

MEMOIRE DE STAGE DE FIN D'ETUDE

Pour l'obtention du

MASTERE PROFESSIONNEL

« Nouvelles Technologies des Télécommunications et Réseaux »

Présenté par :

BEN SAAD Asma

*Conception et développement d'une application de
gestion de données dans un réseau de capteurs mobiles*

Soutenu le : 08/02/2014

Devant le jury :

Mme : Idoudi Hanen Président
Mme : Ben Hsin Ahlem Membre
Mme : Rekaya Houissa Houda Membre

REMERCIEMENTS

Au nom de Dieu le Clément, le Miséricordieux,

La paix et les bénédictions soient sur notre Prophète Muhammad.

J'ai l'honneur et le plaisir d'exprimer ma gratitude et mes remerciements à tous ceux qui ont guidé, aidé et m'ont supervisé durant l'élaboration de ce projet.

Je tiens à remercier Mr. Rjadh Bouhouchi mon encadreur à la UVT et Mr. Nizar Trigui et mon encadreur au Tunisie Télécom qui ont toujours été très serviables et attentionnés tout au long de la réalisation de ce projet, et pour l'inspiration, le soutien et le temps qu'ils ont bien ont gracieusement passé lors de mon stage, ce qui m'a beaucoup aidé à développer ce modeste travail.

Ma gratitude va également à tous ceux qui ont participé de près ou de loin à la réussite de mon stage au sein du Tunisie Télécom

Que MRS les membres du jury trouvent ici l'expression de ma sincère gratitude pour l'honneur qu'ils m'ont donné en acceptant de juger mon travail.

✍ BEN SAAD Asma

DÉDICACE

A ma chère mère, Pour tous vos sacrifices pour moi, nul mot ne saura exprimer mon amour envers vous. Que Dieu vous protège et vous accorde une longue vie.

A mes frères et mes sœurs, Je vous remercie pour votre amour inconditionnel. Que dieu vous garde, Je vous aime et je vous souhaite une vie pleine de succès et de réussite.

A tous mes amis, pour vos encouragements et pour avoir été là pour moi chaque fois que j'en avais besoin.

A tous ceux qui m'aiment, A tous ceux que j'aime Que cette modeste dédicace puisse vous témoigner ma profonde gratitude pour votre immense amour, votre confiance et vos paroles apaisantes qui m'ont toujours fait garder le sourire.

A tous ceux qui ont contribué de près ou de loin pour la réussite de ce projet Veuillez trouver ici le respect et la reconnaissance que J'éprouve pour vous.

Je dédicace ce travail

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Table de matière

Liste des figures.....	vii
Liste des tables	viii
Glossaire	ix
Introduction Générale.....	1
Chapitre 1: Présentation du cadre générale du projet.....	3
Introduction	3
I. Présentation de l'entreprise d'accueil : Tunisie Télécom	3
1. Présentation Générale.....	3
2. Activité de l'entreprise	4
3. Organigramme de l'entreprise	4
II. Présentation du sujet	5
1. Description du sujet	5
2. Définition de Concept de base	6
2.1. Capteur.....	6
2.2. Capteur téléphonique ou Capteur mobile	6
2.3. Réseau de capteur sans fil (WSN).....	7
2.4. Application de capteurs.....	8
III. Mesure de la qualité de l'application : Diagramme de Gantt.....	8
Conclusion	9
Chapitre 2: Etude Préalable	10
Introduction.....	10
I. Analyse de l'existant.....	10
II. Critique de l'existant	13
III. Solution Proposé ou mission du projet.....	13
Conclusion	15
Chapitre 3: Analyse et spécification des besoins	16
Introduction	16
I. Spécification des besoins	16
1. Spécification des acteurs.....	16
2. Spécification fonctionnelle.....	17
3. Spécification non-fonctionnelle.....	18
II. Diagramme de cas d'utilisations	19
1. Diagramme de cas d'utilisation de l'administrateur	20

2. Diagramme de cas d'utilisation du Superviseur	23
Conclusion	25
Chapitre 4: Conception.....	26
Introduction	26
I. Conception Générale	26
1. Le cycle de vie d'un logiciel	26
1.1. Définition	26
1.2. Modèles de cycle de vie du logiciel.....	28
1.2.1. Modèle en V.....	28
1.2.2. Modèle en W	29
2. Méthodologie de conception.....	30
2.1. Merise	30
2.2. UML	30
3. Modèle de conception.....	31
3.1. MVC.....	31
3.2. EAO (Entity Access Object).....	32
3.3. Session Façade	32
II. Conception Détaillé	33
1. Diagramme Dynamique.....	33
1.1. Diagramme de séquence	33
1.1.1. Diagramme de séquence du cas d'utilisation "Authentification"	33
1.1.2. Diagramme de séquence du cas d'utilisation "Ajouter un nouvel utilisateur "	34
1.1.3. Diagramme de séquence du cas d'utilisation "Visualiser statistiques par capteurs "	35
1.2. Diagramme d'activité	35
1.2.1. Diagramme d'activité "Authentification"	37
1.2.2. Diagramme d'activité "Ajouter capture "	38
2. Diagramme Statique.....	39
2.1. Diagramme de classe.....	39
2.2. Diagramme de déploiement	41
Conclusion	41
Chapitre 5 : Développement et Réalisation.....	42
Introduction	42
I. Architecture de développement	42
II. Environnement de développement.....	43
1. Environnement matériel.....	43

2. Environnement logiciel.....	43
2.1. Environnement de développement intégré	43
2.2. Serveurs.....	44
2.2.1. GlassFish Open Source Edition 3.1.2.....	44
2.2.1. MySQL 5.1	46
2.3. Outils de conception (Power AMC).....	47
2.4. Framework et choix techniques	48
2.4.1. JSF 2.1	48
2.4.2. EJB 3.1	49
2.4.3. JPA 2.....	50
2.4.4. PrimeFaces 3.3	50
2.4.5. Log4J	51
III. Travail réalisé.....	52
1. Page d'authentification	52
2. Application Back-office	53
3. Application Front-office.....	55
Conclusion	57
Conclusion et perspectives.....	58
Bibliographie & Netographie	60

