contraintes d'accès utilisateur et sécurité comme:

- premièrement un compte et un seul compte seulement par utilisateur et qui devra être accessible que par la personne concernée en protégeant les comptes avec des mots de passe secrets.
- Deuxièmement, les résultats du système de calcul ne peuvent être consultés que par le professeur et non par les étudiants.
- Une autre contrainte serait que les données collectées ne soient traitées que par le système de traitement et non pas par le professeur qui lui n'a accès qu'au résultat du traitement sous forme de graphique.
- Une dernière contrainte serait que le bipeur ne soit activé que par le système de calcul quand celui-ci prédit un décrochage.

Il faut aussi que notre système est des classes et objets actifs, cela veut dire que nous devons avoir des classes qui initient et contrôlent le flux d'activités comme les classes des différents capteurs qui initient la collecte de données extérieures ou l'étudiant qui initie la collecte de données internes, le système de calcul qui contrôle les données collectées et les traite pour donner des résultats graphiques au professeur.

Une dernière contrainte serait d'éviter la concurrence des états de notre système comme faire en sorte que le système de calcul ne commence le traitement de données qu'après que le capteur et le formulaire les ont collectées et envoyées au traitement et aussi que l'afficheur n'affiche les résultats qu'après que le système de calcul a envoyé les résultats du traitement à "Afficher".

Limite du système:

On peut dire que le système est limité car certaines informations ne sont pas forcément fiables. Un étudiant qui remplit le formulaire peut ne pas être capable de répondre correctement aux questions, ni même décrire correctement leurs états. Un capteur défaillant peut fausser tous les résultats et donc aboutir à une conclusion

fausse. Ici, l'autonomie des capteurs est également très importante. Elle doit être assez puissante pour que le capteur soit opérationnel pendant tout le cours.

Exigences Fonctionnelles:

Ce sont les exigences définissant une fonction du système à développer, entre autres ce que le système doit faire. Pour notre système de détection de décrochage on distingue les exigences suivantes:

- Le système doit pouvoir recevoir les données des capteurs (objet connecté) et cela de manière régulière (par intervalle de 15 seconde environ), et pour ce faire les capteurs doivent être connectés au système de calcul et plus précisément au système IOT (middleware) grâce à un protocole de communication (pour notre système on a choisi le bluetooth) qui devra être constant et sans interruption.
- Le système doit procéder à une authentification avant d'autoriser l'utilisateur à entrer, ce qui se déroulera ainsi:
 1. Tout d'abord l'utilisateur qui va se connecter aura des droits différents selon son type, si c'est un étudiant (après activation du compte) il n'aura droit que de recevoir un formulaire pour le remplir et l'envoyer au système de calcul. Il peut aussi demander à corriger les informations remplies s'il estime qu'il s'est trompé quelque part (car les questions peuvent être très subjectives vu qu'on parle de l'état émotionnel de la personne au moment où elle remplit le formulaire et qui peut changer tout au long du cours). Si c'est un professeur, il sera considéré comme un administrateur il aura donc tous les droits. Notamment le droit d'activer les comptes des étudiants qui viennent d'arriver et de désactiver les comptes qui ne sont plus valides(pour les étudiants qui ne sont plus dans le cours du professeur par exemple). Il aura aussi le droit de consulter les résultats du traitement des données sur l'afficheur et de réagir en conséquence.
 2. Ensuite pour s'authentifier au système, il faudra renseigner son identifiant qui sera unique pour chaque utilisateur (l'étudiant le

recevra après activation du compte par le professeur), et un mot-de-passe que l'utilisateur aura défini (pour des raisons de sécurité le mot de passe devra contenir certaines propriétés que l'on détaillera plus tard).

- Le système doit assurer une interaction en temps réel avec l'environnement et pour cela il doit disposer de quelques fonctionnalités qui assurent cette interaction à temps réel:
 1. Premièrement, il faut que les données envoyées par les capteurs (objet connecté) soient envoyées dans une intervalle régulier (toutes les 15 seconde environ), mais aussi de manière parfaitement synchronisé vers le système de calcul à travers le serveur, cela veut dire que les trois capteurs doivent mesurer et envoyer les données au même moment.
 2. Deuxièmement, le système de calcul doit être opérationnelle tout le temps sans aucune interruption dès l'ouverture de la session administrateur, car même si les capteurs n'envoient les données de mesure que par intervalle régulière, il y a aussi les étudiants qui envoient les formulaires remplis et qu'il faut aussi traiter mais les formulaires seront envoyés de manière aléatoire ce qui fait que le système de calcul doit toujours être opérationnelle.
 3. Troisièmement, les capteurs et le système de calcul doivent être continuellement connectés grâce à la plateforme IOT pour assurer l'envoi des données et les traiter avant l'envoi des prochaines données par ces mêmes capteurs.
 4. Quatrièmement, le serveur doit être opérationnel tout le temps pour qu'il puisse assurer la acheminements des opérations du système.
- Le système doit pouvoir interpréter les données envoyées par les capteurs et les données contenues dans les formulaires remplis par les étudiants, et cela grâce au système de calcul qui va traiter les données de manière synchronisée et grâce à des algorithmes de prédiction de décrochage, et aussi des algorithmes d'apprentissage.
- Le système doit pouvoir éviter la concurrence des états c'est-à-dire faire en sorte que le système de calcul ne commence le traitement de données qu'après que les capteurs les aient collecté et envoyé au traitement et aussi que l'afficheur n'affiche les résultats qu'après que le système de calcul ait envoyé les résultats du traitement à "Afficher".

-
- Le système doit utiliser des algorithmes spéciaux qui donnent au système de calcul des données sur l'heure en indiquant le moment de la journée et aussi quand commence et quand finit le cours du professeur.

Exigence non fonctionnelles:

Nous allons établir maintenant les exigences qui caractérisent des propriétés (qualités) désirées du système tel que la sécurité et la fiabilité... du système et nous avons établi les exigences suivantes:

- Une première exigence fonctionnelle que nous devons satisfaire avec notre système c'est que les capteurs doivent être continuellement et sans interruption alimenté en courant pour qu'il puisse collecter et envoyer les données au système de calcul et cela à intervalle régulier et sans interruption (chaque 15 seconde et pas une de plus).Et cela en:
 1. Prévoyant une batterie de secours qui pourra encore alimenter les capteurs pour une durée de 2 heures environ sans interruption, au cas où il y aurait une panne de courant ou si les capteurs se font débrancher pour une raison ou une autre.
 2. La prise en main de l'alimentation par cette batterie en cas de panne de courant ou autre devra être instantané (en vrai il y a une micro coupure) et cela en utilisant des capteurs avec ce type de batterie donc avec une alimentation GTZ (alimentation Zéro-temps).
 3. Et pour s'assurer que les batteries soient toujours pleines il faut qu'elles se rechargent automatiquement après chaque panne de courant (donc après chaque utilisation de la batterie).
- Une deuxième exigence se porterait sur la fiabilité du système, cela veut dire qu'il faut assurer le bon fonctionnement du système, sans défaillance dans des conditions données et pour une période de temps donnée " T ", pour cela on pose les conditions suivantes:

-
1. Comme on l'a dit plus haut, on doit s'assurer que le système soit continuellement alimenté en énergie et cela en utilisant des capteurs dotés d'une batterie de secours qui alimente le système en cas de coupure d'alimentation et cela pour une période de deux heures, mais pour assurer de la fiabilité de ce processus il nous faut avoir des capteurs avec une batterie très fiable qui garantit une période de fonctionnement assez élevé (garantie de 2 ans environ) avec peu de perte d'énergie si l'on ne s'en sert pas.
 2. Il faudra aussi avoir un système d'alimentation assez fiable, c'est-à-dire qu'il n'y ait pas beaucoup de coupure de courant et ces coupures ne doivent pas durer plus de 2 heures car cela viderait complètement la batterie de secours et on aurait une perte d'énergie et le système s'arrêterait de fonctionner.
 3. Un autre point important pour assurer une bonne fiabilité de notre système c'est l'environnement dans lequel il est utilisé, en effet il nous faut un environnement propice au bon fonctionnement du système, comme par exemple placer les capteurs au milieu de la salle pour avoir des mesures de luminosité et de température homogène et plus fiable ou encore éviter tout type d'interférence qui pourrait brouiller la communication entre les capteurs et le système de calcul, et avoir un environnement très bien isolé pour éviter les bruits extérieurs qui pourraient biaiser les mesures du capteur de bruit.
- Une troisième exigence est que les données des capteurs doivent être fiables et sécurisées. Elles ne doivent pas être biaisées lors des transferts ni affectées par les paramètres extérieurs hors de la salle.
 - Une dernière exigence non fonctionnelle du système est portée sur la sécurité des communications et de l'accès aux données du système et cela en établissent quelque règle de sécurité de base au système:
 1. l'utilisation d'un protocole de communication qui soit sécurisé, dans notre cas on utilisera le bluetooth qui s'impose aujourd'hui comme le choix le plus évident dans le développement d'un système connecté, car il a une très grande portée, peut gérer une très grande quantité de données avec une

grande rapidité de transfert, et une sécurité optimale (connection possible qu'avec un mot de passe).

2. un étudiant ne pourra accéder à son compte que si le professeur active le compte en question.
3. l'étudiant ne recevra son identifiant qu'après activation du compte, et il devra être unique et stocké dans la base de données.
4. Le traitement des données est fait par le système de calcul et par aucun autre acteur.
5. Les résultats du traitement de données ne sont accessibles que par le professeur grâce à l'afficheur.
6. Le système de calcul doit être le seul à activer le bipleur s'il constate un décrochage après traitement des données.

Besoins Matériels du système:

Parmi les choses dont nous avons besoin pour le bon fonctionnement de notre système il y a les besoins matériels et parmi eux on trouve:

Les capteurs et microcontrôleur:

On va devoir utiliser trois types de capteur physique qui prendront des mesures différentes et qui sont le bruit, la lumière ambiante, et la température de la salle, et ces capteurs devront tous être sur la même base pour avoir des données prises dans les mêmes conditions et qui soient cohérentes. Il faudra aussi que les capteurs aient une batterie de secours qui prendra la relève instantanément (zéro temps mort) en cas de coupure de courant. quant à l'emplacement, les capteurs devront être fixés sur le plafond et au milieu de la salle pour garantir l'homogénéité des données collectées et limiter au maximum les risques de perte d'informations, vu que à cet emplacement là, les capteurs couvrent mieux la surface de la salle, enfin les capteurs devront être munis d'un récepteur de réseaux Bluetooth pour qu'on puisse les connecter à la plateforme IOT et qu'ils puissent envoyer les données collectées.

On précise aussi que ses capteurs sont des objets connecté vue qu'il peuvent se connecter au réseau bluetooth pour envoyer les données à la plateforme IOT à travers le serveur, et pour cela ils ont besoin d'un **Microcontrôleur** qui est un circuit intégré qui rassemble les éléments essentiels d'un objet connecté comme le processeur, les mémoires (morte et vive), unités périphériques et interfaces d'entrées-sorties et qui transforme des informations telles que la lumière ou le vent en données informatiques utiles pour un système de calcul.

le Microcontrôleur qu'on a décidé d'utiliser pour notre système est le microcontrôleur ultra-basse consommation ADuCM4050 qui intègre une mémoire SRAM étendue et une mémoire flash pour permettre une prise de décision localisée et faire en sorte que seules les données les plus importantes soient transmises vers le cloud, et qui peut rester en mode basse consommation lorsque les capteurs procèdent à la collecte des données. Il peut ainsi multiplier par plus de 10 les économies d'énergie au niveau système, ce qui rallonge la durée de vie de la batterie.

L'afficheur:

Pour que le professeur puisse consulter les résultats du traitement des données fait par le système de calcul, il doit disposer d'un afficheur qui peut être l'ordinateur de bureau qu'il utilise dans la salle de cours ou une tablette dédiée, et il faut absolument que l'afficheur en question ait un accès internet pour que le professeur puisse se connecter à son compte.

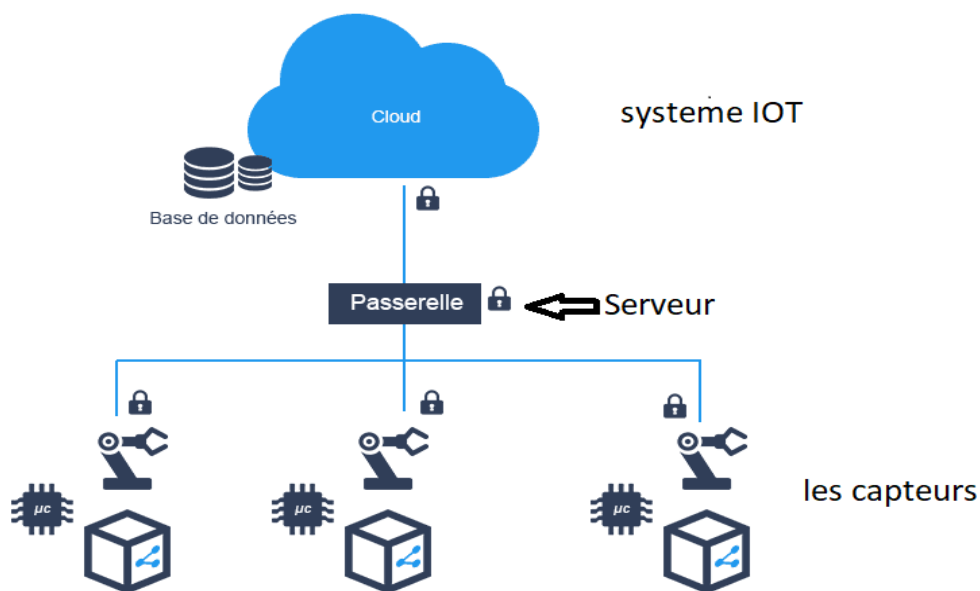
PC et autres périphériques:

Cela nous sera utile pour que les étudiants puissent se connecter à leur compte et recevoir le formulaire qu'ils doivent remplir, et ça sera aussi utile au professeur pour se connecter et pouvoir activer les comptes des nouveaux étudiants.

