

# *Introduction Générale*

Au cours de ces dernières années, Les Nouvelles Technologies de l'Information et de la Communication (NTIC) ont bouleversé notre vie quotidienne.

Le web est sans nul doute une technologie majeure du 21<sup>ème</sup> siècle. Et si sa nature, sa structure et son utilisation ont évolué au cours du temps, force est de constater que cette évolution a également profondément modifié nos pratiques commerciales et sociales.

Depuis quelques années, le besoin d'observer, d'analyser et de contrôler des phénomènes physiques sur des zones étendues est essentiel pour de nombreuses applications environnementales et scientifiques.

Actuellement avec l'utilisation des capteurs environnementaux, il est possible de contrôler et d'enregistrer les conditions environnementales et les activités dans les zones protégées. Ce type de capteurs dédié à la surveillance de l'environnement permet d'étendre le contrôle et la gestion à l'espace environnant, en contrôlant la température et l'humidité ainsi que les incidents inévitables dans la nature générés par les êtres vivants ou des événements atmosphériques en fournissant les valeurs via le web

Ce travail s'inscrit dans le cadre de notre projet de fin d'études pour l'obtention du diplôme universitaire en informatique appliqué à la gestion effectué au sein du centre de développement *tunisiawebdev*. Au cours de notre stage, la société d'accueil nous a confié la conception et la réalisation d'une application web dans un milieu professionnel.

Dans notre projet, nous nous sommes intéressés donc au développement d'une application web qui sert à gérer le flux de données généré par des capteurs mobiles. Nous avons alors eut l'occasion d'étudier les réseaux de capteurs mobiles des techniques de déploiement. Des nouvelles techniques de localisation des capteurs ont été proposées permettant d'optimiser la diffusion de l'information délivrée ainsi que la qualité de cette information. Toutefois, une étude bibliographique a montré que les travaux et les recherches antérieurs sur ce thème restent encore assez limités.

Le présent rapport est organisé en quatre chapitres. une introduction générale, dans laquelle sont décrite la problématique des capteurs, ses applications potentielles et les objectifs de notre travail.

Dans le premier chapitre « Etude Préalable », nous présentons le cadre de notre stage de projet de fin d'étude à savoir l'organisme d'accueil : la société *tunisiawebdev*, le sujet sur lequel portera notre projet. Ainsi que nous nous proposons d'analyser l'existant et nous

donnons un aperçu sur quelques applications web. Enfin, nous critiquons l'existant avec proposition d'une solution.

Le deuxième chapitre «Spécifications des besoins», sera consacré à l'identification des acteurs du futur système, ses besoins fonctionnels et non fonctionnels ainsi que le diagramme de cas d'utilisation général.

Dans le troisième chapitre « Conception », nous expliquons le déroulement de notre application, la méthodologie adoptée, les diagrammes de séquence, ainsi que le diagramme de classe complet.

Le dernier chapitre «Réalisation » présentera les environnements : matériel et logiciel utilisés en donnant une description des interfaces de l'application

Enfin, nous clôturons ce rapport avec une conclusion dans laquelle nous résumons notre travail avec exposition de quelques perspectives futures.

# *Chapitre 1 : Étude*

## *Préalable*

## Introduction

Dans ce chapitre, nous introduisons le cadre de notre projet de fin d'études, à savoir la société *tunisiawebdev* et le sujet qui nous a été communiqué par le responsable de la société. Nous analysons par la suite l'existant tout en proposant une solution.

### I. Présentation de la société :tunisiawebdev

Chaque organisation possède ses propres spécificités et se distingue des autres structures qui l'entourent. Il est donc nécessaire de présenter notre société accueillante sous ses différents aspects organisationnels et fonctionnels afin d'avoir une idée précise sur la nature de ses activités ainsi que les relations, souvent complexes, qu'elle peut entretenir avec son environnement aussi bien interne qu'externe.

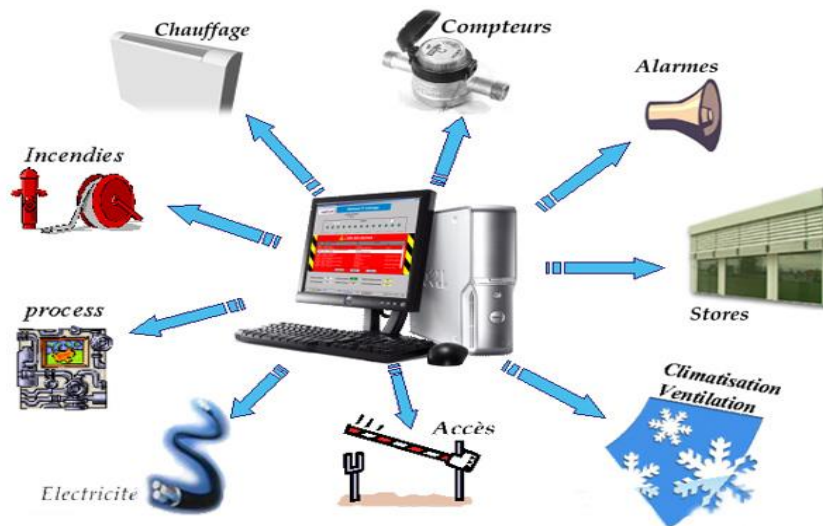
**tunisiawebdev** est un organisme de formation professionnelle à Tunis destiné aux professionnels en informatique. La société est constituée d'ingénieurs informaticiens spécialistes de leur domaine, et qui disposent d'une expérience dans leurs activités d'expert(s).

**tunisiawebdev** offre des formations informatiques indépendantes et larges s'adressant principalement aux professionnels en informatique. Les filières de formation Open Source, UNIX/Linux, Microsoft, Java, Web, Réseaux, et Oracle sont les principaux sujets.

### II. Présentation de projet

#### 1. Sujet :

Notre sujet est intitulé "Conception et développement d'une application web de gestion de données dans un réseau de capteurs ".Il s'agit d'une application qui permet de gérer le flux de données généré par des capteurs tels que la température, la pression et la pollution pour aider les superviseurs de l'environnement à prendre des décisions rapides en cas d'un problème sérieux sans être obligés de quitter leurs postes de travail.



**Figure1 : Aperçu d'un réseau de capteur**

Pour concevoir et développer cette application, nous devons respecter les spécifications suivantes :

- 1- définir et rassembler les exigences (fonctionnelles et non-fonctionnelles).
- 2- trouver les méthodologies de conception.
- 3- choisir et justifier le langage de développement.
- 4- scinder le projet en deux parties:

**La partie Back-office** : est conçue pour l'administrateur, elle permet la gestion des :

- Utilisateurs et leurs privilèges.
- Sessions et leurs journaux.
- Capteurs et leurs paramètres.

**La partie Front-office** : est conçue pour l'utilisateur ou le superviseur, elle permet la surveillance des :

- des états des capteurs et les emplacements.
- lectures des capteurs et de la mesure.
- Captures et les contrôles capteurs (statistiques, alertes).

Mets des références pour toutes les notions ici bas

## 2. Notions de base

### 2.1 Un capteur :

Un **capteur** est un dispositif transformant l'état d'une grandeur physique (température, pression, position, concentration, etc.) Observée en une grandeur utilisable ou signal, telle

qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille. Ce signal peut être lu par un appareil ou par un observateur. Il nous permet de surveiller et de réagir en monde réel.

## **2.2 Les réseaux de capteurs :**

Les réseaux de capteurs sont des systèmes qui regroupent plusieurs capteurs afin de couvrir une zone cible. Cette zone peut être géographique ou délimitée par un système plus ou moins étendu. Les réseaux de capteurs connaissent de multiples applications telles que la météorologie environnementale, urbaine ou industrielle.

En réseau, les capteurs peuvent interagir entre eux et avec un système externe (par exemple Internet) par le biais de communication sans fil. Les réseaux de capteurs sans-fils concentrent les dernières avancées technologiques et représentent l'opportunité de nouvelles applications.

## **2.3 Les capteurs mobiles :**

De nos jours, la plupart des Smartphones sont équipés de nombreux capteurs embarqués qui ont une capacité moyenne de calcul.

Le Smartphone est un véritable micro-ordinateur de poche doté de fonctionnalités multimédias avancées et truffés de capteurs :

- GPS
- accéléromètre
- Gyromètre (confondu avec le Gyroscope)
- magnétomètre
- capteurs de pression
- capteurs de lumière ambiante
- capteur de proximité.

Le Smartphone combine les données captées par ces différents dispositifs(capteurs) afin d'estimer une probabilité de localisation en fonction du temps, et de la direction prise à partir d'un point de départ prédéterminé pour des utilisations variées (boussole, navigation, console de jeux portable, etc).

<b>Caméra 1</b> (arrière)  <b>Caméra 2</b> (avant) <b>Écran tactile</b> <b>Microphones</b>  <b>Détecteur de proximité</b> <b>Capteur de luminosité</b>	⇒ photo, vidéo, lecteur flash code, loupe, scanner, mesures, Réalité Augmentée. ⇒ auto-photo, visiophonie ⇒ clavier, dessin, jeux ⇒ téléphone, dictaphone, vidéo  ⇒ pour désactiver l'écran tactile ⇒ pour s'adapter à l'éclairage ambiant
<b>Magnétomètre</b> *  <b>Gyroscope</b> *  <b>Accéléromètre</b> *  * 3 axes donc 3 capteurs <b>Capteur de pression</b>  Les antennes: <b>GPS</b> <b>2G, 3G, 4G</b> <b>Wifi</b>  <b>Bluetooth</b>	⇒ boussole, détecteur de métaux, Réalité Augmentée.  ⇒ jeux, localisation, Réalité Augmentée.  ⇒ jeux, orientation Smartphone, niveau, Réalité Augmentée. stabilisation d'image.  ⇒ Baromètre, altimètre.  ⇒ localisation ⇒ téléphone, internet (qq km) ⇒ communication moyenne distance (20 m), internet, imprimante. ⇒ communication courte distance (qq m), oreillette, imprimante.
<b>NFC</b>	⇒ communication ultra distance courte (10 cm), porte-monnaie et carte d'abonnement virtuels
<b>Radio FM</b>	⇒ radio

**Tableau 1: Les types de capteurs intégrés dans un Smartphone et leurs utilisations**

## 2.4 Application de capteurs

Aujourd'hui à travers la technologie des capteurs, nous pouvons vivre dans des villes intelligentes.

Les réseaux de capteurs sont de plus en plus utilisés dans l'environnement grâce notamment aux derniers développements réalisés dans le domaine des technologies sans-fils (« Wireless »). En effet, le besoin d'observer, d'analyser et de contrôler des phénomènes



physiques sur des zones étendues est essentiel pour de nombreuses applications environnementales et scientifiques.

Cette nouvelle manière d'envisager la météorologie fait émerger de nouvelles problématiques technologiques, par exemple sur l'autonomie énergétique des capteurs, et les nouveaux types d'application nous permettant de mieux connaître notre environnement et d'anticiper les problèmes de sécurité, de pollution, de risques naturels, de défaillances, de maintenances, ou plus généralement de tous phénomènes non désirés qui pourraient être anticipés.

### III. Etude de l'existant

On peut constater que notre pays est en train de rencontrer une variété de problèmes. Les **activités humaines** sont à l'origine de ces problèmes environnementaux qui affligent la planète et les êtres humains.

Ces activités humaines ont trois catégories principales d'impact:

- L'épuisement des ressources
- La pollution
- La destruction des habitats


Ces trois conséquences directes des activités humaines sont à l'origine des diverses crises environnementales qui ont des effets locaux autant que globaux. Par suite, ces crises sont regroupées sous le terme de crise environnementale globale.

D'une façon générale, les problèmes environnementaux peuvent inclure :

- ✚ L'érosion de la diversité biologique (incluant la perte des services éco-systémiques)
- ✚ La dégradation des océans (notamment l'acidification)
- ✚ Le réchauffement planétaire et les changements climatiques (incluant la hausse du niveau des océans)
- ✚ L'altération des cycles biogéochimiques (azote, phosphore, eau)
- ✚ La diminution de la qualité des sols
- ✚ La déforestation
- ✚ La désertification
- ✚ La raréfaction de l'ozone stratosphérique (trou de la couche d'ozone)

Face aux problèmes environnementaux, beaucoup cherchent à trouver des solutions technologiques capables de transcender les limites de notre modèle actuel. En effet, Les nouvelles technologies sont désormais accessibles à tous et permettent aujourd'hui d'accéder à de nombreuses données scientifiques. Grâce aux images satellitaires, par exemple, avec l'arrivée de Google Earth en 2005, toute personne bénéficiant d'un ordinateur et d'une connexion internet peut se rendre compte du monde et de son environnement.