Liste des figures

Figure 1:Organisation interne de Tunisie Télécom	4
Figure 2 : Aperçu d'un réseau de capteur	5
Figure 3:Diagramme du Gantt.....	8
Figure 4 : Interface Graphique pour utiliser " ProSense Common Gateway"	12
Figure 5 : Interface Graphique pour utiliser " Ubiquitous Mobile Awareness "	12
Figure 6 : Architecture du système	14
Figure 7 : Catégorie des utilisateurs.....	16
Figure 8:Diagramme de cas d'utilisation général	18
Figure 9:Diagramme de cas d'utilisation général de l'administrateur	20
Figure 10 : Diagramme de cas d'utilisation "Gestion des Utilisateurs"	21
Figure 11:Diagramme de cas d'utilisation "Gestion des Capteurs"	22
Figure 12:Diagramme de cas d'utilisation général du superviseur	23
Figure 13:Diagramme de cas d'utilisation "Gérer les Captures"	24
Figure 14: Le modèle en V	28
Figure 15: Le modèle en W	29
Figure 16: Le modèle MVC	32
Figure 17:Diagramme de séquence du cas d'utilisation "Authentification"	33
Figure 18:Diagramme de séquence du cas d'utilisation "Ajouter un nouvel utilisateur "	34
Figure 19:Diagramme de séquence du cas d'utilisation "Visualiser statistiques par capteurs "	35
Figure 20:La structure du diagramme d'activité	36
Figure 21:Diagramme d'activité "Authentification"	37
Figure 22:Diagramme d'activité "Ajouter capture"	38
Figure 23:Diagramme de Classe.....	40
Figure 24:Diagramme de déploiement	41
Figure 25:Architecture de développement	42
Figure 26:PC DELL.....	43
Figure 27:l'IDE Netbeans.....	44
Figure 28: GlassFish Open Source Edition.....	46
Figure 29:L'espace de travail de MySQL 5.1	47
Figure 30:L'espace de travail de PowerAMC.....	48
Figure 31:L'utilisation au cours des dernières années.....	51
Figure 32:L'utilisation au cours des derniers mois	51
Figure 33:Page d'authentification	52
Figure 34:Interface d'accueil Administrateur	53
Figure 35:Interface Ajout d'un nouvel utilisateur	53
Figure 36:Interface Afficher tous les capteurs	54
Figure 37:Interface Afficher les détails d'un utilisateur	54
Figure 38:Interface d'accueil Superviseur	55
Figure 39:Interface Ajouter nouvelle capture	55
Figure 40:Interface Simuler captures du température	56
Figure 41:Interface Générer les statistiques de température.....	56
Figure 42:Interface Générer les statistiques de température.....	57

Liste des tables

Tableau 1:Les types de capteurs pris en charge par les systèmes d'exploitation mobiles	7
Tableau 2:Description textuelle de cas d'utilisation "Ajouter un nouvel utilisateur"	21
Tableau 3:Description textuelle de cas d'utilisation "Afficher les détails d'un capteur "	22
Tableau 4:Description textuelle "Consultez les observations des capteurs"	25

Glossaire

Acronyme	Description
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
DAO	Data Access Object
EAO	Entity Access Object
EJB	Enterprise JavaBeans
GUI	Graphical User Interface
HTML	Hypertext Markup Language
JDBC	Java DataBase Connectivity
JEE	Java Enterprise Edition
JPA	Java Persistence API
JSF	Java Server Faces
MVC	Model View Controller
SQL	Search and Query Language
UMA	Ubiquitous Mobile Awareness
UML	Unified Modeling Language
XML	Extensible Markup Language
ORM	Mapping Objet-Relationnel
CMP	Container-Managed Persistence

Introduction Générale

En tournant les différentes pages des époques passées, tout le monde s'est aligné pour sauver et maintenir l'équilibre de notre éco-système. Plusieurs études et recherches ont été effectuées pour réviser les ressources et les moyens déployés afin de contrôler et superviser l'état de stabilité de notre système écologique.

Pour mettre en marche un système de contrôle et de mesure à distance, le développement d'une architecture compatible est devenu impératif d'où vient l'intégration de la technique des réseaux à capteurs sans fil ou (Wireless Sensors Networks). Les réseaux de capteurs sans fil ont été récemment appliqués à une grande variété d'applications tant civiles que militaires. Ils sont constitués d'un grand nombre de nœuds de capteurs généralement déployés dans des environnements différents et visent essentiellement à détecter des événements ou de phénomènes naturels, de collecter, traiter et transmettre des données détectées aux utilisateurs concernés.

L'utilisation de ce type de capteurs pour la surveillance d'environnement semble être un bon moyen afin de contrôler à distance les conditions physiques mais leur intégration dans le Web qui les a rendus accessibles sur Internet a confronté plusieurs obstacles. Par contre, nous avons besoin d'avoir un mécanisme simple réalisable pour profiter des avantages de ces nouvelles techniques.

Actuellement, les capteurs sont entre les mains de 5,6 milliards de personnes à travers le monde via les téléphones mobiles, il est donc intéressant d'utiliser ces capteurs pour mesurer différents paramètres aux différents emplacements.

L'objectif de ce projet est la conception et le développement d'une application de gestion du trafic de données transmises par les capteurs mobiles tels que la température, pression, de mouvement ou de polluants, à des emplacements différents.

Ce rapport présente mon travail, il contient quatre chapitres :

- + Le premier chapitre constitue le cadre général du projet, y compris l'organisation hôte de l'entreprise d'accueil "Tunisie Télécom" et la définition des concepts de base.
- + Le deuxième chapitre décrit l'état de l'existant, ainsi la solution proposée.
- + Le troisième chapitre présente les acteurs de l'application, l'analyse des besoins (fonctionnels et non-fonctionnels).
- + Le quatrième chapitre présente la phase de conception lorsque nous avons décrit la méthodologie adoptée, et les diagrammes UML détaillés pour la modélisation du système.
- + Le cinquième chapitre présente les environnements matériels et logiciels utilisés en donnant une description des interfaces de l'application.

Enfin, le rapport du mon mémoire de mastère sera clôturé par une conclusion où notre travail sera résumé en exposant quelques perspectives d'avenir.

Chapitre 1: Présentation du cadre générale du projet

Introduction

Dans ce chapitre, nous commencerons par présenter l'entreprise Tunisie Télécom au sein de laquelle nous effectuons notre stage. Ensuite, nous donnerons une vue générale sur le sujet, quelque concept de base et l'importance du réseau à capteurs. Enfin, nous allons présenter les mesures de qualité de l'application.

I. Présentation de l'entreprise d'accueil : Tunisie Télécom

1. Présentation Générale

Tunisie Télécom est le leader sur le marché des télécommunications en Tunisie, administré directement par le ministère de Télécommunications jusqu'en 1995.

Aujourd'hui, Tunisie Télécom est un chef de file sur le marché des télécommunications en Tunisie grâce à sa présence dans les segments de la téléphonie fixe et mobile en outre l'accès à Internet. Il cible le grand public, les entreprises et les opérateurs tiers.