Le serveur:

Est un dispositif informatique qui va nous offrir plusieurs services comme: la gestion des périphériques, la gestion d'authentification et du contrôle d'accès, et un accès au WEB.

Du coup c'est grâce au serveur qu'on va pouvoir intégrer un système d'authentification, et il va faire aussi faire office de passerelle entre les objets connectée (les capteurs) et le système IOT (middleware) et la SGBD, comme le montre le schéma ci-dessous:



Objet connecté:

Un objet connecté est un appareil informatique capable de transmettre ou recevoir des données via un réseau de communication avec un autre objet ou avec tout un réseau d'objets connecté.

Dans ce projet notre objet connecté est divisé en trois parties: capteur, microcontrôleur et carte réseau; qui sert dans l'ordre à capter des mesures de température, bruit et luminosité, puis à les convertir en données utilisables par le

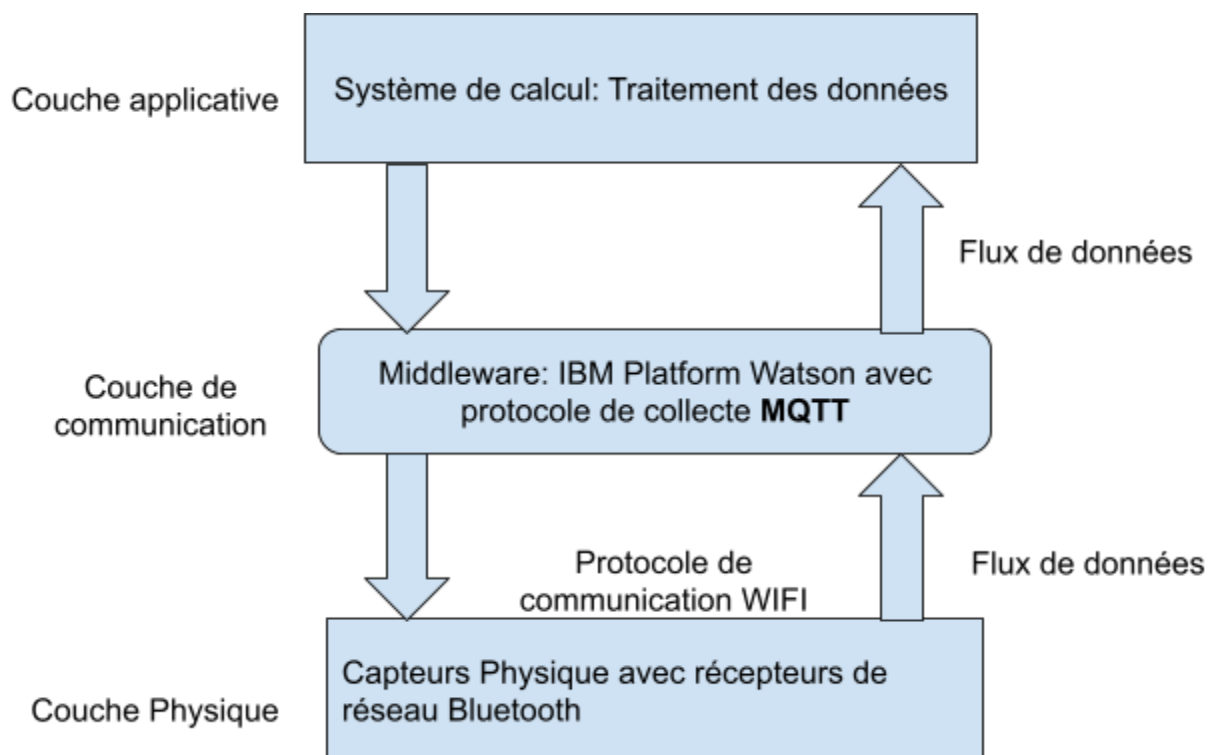
microcontrôleur et enfin les données créées sont envoyées à la plateforme IOT grâce à la carte réseau.

Besoins logiciels du système:

Nous allons voir dans cette partie les différents logiciels et protocoles de communication que nous devons utiliser pour assurer les fonctionnalités de notre système, et tout d'abord nous allons commencer par: la plateforme IOT que nous avons choisie pour représenter notre système de calcul et base de données.

IBM Platform Watson:

Ce logiciel est un middleware il va jouer le rôle d'intermédiaire entre les capteurs et le système de calcul, il va nous permettre de faciliter la connexion des capteurs au réseau et de faciliter la création d'applications de gestion et de contrôle des capteurs, et donc de créer un écosystème IOT plus facile à gérer.



Pourquoi a-t-on choisi celui-ci ?

On a choisi IBM Platform Watson car il propose une gestion complète des périphériques qu'on souhaite faire communiquer entre eux, et qui utilise MQTT comme protocole de collecte des données (celle du formulaire entre autres), et tout ça gratuitement pour l'abonnement de base.

Arduino:

Arduino est un logiciel de programmation des modules informatiques ex: objet-connecté, dont l'interface graphique est appelé **Arduino IDE**, qui est une application multiplateforme servant d'éditeur de code et de compilateur, elle nous sera utile durant notre projet pour dire au capteurs de prendre des mesures régulières de température, bruit et luminosité et ensuite de les convertir en données utile grâce au microcontrôleur et enfin de les envoyer à la carte réseau qui enverra ses données au système, sachant que les programmes qu'on devra concevoir devront être en langage C/C++ vu que c'est les deux seuls langages reconnus par cette IDE.

Malheureusement à cause de la crise que nous vivons actuellement on ne pourra pas utiliser cette solution pour capter nos données vu qu'on ne dispose d'aucun capteur, à la place on va simuler ses données avec des simulateurs de données (pour la phase de test) mais on approfondira plus cette démarche dans la partie conception et implémentation.

Un exemple de code pour les capteurs de température:

_____ Pour cet exemple, Je propose la création très simple d'un capteur de température, piloté avec un Arduino Uno et nous utiliserons l'entrée Analogique **A0** de l'Arduino Uno LM35.

/ Capteur de température

```
void setup() {  
  
    // Initialisation du moniteur série  
  
    Serial.begin(9600);
```

```
}

void loop() {

    // Lecture de la valeur sur la broche analogique A0

    int valeur = analogRead(A0);

    // Calcul de la température en degré Celsius

    float temperature_celsius = valeur * 100.0 / 1023.0;

    // Affichage de la température sur le moniteur série

    Serial.print("Temperature : ");

    Serial.print(temperature_celsius);

    Serial.println(" Celcius");

    Serial.println("-----");

    // Delais de 250 millisecondes

    delay(250);

}
```

Protocole de communication:

_____ Nous allons utiliser pour notre système le réseau Bluetooth pour connecter les capteurs à notre middleware, car le bluetooth est un protocole de communication à courte distance ce qui est parfait pour notre système vue que ce réseau peut couvrir toute une salle de cours, en plus de ça il est très facile à mettre en oeuvre dans un système IOT car c'est une technologie qu'on maîtrise dans le domaine des objet connecté étant donné qu'elle est sortie il y'a 26 ans, en plus du faite qu'il est adaptée au petit bloc de données ce

qui est encore une fois très pratique pour notre système vue que les capteur n'envoie que des mesure de température, bruit et luminosité.

Il est important de souligner qu'on veut utiliser la version BLE du bluetooth qui consomme très peu d'énergie, le petit problème qu'on peut rencontrer avec le bluetooth c'est les problèmes de sécurité mais vue qu'on fait un système en locale seulement ça ne devrait pas poser de problème.

Protocole de collecte:

_____ Pour le protocole de collecte vue qu'on utilisera IBM Platform Watson on va donc utilisé MQTT qui est un protocole de messagerie publish-subscribe c'est à dire que les message émis par l'émetteur n'arrive pas a un unique destinataire mais a une catégorie associé au message comme les hashtag, et ce protocole est basé sur le protocole TCP/IP qui garantie l'arrivé du message au destinataire ce qui fait de MQTT un très bon protocole pour notre système car il garantit l'arrivé des données au système IOT (middleware) via le serveur.

Question implementation nous utiliserons une des nombreuses bibliothèques disponibles pour programmer des clients MQTT (ex: la bibliothèque Paho), pour la plupart en langage JAVA ou JavaScript, sur la plateforme Arduino IDE.

Application mobile:

_____ Dans notre système nous aurons besoin d'une application mobile qui va servir de passerelle entre l'utilisateur et le serveur, grâce à cette appli les étudiants pourront se connecter au serveur ou créer un compte puis recevoir le formulaire à remplir et l'enseignant pourra s'authentifier à son compte pour accéder à l'afficheur et au compte à activer.

Sachant que le protocole de collecte utilisé pour notre application est le MQTT qui se charge d'envoyer les informations remplit au serveur qui va fonctionner comme un broker de message (courtier a message) et ce protocole est le mieux adapté car une session MQTT est divisée en quatre étapes : connexion, authentification, communication et

terminaison ce qui constitue les quatre étapes qu'on souhaite intégrer dans notre application.

Page du Formulaire:

_____ Quand un étudiant qui va se connecter à son compte le serveur lui envoie une page web avec le formulaire à remplir et a envoyé au système IOT, et le formulaire contient des demandes de renseignements sur l'état interne de l'étudiant (très subjectif), et aussi des informations un peu plus concrètes, et le formulaire se présente ainsi. Voir [Annexe 1](#)

Analyse et Spécification des fonctionnalités du système:

Pour analyser et spécifier les fonctionnalités de notre système au repos nous allons identifier les acteurs qui vont interagir avec ce dernier, puis on va établir un diagramme de cas d'utilisation qui nous donnera une vision globale des fonctionnalités du système, et enfin nous allons écrire quelques scénarios possibles durant l'utilisation du système avec les pré et post-conditions et les diagrammes de séquence correspondants.

Quant à l'analyse du système en activité nous allons établir des diagrammes d'activités et d'états de transition qui nous donneront l'enchaînement de certaines fonctionnalités et transitions d'état de système.

Les acteurs du système:

Maintenant nous allons définir les différents acteurs qui modélisent le type de rôle joué par une entité qui interagit avec notre système.

Dans notre système une interaction représente un échange de données et d'informations entre deux entités, et les acteurs sont toujours externes au système. Un acteur peut représenter le rôle joué par un humain ou un robot (ex: un sous-système, un appareil informatique...etc), et dans notre système, on distingue deux types d'acteurs:

Les acteurs primaires:

Ce sont les acteurs qui initient une interaction ou opération au sein du système (les objets déclencheurs du traitement autonomes) et cela peut être un acteur humain ou non-humain, et dans le cas de notre système on peut distinguer les acteurs primaires suivant:

Les capteurs(acteur non-humain): qui font en sorte de prendre des mesures différentes selon le type de capteur, de les collecter et de les envoyer au système de calcul (sous forme de signaux) pour qu'elle soit traitée, à savoir que ses données sont collectées et envoyées de manière synchronisée entre les capteurs et par intervalle régulière (chaque 15 seconde), et ces capteurs qu'on va utiliser sont en nombre de trois:

- ❑ capteur de température pour les mesures de température ambiante
- ❑ capteur de bruit pour les mesures de bruit
- ❑ capteur de lumière pour des mesures de luminosité ambiante.

L'utilisateur(acteur humain): cet acteur initie l'opération d'authentification et de création d'un compte utilisateur et on dénombre deux sortes d'utilisateur qui ont des droits différents selon le type de compte auquel ils peuvent accéder:

- ❑ L'enseignant qui est aussi considéré comme l'administrateur du système, a comme tâche après authentification d'activer les nouveaux comptes étudiant et de consulter les résultats du traitement sur l'afficheur.
- ❑ L'étudiant quant à lui après authentification va recevoir un formulaire qu'il doit remplir et envoyer au système de calcul pour qu'il traite les données saisies par l'étudiant et cela pour chaque authentification, mais bien sûr si le compte a été activé au préalable par le professeur.

Les acteurs secondaire:

Contrairement aux acteurs primaires, ils n'ont pas la possibilité de commencer les interactions avec le système, ils attendent d'abord l'arrivée des signaux particuliers pour commencer leur action, dans notre système les acteurs secondaires attendent

généralement l'arrivée des données (traité ou pas) pour commencer leur interaction ou bien serve juste à stocker ces dernières

Le système de calcul(acteur non-humain): Cette acteur-là attend d'abord l'arrivée des données collectés par les capteurs ou les données saisies sur le formulaire par l'étudiant avant de commencer à les traiter avec des algorithmes de prédiction et d'apprentissage, puis après traitement les envoie à l'afficheur, ou parfois quand il détecte un potentiel décrochage il active le bipteur pour prévenir le professeur de cela.

La base de donnée "SGBD" (acteur non-humain): Quand à elle va servir à stocker les résultats des traitements qu'effectue le système de calcul pour qu'il puisse les réutiliser pour les traitements suivants, car on l'a dit plus haut le système de calcul utilise des algorithmes de prédiction qui s'appuient sur les données déjà traitées et connues, en plus pour afficher les résultats du traitement sur l'afficheur on a besoin de prendre ses données de la SGBD.

L'afficheur(acteur non-humain): il va servir à afficher les résultats du traitement fait par le système de calcul et comme dit plus haut ne pourra être consultable que par le professeur grâce à l'application.

Le système IOT(acteur non-humain): qui va servir d'intermédiaire entre les capteurs physiques et le système de calcul. En vérité, ce sera lui qui va collecter les données mesurées par les capteurs grâce à un protocole de communication.

Le serveur: qui va permettre à plusieurs utilisateurs d'accéder au système en même temps (gestion d'authentification), et aux capteurs d'envoyer les données au middleware (il a le rôle de passerelle).

Les opérations du système:

Nous allons maintenant voir les différentes opérations que peut effectuer notre système, et qui sont:

- Le système peut effectuer des mesures de bruit ambiant, de lumière ambiante et de température grâce aux capteurs.

-
- Une autre opération c'est la création de compte par un utilisateur qui peut être un professeur ou un étudiant.
 - Opération d'authentification au système par un utilisateur, opération qui n'est possible qu'après création du compte.
 - L'opération de remplissage d'un formulaire par un étudiant et l'envoyer au système à travers le serveur.
 - Le traitement des données est l'une des opérations les plus importantes car c'est au niveau du traitement que le système va pouvoir prédire un décrochage.
 - le bipage qui consiste à activer le bipeur en cas de prédiction d'un décrochage.
 - L'affichage qui consiste à afficher les résultats du traitement.