Dans cette optique de recherche, diverses applications ont été cadrées comme :

 **the SUPREME system (Système de surveillance et de prévention des impacts sanitaires des événements météorologiques extrêmes)** est une application web open source intégrée qui exploite les logiciels libres pour la surveillance en temps réel et la prévention des impacts des événements météorologiques extrêmes sur la santé publique.

Sa première utilisation sur le terrain impliquait des vagues de chaleur. Ce système d'aide à la décision repose sur des logiciels libres et comprend quatre modules:

- ➔ acquisition et intégration des données
- ➔ analyse et signalisation des risques
- ➔ application cartographique
- ➔ diffusion de l'information.

Le système est mis à la disposition des spécialistes de la santé par l'intermédiaire d'un portail d'information sécurisé sur Internet et permet d'accéder aux prévisions météorologiques, aux indicateurs historiques et en temps réel (y compris les décès et les hospitalisations), aux alertes et aux diverses données cartographiques utilisées pour mener des activités de prévention et lancer des mesures d'urgence.



**Figure : Page d'accueil du système SUPREME, volet surveillance**

L'interface se compose de la zone d'affichage de la carte, des outils de navigation cartographique (en haut à gauche), un diagramme de statistique (à droite) et du menu (en haut à droite) permettant de fixer les paramètres de statistiques et de partager la carte

<http://www.ors-idf.org/dmdocuments/2015/cahier/N73.pdf>

🚦 **CityEvents** est une application contextuelle et adaptée aux contenus. Il décrit un système multimédia mobile pour afficher des événements culturels géolocalisés. Il fournit les informations nécessaires à l'utilisateur d'une manière simple et intuitive, en captant les intérêts de l'utilisateur et en essayant de suggérer des événements à proximité présentés dans une carte visuelle qui offre des informations détaillées sur demande.

Une liste d'événements commandés par intérêt est également disponible. La quantité des données transférées entre le client et le serveur est prise en compte et est réduite par des mécanismes de mise en cache et un système de compression approprié sur le côté serveurs. Le client ne fera que des demandes au serveur si nécessaire, dans des threads d'arrière-plan afin d'améliorer l'expérience utilisateur.

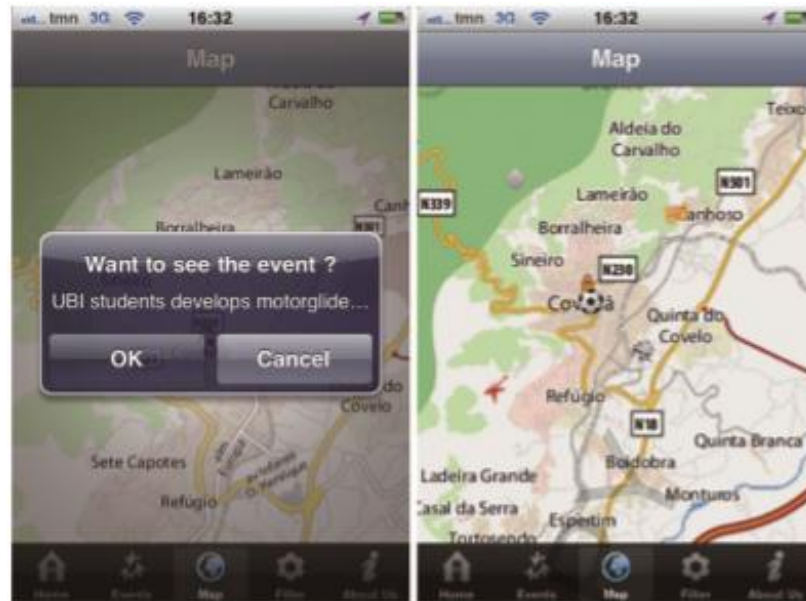


Figure : Cartes avec des événements d'alerte et de notation

<http://netgna.it.ubi.pt/files/mis00157.pdf>

## 1. Critique de l'existant :

La critique des applications constitue une étape utile et importante. Elle a pour but de porter un jugement objectif afin de déceler les insuffisances éventuelles rencontrées au cours de l'étude de l'existant en vue de proposer une application plus fiable que l'ancienne.

Au terme de l'analyse de l'existant, il convient d'avouer que :

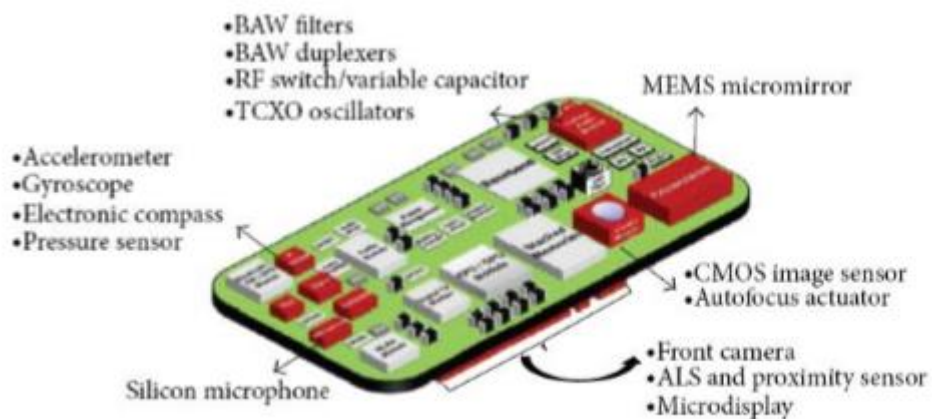
- ✚ Le coût des capteurs exploitées pour ces applications est élevé et le fait d'utiliser un grand nombre des capteurs statiques demandé pour couvrir la même superficie couverte par un unique périphérique mobile peut être trop coûteux à déployer.
- ✚ La grandeur est quantifiée par un nombre défini de positions, donc la résolution est toujours finie.

## 2. Solution Proposé :

La majorité des appareils modernes sont bien plus que des simples outils pour communiquer ou naviguer sur internet. Ils ont des capacités sensorielles, matérialisées par leurs capteurs.

Ces capteurs nous fournissent des informations brutes avec une grande précision, qu'il est possible d'interpréter pour comprendre les transitions d'état que vit le terminal. On trouve par

exemple des accéléromètres, des gyroscopes, des capteurs de champ magnétique, etc. Tous ces capteurs nous permettent d'explorer de nouvelles voies, d'offrir de nouvelles possibilités aux utilisateurs. En effet, les téléphones mobiles peuvent fournir une couverture là où des capteurs statiques sont difficiles à déployer ainsi que l'utilisateur aura la possibilité d'utiliser son terminal pour améliorer les fonctionnalités de l'application.



**Figure 2 : Capteurs intégrés dans un Smartphone**

<http://www.internetactu.net/2015/01/22/applications-de-sante-13-que-captent-les-capteurs/>

Notre solution a pour objectif d'exploiter les différents capteurs existant dans les Smartphones modernes. En effet, elle consiste en la conception et le développement d'une application qui va nous permettre de collecter les données détectées par les capteurs mobiles afin de mesurer en temps réel la pression atmosphérique, la température, l'humidité, la qualité de l'air, le taux de CO<sub>2</sub> ou encore le niveau sonore afin de fournir une prévision détaillée...

### **Conclusion :**

Dans ce chapitre, nous avons mis en place l'organisation où j'ai passé mon stage, présenté le cadre de sujet de notre projet, une analyse de l'existant et sa critique et on se termine par proposition d'une solution. Dans le prochain chapitre, nous présentons une identification complète des exigences fonctionnelles et non-fonctionnelles ainsi qu'une analyse détaillée pour les cas d'utilisation de notre système.

# *Chapitre 2 : Spécifications des besoins*

## Introduction :

Dans le cadre de ce chapitre, nous allons présenter la spécification de besoins qui consiste en la qualification des besoins fonctionnels et non fonctionnels attendus du système afin de mieux comprendre le projet.

### I. Spécification des besoins :

#### 1. Les acteurs :

Un acteur représente l'abstraction d'un rôle joué par des entités externes. Dans notre application, nous distinguons principalement deux types d'utilisateurs, avec des exigences différentes (administrateur et superviseur)

✚ L'utilisateur ou superviseur : c'est la personne qui utilise l'application pour consulter les informations.

✚ L'administrateur : c'est la personne qui gère l'administration de l'application.

#### 2. Spécification fonctionnelle :

Notre application offre plusieurs fonctionnalités aux utilisateurs. Elle permet à l'administrateur de :

- Gérer les comptes utilisateurs (ajouter, modifier, supprimer, consulter).
- Gérer les capteurs et leurs paramètres (ajouter, modifier, supprimer, consulter).
- Gérer les sessions
- Gérer les types des capteurs et leurs paramètres standards ...

Elle permet au superviseur de :

- Rechercher des données des capteurs à l'aide d'un filtre temporel ou spatial.
- Localiser les capteurs.
- Visualiser les informations du capteur.
- Visualiser les statiques.
- Éditer des alertes qui seront émises par des capteurs ou détecteurs en se basant sur certains critères.
- Gérer les captures

Le diagramme de cas d'utilisation ci-dessous illustre les différentes fonctionnalités offertes aux utilisateurs (administrateur ou superviseur) à travers notre application.

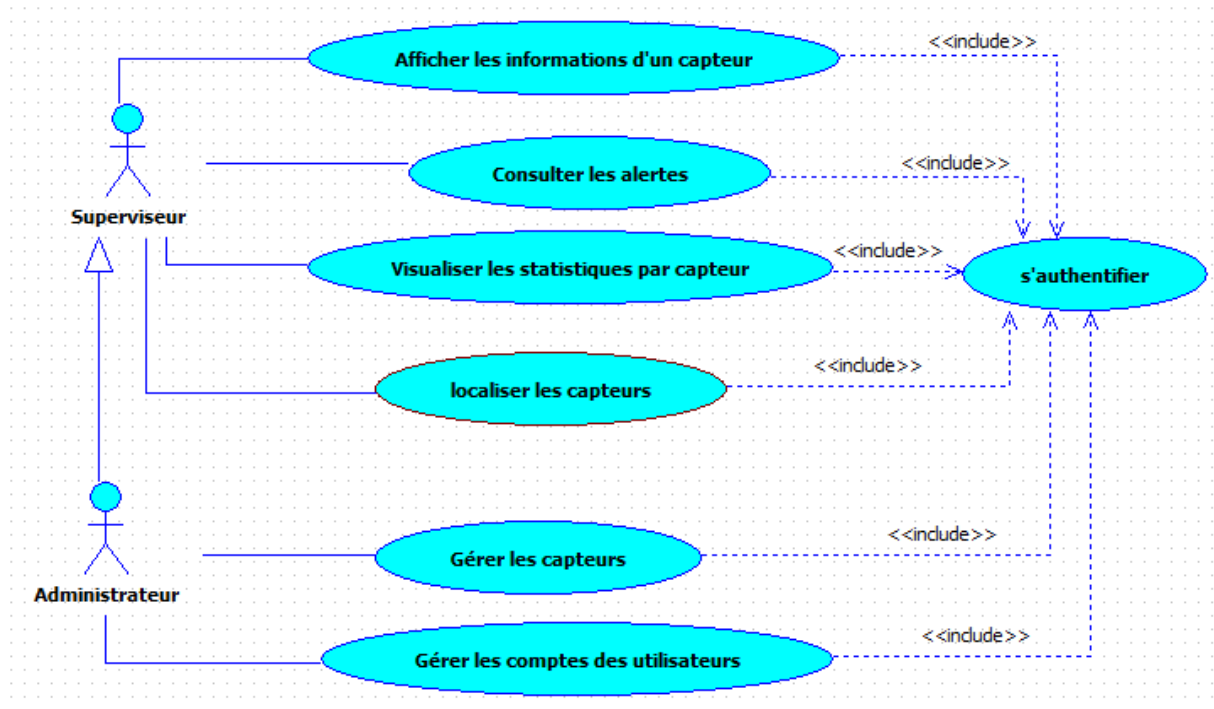


Figure : Diagramme de cas d'utilisation général

## IV. Diagrammes de cas d'utilisations

### 1. Diagramme de cas d'utilisation de l'administrateur

L'administrateur est le responsable de l'application Back-office .C'est le gérant de l'application, il a une visibilité totale sur les bases de données. Il a pour tâches de gérer tout le système. Il spécifie les utilisateurs et les droits de chacun. Il gère les dispositifs de détection et leurs types, les unités de mesure utilisées, les villes et les différents états (état de l'utilisateur, l'état du privilège, et l'état du capteur)