Tunisie Télécom est aujourd'hui organisée autour de deux pôles d'activité :

✚ Le pôle « **détail** » regroupe les services de téléphonie mobile, de téléphonie fixe, d'Internet (destiné au grand public et aux entreprises) et les services de transmissions de données (destinée exclusivement aux entreprises).

✚ Le pôle « **opérateur et international** » regroupe les services

- ☐ interconnexions nationales
- ☐ terminaison
- ☐ transit
- ☐ roaming-in.

2. Activité de l'entreprise

En tant qu'opérateur, Tunisie Télécom est chargée de :

- + L'installation, la maintenance et l'exploitation de réseaux publics de télécommunications.
- + Offrir des services de télécommunications publics ou privés qui correspondent aux différents besoins sociaux et économiques.
- + La promotion des nouveaux services de télécommunication.
- + La contribution à l'élaboration d'études et de recherches scientifiques liées au secteur des télécommunications.
- + La participation à l'effort national de l'enseignement supérieur en matière de télécommunications.
- + L'application des conventions et traités des organisations internationales et régionales spécialisées dans le domaine des télécommunications.
- + Promouvoir la coopération à tous les niveaux dans tous les domaines des télécommunications.

3. Organigramme de l'entreprise

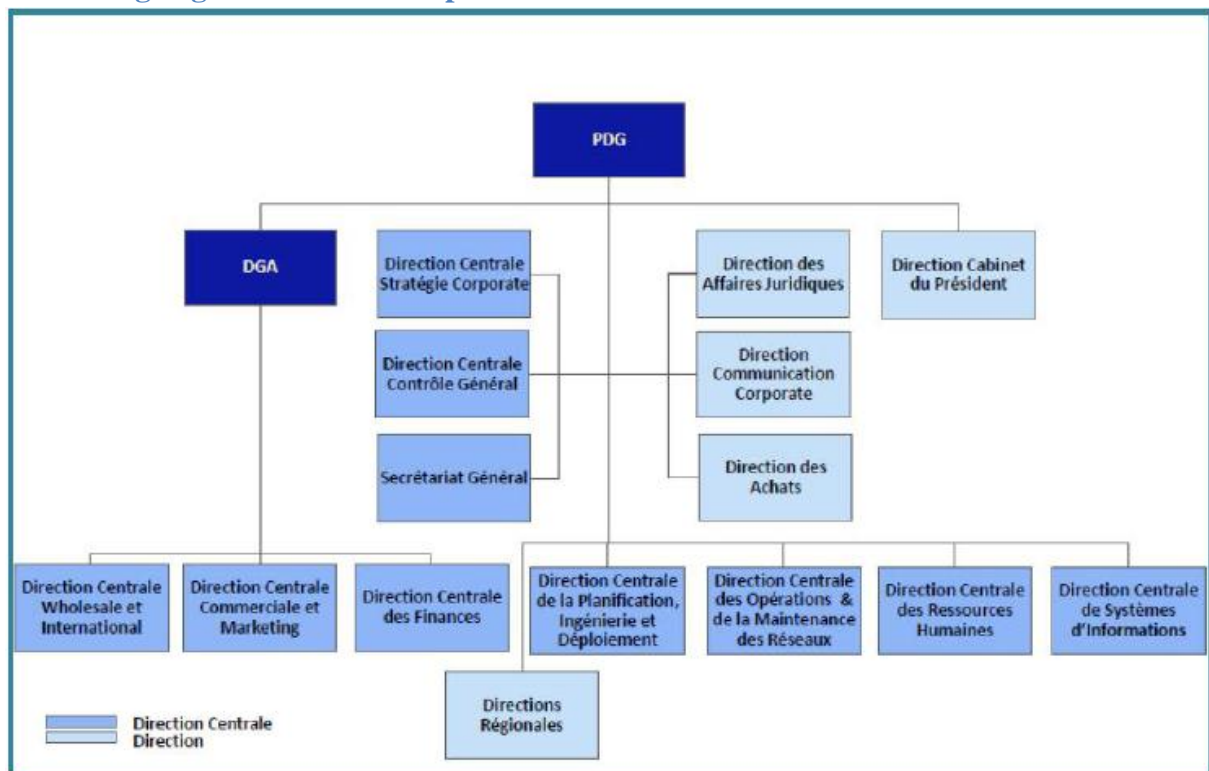


Figure 1: Organisation interne de Tunisie Télécom

II. Présentation du sujet

Cette partie vise à donner un aperçu du thème du projet et son plan de réalisation. Afin d'atteindre les objectifs souhaités, nous avons besoin d'abord de préciser les principaux objectifs du projet, puis nous avons mis en place un processus qui permet de surveiller la progression du travail.

1. Description du sujet

Notre sujet intitulé *"Conception et développement d'une application de gestion de données dans un réseau de capteurs mobiles"*.

Cette application permet de gérer le flux de données généré par des capteurs mobiles tels que la température, la pression et la pollution pour aider les superviseurs de l'environnement de prendre des décisions rapides en cas d'un problème sérieux sans être obligés de quitter leurs postes de travail.

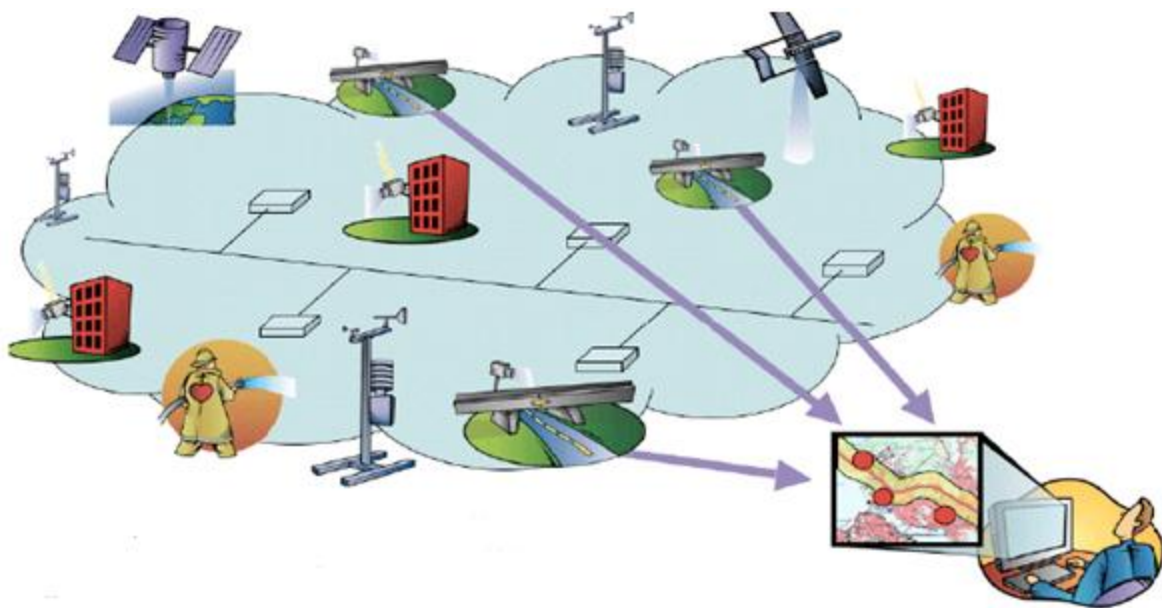


Figure 2 : Aperçu d'un réseau de capteur

Notre produit final doit être une application qui gère le trafic des informations envoyées par les capteurs téléphoniques et permet aux superviseurs de faire décisions concernant des problèmes environnementaux de leurs postes de travail.