Diagramme de cas d'utilisation:

_____ Maintenant nous allons établir le diagramme de cas d'utilisation de notre système qui donne une vision global du comportement fonctionnel du système logiciel, et comme vous pouvez le voir en [Annexe 2](#) il est présenté ainsi:

- Nous avons mis les acteurs principaux à gauche du package nommé "système anti décrochage" et les acteurs secondaires à droite de celui-ci.
- Nous avons aussi mis les cas d'utilisation du système à l'intérieur du package et nous avons tous reliés avec des flèche qui représente les relations entre les acteurs, les cas d'utilisation ou les deux, et parmi ses relations on trouve:

1. Une relation de généralisation entre l'utilisateur, le professeur et l'étudiant c'est-à-dire que l'utilisateur est une généralisation du professeur et de l'étudiant car l'utilisateur peut être soit l'un soit l'autre.
2. On a aussi deux relations d'extension la première entre le bipeur et le traitement de donnée car le bipeur est une option du traitement.

Et la deuxième relation c'est entre la création de compte et le login car le login est une option de la création, c'est-à-dire qu'on ne peut pas se connecter(login) si on crée pas de compte .

-
3. Une relation d'inclusion entre le remplissage du formulaire et l'authentification qui signifie que l'authentification est obligatoire si l'étudiant veut remplir le formulaire.

Diagramme de séquence et Scénario :

Le diagramme de séquence montre l'interaction entre les acteurs de notre système. Un acteur du système peut être sollicité par l'intermédiaire de ses opérations. C'est la seule manière d'adresser un objet. On dit qu'on envoie de message à l'objet. Ce message est dénoté par le nom d'une opération. D'une manière graphique on représente l'utilisation d'un objet dans le cadre de l'envoi d'un message.

scénario avec les acteurs principaux:

Scénario 1: capture de la température

	Scénario
Nom	Capteur de température
Objectif	prendre les mesure de température grâce au capteur et les envoyer au système de calcul
Pré-condition	<ul style="list-style-type: none">• le capteur doit être fonctionnel(avoir de l'énergie en continue et le mettre dans un environnement sans interférence) et doit être aussi allumé.• Le capteur doit être connecté au réseau bluetooth et en continue durant toute la durée de la collecte.• Les mesures du capteur doivent être homogène à tout point de la salle et pour toutes les autres mesures
Post-condition	<ul style="list-style-type: none">• continuer de prendre des

	<p>températures à intervalle régulier (toutes les 15 secondes).</p> <ul style="list-style-type: none"> • Le capteur est à un endroit fixe de la pièce et ne bouge jamais. • après utilisation éteindre le capteur(il arrête donc de prendre des mesure de température)
scénario nominal	<p>Étape 1: le capteur allumé (par l'utilisateur), se connecte automatiquement au réseau bluetooth et à la plateforme iot et commence à prendre des mesures de température</p> <p>Étape 2: le capteur prends la température toutes les 15 seconde</p> <p>Étape 3: il envoie les informations mesurées au serveur qui les envoie au système IOT(middleware).</p>

Diagramme de séquence de la capture de la température en [Annexe 3](#)

Scénario 2: Authentification du professeur

	Scénario
Nom	Authentification du professeur
Objectif	Récupérer les données
Pré-condition	<ul style="list-style-type: none"> • Il faut que le professeur est créé un compte au préalable
Post-condition	<ul style="list-style-type: none"> • Le professeur doit relever les informations en temps réel, activer les sessions des étudiants
Scénario nominal	Étape 1: le professeur se connecte au réseau, en saisissant son identifiant et son

	<p>nom, prénom</p> <p>Étape 2: les données saisie sont envoyé au serveur</p> <p>Étape 3: le serveur vérifie sur la base de données si ça correspond avec les données d'authentification déjà enregistré et constate que oui.</p> <p>Étape 4: le professeur entre dans sa session en tant qu'administrateur</p> <p>Étape 5: le professeur reçoit les données en temps réel et les consulte sur l'afficheur, et peut activer les sessions des nouveaux étudiants.</p>
--	---

Diagramme de séquence d'authentification du professeur en [Annexe 4](#)

Scénario 3: création de compte par un étudiant

	Scénario
Nom	création d'un compte étudiant
Objectif	se créer un compte et remplir le formulaire
Pré-condition	<ul style="list-style-type: none"> • Il faut que l'étudiant renseigne son nom et prénom et un mot de passe pour créer un compte sur le réseau. • Le professeur doit activer le compte de l'étudiant. • Le système attribue un identifiant unique à l'étudiant.
Post-condition	<ul style="list-style-type: none"> • L'étudiant doit remplir le formulaire et l'envoyer au système de calcul.
Scénario nominal	Étape 1: L'étudiant crée un compte en renseignant son nom, prénom et mot de passe.

	<p>Étape 2: les informations renseignés par l'étudiant sont envoyés au serveur qui les envoie à son tour vers la base de donnée</p> <p>Étape 3: l'étudiant se voit attribuer un identifiant unique lorsque le professeur aura activé le compte.</p> <p>Étape 4: l'étudiant reçoit le formulaire et doit le remplir intégralement.</p> <p>Étape 5: l'étudiant envoie le formulaire au système de calcul à travers le serveur.</p>
--	--

diagramme de séquence de création de compte par un étudiant en [Annexe 5](#)

Scénario avec les acteurs secondaires:

Scénario 4: le stockage de donnée

	Scénario
Nom	Stockage de données
Objectif	Stocker les données traitées dans la base de donnée
Pré-condition	<ul style="list-style-type: none"> les données doivent être cohérentes c'est à dire non modifiées et non biaisées par des éléments extérieurs. Les données doivent être au préalable traitées par le système de calcul. La base doit avoir assez d'espace pour sauvegarder toutes les données qu'elle recevra.
Post-condition	<ul style="list-style-type: none"> Le système doit avoir accès aux données précédemment sauvegardées dans la base. La base doit être continuellement

	en service sans interruption pour que le système puisse fonctionner.
Scénario nominal	<p>Étape 1: le système de calcul envoie les données traitées au SGBD à travers le serveur.</p> <p>Étape 2: La base vérifie si les données n'ont pas été biaisées.</p> <p>Étape 3: les données sont stockées dans la SGBD et prêtes à être réutilisées</p>

Diagramme de séquence pour le stockage de données en [Annexe 6](#)

Scénario 5: le traitement de données

	Scénario
Nom	système de calcul.
Objectif	Traiter les données mesurées par les capteurs.
Pré-condition	<ul style="list-style-type: none"> • Les mesures des capteurs doivent être envoyées à intervalle régulier (15 secondes environ). • Les capteurs doivent être connectés avec le système de calcul grâce au middleware (plateforme IOT).
Post-condition	<ul style="list-style-type: none"> • Le système de calcul doit être continuellement en service et sans interruption pour que le système puisse fonctionner • Le système de calcul doit pouvoir réutiliser les données précédemment traitées pour les nouveaux traitements à venir.
Scénario nominal	Étape 1: les capteurs envoient les données

	<p>mesurées au système de calcul (via le middleware) à travers le serveur.</p> <p><u>Étape 2:</u> le système de calcul traite ses données grâce à des algorithmes de prédiction.</p> <p><u>Étape 3:</u> il envoie les résultats du traitement au serveur qui les envoie ensuite à l'affichage et à la base de données.</p>
--	--

Diagramme de séquence pour le traitement de donnée en [Annexe 7](#)

Diagramme des classes :

Jusque là on a modélisé notre application par du texte et de figure guidée par le cahier de charge et le diagramme de cas d'utilisation. Ce choix est pédagogique pour la compréhension du client et pour aussi toute l'équipe de développement. Le passage du monde réel en programmation informatique nécessite une autre représentation des acteurs du système et leurs actions. Ici on utilise le concept objet. Un objet regroupe beaucoup d'informations qui le caractérise de manière unique du système . Son type, ses services et ses propriétés sont modélisés dans une représentation abstraite appelée classe.

On voit sur notre diagramme de classe en [Annexe 8](#) plusieurs associations entre les acteurs de l'application. On a aussi des relations de généralisation entre la classe Etudiant et la classe Personne. Il y a partage d'opération commune donné par la notion d'interface de la classe de l'interface **OperationCapteurs**. Chaque capteur est une spécialisation de la classe abstraite capteur.

Définition des classes et leur description

Dans cette partie on va définir le rôle de chaque classe et sa description.

• **La classe Personne :** cette classe générique décrit les acteurs humains. Le professeur, l'élève et l'administrateur(root). Le root est une réflexivité sur la classe Professeur.

• **La classe Etudiant :** Acteur humain après inscription et validation par le root, il fait que fournir ses informations personnelles et renseigner son état. Il n'intervient plus dans le système. C'est le professeur qui s'occupe du reste.

• **La classe Professeur :** Le professeur gère les inscriptions des étudiants. on parlera de root pour le prof que quand il veut accéder à la base de données. Sa communication avec les étudiants (envoi de formulaire, validation d'une inscription) se fera par l'intermédiaire du serveur. Il est l'observateur, interprétation après traitement des données par le système de calculs. Il est en interaction permanente avec le système. Il envoie des informations et attend des informations.

• **La classe Afficheur :** C'est l'interface du professeur pour observer les résultats du traitement. C'est une implémentation d'une interface graphique de Java. Il est en relation avec le serveur et avec le système de calculs pour récupérer les données à afficher.

• **La classe abstraite Capteurs :** C'est une généralisation de la classe des capteurs. chaque capteur est une spécialisation du premier.

• **Interface OperationCapteur :** C'est une interface qui regroupe toutes les opérations de service offertes et fournies par les capteurs.

• **La classe IOT :** C'est la classe qui récupère des données fournies par les non humains et les transmet dans le système de calculs. C'est comme une base de données ou cloud pour les systèmes communiquant. Il joue le rôle d'intermédiaire entre les capteurs et les autres services de stockage et de traitement de données.

• **La classe Serveur :** Il est indispensable pour récupérer des données ou envoyer des données dans la base. C'est la classe qui fait plus d'association avec les autres classes du système. Le formulaire de l'étudiant passe forcément par lui.

• **La classe Base de données :** C'est une classe de stockage de données en association qu'avec le serveur et le root.

• **La classe Microcontrôleur :** Cette classe récupère les données des capteurs pour les transmettre au serveur.

- **La classe Bluetooth :** Cette classe sert de passerelle de transmission de données sur le réseau sans fil.

- **La classe Application :** cette classe joue le rôle d'une application qui gère les données et la mise en œuvre des services de système.

ANNEXE 8: Le diagramme de classe

Les contraintes en temps réels:

Notre système devra bien sûr répondre à certaines contraintes très importantes pour assurer son bon fonctionnement. Les communications entre les différents acteurs primaires ou actifs sont en temps réel et asynchrone. Les capteurs doivent envoyer régulièrement des données mesurées à l'extérieur du système de calculs sans attendre aucun autre retour de celui-ci ou de dépendance quelconque d'un autre objet du système. Les supports du parallélisme sont assurés dès lors qu'on déclare certaines classes actives.

• Contraintes temporelles :

C'est la durée entre deux événements. Il est nécessaire de préciser que la notion de temps réel est le fait que les ensembles d'objets communiquent en même temps. Mais un objet identifié de manière unique du système ne peut pas effectuer deux opérations en parallèle mais en continu. Donc une opération périodique ou répétitive est maintenue. Ainsi on est libre de fixer cet intervalle de renouveler ces opérations. Dans le cas des capteurs on a choisi que chaque 5 minutes les capteurs envoient des mesures à l'objet IOT.

Modélisation dynamique des réseaux de capteurs :

On a utilisé jusque-là le langage d' UML pour modéliser la structure et le comportement de notre système. Ce dernier a des limites pour modéliser certains systèmes compliqués d'ordre technique.

Langage de modélisation:

Pour modéliser dynamiquement on va utiliser un nouveau langage SysML plus adapté que UML. Car UML a ses limites de modélisation. Il n'est utilisé que pour des applications logiciels. Pour les réseaux sans fils (capteurs), systèmes embarqués, objets communicants, et en systèmes , UML ne suffit pas pour les modéliser. Car ces entités du système dépendent d'autres facteurs techniques et physiques pour fonctionner comme l'énergie, si un capteur tombe en panne la relève est assurée automatiquement. Toutes ces fonctionnalités ne sont pas prévues dans la conception de UML d'où l'utilisation de SysML. Mais le concept de UML est toujours là.

Concepte de modélisation SysML :

SysML utilise les concepts de blocs ou nœuds(qu'on a expliqué dans le diagramme de déploiement) comme entité de bases de la structure du système. Les blocs sont décomposables en composants qui sont équivalents des classes dans UML. Ces blocs peuvent posséder des comportements. Les composants d'un même bloc sont reliés par des connecteurs. Un connecteur ne dit rien sur la nature de la liaison. Les composants possèdent aussi des ports qui indiquent la nature et type de relation ou grandeur physique entre différents composants du système. Pour la modélisation dynamique SysML utilise les diagrammes blocs internes, diagramme d'état, diagramme déploiement . . en spécifiant les propriétés des ports et des services requis et fournis.

Vision dynamique du système:

Comportement du système :

Le système collecte les informations reçues par les différents acteurs. Notamment, les capteurs qui eux sont des acteurs primaires. Grâce aux informations reçues, le système de calcul peut déduire de la situation réelle. Cette situation pourra aboutir ou non à un bip pour signifier un décrochage. Tous les acteurs peuvent envoyer des données dans la base, et ensuite les calculs se feront. Concernant les utilisateurs après remplissage du

formulaire, le système comme d'habitude ira stocker ces informations et le professeur lui en aura l'accès. En choisissant le réseau wifi, nous allons pouvoir interagir simultanément et efficacement mais surtout très rapidement grâce à la performance de ce réseau, une performance qui sera déjà assurée dès le début du lancement.

Les informations seront donc stockées et envoyées, cela aboutirait à un résultat qui permettrait de voir s'il y a bien eu un décrochage ou pas.