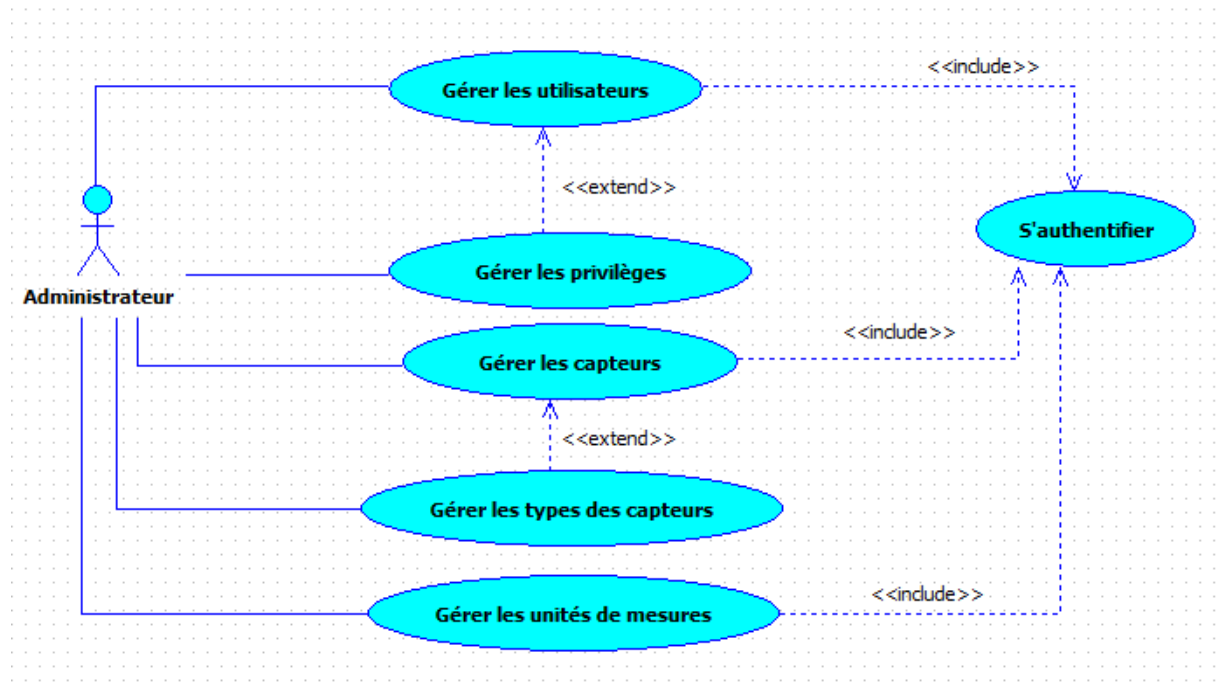


Figure : Diagramme de cas d'utilisation de l'administrateur

### 1. Cas d'utilisation Gestion de l'utilisateur :

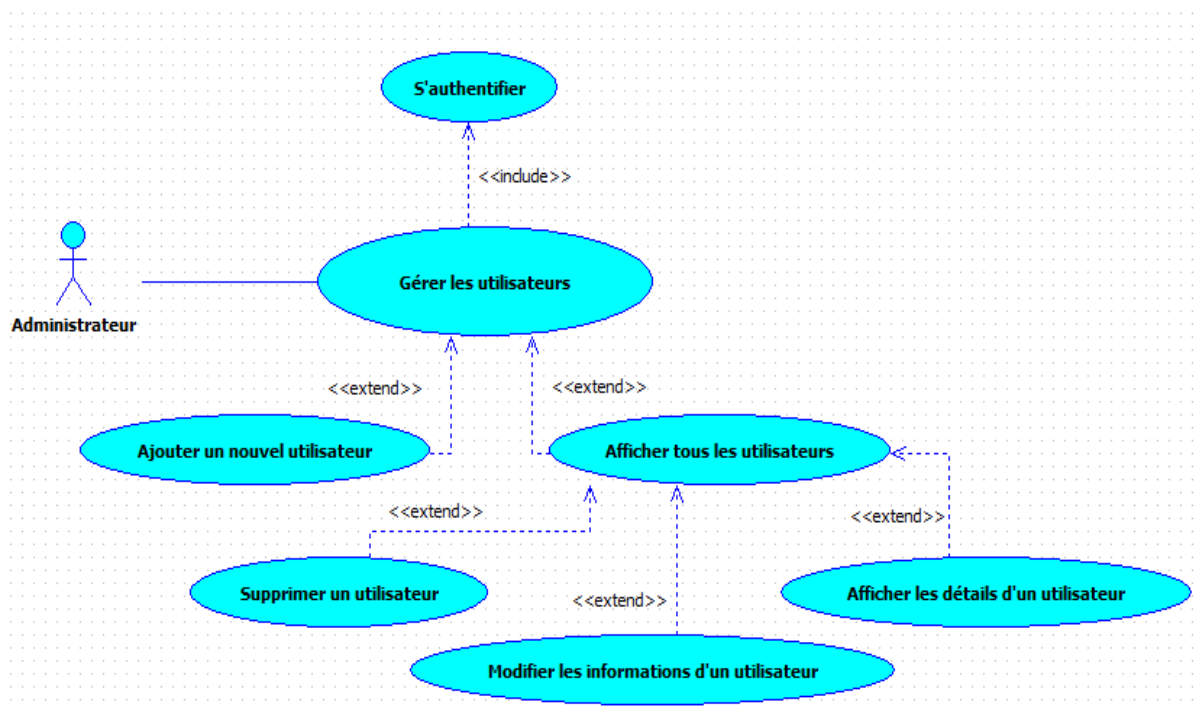
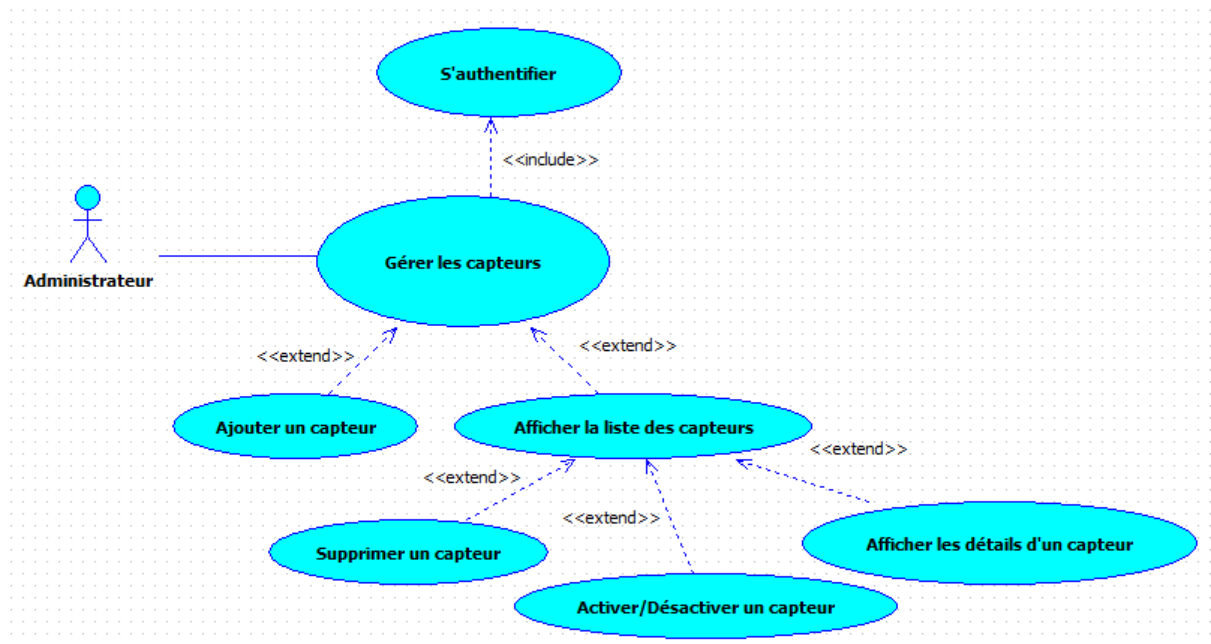


Figure : Diagramme de cas d'utilisation “Gestion des Utilisateurs”

Titre :	Ajouter un nouvel utilisateur		
Objectif	permet à l’administrateur d’ajouter un nouvel utilisateur à l'application en précisant son privilège associé (Superviseur, Administrateur).		
Résumé	L'administrateur doit remplir le formulaire d'ajout avec les informations adéquates et attribuer le(s) privilège(s) adéquat(s).		
Acteur	Administrateur		
Description d’enchainement			
Préconditions		Post-Conditions	
L'administrateur doit avoir des privilèges assez élevés quand il s'authentifie.		Le nouvel utilisateur est ajouté à la base de données	
Scénario principal			
<div>1. remplit le formulaire et clique sur le bouton« Ajouter »</div> <div>2. Le système valide les entrées</div> <div>3. Le système enregistre le nouvel utilisateur dans la base de données</div> <div>4. Le système redirige l'administrateur à la page de la liste des utilisateurs restants</div>			

## 2. Cas d'utilisation Gestion des capteurs :



**Figure : Diagramme de cas d'utilisation “Gestion des Capteurs “**

<b>Titre :</b>	Afficher les détails d'un capteur		
<b>Objectif</b>	Ça consiste en la consultation des détails d'un capteur particulier à partir de la liste des capteurs		
<b>Résumé</b>	L'administrateur doit afficher la liste des capteurs, puis il choisit le capteur qu'il veut afficher ses détails et il clique sur le bouton d'affichage		
<b>Acteur</b>	Administrateur		
<b>Description d'enchainement</b>			
<b>Préconditions</b>		<b>Post-Conditions</b>	
L'administrateur est authentifié, la liste des capteurs s'affiche		L'administrateur dispose de toutes les informations du capteur sélectionné	
<b>Scénario Principal</b>			
<div>1. L'administrateur clique sur le lien de la liste des capteurs</div> <div>2. Le système charge les capteurs enregistrés</div> <div>3. L'administrateur sélectionne le lien « Afficher capteur »</div> <div>4. Le système charge la requête et renvoi les données de capteur sélectionné</div>			

## 2. Diagramme de cas d'utilisation d'utilisateur :

Le superviseur est responsable de l'application Front-Office, comme la gestion des paramètres de base du capteur, l'emplacement des capteurs, les différentes lectures des captures et les types de simulation des capteurs.