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 droite.
- 3- choisir et justifier le langage de développement de codage.
- 4- scinder le projet à deux phases importantes :

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

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

La partie **Front-office** est conçue pour l'utilisateur ou Superviseur, il permet la surveillance :

- Des états du capteur et les emplacements.
- Lectures de capteurs et de la mesure.
- Captures et les contrôles capteurs (statistiques, alertes).

2. Définition de Concept de base

Dans cette partie, nous allons présenter les concepts de base liés aux réseaux de capteurs. Ces concepts sont considérés comme des moyens pour découvrir les données et les services.

2.1. Capteur

Un capteur ou un détecteur est un convertisseur qui permet de mesurer une grandeur physique, puis la transformer en un signal qui peut être lu par un appareil ou par un observateur. Il nous permet de surveiller et de réagir en monde réel.

2.2. Capteur téléphonique ou Capteur mobile

De nos jours, la plupart des téléphones mobiles sont équipés d'un ensemble riche de capteurs embarqué qui ont une capacité de calcul moyenne.

En outre, ils sont entièrement en réseau, et avec les réseaux 3G/4G modernes leur bande passante est raisonnable.

La plupart des capteurs ont des radios wifi 802.11, et peuvent avoir des connexions continues Wi-Fi dans les milieux urbains, avec une bande passante de 1-50 Mbps.

Ces téléphones ont une capacité de traitement raisonnable sur les processeurs de faible puissance modernes. Il est supposé que les téléphones sont équipés de capteurs intégrés tels que GPS ,802.11x, Bluetooth (classe 1,2 ou 3), de la température, de l'orientation, de l'accélération, microphone audio, et l'appareil photo (images fixes ou vidéo).

Un des avantages de l'utilisation de capteurs mobiles, c'est qu'ils sont facilement accessibles par les techniques 3G/4G sans être obligé d'utiliser des passerelles ou des procurations pour recueillir des données détectées.

Type de Capteur	iOS 5	Android	Windows 8
Capteur de lumière ambiante	✓	✓	✓
Audio(Microphones)	✓	✓	✓
Caméra (s)	✓	✓	✓
Capteur d'humidité		✓	
Capteur inertiel de mouvement			
✚ Accéléromètre	✓	✓	✓
✚ Magnétomètre	✓	✓	✓
✚ Gyroscope	✓	✓	✓
Capteur de pression		✓	
Capteur de proximité	✓	✓	✓
Capteur de Température	✓	✓	✓

Tableau 1:Les types de capteurs pris en charge par les systèmes d'exploitation mobiles

2.3. Réseau de capteur sans fil (WSN)

Réseau de capteurs sans fil (WSN) est constitué de capteurs autonomes répartis dans l'espace pour surveiller les conditions physiques ou environnementales telles que la température, bruit, pression... en passant leurs données via le réseau à un emplacement principal.

Ce réseau se compose généralement de nombreux systèmes embarqués visés comme des nœuds de capteurs, chacun de ces nœuds se compose d'un petit microprocesseur, un récepteur, un certain nombre de capteurs et c'est généralement alimenté par batterie.

2.4. Application de capteurs

Aujourd'hui à travers la technologie des capteurs, nous pouvons vivre dans des villes intelligentes. Les capteurs nous aident à améliorer notre mode de vie, plusieurs domaines d'application utilisent ces technologies.

Nous citerons une grande variété de domaine par exemple le transport, les réseaux sociaux, surveillance de l'environnement (détection des feux de forêt, la pollution de l'air...), la santé et le bien-être.

III. Mesure de la qualité de l'application : Diagramme de Gantt

Pour planifier notre projet et pour suivre l'avancement, nous avons utilisé le diagramme de GANTT. Cette méthode visuelle est efficace puisqu'elle représente graphiquement l'avancement du projet. Le diagramme de GANTT répond à deux objectifs : planifier de façon optimale et communiquer sur le planning établi. Elle nous permet de lister les différentes tâches à faire ainsi que la date de début et de fin de cette tâche.

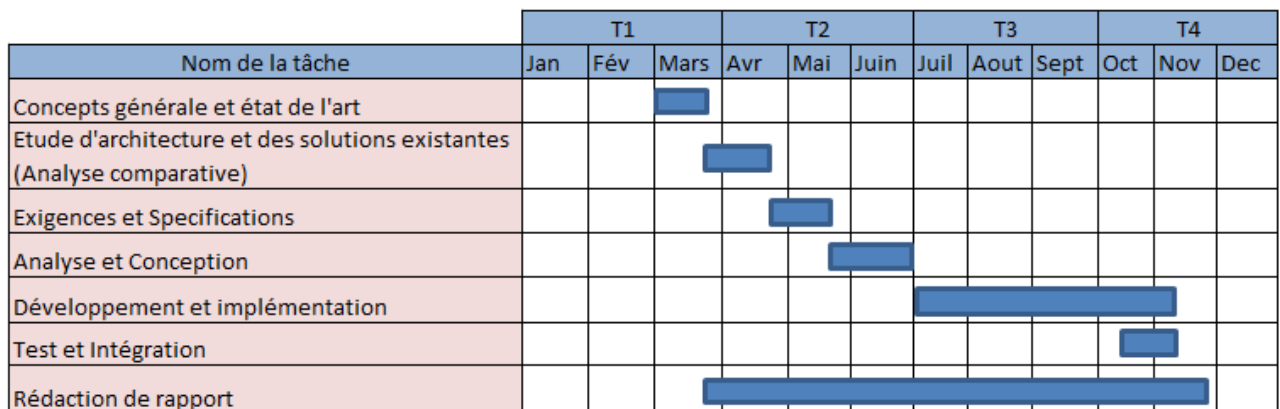


Figure 3:Diagramme du Gantt

Conclusion

Dans ce chapitre, nous avons mis en place l'organisation où j'ai passé mon stage. Ensuite, nous avons présenté notre mission et nous avons fini par une présentation générale du projet, tout en mettant l'accent sur la mesure de la qualité de l'application.