Contrôles de l'application:

l'application doit être contrôlée en permanence. Les données reçues doivent être des données cohérentes et bien mises à jour à chaque instant. Pour cela, chaque capteur doit être vérifié, ils doivent être opérationnels et les mesures prises doivent être vérifiées pour que les résultats de calculs ne soient pas erronés. Les données sont le principal enjeu, car le système dépend pratiquement de ça. C'est pour cela qu'il faut à tout prix, avoir des données bien cohérentes et à jour. Pour ce faire, le contrôle de l'application se fera automatiquement, grâce à un système de mise à jour et de vérification automatique des données, mais surtout de contrôle des capteurs. Il existe deux systèmes de stockage, le premier concerne le stockage des données locales, c'est-à-dire les données qui sont traitées notamment les mesures. Le deuxième stocke toutes les données de la plateforme. On peut constater qu'il y a une relation de dépendance entre les objets connectés et les capteurs, mais également avec le serveur. On voit alors la notion d'agrégation. Les capteurs sont reliés au serveur et le serveur à la base. Et les données sont stockées dans l'IBM cloud. Ce sont les données obtenues grâce au formulaire.

Diagramme d'état de transition:

Les diagrammes d'états-transitions d'UML décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements discrets (méthode). Ils spécifient habituellement le comportement d'une

instance de classeur (classe ou composant), mais parfois aussi le comportement interne d'autres éléments tels que les cas d'utilisation, les sous-systèmes, les méthodes.

Il m'a été demandé de faire le diagramme d'état de transition pour la création de compte, le diagramme représente la création du compte d'un étudiant, qui sera ensuite validé ou pas par l'administrateur, qui ici est le professeur. Après avoir créé ce compte, l'étudiant pourra ensuite envoyer le formulaire. L'étudiant peut ne pas réussir à créer un compte du premier coup, alors il pourra refaire l'opération.

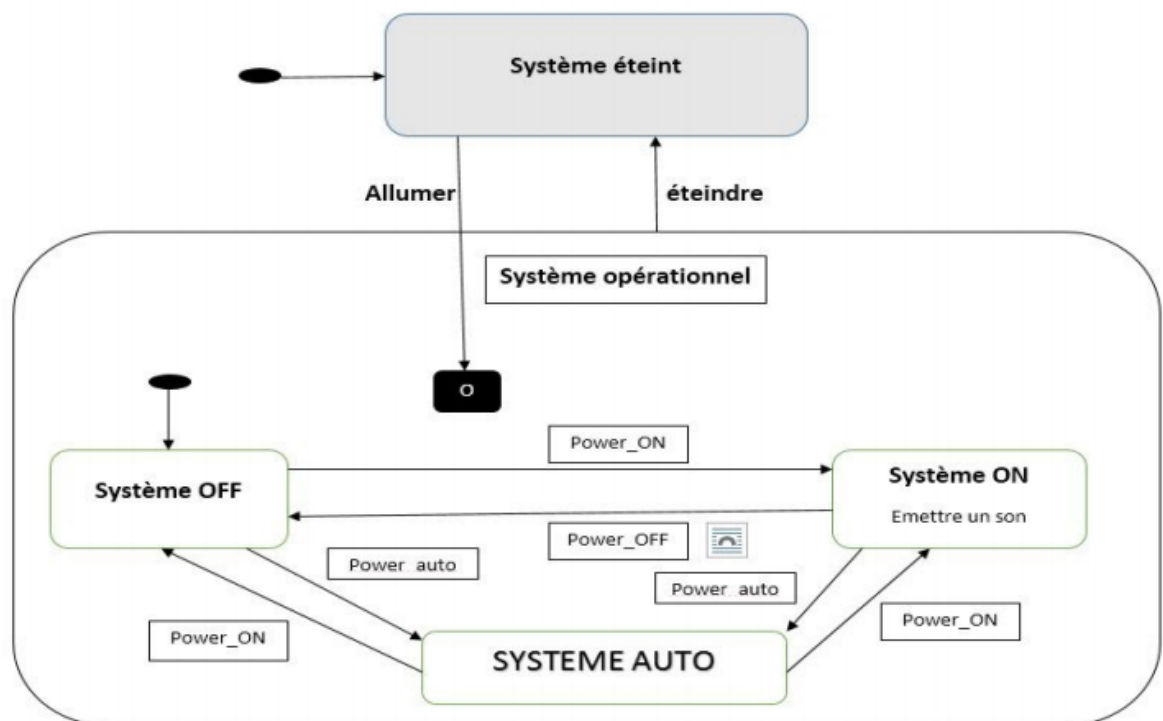


FIGURE 9 – Le Diagramme du fonctionnement du système ON/OFF

Diagramme d'activité:

Nous allons établir dans cette partie les diagrammes d'activité qui nous illustrent les cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de

représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Dans notre système on a établi deux diagramme d'activité, le premier sur les capteurs qui envoie les mesure au système de calcul qui les traite puis les stocke dans la SGBD et les affiche sur l'afficheur, diagramme à voir en [Annexe 9](#).

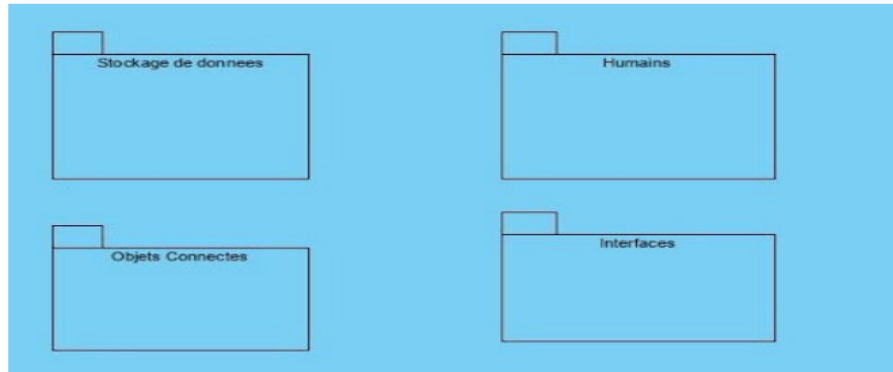
Quand au deuxième diagramme il porte sur l'authentification d'un utilisateur qui peut soit être: le prof qui ensuite active les compte étudiant et consulte l'afficheur ou bien sa peut etre un étudiant qui se connecte et remplit un formulaire pour l'envoyer ensuite au système de calcul qui traite les information de l'étudiant et les stock dans la SGBD,diagramme à voir en [Annexe 10](#).

Diagramme de packages :

L'ensemble des packages définit l'architecture du logiciel. Il s'agit d'un mécanisme général regroupant d'éléments tels que blocs , interface,acteurs, cas d'utilisation. Il constitue un espace de noms(namespace) pour les éléments qu'il contient. La façon d'organiser son code en packages est relative. On peut les regrouper en fonction des exigences du système, les blocs extérieurs, de cas d'utilisation . . . Mais ici on va organiser notre code en fonction du type des acteurs et leur relation logique.

• Les quatres de packages de notre application:

On va tenter d'organiser le code dans quatres packages différents comme l'indique le schéma. Et ensuite on va définir leur contenu.



• Les relations de contenance entre les packages et les classes:

Nous avons vu que les blocs peuvent être reliés par des associations, de compositions, de généralisations, etc. Ces relations entre éléments induisent des relations de dépendances entre les packages englobants.

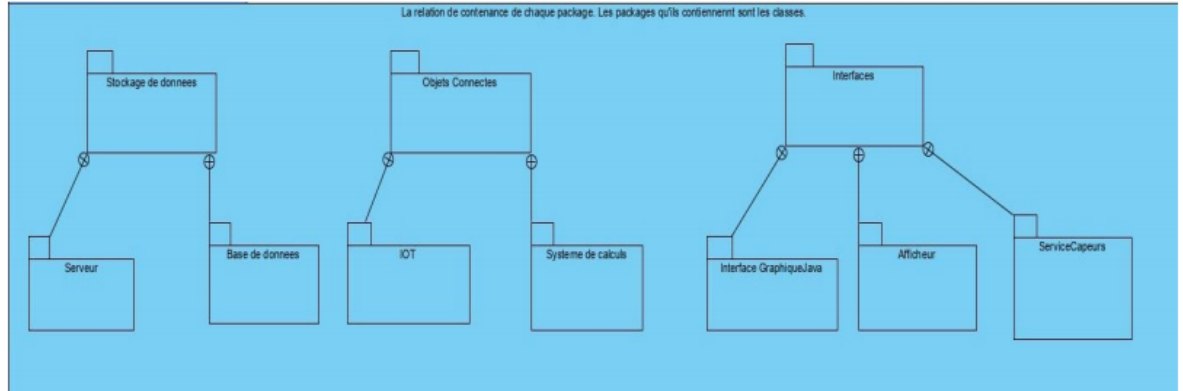


FIGURE 17 – Les relation d'inclusion entre les classes et packages

• La relation entre classe implique relation entre packages:

Ce dessin illustre les contenus des contenants. Ici les contenus c'est les classes de notre système. La relation des classes implique la relation des packages comme illustre le schéma ci-dessous.

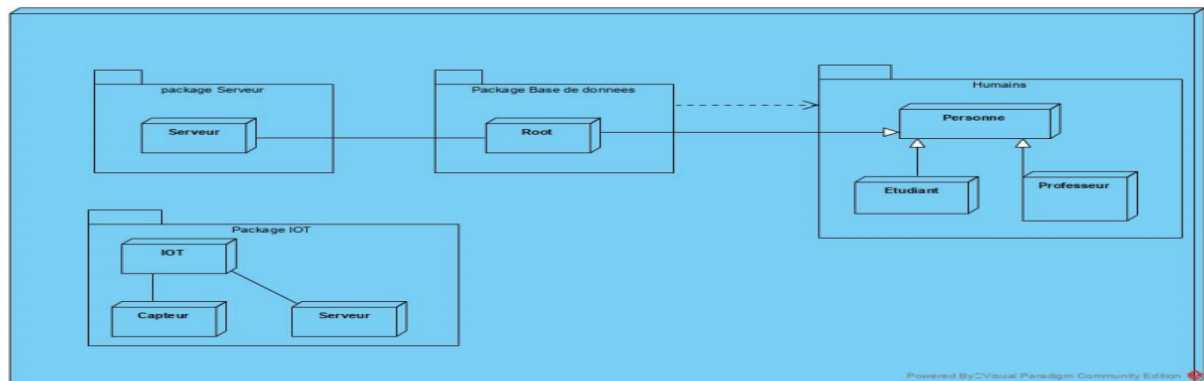


FIGURE 18 – La relation des classes implique la relation des packages

• L'ensemble des classes dans leurs packages:

Ce diagramme illustre la relation entre les packages due à la relation entre des acteurs.

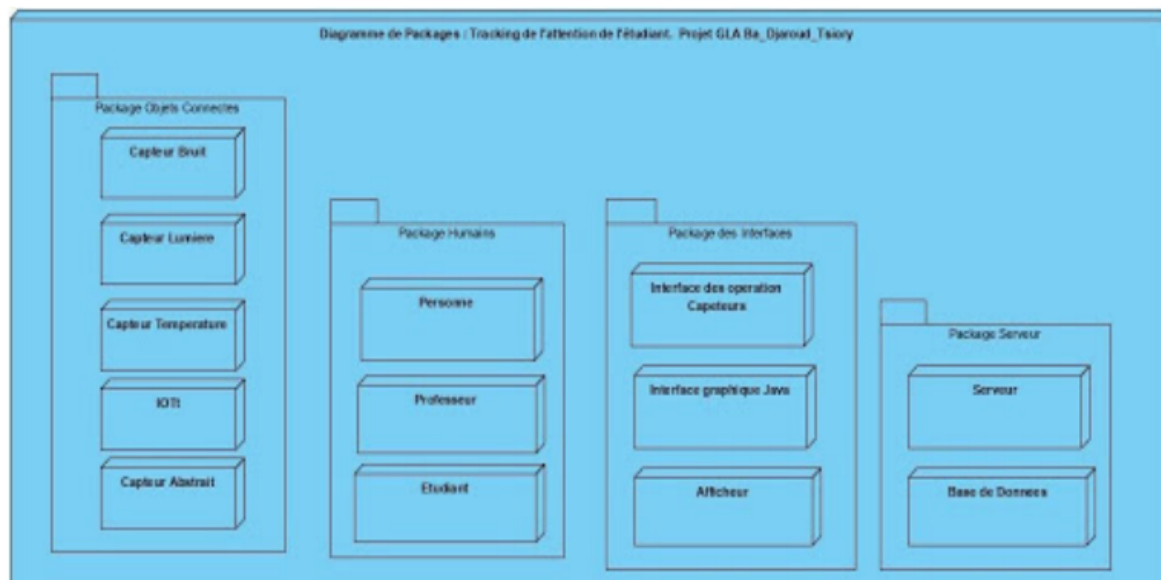


FIGURE 19 – L'ensemble des classes dans leurs packages

Les diagrammes des noeuds et composants de notre systemes:

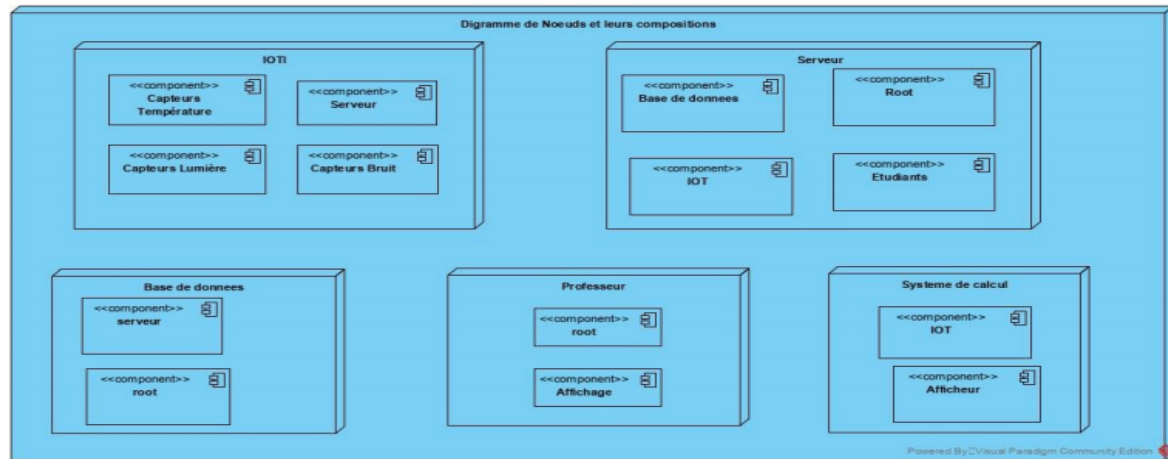


FIGURE 20 – Diagramme des noeuds et leurs composants

Un composant est une unité autonome représentée par un classeur structuré, stéréotypé « component », comportant une ou plusieurs interfaces requises ou offertes. Son comportement interne, généralement réalisé par un ensemble de classes, est totalement masqué : seules ses interfaces sont visibles. La seule contrainte pour pouvoir substituer un composant par un autre est de respecter les interfaces requises et offertes.