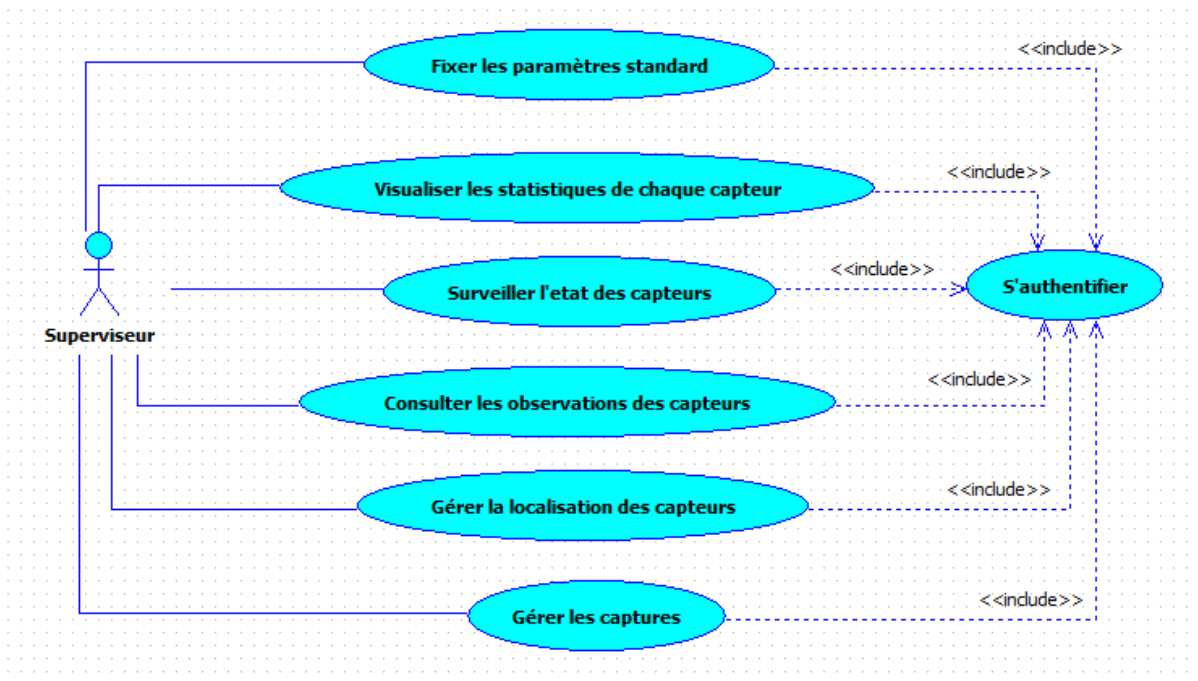


Figure : Diagramme de cas d'utilisation De superviseur

## 1. Cas d'utilisation Gérer les captures :

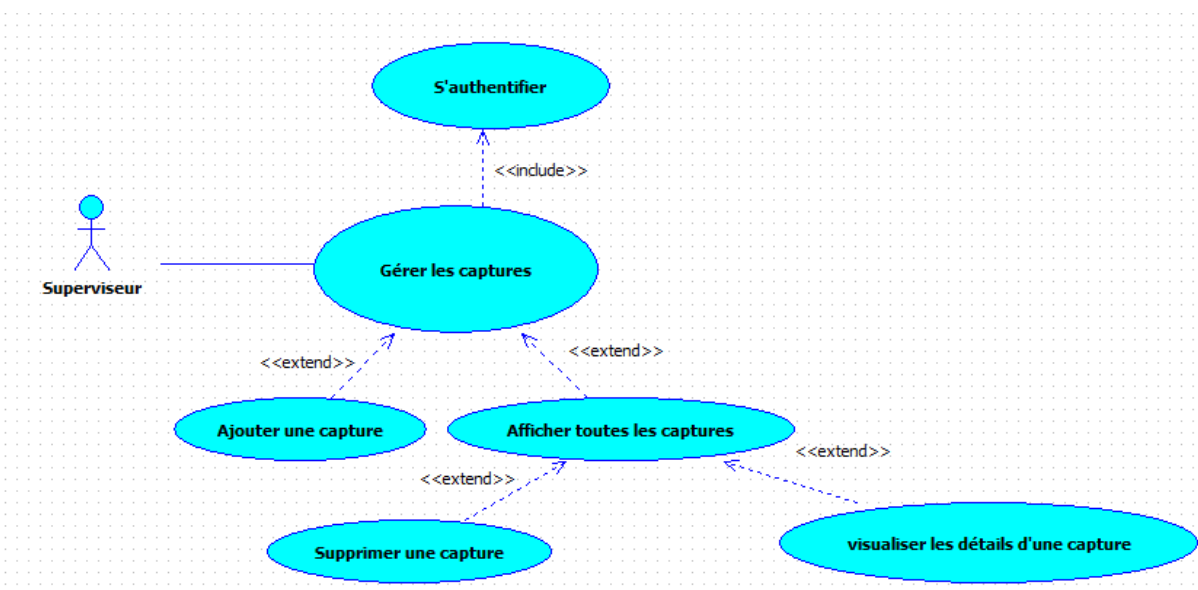


Figure : Diagramme de cas d'utilisation 'Gérer les captures '


Titre :	Consulter les observations des capteurs		
Objectif	Ça consiste à consulter à observer les capteurs sur une carte géographique avec des marqueurs de couleur qui indiquent l'état des captures et leurs lieux.		
Résumé	Le superviseur doit cliquer sur le lien de simulation et choisir le dispositif de détection ou de type ou l'emplacement pour obtenir les observations recherchées sur le plan présenté par différents marqueurs. Pour afficher les détails d'observation, il doit cliquer sur les marqueurs de cartes.		
Acteur	Superviseur		
Description d'enchainement			
Préconditions		Post-Conditions	
Le superviseur doit être authentifié		Le superviseur reçoit une carte avec des marqueurs de couleurs présentant les observations des capteurs.	
Scénario Principal			
<div>1. Les clics de superviseur sur le lien de simulation</div> <div>2. Le contrôleur choisit le type de simulation et / ou le dispositif de capteur, et / ou de localisation</div> <div>3. Le système analyse la requête et renvoie le résultat sur la carte</div> <div>4. Le superviseur peut cliquer sur un marqueur pour afficher les détails d'observation</div>			

**Description textuelle "Consultez les observations des capteurs"**

### 3. Spécification non fonctionnelle

Après avoir déterminé les besoins fonctionnels, nous présentons ci-dessous l'ensemble des contraintes à respecter pour garantir la performance du système, et donc fournir un produit performant qui respecte les exigences de l'utilisateur et qui peut faire face à des risques de panne ou de dysfonctionnement.

Ce sont les besoins qui permettraient d'améliorer la qualité des services d'application comme la connectivité et l'ergonomie des interfaces ainsi que l'amélioration du temps de réponse, Parmi ces besoins, nous citons :

 Facilité d'utilisation: l'application doit être simple, facile à comprendre par l'utilisateur de

l'ordinateur.

- + Conception : l'interface utilisateur doit être d'une belle apparence, facile à utiliser et suit les normes des modèles d'applications.
- + Fiabilité : notre application doit être digne de confiance et fiable et fournit toutes les fonctions nécessaires pour les utilisateurs
- + Sécurité : l'application doit garantir la sécurité des données et la vie privée en utilisant des sessions afin d'éviter la perte de données ou leur modification par des pirates.
- + Performance : la vitesse du fonctionnement de système doit être à un niveau supérieur pour garantir une bonne qualité de service.
- + Évolutivité : la gestion efficace des ressources mémoire.
- + Optimisation : Prendre soin d'utiliser aussi peu de puissance de traitement et autonomie de la batterie que possible en raison de la faible capacité de la batterie d'un téléphone mobile.
- + Efficacité : le logiciel doit avoir des résultats concis.
- + Maintenabilité : Les différents modules de l'application doivent être lisibles et compréhensibles afin de les maintenir et de les mettre à jour rapidement et facilement.
- + Réutilisation : cet outil doit permettre l'ajout et l'édition des extensions futures.

## Conclusion

Dans ce chapitre, nous avons expliqué clairement les différents besoins fonctionnels et non-fonctionnels du système. Puis, nous avons étudié le comportement en utilisant quelques cas d'utilisation.

## *Chapitre 3 : Conception*





## **Introduction :**

Dans le cadre de ce chapitre, nous nous intéressons à la conception, qui est une phase initiale dans la création et la mise en œuvre de notre projet. Cette activité d'analyse permet de traduire les besoins fonctionnels et les contraintes dans un langage plus professionnel et compréhensible par tous les individus intervenants dans la réalisation et l'utilisation de l'application. En fait, elle représente une étape importante de réflexion dans le cycle de développement logiciel après la phase de spécification des besoins.

Tout au long de ce chapitre, nous allons présenter en détails cette phase en la subdivisant en deux parties : la première dédiée à la conception globale du projet et la seconde sera consacrée à la conception détaillée en se basant sur quelques diagrammes UML statiques et dynamiques.

## **I. Conception Globale :**

### **1. Choix de méthode de conception :**

Afin de réaliser un bon système, une étude et une conception normalisée selon la norme de modélisation universellement reconnue est nécessaire. Une étude comparative entre UML et Merise permet de choisir un langage de modélisation.

#### **2.1 MERISE :**

(Méthode d'Etude et de Réalisation Informatiques pour des Systèmes d'Entreprise) est une méthode de conception, de développement et de réalisation de projets informatiques. Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. C'est d'ailleurs son point fort. Merise est organisée sous la forme de trois cycles, effectués simultanément : le cycle de vie, le cycle de décision et le cycle d'abstraction

#### **2.2 UML :**

(Unified Modeling Language) est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée afin de visualiser la conception

d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

UML utilise des diagrammes pour représenter chaque aspect d'un système c'est à dire statique, dynamique, etc., tout en s'appuyant sur la notion de l'orienté objet qui est un véritable atout pour ce langage.

UML est idéal pour :

- concevoir et déployer une architecture logiciel développée dans un langage objet (Java, C++, PHP..).
- pour modéliser les données (les diagrammes de classe réduit sans méthodes et stéréotypé en entités), mais avec des lacunes que ne présentait pas l'entité relation de Merise.
- pour modéliser le fonctionnement métier (le diagramme d'activité et de cas d'utilisation) qui sont des formalismes très anciens qu'avait, en son temps, amélioré Merise



Figure : Unified Modeling Language

Nous pouvons donc dire que la méthode UML répond à toutes les questions que l'on doit se poser pour assurer parfaitement la modélisation de notre projet.

Suite à l'étude ci-dessus, nous allons donc adopter UML comme langage de modélisation puisque nous allons utiliser le concept de l'orienté objet avec le PHP comme langage, pour développer notre application web.

.

## 2. Le cycle de vie d'un logiciel

Le « **cycle de vie d'un logiciel** » (*software lifecycle*), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la **validation** du

développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la **vérification** du processus de développement.

La séquence et la présence de chacune de ces activités dans le cycle de vie dépend du choix d'un modèle de cycle de vie entre le client et l'équipe de développement.

Afin de répondre au mieux aux besoins cités précédemment, nous sommes orientés vers l'utilisation d'un processus de développement en V qui demeure actuellement le plus utilisé.

Le modèle du cycle en V a été imaginé pour pallier le problème de réactivité du modèle en cascade. Ce modèle est une amélioration du modèle en cascade qui permet en cas d'anomalie, de limiter un retour aux étapes précédentes. Les phases de la partie montante doivent renvoyer de l'information sur les phases en vis-à-vis lorsque des défauts sont détectés afin d'améliorer le logiciel.

De plus le cycle en V met en évidence la nécessité d'anticiper et de préparer dans les étapes descendantes les « attendus » des futures étapes montantes : ainsi les attendus des tests de validation sont définis lors des spécifications, les attendus des tests unitaires sont définis lors de la conception, etc.

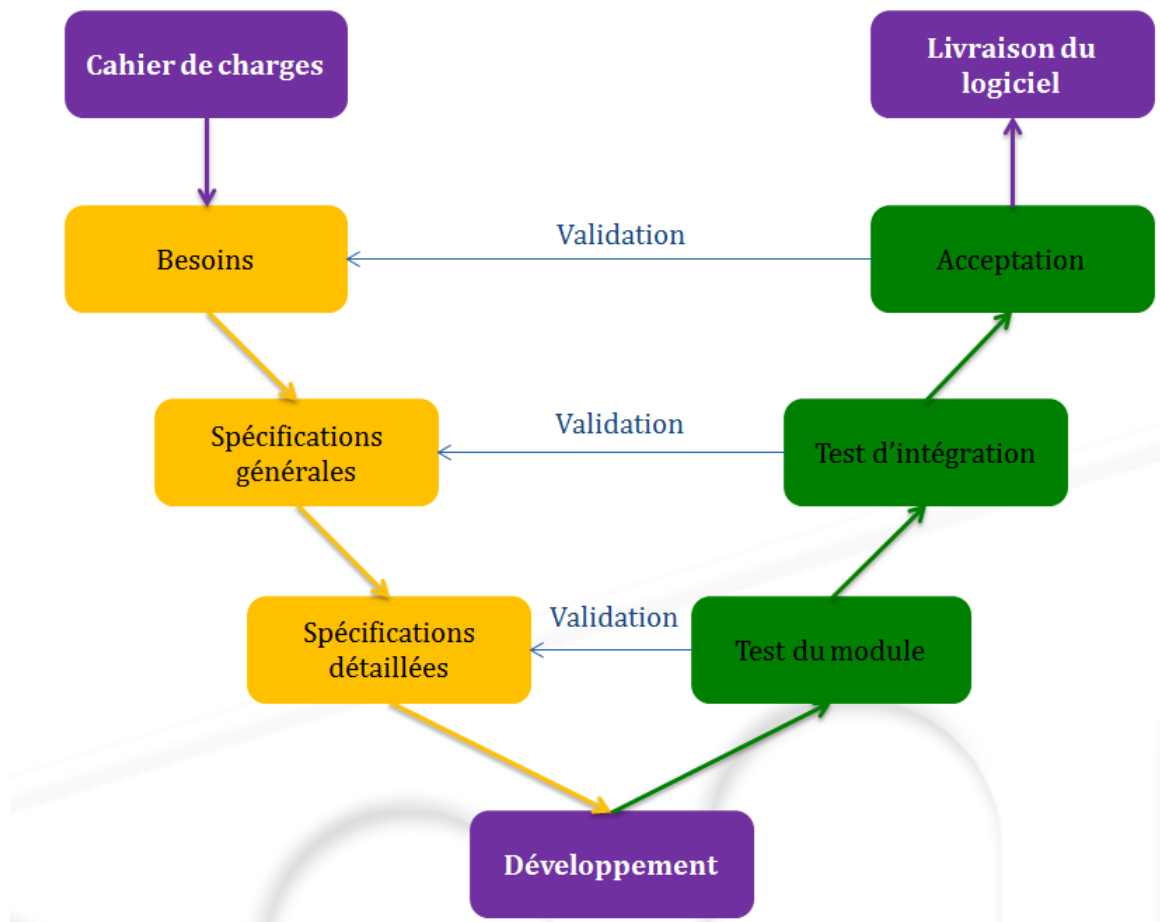


Figure : Modèle du cycle de vie en V

### 3. Le choix de l'atelier de génie logiciel :

Il existe différents logiciels utilisés pour la conception des projets. Parmi ces logiciels, on cite Modelio free Edition, PowerAMC, AgroUML, etc.