Chapitre 2: Etude Préalable

Introduction

L'étude de l'existant est une phase primordiale qui sert à analyser en détail ce qui existe dans le but de dégager ses limites et ce que nous proposons comme solutions. Dans ce chapitre, nous allons préciser le contexte de l'application afin de bien identifier les besoins dans le chapitre suivant.

I. Analyse de l'existant

En prenant à titre d'exemple, notre pays, comme ses semblables, on peut constater qu'elle est en train de rencontrer une variété de problèmes, assez critiques, générés par l'être humain qui n'a pas su tenir une protection saine et correcte pour notre système écologique.

À long terme, ces contaminations d'ordre humaines, peuvent causer des dommages au système écologique ainsi qu'à notre globe terrestre. Essayant de voir de plus près quelques recherches et statistiques récentes qui reflètent bien les conséquences désastreuses de la mauvaise gérance de notre environnement :

- ✚ La quantité du **GES** (gaz à effet de serre qui sert à absorber le rayonnement infrarouge) est actuellement au-dessous du seuil qui peut potentiellement causer un changement climatique dangereux. Nous sommes déjà à risque.
- ✚ Au niveau de la température, la tendance ne cesse d'augmenter et selon les statistiques une augmentation de 6 degrés Celsius (en 2050) est prévue, ce qui peut engendrer des conséquences assez désastreuses pour la planète.
- ✚ Durant les années 1990, les émissions de Carbone (CO₂) ont augmenté de moins 1% ce qui paraît parfait. En revanche, depuis l'entrée du III^{ème} millénaire, les émissions ont subi une montée anormale d'à peu près 3.5%.

- ✚ Si la quantité du CO₂ dans l'atmosphère est réduite à 350 parties par millions, ça va causer des dommages énormes, irrécupérables et irrévocables à la planète.

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

- ✚ Les catastrophes naturelles (par exemple, les inondations, les ouragans, les tempêtes provoquées par des processus naturels) habituellement imprévisibles (même s'il peut y avoir quelques avertissements à court terme), incontrôlables, destructrices, aigus, mais ils sont prévisibles.
- ✚ Dégradation de l'environnement à long terme (par exemple, l'air, l'eau, le bruit et la pollution lumineuse, réchauffement de la planète).

Des multiples recherches et solutions ont été proposées pour faire face aux désastres naturels qui peuvent, à long terme, créer une dégradation environnementale et superviser les potentiels dommages infligés.

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

- ✚ l'application **PCG** (ProSense Common Gateway) est une liaison entre l'application de l'utilisateur et les divers réseaux de capteurs sans fil et qui permette à ces applications un accès facile aux données des capteurs déployées partout.

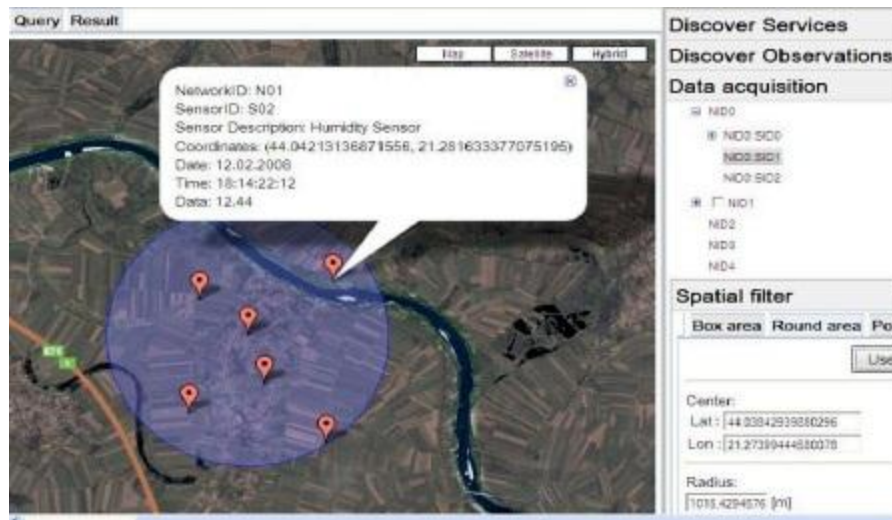


Figure 4 : Interface Graphique pour utiliser " ProSense Common Gateway"

- ✚ l'application **UMA** (Ubiquitous Mobile Awareness) visualise à l'instant les informations des capteurs relatives à des objets dans Google Maps et prévient l'utilisateur des changements tels que le passage d'un autre objet appartenant au groupe de l'utilisateur.



Figure 5 : Interface Graphique pour utiliser " Ubiquitous Mobile Awareness "

II. Critique de l'existant

Cette méthode de monitoring représente un ensemble de limites et voici les problèmes qu'on pouvait rencontrer durant la collecte sur terrain :

- ✚ Ça sollicite un effort titanesque et un temps énorme durant les processus de collecte des données.
- ✚ L'acte de la prise des mesures peut à son tour changer les propriétés de la chose que vous êtes en train de mesurer.
- ✚ Le fait d'utiliser un capteur statique en une entité unique doit acquérir les droits d'accès pour placer les capteurs dans le domaine de la couverture exigée par une application, comme l'exemple d'une combinaison des stations de métro, les parcs publics et centres commerciaux. Le nombre des capteurs statiques demandé pour couvrir la même superficie couverte par un unique périphérique mobile peut être trop coûteux à déployer.

III. Solution Proposé ou mission du projet

Le temps où on trouve une difficulté de l'utilisation des capteurs, les téléphones mobiles de nos jours sont riches en termes de capteurs intégrés. En maniant des téléphones mobiles, on peut acquérir plusieurs avantages. 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. Ainsi, un nombre important de Smartphones déjà existant ont l'avantage d'avoir la détection physique de l'infrastructure. En déduction, les capteurs dans les smartphones n'exigent jamais un déploiement.

Notre solution consiste à renforcer l'utilité de l'existant de terminaux mobiles. En effet, notre travail est la conception et le développement d'une application qui gère le flux de données approvisionnement de la circulation par les capteurs mobiles tels que la température, la pression et la pollution, à des emplacements différents.

L'émergence et la consolidation des applications de services et des périphériques capteurs ont augmenté les prestations, telles que l'adaptabilité, la flexibilité et l'évolutivité pour la construction de notre application.