Le diagramme de déploiement:

Le diagramme de déploiement présenté en [Annexe 11](#) illustre comment notre système est connecté avec les différents composants physiques et matériels. Ces derniers ne figurent pas forcément sur les diagrammes de cas d'utilisation ou d'états ou de classes. Le serveur représente ici un nœud qui contient le plus de composants internes ou d'association. C'est lié à son indispensable rôle d'intermédiaire entre la base et les autres composants. Sauf l'administrateur qui a accès directement à la base, sinon toute

consultation de la base passe par le serveur. Il contient des fichiers .html, .php, .js On va détailler dans les prochaines lignes un peu son comportement.

On voit sur le schéma en [Annexe 12](#), le comportement réel du serveur à la suite d'une requête. Le serveur contient un serveur web qui exécute les pages ou requêtes dynamiques en php et les transforme en page html pour envoyer le contenu au navigateur ou client. Si la requête demande des pages statiques html il va les chercher directement à la base.

Chapitre 3 : conception et solution préconisée

motivation:

Vu qu'on est confronté à plusieurs contraintes techniques, matérielles et logicielles pour l'implémentation. Pour avoir une solution satisfaisante répondant aux besoins du client , on doit faire appel à des librairies existantes, outils de développements(SDK), des apis, des modules et tout autres projets qui apportent des solutions partielles ou totales à l'implémentation de notre application. Vu qu'on était tombé d'accord sur certaines solutions apportées dans l'analyse concernant le protocole(MQTT), la connexion à distance wifi .. Il ne reste qu'à voir leurs principes de fonctionnement et ensuite de chercher des APIS, des bibliothèques, des modules, des projets, des open source, des IDE qui sont compatibles avec ce qu'on a et ce qu'on sait déjà.

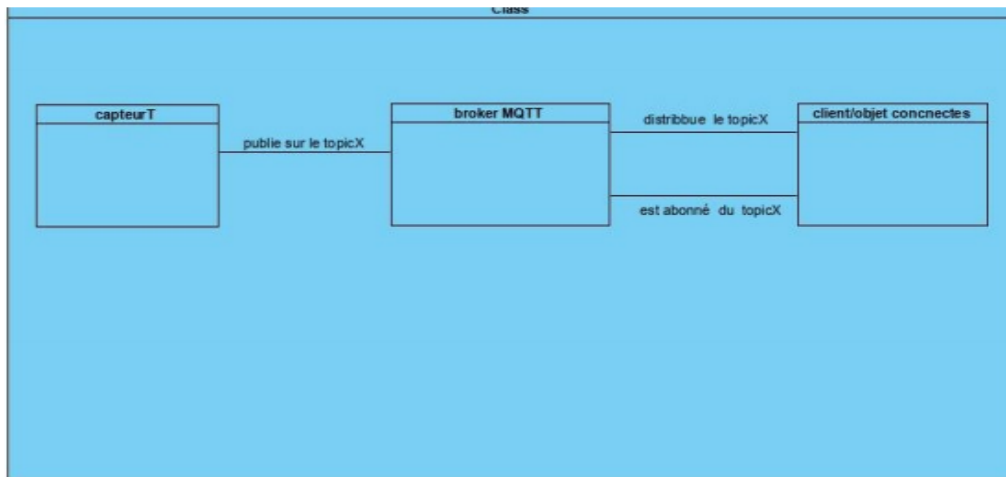


FIGURE 23 – Le serveur MQTT qui reçoit un message et le redistribue

PAHO et Mosquitto:

Le principe de communication du MQTT:

Comme rappelé précédemment, le protocole MQTT est bien adapté pour les API de communication qui ont beaucoup de contraintes d'environnement. Il fonctionne sur un principe simple de souscription/publication entre les clients (qui sont ici les objets connectés). C'est juste une architecture client/serveur.

Le serveur centre des distributions des messages indépendamment du développeur. Tout objet inscrit ou abonné à un "topic" ou canal reçoit des messages publiés sur ce canal. Le canal c'est juste un nom: un nom d'utilisateur, numéro port, ou une adresse ou chemin sous forme de chaîne de caractères. Donc c'est des commandes faciles d'utilisation pour cibler les objets à envoyer des messages.

Publication

Souscription

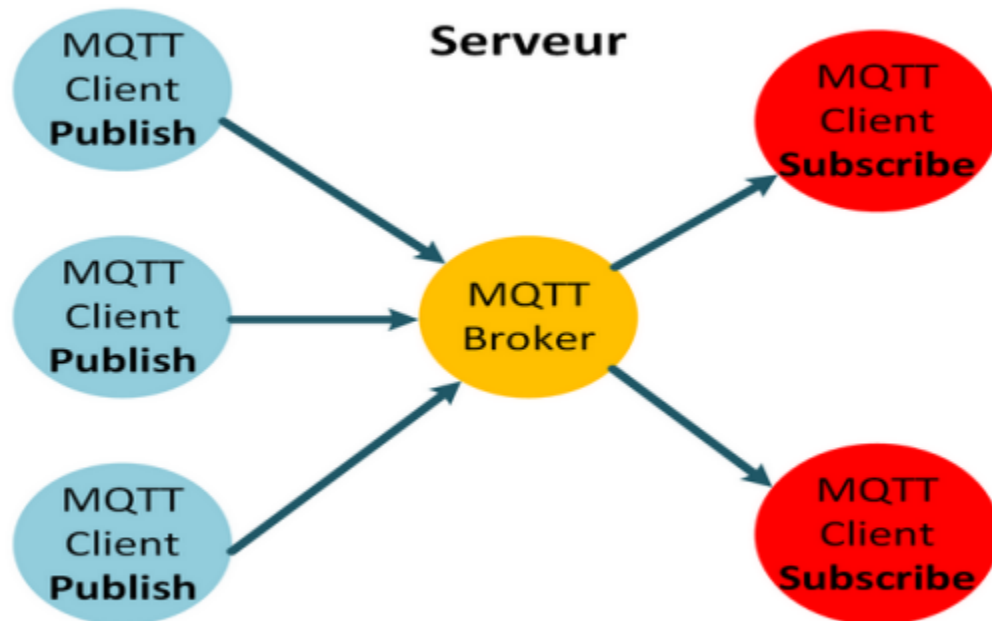


FIGURE 24 – Le principe de communication du protocole MQTT

Langage ,Librairies et Environnement de travail:

Il y a plusieurs librairies ou bibliothèques ou IDE qui offrent différentes possibilités d'implémentation dans tous les langages. Ici vu qu'on privilégie l'orienté objet et java , ce dernier possède des projets eclipse sur les objets connectés que nous allons développer dans les lignes suivantes.

Gestion des dépendances dépendance : MAVEN

Vu que notre application fera l'objet d'intégration de plusieurs autres logiciels on aura besoin d' un module pour gérer les dépendance et les conflits. Eclipse PAHO est un projet eclipse de MQTT avec comme option de création de projet "MAVEN PROJECT". Ce dernier est un open source ou module à configurer. Sa configuration est très facile. C'est

comme un open source. Le code existe déjà après création du projet Maven. Il a une structure différente des projets java qu'on a l'habitude de voir. La seule chose importante à retenir est qu'il génère automatiquement un fichier pom.xml. C'est ce fichier qui gère les dépendances de notre projet. C'est très utile pour les projets qui utilisent plusieurs technologies. Donc MAVEN seul a réglé plusieurs choses.

Mosquitto est une API de éclipse PAHO qui contient des bibliothèques qui implémentent MQTT. Il peut jouer le rôle du serveur et est au cœur des services de distribution de message. Le package "org.eclipse.paho.client.mqttV3" de paho contient deux grande classe d'implémentation :

- **La classe Mqtt Client**
- **La classe MqttAsyncClient**
- **L'interface IMqttClient**

Ces deux classes contiennent le nécessaire pour simuler les clients et communiquer entre objets connectés et le serveur en synchrone et/ou asynchrone..ET les deux premières classes sont une implémentation de la troisième qui est une interface de MQTT paho.

Les commandes de mosquitto:

pour communiquer en local Pour communiquer entres clients mosquitto dispose deux commandes :

- **mosquitto-pub** : pour publier un message.
- **mosquitto-sub** : pour souscrire à un topic.

ils prennent des paramètres :

- **h** : nom du broker.
- **p** : indique le nom du port par défaut c'est 1883..
- **t** : le topic sur le quel on publie le message.
- **m** : message à envoyer texte, valeur ou Json.

Exemple de messages du capteur : Si le capteur est connecté au serveur (-h) localhost et est abonné sur le topic (-t) capteurT et publie le message (-m), 22.5 degrés Celsius s'écrit :
`mosquitto-pub -h localhost -t "capteurT" -m 22.5`

Pareil pour s'abonner à un topic capteurT , on écrit :

`mosquitto-sub -h localhost -t "capteurT" -m 22.5`

Pour la base de donnée:

Pour la base de données On a fait appel à la base de données local MYSQL. Eclipse Paho intègre bien une interface qui permet d'implémenter ou d'interagir avec la base de données.

Les autres apis pour les besoins matériels et leur configuration :

Les autres apis sont ESP8266, ESPEasy, NODE-RED, MQTTBox, le Virtual port Serial port Driver 9.0 ce sont des besoins matériels pour simuler notre application.

- **Le ESP8266 :** est une micro puce qui a des propriétés de TCP/IP et de microcontrôleur. Il est téléchargeable sur le site de EXPRESSIF.

- **ESPEasy :** Il est utile pour flasher le microcontrôleur. Il est paramétré pour régler les COM PORTS, le flux d'octets et la capacité maximale en octets du microcontrôleur ainsi les autres paramètres de Qualité of Services (Qos).

- **Le Node-Red et le MQTTBox :** Tous les deux sont des interfaces de visualisations. Ils instancient des clients ou les objets de connexion. Et permet de voir de manière globale tous les messages échangés.

Chapitre 4 : Apport du projet et Conclusion

Confrontation avec la réalité:

Par manque de temps et à cause du confinement dû à la crise sanitaire, on a pas pu mettre en pratique notre solution avec de vrais capteurs qu'on aurait disposé dans une salle de cours, mais à la place et comme expliqué durant le mémoire nous avons opté pour une simulation avec les apis mosquito sous éclipser paho, ce qui nous a permis de tester notre système, et les résultats sont plutôt prometteurs car ça répond bien mais nous en serons certains que a la fin de la crise pour mettre en place notre solution de capteurs dans une salle de cours et assurer une contraintes essentielle de notre système c'est à dire assurer des donnée fiable et homogène.

Dans ce projets j'ai mis à contribution mes connaissance en UML pour proposer une analyse fonctionnel pour le système et des diagrammes de séquences et de cas d'utilisation pour illustrer les fonctionnalités sous forme statique mais aussi sous forme dynamique grâce aux diagramme d'activité de packages et de déploiement que nous debriefons bien sur a chaque réunion et apportant les modifications nécessaire.

Dans le future:

Durant les simulation nous avons détecté aucune defaillance du systeme et a l'aire de très bien fonctionner mais zocus réfléchit en ce moment a la suite c'est à dire mettre en place l'infrastructure physique du projet avec de vrais capteurs et une vrais base de données temps réel et pour ce faire notre travail va se concentrer dans un premier temps à la mise en place de cette infrastructures en plaçant les capteurs de manière optimale et commencer à déployer l'application aux professeurs puis aux élèves dans le bits de tester cette dernière et le système d'authentification et aussi si la collecte des données du formulaire ce passe bien avec un flux conséquent; deuxièmement notre activité principale va se porter sur la maintenance du système et faire en sorte qu'il soit toujours

opérationnelle et aussi d'améliorer l'expérience en proposant une interface de plus en plus pratique et en améliorant certains composants pour une collecte de données plus fiable.

Nous avons remarqué durant ce projet et plus particulièrement durant les simulations l'importance d'une planification structurée qui définit l'objectif et propriété de chaque opération, mais il a aussi mis en évidence l'importance de la communication pour la cohésion entre les équipes et pour faire un travail efficace.

Zocus prévoit aussi de développer une solution qui a été proposée dans les hypothèses de recherche et qui est de développer un produit (machine) qui va englober tous les capteurs dont on a besoin pour le bon fonctionnement du système ainsi qu'une application dédiée et beaucoup plus accessible. En d'autres termes rendre le système de détection beaucoup plus accessible même pour les particuliers car en aura plus besoin de mettre en place une infrastructure complexe dans une pièce dédiée (une salle de cours entre autre) mais selon la première estimation cela provoquera une baisse légère de fiabilité des données collectées et toute la complexité du projet sera de faire en sorte qu'il y ait le moins de perte d'homogénéité et de précision des données externes.

Apports et difficultés du projet:

J'ai eu l'occasion d'effectuer mon stage au sein d'une entreprise moderne et avec un service informatique fraîchement ouvert. En effet ce projet est l'un des premiers à avoir été réalisé au sein de ce service, ce qui m'a permis de bien m'intégrer au sein du groupe vu qu'on était tous nouveaux.