Notre outil de conception UML est le logiciel StartUML, qui est un outil de modélisation cédé comme open source par son éditeur, sans limite et gratuit.

### Conception Détaillée :

Une fois la conception globale déterminée, nous procédons à la conception détaillée durant laquelle on détermine les spécifications de chaque élément projet. C'est une vue plus profonde de l'architecture du projet en présentant quelques-uns des diagrammes UML statiques et dynamiques.

#### 1. Diagramme de séquence :

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UnifiedModelingLanguage.

Un diagramme de séquence complète un cas d'utilisation :

- Les diagrammes de cas d'utilisation modélisent à QUOI sert le système, en organisant les interactions possibles avec les acteurs.
- Alors que Les diagrammes de séquences permettent de décrire COMMENT les éléments du système interagissent entre eux et avec les acteurs :
  - ▣ Les objets au cœur d'un système interagissent en s'échangeant des messages.
  - ▣ Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

Après l'étude des cas d'utilisation, nous avons pu dégager les diagrammes de séquences correspondants dont voici les plus importants :

### **II.1.1 Diagramme de séquence du cas « authentification »**

Dans ce diagramme, nous montrons les interactions entre les acteurs et le système :

- ❖ L'administrateur (utilisateur de l'application) saisit son login et son mot de passe.
- ❖ L'administrateur valide la saisie en cliquant sur le bouton « connexion ».
- ❖ Le système vérifie les paramètres saisis.
- ❖ Le système accède à la table utilisateur dans la base de données pour vérifier les paramètres.
- ❖ le système affiche la page d'accueil selon le droit d'accès
- ❖ si login et mot de passe sont invalides, le système retourne au page login avec affichage d'un message d'erreur

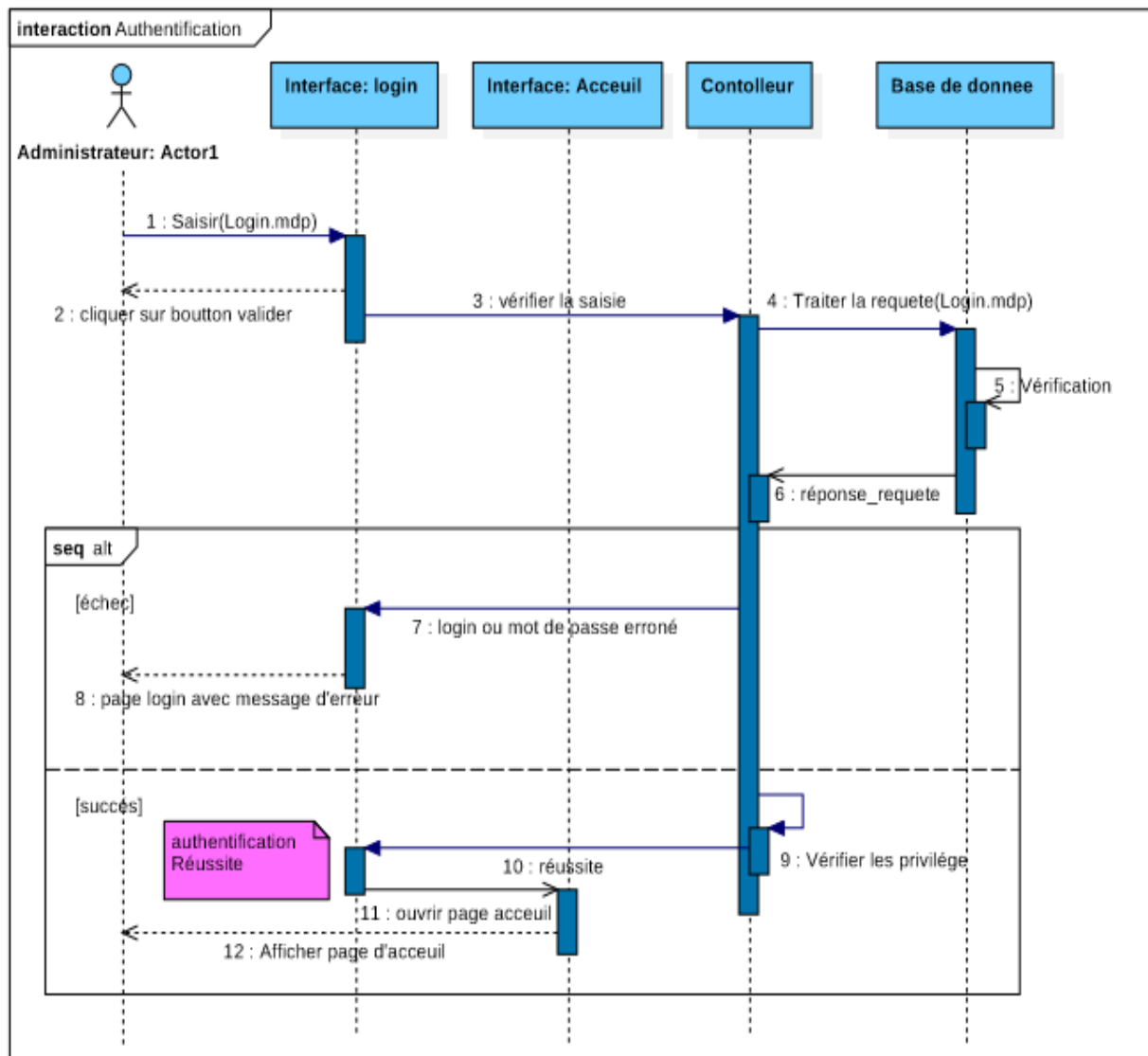


Figure : Diagramme de séquence « authentification »

## II.2 Diagramme de séquence du cas « ajout d'un nouvel utilisateur avec gestion des privilèges »

Dans ce diagramme, nous montrons les interactions entre les acteurs et le système :

- ❖ L'administrateur saisie les données concernant le nouvel utilisateur ajouter
- ❖ L'administrateur demande l'enregistrement avec une clique sur le bouton Ajouter.
- ❖ Le système prend en charge la vérification du type des données et l'insertion dans la base de données
- ❖ si l'utilisateur n'existe pas, le système prend en charge la demande d'ajout.
- ❖ Le système accède à la table Utilisateur, ajoute le nouvel utilisateur et affiche un message de confirmation informe l'utilisateur que l'enregistrement est réalisé avec succès

- ❖ Si l'utilisateur existe, le système affiche un message d'erreur

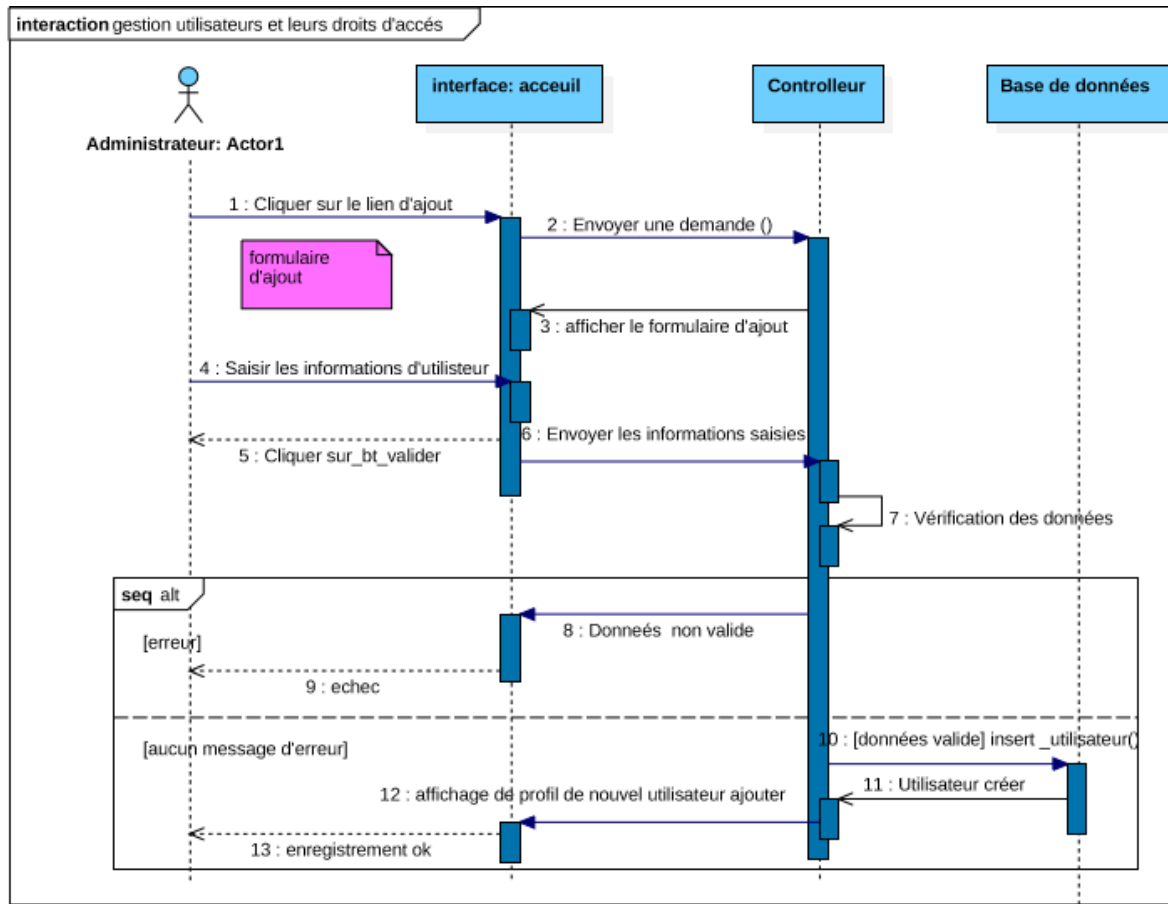


Figure: Diagramme de séquence « Gérer les utilisateurs et les droits d'accès »

## II.2 Diagramme de séquence des cas « Consulter, modifier et supprimer un utilisateur »

Dans ce diagramme :

- ❖ L'administrateur demande la liste des utilisateurs. Une requête sera passée à la base de données pour préparer la liste des objets utilisateurs qui sera chargé par le contrôleur et envoyer par la suite au page d'affichage de liste utilisateurs
- ❖ Une fois la liste est affichée, l'utilisateur peut sélectionner un utilisateur pour le modifier ou le supprimer.
- ❖ La modification génère la page New\_User avec l'utilisateur sélectionné en paramètre pour lister les informations de l'utilisateur sélectionné. Une fois les modifications sont effectuées, l'objet utilisateur modifié sera passé pour exécuter la requête SQL correspondante à la base de données.
- ❖ De même pour la suppression.