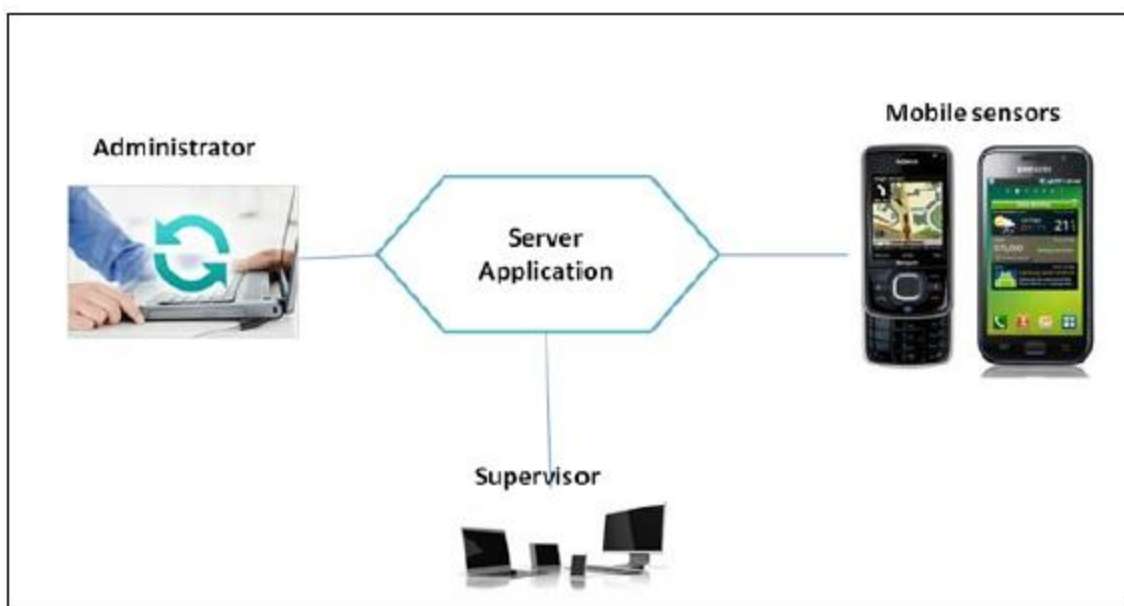


Figure 6 : Architecture du système

L'architecture suivante résume les relations entre les technologies mentionnées. Ainsi, notre système constitue par trois entités :

- ❏ Capteurs mobiles qui détectent le monde physique comme la température, le son, les niveaux de pollution et l'humidité.
- ❏ L'utilisateur (administrateur ou superviseur) qui peut accéder aux informations des capteurs via une interface utilisateur graphique.

- Application de service qui stocke les informations des capteurs et de le rendre disponible aux applications.

Conclusion

Dans ce chapitre, nous avons commencé par donner une analyse de l'existant et sa critique, puis nous avons présenté les exigences fonctionnelles du système ainsi que son architecture.

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 3: Analyse et spécification des besoins

Introduction

Après avoir présenté le projet dans les chapitres précédents, nous avons acquis suffisamment de connaissances sur la solution proposée, ce qui nous rend capables de préciser les besoins de notre projet.

Dans cette partie, nous avons présenté et isolé les besoins des différents types d'utilisateurs, sachant que cette étude dépassera le cadre de projet présent.

Notre sujet a un caractère ouvert, continu et évolutif, donc nous prévoyons que le projet sera poursuivi notamment avec la mise en place de la quatrième génération de réseaux mobiles.

I. Spécification des besoins

1. Spécification des acteurs

En fonction des besoins des différentes applications, on distingue deux types d'utilisateurs, avec des exigences différentes (administrateur et superviseur).

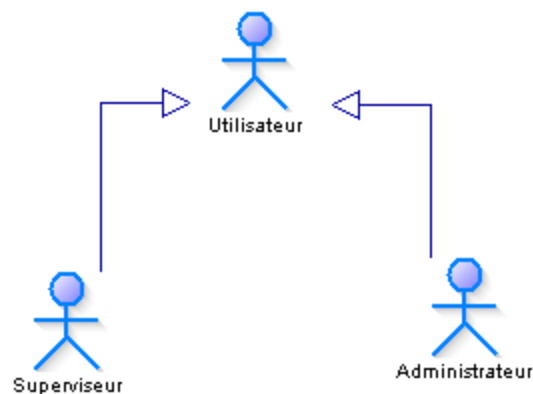


Figure 7 : Catégorie des utilisateurs

- L'utilisateur ou superviseur : toutes les personnes, qui utiliseront l'application, ayant un smartphone Android ou un PC.

- L'administrateur de l'application est une personne qui a accès par des interfaces, les tables de base de données utilisées à notre application.

2. Spécification fonctionnelle

Les besoins fonctionnels sont les services offerts par notre système. Ils capturent les tâches ou activités qui doivent être formulées par l'application pour différents types d'utilisateurs.

Notre système permet à l'administrateur de :

- ✚ Gérer les comptes utilisateurs, y compris les superviseurs (ajouter, modifier, supprimer, consulter).
- ✚ Gérer les capteurs (ajouter, modifier, supprimer, consulter).

Notre système permet au superviseur de :

- ✚ Rechercher des données de 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.

Voici le diagramme de cas d'utilisation général pour notre système présentés par cette figure :