Le rythme de 3 semaines en entreprise m'a beaucoup aidé à assimiler rapidement la méthode de travail en entreprise (ici le cycle en V), mais le travail à distance due à la crise sanitaire a beaucoup compliqué les choses car on ne discutait la plupart du temps que par mail ce qui m'obligeait à toujours regarder si il n'y a pas de nouvelle modification dans la conception du système et quand cela arrivait je devais réagir très rapidement car le moindre changement ou ajustements dans les fonctionnalités m'obligeait à revoir les différents diagrammes que j'avais pu faire.

Entre autres, ce projet de stage m'a permis de découvrir l'internet des objets connectés et leur fonctionnement en allant de la conception à l'implémentation sans oublier les besoins matériels ou/et logiciels et leurs intégrations. C'est un vrai travail d'ingénieur. Il est plein d'aspects techniques dans le sens qu'il fait intervenir des composants électroniques dont leur configuration est fastidieuse. Son besoin en matériel est énorme. Ce qui a provoqué notre limite et notre difficulté à aller au bout du projet. Le confinement à limiter notre champ de recherche qui s'est basé essentiellement sur internet et nos échanges sur les plateformes de télétravail. Même pour simuler notre application, on a dû faire face à des difficultés notamment avec les interfaces de eclipse paho. L'installation et la configuration de ces bibliothèques ou apis sont très difficiles et prennent beaucoup de temps. Il nous a aussi permis de pouvoir faire une analyse profonde d'un cahier de charge. Et d'en sortir tous les besoins fonctionnels et non fonctionnels ainsi que les contraintes techniques.

Conclusion :

Dans le domaine cognitif et de la concentration, les objets connectés ainsi que la gestion et le traitement de données de manière automatique peuvent représenter un enjeu majeur dans la compréhension cognitive et les raisons qui induit un décrochage.

C'est la question que s'est posée l'entreprise Zocus et a mis à profit les technologies au service des sciences cognitives en réalisant ce projet et j'ai eu la chance d'y participer et d'y contribuer avec mes connaissances et compétences acquises au bout de quatre années d'étude en informatique.

C'était une vraie chance pour moi de mettre à profit mes connaissances en génie logiciel face aux exigences de l'entreprise. j'ai pu résoudre le problème en quatre étapes: d'abord avec une brève présentation de l'entreprise et de la problématique posée par cette dernière, ensuite j'ai fait une analyse fonctionnel du système en mettant en lumière les exigences, les contraintes et les besoins du système, après je suis passé à la conception du système en élaborant les différents diagrammes (dynamique et statique) représentant ce dernier et enfin j'ai proposé une solution qui consistait à faire des simulations de données pour voir si le système répondait bien et permettre à l'équipe de développement de travailler malgré la distanciation sociale et l'absence de capteur physique.

Finalement le bilan du projet est très positif car grâce au système mis au point par nous il sera à présent très facile à un professeur ou à un administrateur de zocus de savoir à quel moment un étudiant a décroché et même de pouvoir analyser les facteurs

qui rentre le plus en considération dans un décrochage et pouvoir apporter les solutions appropriée à chaque classes.

Le gros plus pour moi a été de pouvoir travailler avec d'autre salarié de l'entreprise et en groupe ce qui a stimulé ma capacité à m'intégrer au sein d'un groupe de travail et pouvoir associées nos idées pour donner vie à un projet, et le faite de jongler entre le travail et les cours en plus des changements de dernières minutes sur la conception du système m'a permis de m'apprendre à bien m'organiser et à mieux gérer mon temps.

Ce projet m'a énormément apporté en expérience de travail et professionnalisme.

Annexe:

Annexe 1: Formulaire À remplir par l'étudiant

Formulaire Etudiant :

Information concrète :

Nom de l'étudiant :

Prénom de l'étudiant :

Date de Naissance :

Sexe de l'étudiant : Femme

Nom du cours : Choisissez un élément.

De quel type de cours s'agit-il ? Choisissez un élément.

Durée du cours : 1 heure

Heure de début du cours : Choisissez un élément.

Information sur les facteurs interne de l'étudiant :

- Etes-vous motivé par se cours : Choisissez un élément.
- Aimez-vous se cours : Choisissez un élément.
- Quel degré d'intérêt portez-vous a se cours : Choisissez un élément.

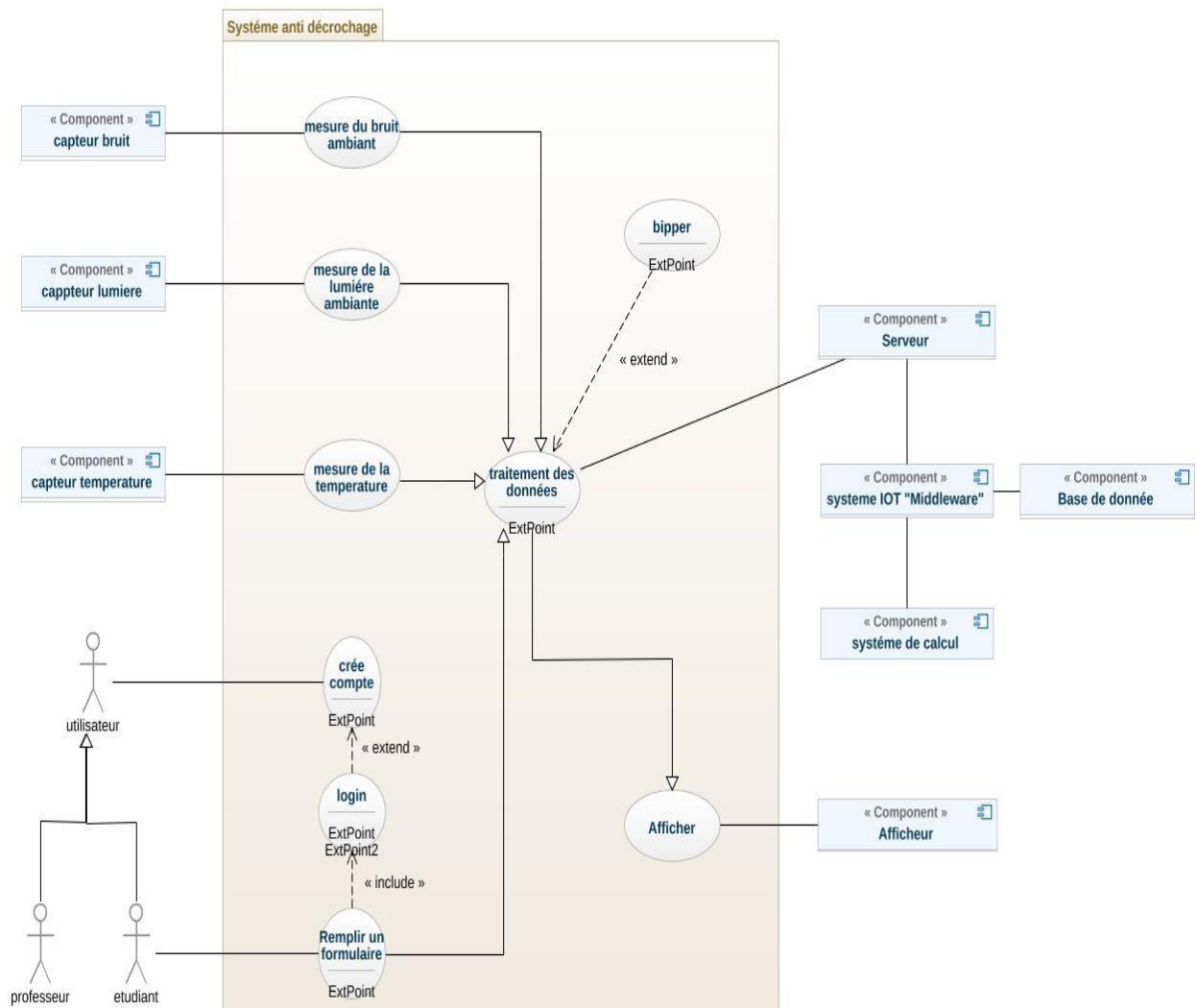
- Renseigné votre état mental en ce moment : Plusieurs réponse possible

Stressé(e) ☐
Triste ☐
Furieux(se) ☐
Seul(e) ☐
Préoccupé(e) ☐
Joyeux(se) ☐
Tout ☐

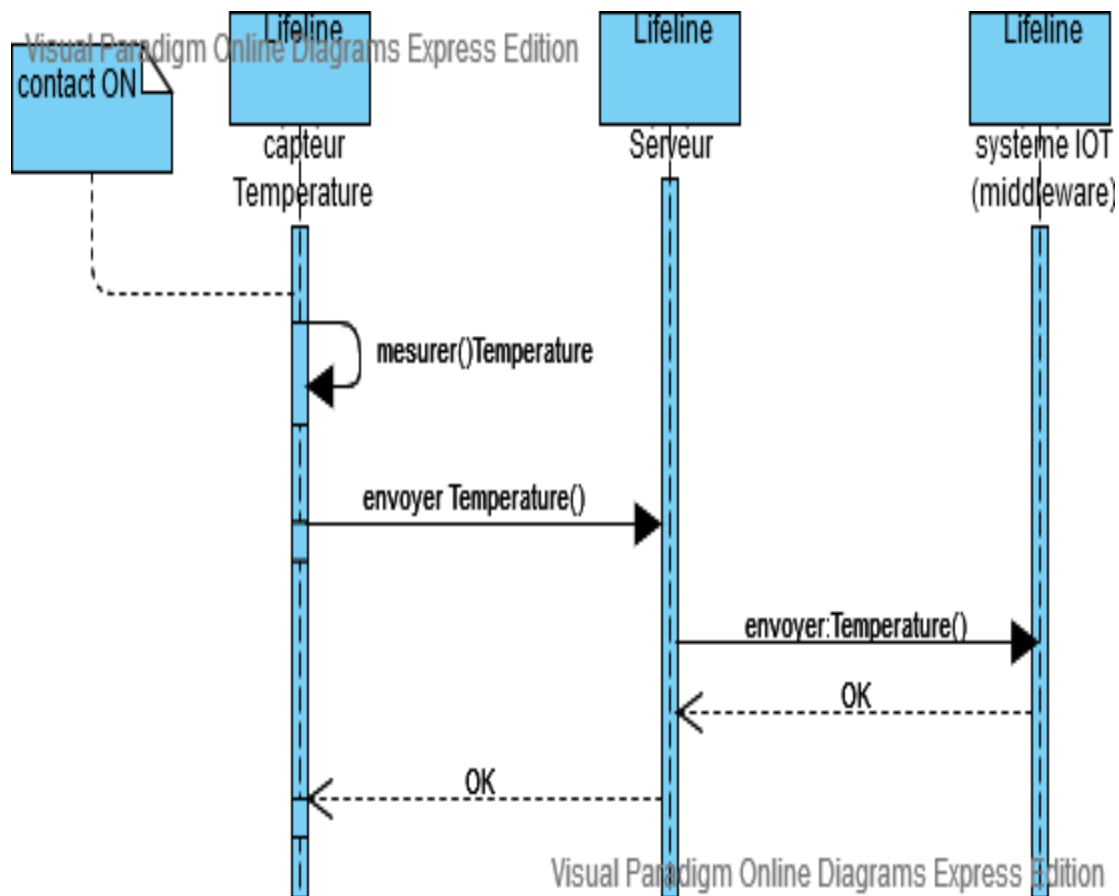
- Renseigné votre état physique actuelle : Plusieurs réponse possible

Fatigué(e) ☐
En forme ☐
Malade ☐
Tous ☐

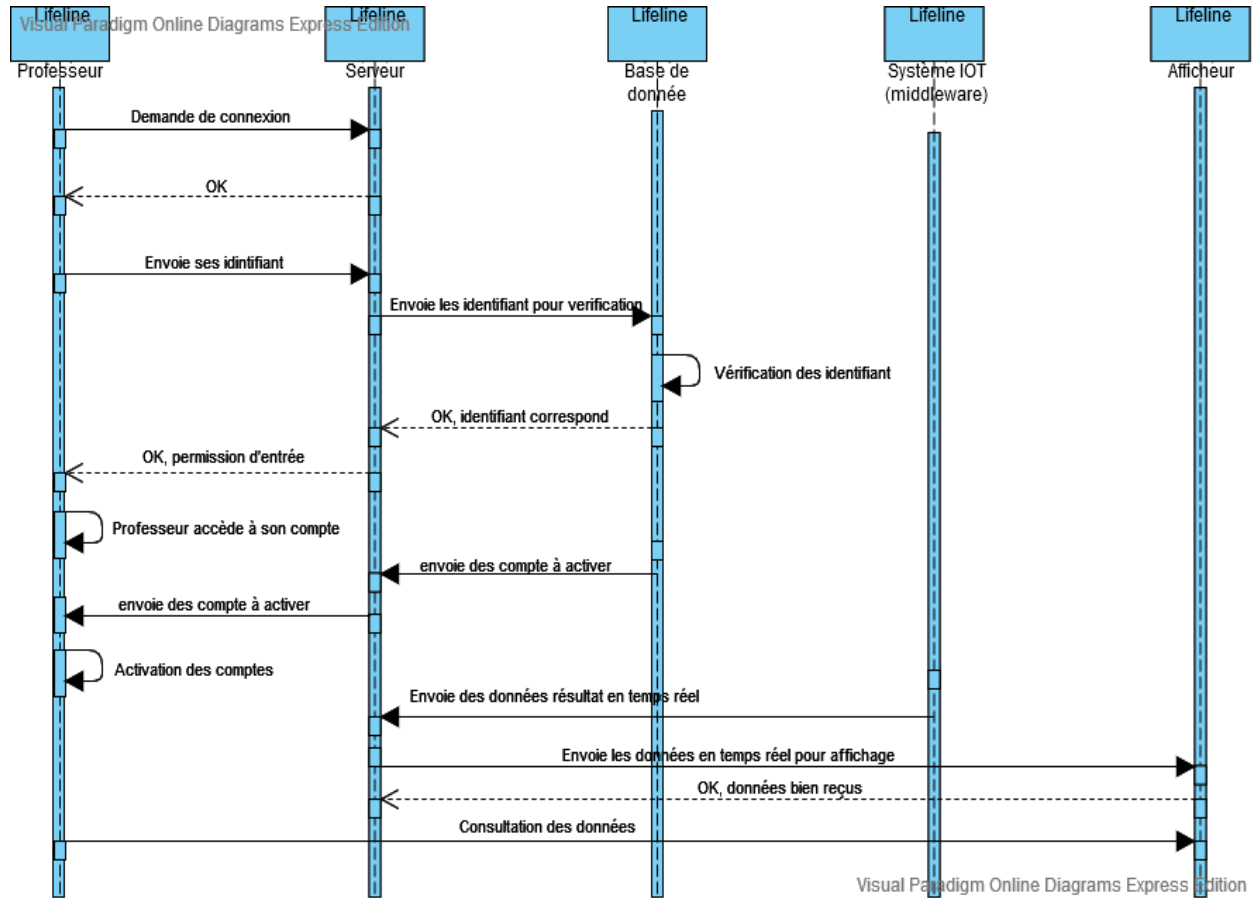
Annexe 2 : Diagramme de cas d'utilisation