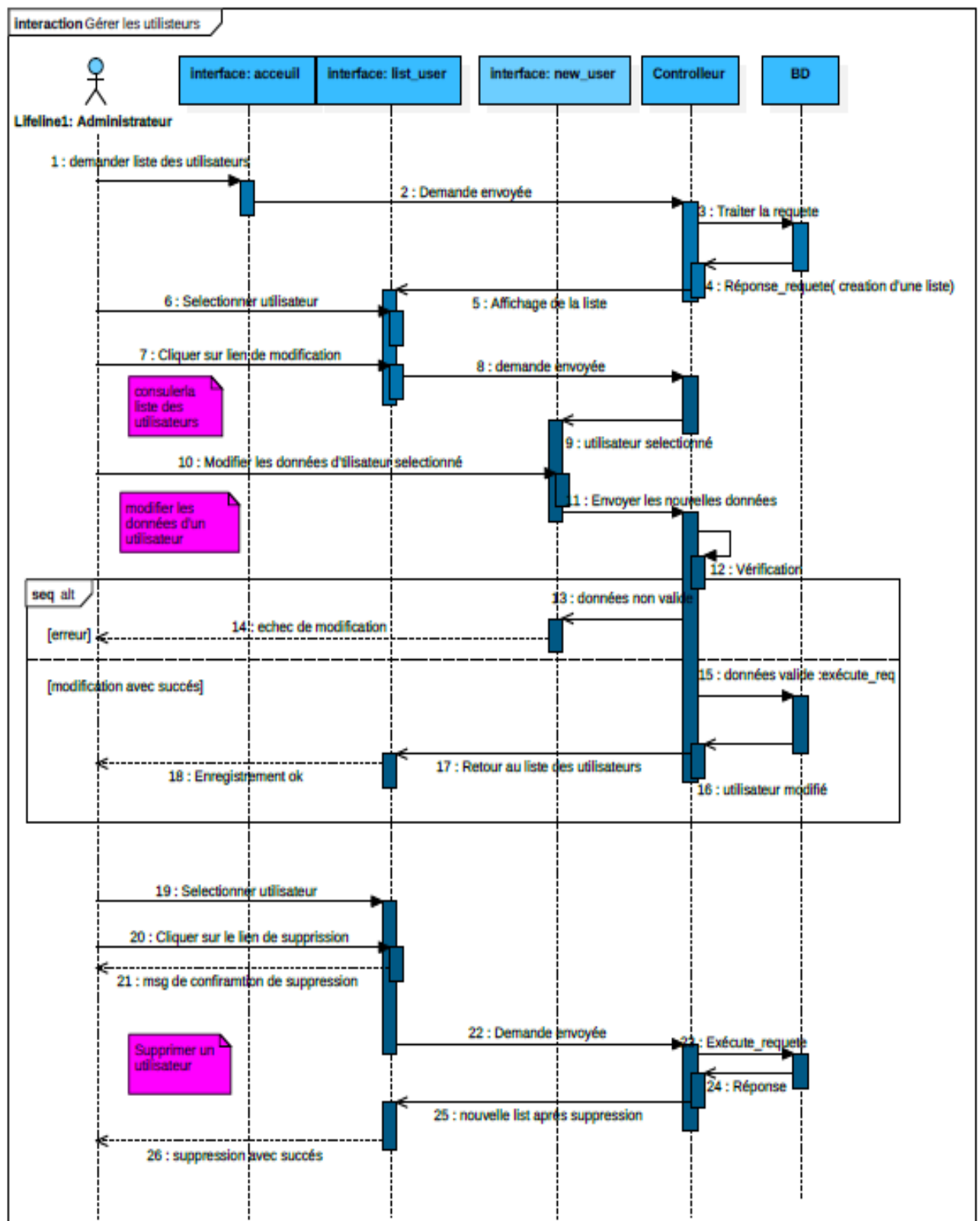


Figure: Diagramme de séquence « Gérer les utilisateurs : consulter, modifier et supprimer»

## II.2 Diagramme de séquence des cas «Consulter, modifier et supprimer un utilisateur »

Dans ce diagramme :



- ❖ Le superviseur choisi parmi la liste un capteur
- ❖ le système prend en charge la demande, envoyé la requête a la base de données et reçoit une liste des valeurs de ce capteur
- ❖ le système ajoute ces valeurs au modèle de statistiques et affiche sur l'écran sous la forme d'un diagramme

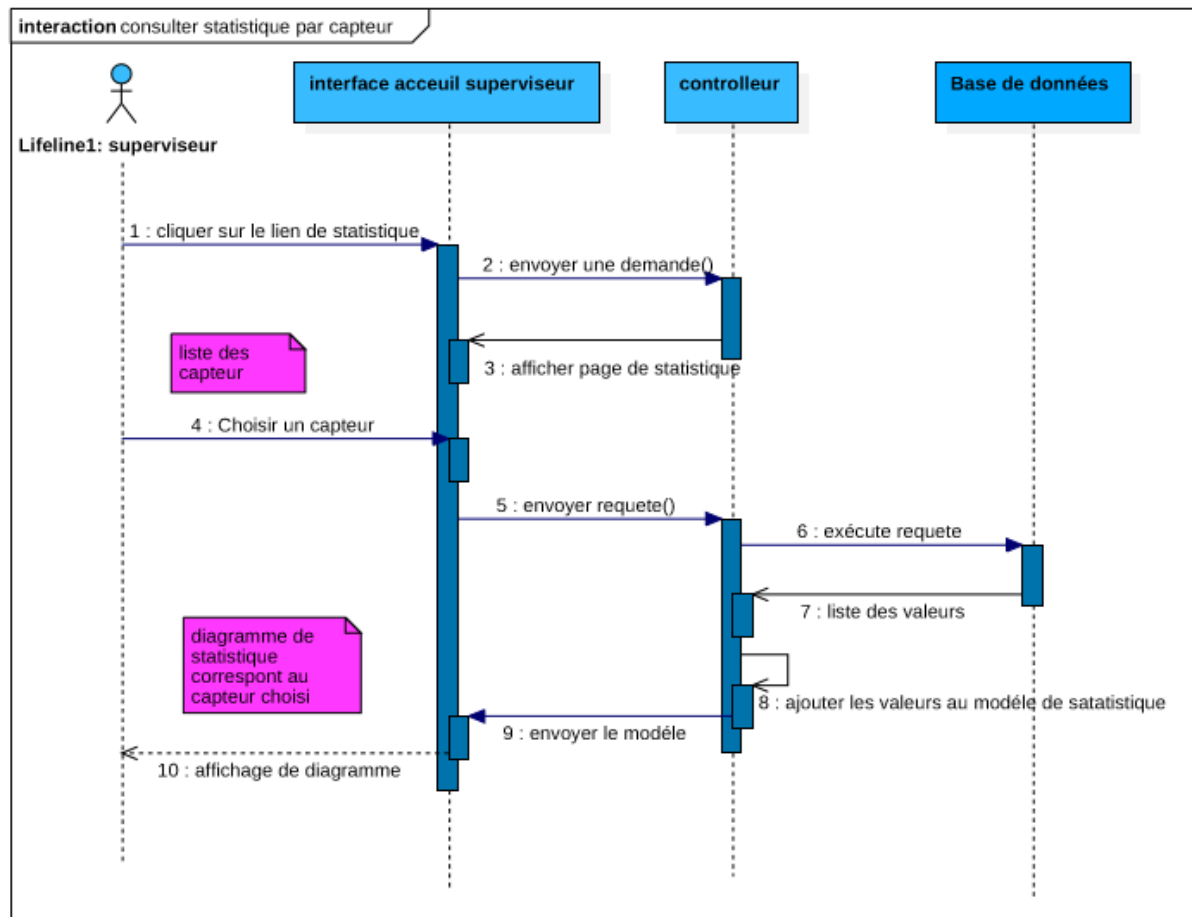


Figure: Diagramme de séquence « consulter statistique par capteur »

## II.2 Diagramme de classe :

Le diagramme de classe est une modélisation statique du système en termes de classes et de relations entre ces classes. Son intérêt réside dans la modélisation des entités du système d'information.

Le diagramme de classes se base sur les concepts suivants :

- Classe : description formelle d'un ensemble d'objets ayant une sémantique, des propriétés et un comportement communs.

- Association : est la réunion, le rapprochement de plusieurs entités ; c'est une relation sémantique entre deux ou plusieurs classes.
- Propriété (attribut) : élément permettant de décrire une classe ou une association.
- Méthode : opération réalisée par les objets d'une classe.

Le diagramme ci-dessous nous donne une idée sur l'organisation des différentes classes constituant notre application :

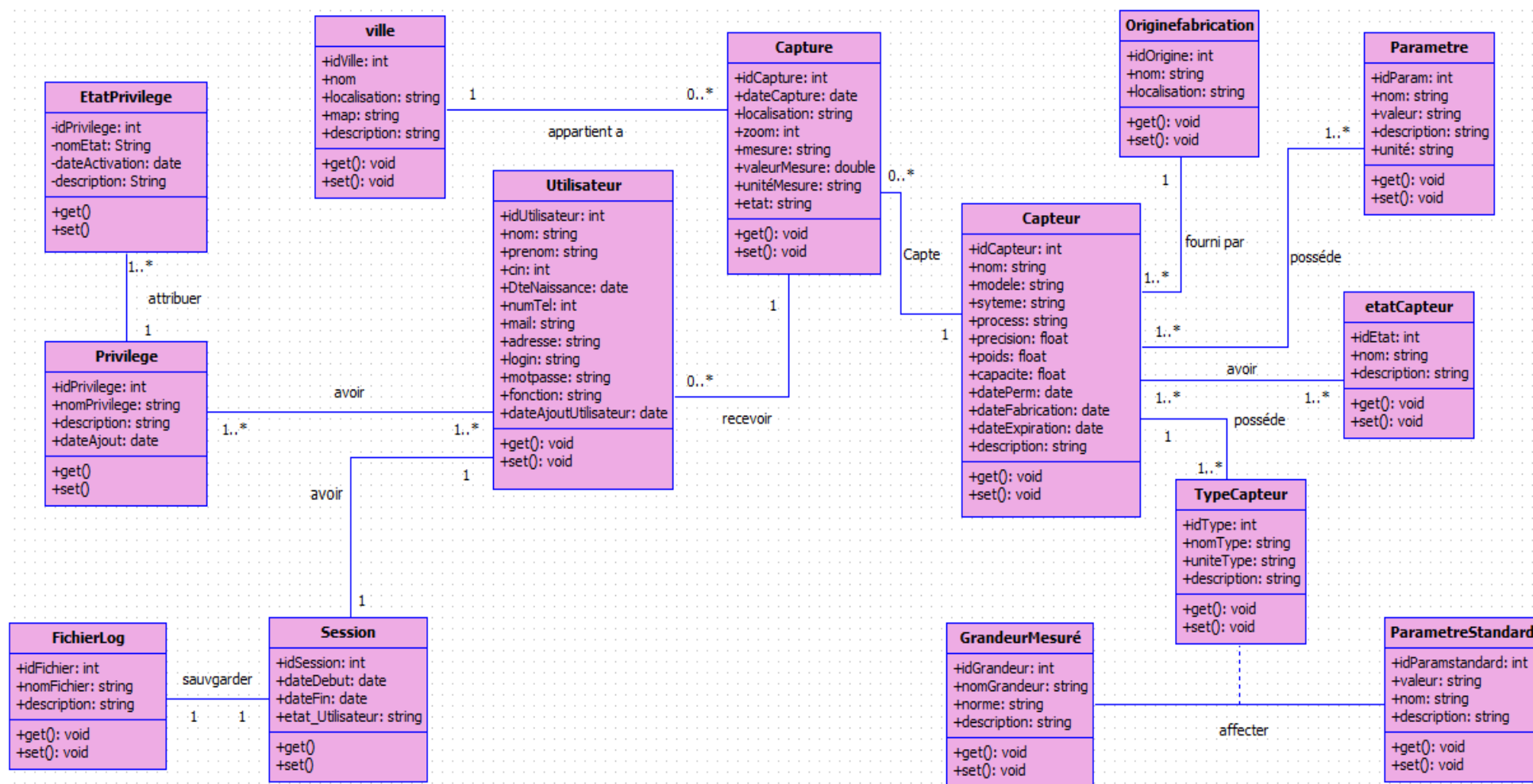


Figure :Diagramme de classe

## Les détails des classes :

**La classe utilisateur :** est la classe qui contient toutes les informations concernant un utilisateur de l'application web.

**La classe capteur :** est la classe qui présente toutes les caractéristiques nécessaires d'un capteur fournie par le fabricant du matériel

Les deux classes utilisateur et capteur représentent les éléments principaux de notre application web. Ils sont en relation indirect : un utilisateur utilise un capteur pour capter des informations. Ces deux classes ont des relations avec toutes les autres classes intermédiaires de notre application.