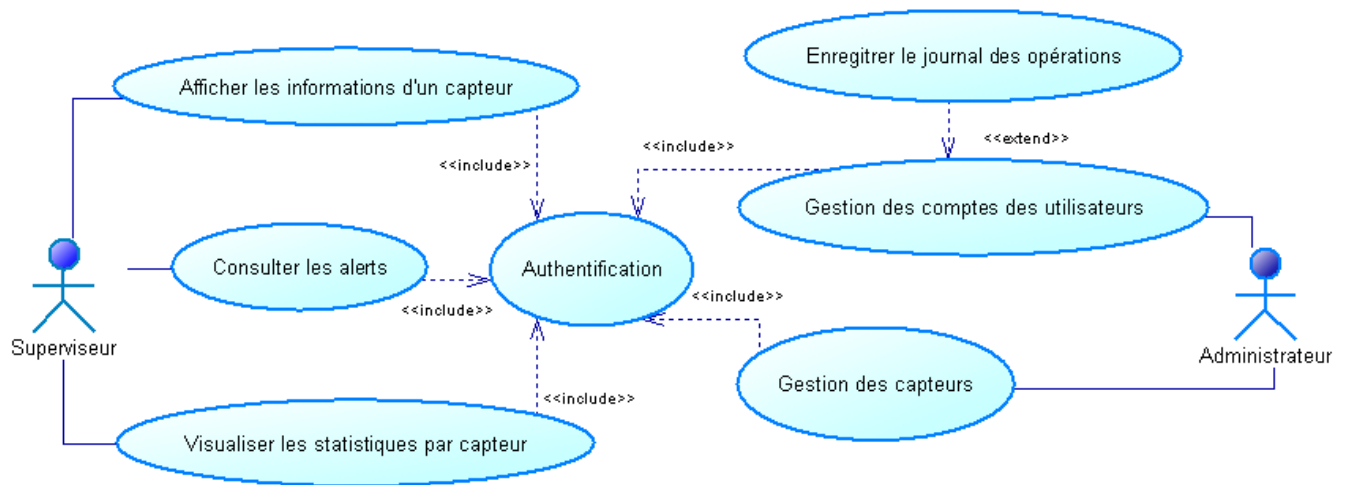


Figure 8:Diagramme de cas d'utilisation général

3. Spécification non-fonctionnelle

Dans cette partie, nous traitons la phase des exigences non-fonctionnelles, cette phase qui est très importante et qui complète les besoins fonctionnels. En effet, les spécifications non fonctionnelles offrent souvent beaucoup de contraintes qui peuvent influencer les choix futurs.

L'incorporation des exigences non fonctionnelles telles que les contraintes liées au temps, la sécurité et l'efficacité de l'opération sont extrêmement délicates. Nous serons maintenant intéressés à identifier les besoins non fonctionnels de notre application.

- ✚ Facilité d'utilisation : l'application doit être simple, facile à comprendre par n'importe quel utilisateur de téléphone mobile Android ou l'utilisateur de l'ordinateur.
- ✚ Conception : l'interface utilisateur doit être d'une belle apparence, facile à utiliser et suivre les modèles d'application des normes.
- ✚ Fiabilité : notre application doit être digne de confiance et fiable et fournit toutes les fonctions nécessaires de sécurité pour les utilisateurs.



- + Sécurité : l'application doit garantir la sécurité des données et la vie privée en utilisant sessions afin d'éviter la perte de données ou leur modification par des pirates.
- + Performance : la vitesse de fonctionnement de votre 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 mettre à jour rapidement et facilement.
- + Réutilisation : cet outil doit permettre l'ajout et l'édition des extensions futures.

II. Diagramme de cas d'utilisations

Les exigences fonctionnelles globales sont représentées par le modèle de cas d'utilisation et la description textuelle.

Le modèle de cas d'utilisation se concentre sur les facteurs critiques de succès du système, en termes de fonctionnalités ou caractéristiques que les utilisateurs ont besoin d'interagir avec. Ainsi, nous avons proposé d'analyser les différents cas d'utilisation que nous avons trouvée juste à travers l'activité de capture des besoins afin de déterminer les différentes classes impliquées dans chaque cas et pour identifier la séquence des cours d'eau pour la réalisation de chaque scénario.

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

L'administrateur est responsable pour de l'application Back-office, comme la gestion des utilisateurs, les privilèges, les dispositifs de détection et de types, les unités de mesure utilisées, les villes et les différents états (état de l'utilisateur, l'état de privilège, et l'état du capteur).