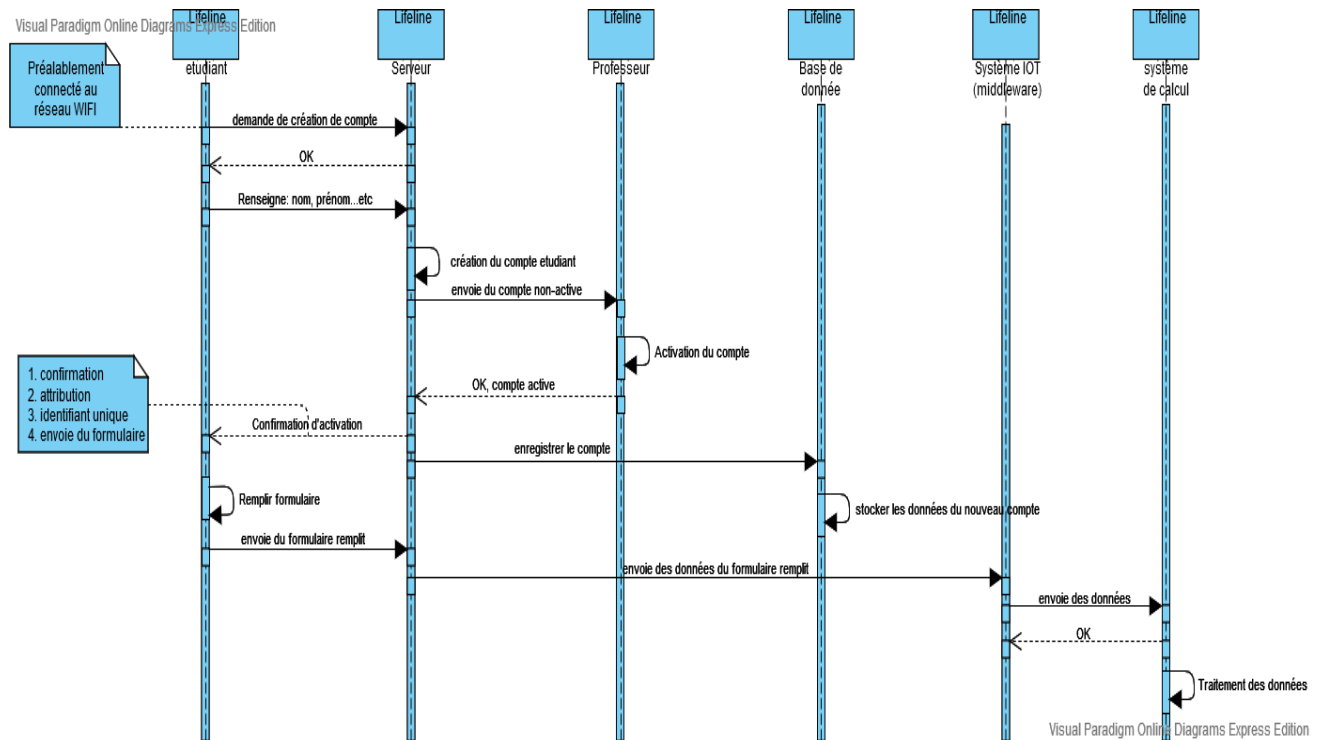
Annexe 3: Diagramme de séquence de la capture de la température



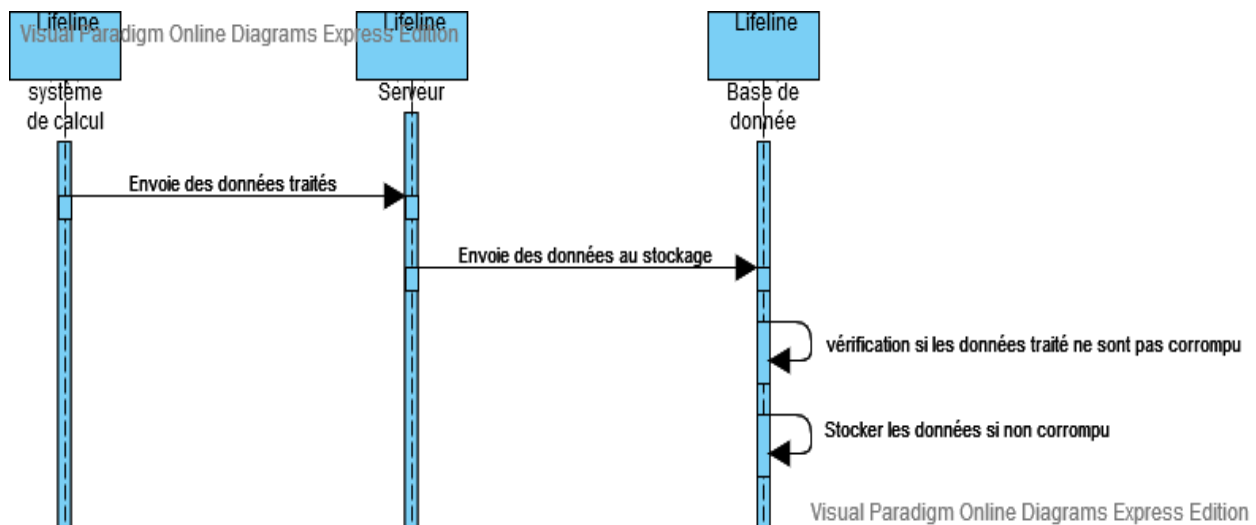
Annexe 4: Diagramme de séquence d'authentification du professeur



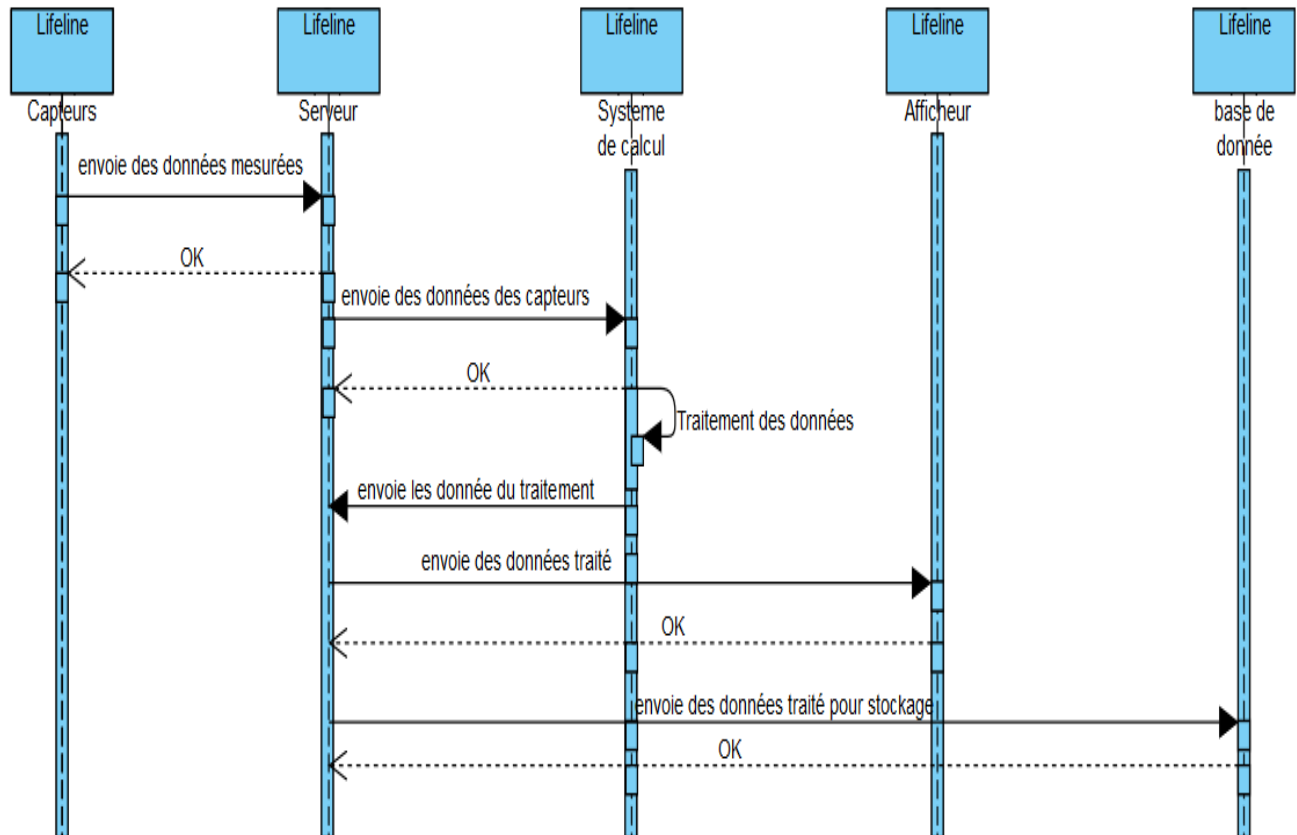
Annexe 5: Diagramme de séquence de création de compte par un étudiant



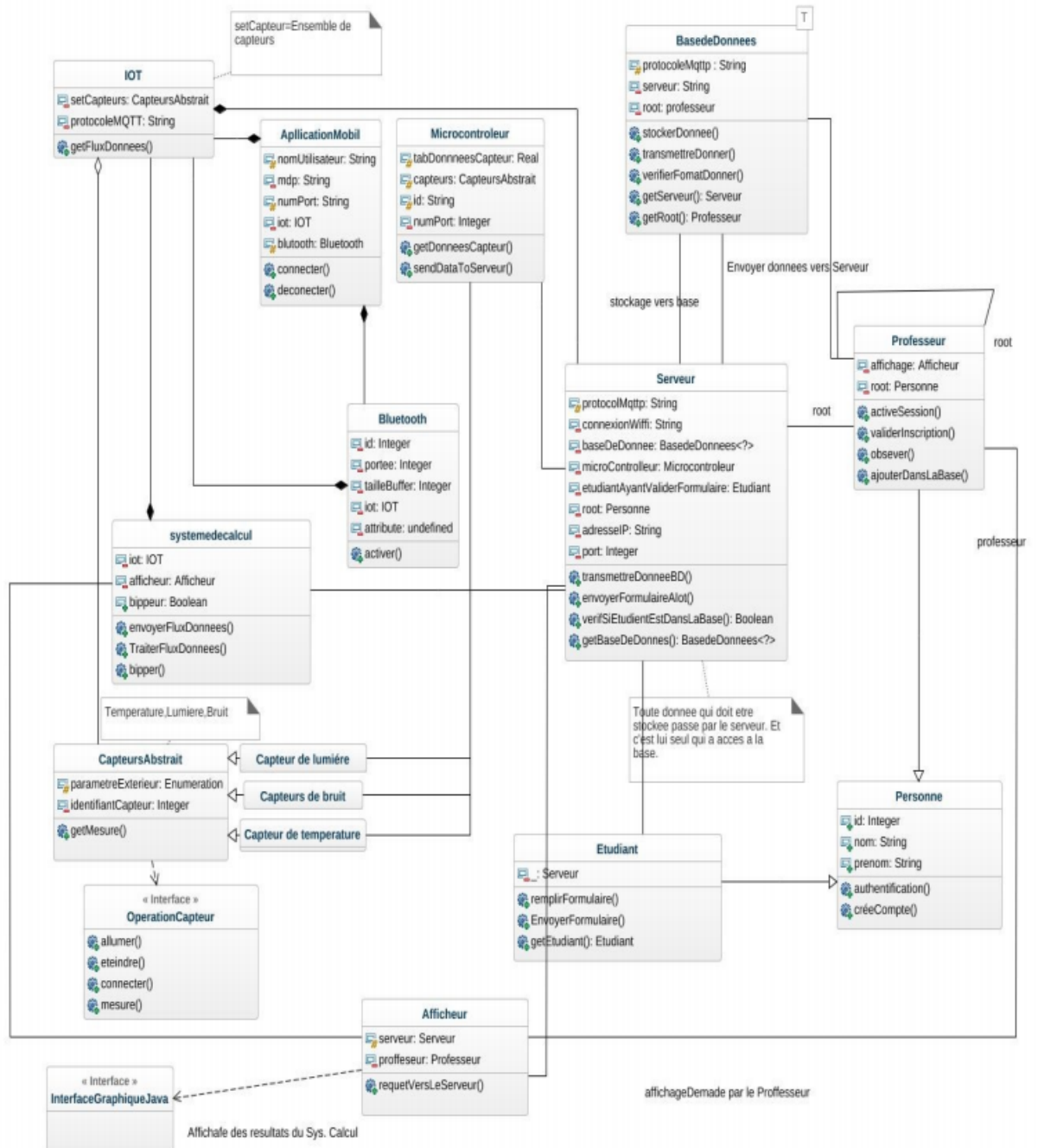
Annexe 6: Diagramme de séquence pour le stockage de donnée