Les classes en relation avec l'utilisateur :

**La classe privilège :** cette classe décrit l'ensemble des privilèges, c'est une classe de jointure entre la classe d'utilisateur et le classe EtatPrivilège.

**La classe EtatPrivilège :** cette classe permet d'attribuer ou retirer un privilège aux utilisateurs, elle gère les privilèges et sélectionne les privilèges actifs

**La classe EtatUtilisateur :** cette classe permet d'indiquer si un utilisateur est actif ou pas et détermine la date d'activation de ce utilisateur.

**La classe session :** cette classe permet de sauvegarder des informations sur les opérations d'un visiteur (utilisateur) de l'application web dans un fichier. Une session possède une date de démarrage et date de fin puisque une session sera détruite automatiquement lorsque le visiteur quittera l'application et par conséquent toutes les données enregistrées dans le fichierLog seront détruites

**La classe fichierLog :** cette classe représente le fichier dans lequel sont enregistrées les informations d'une session actif. Un fichier est créé lors du démarrage d'une session.

**La classe ville :** cette classe permet de fixer l'emplacement de la capture par rapport à la localisation d'utilisateur.

*Les classes en relations avec le capteur :*

**La classe capture :** contient les signaux électriques analogiques numérisés sous forme d'image numérique, ces images réalisées par un capteur.

**La classe OrigineFabrication :** rassemble les informations concernant le fabricant d'un matériel (capteur).

**La classe Paramètre :** cette classe contient les paramètres de configuration et d'adaptation du fonctionnement d'un capteur (comme : signalisation sonore, mode d'envoi, sensibilité du mouvement ...)

**La classe EtatCapteur :** cette classe permet d'indiquer si le capteur est actif ou non

**La classe TypeCapteur :** contient les différents types de capteurs, on peut caractériser le capteur selon deux critères : grandeur de mesure et paramètre standard

**La classe GrandeurMesuré :** dans cette classe on parle de l'unité de mesure du signal détecté.

**La classe paramètreStandard :** on parle des paramètres d'origine, c.-à-d. les informations délivrées par un capteur logique.

### Implémentation de la base de données :

**Utilisateur** (id, nom, prenom, cin, dteNaissance, numTel, mail, adresse, login, mdp, role, dteAjout, Privilège\_id#),

**Privilège** (id, nomPrivilège, dateAjout, etatPrivilège\_id#)

**Ville** (id, nom, Localisation, map),

**Session** (id, dateDebut, dateFin, etatUtilisateur, Utilisateur\_id#),

**fichierLog** (id, nomFichier, description),

**Capture** (id, dateAjout, Localisation, zoom, mesure, valeurMesuré, unité, etat, Capteur\_id#, Ville\_id #),

**Capteur** (id, nom, modele, système, process, précision, poids, capacité, dteDebutUtilis, dteFabrica, dteExpirat°, EtatCapteur\_id#, TypeCapteur\_id#, OrigineFabricat°\_id#, Paramètre\_id#),

**TypeCapteur** (id, nomType, unité, description),

**Paramètre** (id, nom, valeur, unité),

**OrigineFabrication** (id, nom, localisation),

**GrandeurMesure** (id, nom, norme, description),

**ParamètreStandard** (id, valeur, nom, description).

### **Conclusion :**

A l'issue de ce chapitre, nous avons donné un aperçu sur la démarche conceptuelle adoptée. Cette phase de conception était importante pour pouvoir visualiser le fonctionnement de notre application d'une façon abstraite, et à partir de laquelle nous avons pu passer à la phase de réalisation.

# *Chapitre 3 : Réalisation*

## Introduction :

Dans cette partie, nous présentons, tout d'abord, l'architecture sur laquelle nous avons développé notre application. Ensuite, nous décrivons l'environnement de travail que nous avons utilisé pour réaliser notre projet et qui se compose en : environnement matériel et logiciel. Nous justifions les choix du langage de programmation et la technologie de développement appliqué. Nous illustrons par la suite, quelques captures d'écran de l'application représentant les fonctionnalités réalisées.

### I. Architecture logiciel :

Notre application web sera déployée sur une architecture 3-Tiers. Cette architecture peut être décrite par la figure ci-dessous :

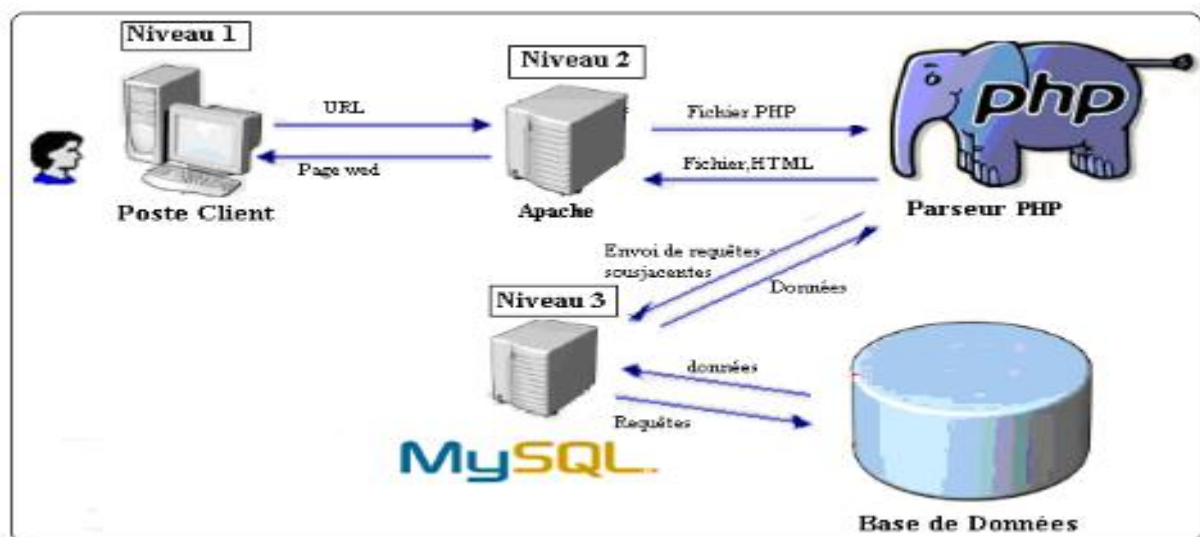


Figure: Architecture 3 tiers

Cette configuration que l'on met ci-dessus est souvent nommée "3-tiers" (3-tier en anglais) car on peut décomposer fonctionnellement notre application en 3 couches logicielle. Donc c'est une architecture partagée entre :

- **Le Client :** Dans un réseau informatique un client est l'ordinateur et le logiciel qui envoient des demandes à un serveur. L'ordinateur client est généralement un ordinateur personnel ordinaire, équipé de logiciels relatifs aux différents types de demandes qui vont être envoyées, comme par exemple un navigateur web, un logiciel client pour le World Wide Web.



- **Le Serveur d'application** : Dans un réseau informatique, un serveur est à la fois un ensemble de logiciels et l'ordinateur les hébergeant dont le rôle est de répondre de manière automatique à des demandes envoyées par des clients ordinateurs et logiciel via le réseau. Le serveur d'application permet ainsi au client de communiquer avec le serveur de bases de données.
- **Le Serveur de base de données** : Dans un réseau informatique, un serveur de base de données fournissant au serveur d'application les données dont il a besoin.

## II. L'architecture conceptuelle

### 1. Qu'est-ce qu'un Framework ?

Un Framework est un cadre de travail basé sur une architecture applicative. Il propose ainsi un ensemble d'outils et de composants logiciels cohérents tels qu'une bibliothèque logicielle réutilisable. Sa structure technique et logique impose au développeur le respect de certaines pratiques et normes de développement.



Figure : exemples des Framework PHP

### 2. L'objectif d'un Framework

Un Framework sert de fondation et de squelette à une application. Toute la partie technique (architecture, structure et bibliothèque) est gérée en toute transparence. Sa conception est faite de telle manière à faciliter le travail du développeur et lui permettre de ne se concentrer que sur l'essentiel, c'est-à-dire le développement des composants métiers

### 3. Symfony2 :

Symfony2 est un puissant Framework de développement PHP Web, une boîte à outils faite en PHP dans le but de nous simplifier la réalisation et le développement des applications web

rapidement et des sites complexes. Mais de façon structurée et avec un code clair et maintenable.

#### 4. L'architecture conceptuelle de symfony2 :

Le Framework propose une organisation du projet, gère la cohérence entre les modules, intègre tous les composants formant la structure technique et s'assure du bon fonctionnement de l'application. Aujourd'hui, on ne peut parler de Framework sans citer le MVC.

Dans notre projet, nous nous intéressons à développer une application web avec le Framework symfony2. Sachez que Symfony2 respecte bien entendu l'architecture MVC.

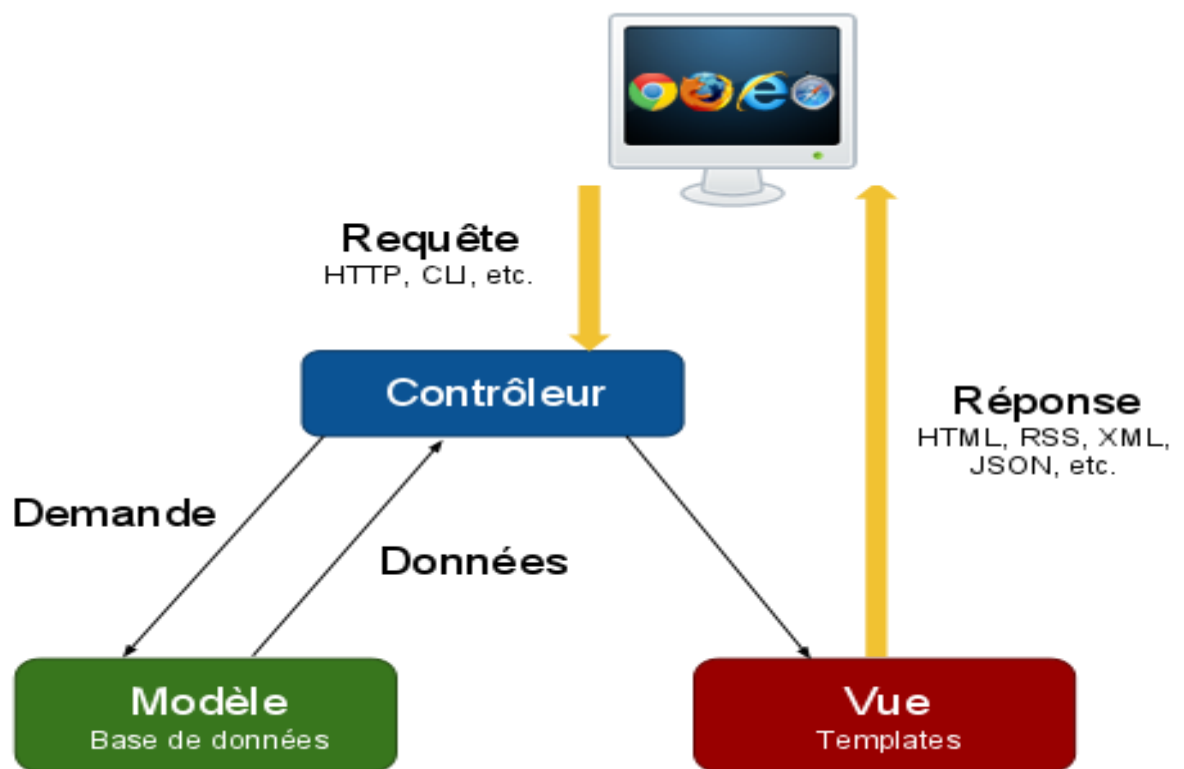


Figure : la modélisation de MVC

MVC signifie « Modèle / Vue / Contrôleur ». C'est un découpage très répandu pour développer les sites Internet, car il sépare les couches selon leur logique propre :

- **Le Contrôleur** (ou Controller) : son rôle est de générer la réponse à la requête HTTP demandée par le visiteur. Il est la couche qui se charge d'analyser et de traiter la requête de l'utilisateur. Le contrôleur contient la logique de notre application web et va se contenter « d'utiliser » les autres composants : les modèles et les vues.