Annexe 7: Diagramme de séquence pour le traitement de donnée



Annexe 8: Diagramme de classe du système



Annexe 9: diagramme d'activité de flot de données des capteurs

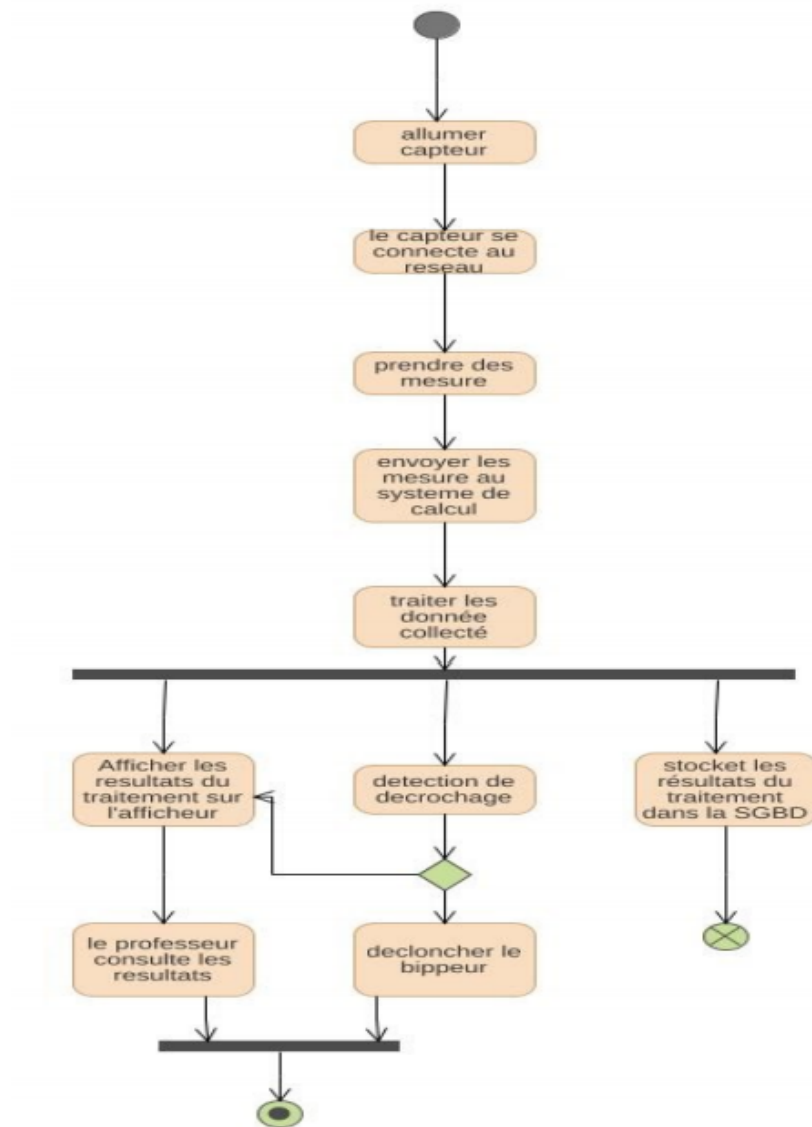
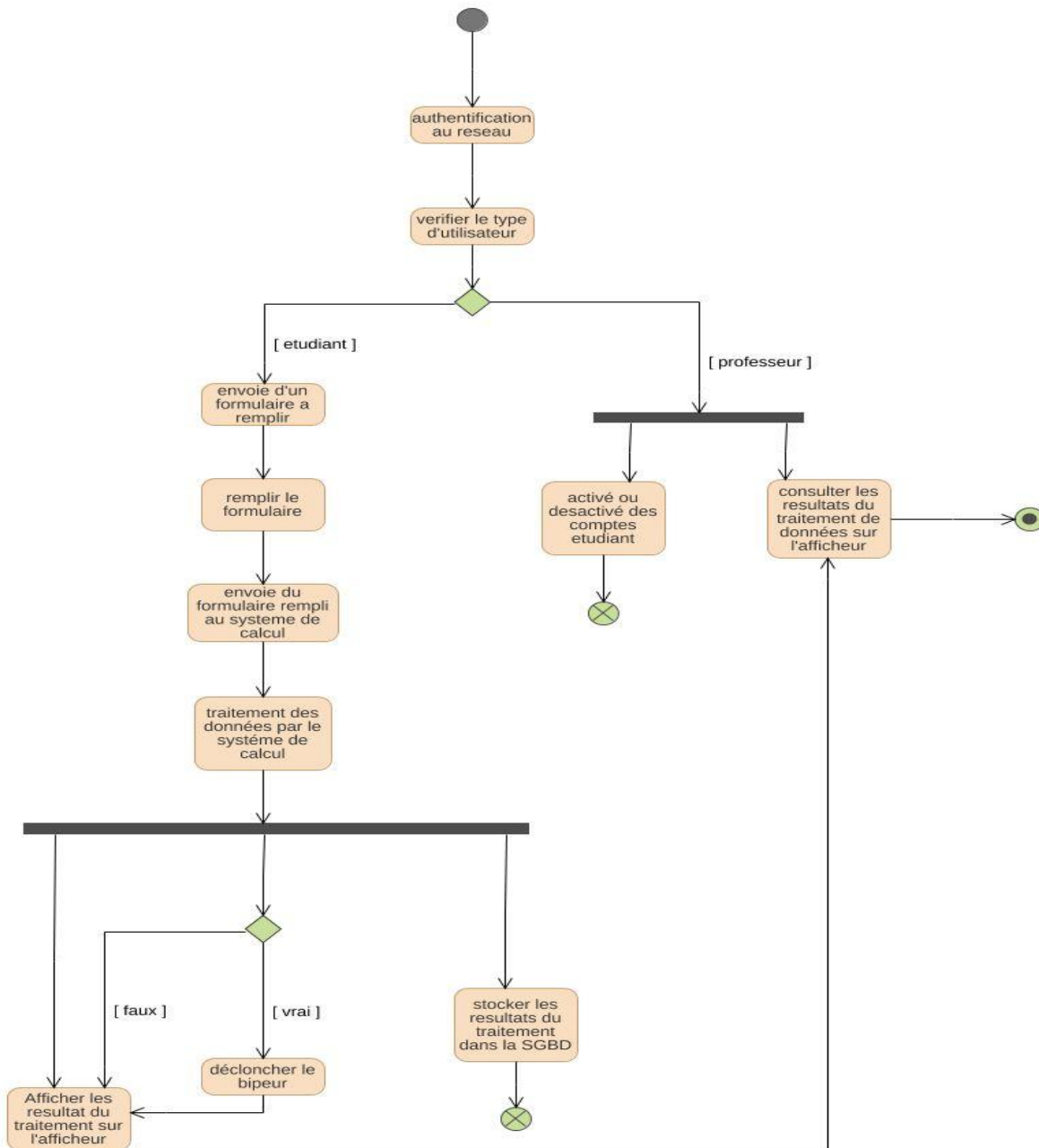


FIGURE 12 – Le Diagramme d'activité de flot de données des capteurs

Annexe 10: Diagramme d'activité d'authentification d'un utilisateur



Annexe 11: Diagramme de déploiement du système

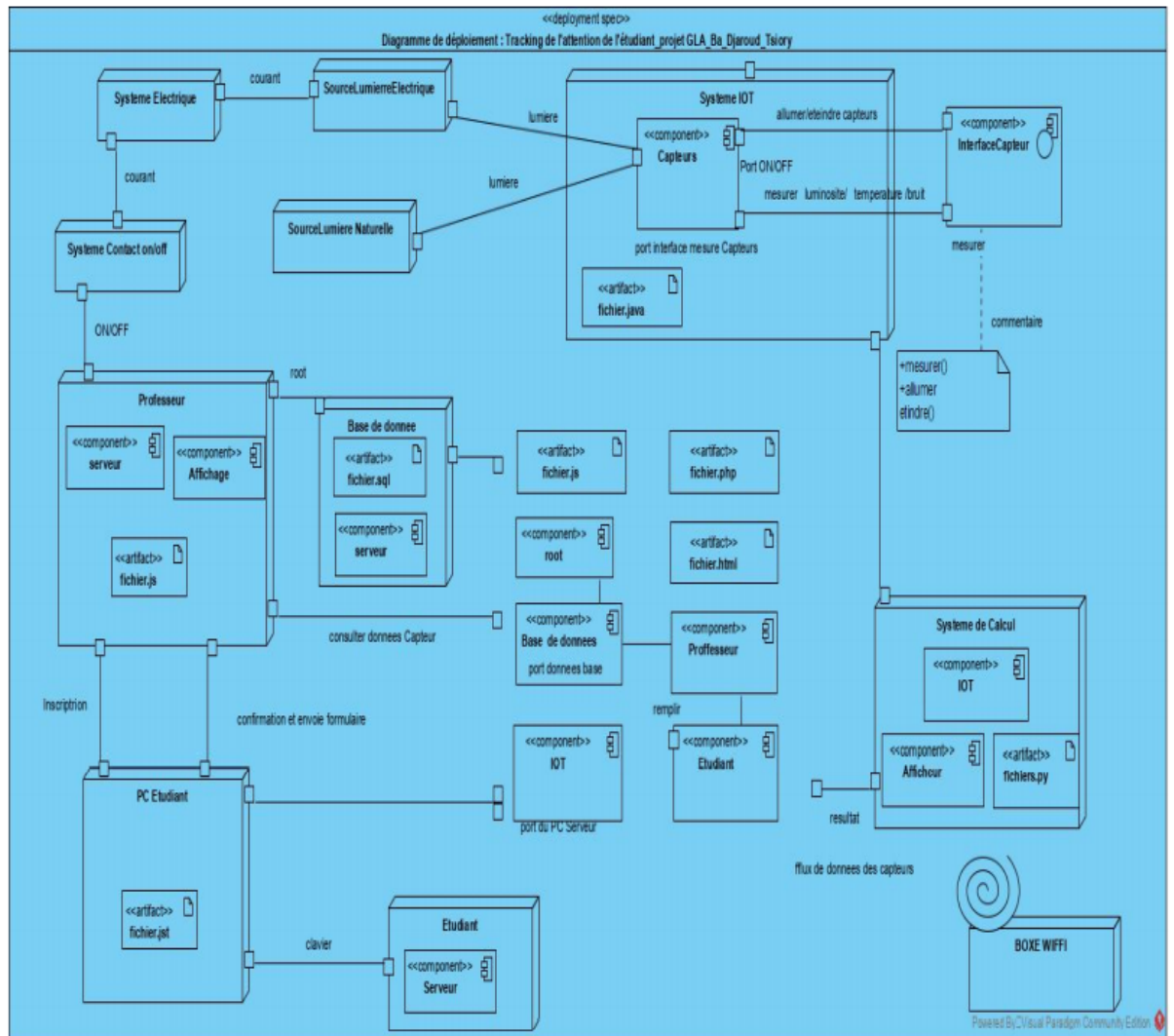


FIGURE 21 – Le diagramme de déploiement de notre système

Annexe 12: Comportement du serveur

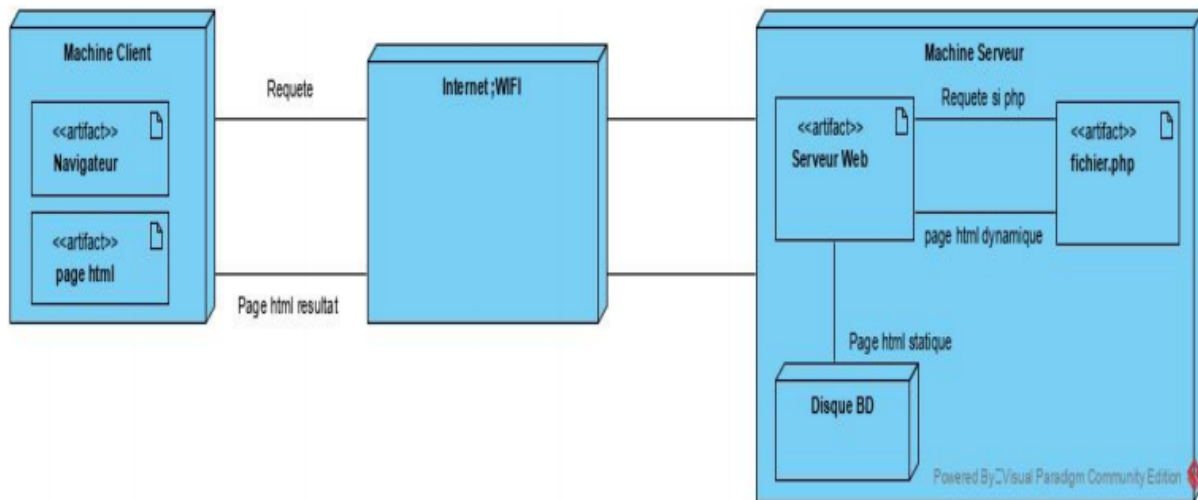


FIGURE 22 – Le serveur , le coeur de la communication

Bibliographie:

Source internet:

1. <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-etats-transitions>
2. <https://vipress.net/microcontroleur-iiot-divisant-10-consommation-niveau-au-systeme-analog-devices/>
3. <https://fr.wikipedia.org/wiki/Microcontr%C3%B4leur>
4. <https://fr.wikipedia.org/wiki/Microcontr%C3%B4leur>
5. <https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation>
6. <https://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours04.html>
7. <https://www.institut-numerique.org/32-modelisation-des-differents-services-de-la-plateforme-51f7a7fa11a36>
8. <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/diagramme-de-classe/>
9. <http://www.uml-sysml.org/diagrammes-uml-et-sysml/diagramme-uml/diagramme-de-classe/>
10. <https://www.youtube.com/watch?v=mz8BtQX45e8>
11. https://fr.wikipedia.org/wiki/Diagramme_%C3%A9tats-transitions
12. <https://www.codeflow.site/fr/article/java-mqtt-client>
13. <https://www.lemagit.fr/conseil/Internet-des-Objets-bien-comprendre-MQTT>
14. <https://www.ibm.com/fr-fr/cloud/ai>
15. https://en.wikipedia.org/wiki/Eclipse_Paho
16. https://iot.goffinet.org/iot_protocole_mqtt.html

Livres consultés:

- UML ET JAVA 3ème édition, Michel Lai, édition: Dunod, 2004.
- UML 2.5 par la pratique , Pascal Roques, édition: eyrolles, 2018.
- Modélisation de systèmes complexe avec SysML, Pascal Roque, édition: eyrolles