- **Le Modèle** (ou Model) : son rôle est de gérer les données. C'est le modèle qui sait comment les récupérer, généralement via une requête au serveur SQL. il permet au contrôleur de manipuler les données, mais sans savoir comment les sont stockés, gérés, etc. C'est une couche d'abstraction.
- **La Vue** (ou View) : son rôle est d'afficher les pages. En pratique, c'est le designer d'un projet qui travaille sur les vues. La vue formate les résultats du traitement retournés par le modèle pour les présenter à l'utilisateur

Afin de bien visualiser tous les acteurs que nous avons vus jusqu'à présent, nous proposons un schéma du parcours complet d'une requête dans Symfony2 :

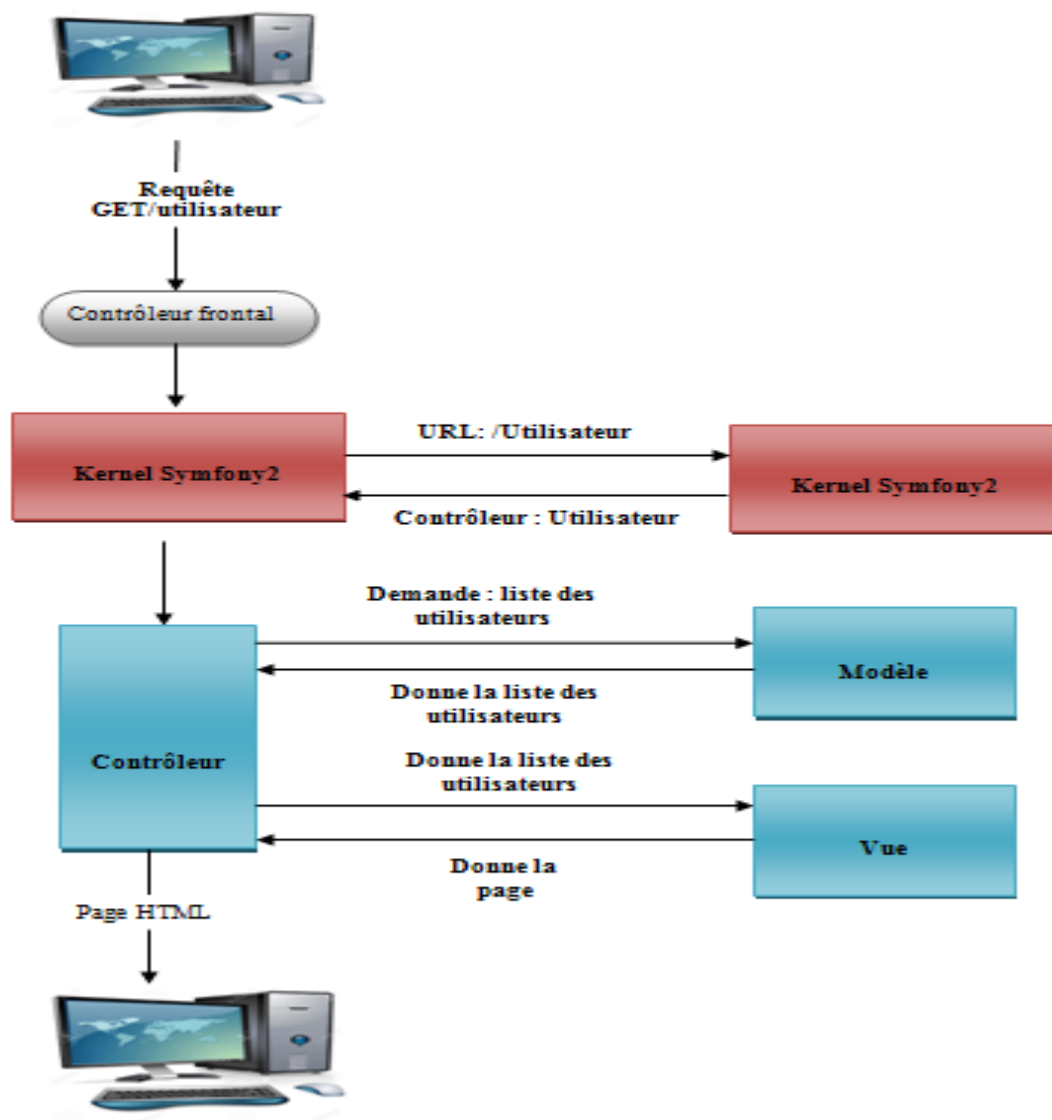


Figure : Parcours Complet d'une requête dans symfony2

## II. Environnement du travail :

### 1. Environnement matériel :

Pour développer notre application nous avons utilisé un ordinateur portable HP avec les spécifications suivantes :

- ❏ Un processeur : Intel(R) Core(TM) i3-321 CPU @ 1.80GHz 1.80 GHz.
- ❏ Système d'exploitation : Windows 10.
- ❏ Une mémoire vive(RAM) de 4 GO.
- ❏ Type de système : Système d'exploitation 32 bits, processeur x64

### 2. Environnement logiciel :

Le développement de notre application nécessite aussi des outils logiciels qui sont installés dans notre environnement de travail et que nous avons utilisés durant la réalisation du projet :

- ❏ Environnement du développement : netbeans
- ❏ Plateforme de développement : PHP5
- ❏ Frameworks : Symfony2
- ❏ Serveur web : Apache
- ❏ Système de gestion de base de données : phpmyadmin
- ❏ Méthodologies de conception : Modèle en V
- ❏ Langage de modélisation : UML
- ❏ Outils de modélisation : StarUML

#### ➤ Environnement du développement :Netbeans IDE 7.4

NetBeans est une plate-forme de développement logiciel en Java. Cette plate-forme permet aux applications d'être développées à partir d'un ensemble de composants logiciels modulaires. Les applications basées sur NetBeans, y compris l'environnement de développement intégré (IDE) NetBeans, peuvent être étendues par des développeurs tiers.

L'IDE NetBeans est principalement destiné au développement en Java, mais supporte également d'autres langues, en particulier PHP, C / C ++ et HTML5.

NetBeans est multi plate-forme et s'exécute sur Microsoft Windows, Mac OS X, Linux, Solaris et d'autres plates-formes prenant en charge une JVM compatible.

## ➤ Système de gestion de base de données : MySQL

MySQL est le serveur de base de données de notre application, c'est est une base de données relationnelle libre qui a vu le jour en 1995 et très employée sur le Web, souvent en association avec le langage *PHP* et le serveur web *Apache*. MySQL fonctionne indifféremment sur tous les systèmes d'exploitation.

Avec le Framework Symfony2, Pour créer une base de données de type MySQL, il faut éditer le fichier **Symfony2/app/config/parameters.ini** avec « Symfony2 », étant le nom du répertoire de l'installation Symfony2.

Dans ce fichier, on définit le nom, le type et d'autres paramètres de la base de données. Ensuite, il faut taper la commande suivante dans la console pour généré la base : **php app/console doctrine:database:create** :

```
parameters:
    database_host: 127.0.0.1
    database_port: null
    database_name: projetfinetude
    database_user: root
    database_password: null
    mailer_transport: smtp
    mailer_host: 127.0.0.1
    mailer_user: null
    mailer_password: null
    secret: ThisTokenIsNotSoSecretChangeIt
```

Figure : les paramètres de la base de données de notre application (fichier parameter.yml)

## ➤ Plateforme de développement :

- **PHP :**

Hypertext Preprocessor, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale.

- **Frameworks : Symfony:**

Quand on parle de PHP orienté objet on parle de Framework de développement, et Symfony est l'un de puissant Framework qui permette de réaliser des sites complexes et des applications web rapidement, mais de façon structurée et avec un code clair et maintenable.

Symfony, créé en 2005 par le français Fabien Potencier, cet framework PHP reconnu internationalement représente depuis quelque temps la référence en la matière. L'un des meilleurs points forts de Symfony est son architecture flexible, basée sur tout un ensemble de composants indépendants et réutilisables.

Symfony Très populaire parmi les entreprises de part de sa maturité ainsi que ses très bonnes performances, il est actuellement en version 3.1 et les recherches d'innover et d'amélioration continue de la productivité.

### Les Points forts de Symfony :

- Symfony est un Framework mature et réputé, maintenu par la société SensioLabs et de nombreux autres contributeurs
- Symfony Basé sur un ensemble de composants indépendants et réutilisables, qui sont devenus un standard parmi les développeurs d'applications PHP
- Symfony a une Grande flexibilité
- Symfony est un Moteur de Template intégré : **Twig**
- Symfony, Généralement associé à l'ORM Doctrine, mais le développeur pourra très bien en choisir un autre
- Une communauté de plus de 300 000 développeurs
- Symfony a Une documentation très détaillée
- Avec symfony, Beaucoup de bundles (plugins) à disposition du développeur

<http://projectsquare.io/principaux-frameworks-php-2016/>

- Les bundles

Symfony fonctionne en bundles, c'est-à-dire avec une liste de fonctionnalités de notre application, où bien encore des blocs d'une application .La nomenclature suivie est namespace/nomBundle, le namespace pouvant être par exemple le nom d'une appli

Pour créer un bundle:

- Depuis la console: `php app/console generate:bundle`
- Il demande un namespace: `INRA/FormationBundle`
- Symfony nous guide dans la suite des étapes

[file:///C:/Users/Standard/Downloads/sicpa\\_codex\\_symfony2\\_MR.pdf](file:///C:/Users/Standard/Downloads/sicpa_codex_symfony2_MR.pdf)

- Twig :

Symfony utilise la bibliothèque Twig pour faire ses vues. En gros Twig est un langage permettant de mixer du code html avec des variables, sans pour autant dénaturer complètement le code

Les modèles Twig sont destinés à être simples et ne traiteront pas les tags PHP. C'est par conception: le système de modèle Twig est destiné à exprimer la présentation, et non pas la logique du programme. Plus vous utilisez Twig, plus vous apprécierez et bénéficierez de cette distinction

Twig peut également faire des choses que PHP ne peut pas les faire, telles que :

- le contrôle des espaces blancs,
- la sauvegarde sur un fond blanc,
- l'échappement HTML automatique,
- l'échappement de la sortie contextuelle manuelle et l'inclusion de fonctions personnalisées et de filtres qui affectent uniquement les modèles.

Twig contient de petites fonctionnalités qui rendent les modèles d'écriture plus faciles et plus concis

<http://symfony.com/doc/current/templating.html>

- CSS 3 :

« [Cascading Style Sheet](#) » définit la structure et la présentation d'un contenu de site web et n'a rien à voir avec le design.

Une feuille CSS peut contrôler la police, le positionnement, la couleur, et le style des informations de tout un site.

**6 avantages d'utilisation de CSS :**

<https://www.cssdebutant.com/debuter-en-css-points-forts-du-css.html>

- Les pages web se chargent plus rapidement et utilisent moins de bande passante
- Le CSS est compatible avec tous les navigateurs
- Le CSS peut être utilisé pour créer des pages web faciles à imprimer
- Le CSS permet à l'utilisateur de personnaliser une page web
- Le CSS facilite le positionnement dans les moteurs de recherche
- Le CSS facilite la portabilité du contenu

CSS3 est la dernière évolution du langage Cascading Style Sheets et vise à étendre CSS2.1. Il apporte beaucoup de nouveautés tant attendues, comme des coins arrondis, des ombres, des gradients, des transitions ou des animations, ainsi que de nouvelles mises en page comme les multi-colonnes, les configurations de boîtes flexibles ou de grille.

- **HTML5**

L'HyperText Markup Language, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte. [https://fr.wikipedia.org/wiki/Hypertext\\_Markup\\_Language/](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language/)

HTML5, il s'agit de nouveaux, qui vont nous permettre de faire des choses assez intéressantes qui n'étaient pas ou peu possibles avant.

Le HTML5 est alors pas seulement le successeur de HTML 4, il est bien plus que ça. Tant que les langages HTML 4 et autres XHTML se focalisaient juste sur le contenu des pages Web, le HTML5 se focalise sur les applications Web et l'interactivité, sans toutefois délaisser l'accessibilité et la sémantique. Le HTML5 se positionne également comme concurrent des technologies Flash et Silverlight .

- **ORM :**

**Object-Relational Mapping** est une technique de programmation informatique qui crée l'illusion d'une base de données orienté objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé. On pourrait le désigner par « correspondance entre monde objet et monde relationnel ».



[https://fr.wikipedia.org/wiki/Mapping\\_objet-relationnel](https://fr.wikipedia.org/wiki/Mapping_objet-relationnel)

- JavaScript :

JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés.

<http://www.additeam.com/SSII/javascript/>