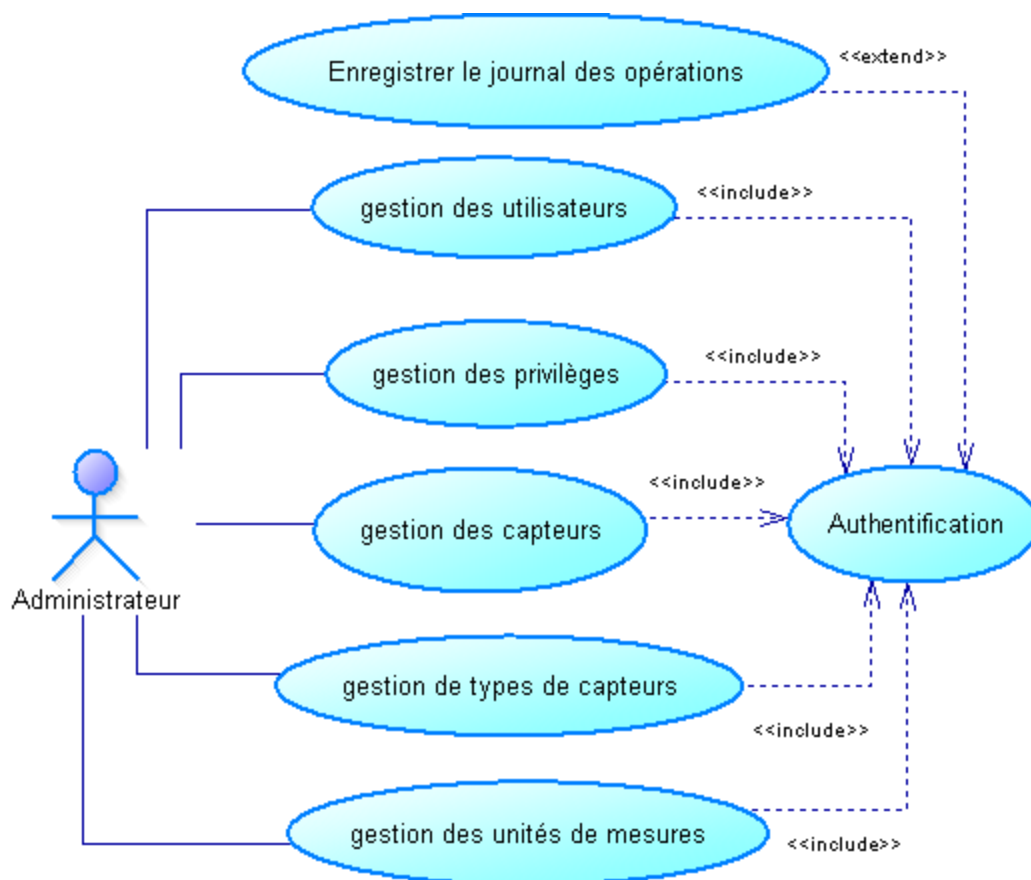


Figure 9:Diagramme de cas d'utilisation général de l'administrateur

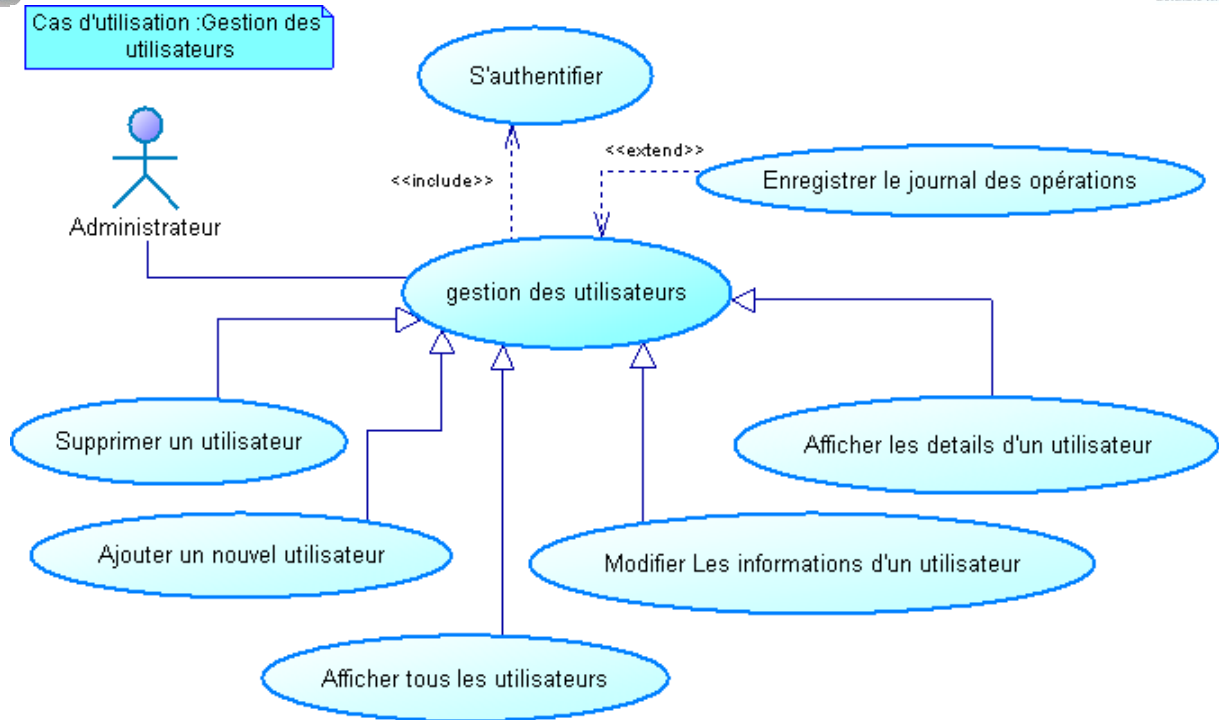


Figure 10 : Diagramme de cas d'utilisation “Gestion des Utilisateurs”

Résumé de l'identification	
Nom du cas	Ajouter un nouvel utilisateur
Objectif	Ça consiste à 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 privilège adéquat.
Acteur	Administrateur
Description d'enchainement	
Préconditions	Post-Conditions
L'administrateur doit avoir le privilège plus élevé quand il s'authentifie.	Le nouvel utilisateur est ajouté à la base de données
Scénario Principal	
<ol style="list-style-type: none"> 1. L'administrateur remplit le formulaire et cliquez sur le bouton Ajouter 2. Le système valide les entrées 3. Le système enregistre le nouvel utilisateur dans la base de données 4. Le système redirige l'administrateur à la page de liste des utilisateurs 	

Tableau 2:Description textuelle de cas d'utilisation "Ajouter un nouvel utilisateur"

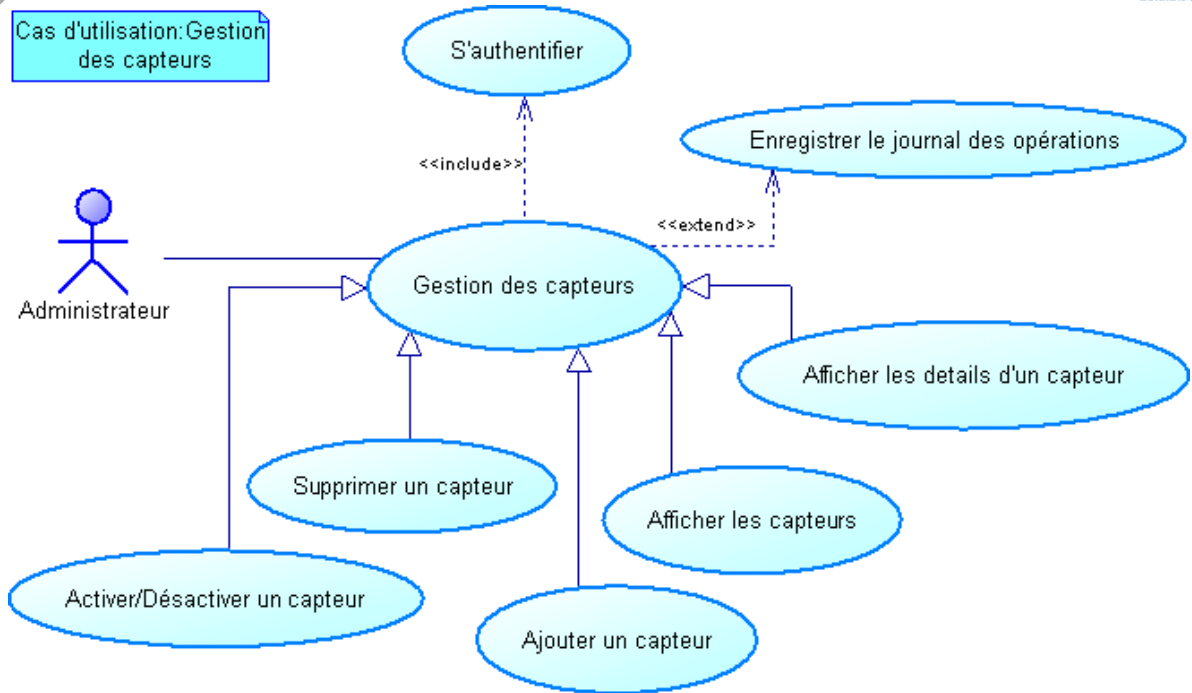


Figure 11:Diagramme de cas d'utilisation “Gestion des Capteurs”

Résumé de l'identification	
Nom du cas	Afficher les détails d'un capteur
Objectif	Ça consiste à la consultation des détails d'un capteur particulier à partir de la liste des capteurs
Résumé	L'administrateur doit afficher la liste des capteurs, puis il choisit le capteur qu'il veut afficher ces détails et il clique sur le bouton d'affichage
Acteur	Administrateur
Description d'enchainement	
Préconditions	Post-Conditions
L'administrateur est authentifié, la liste des capteurs s'affiche	L'administrateur dispose toutes les informations du capteur sélectionné
Scénario Principal	
<ol style="list-style-type: none"> 1. L'administrateur clique sur le lien de la liste des capteurs 2. Le système charge les capteurs enregistrés 3. L'administrateur sélectionne le lien Afficher capteur 4. Le système redirige l'administrateur à la page d'informations du capteur 	

Tableau 3:Description textuelle de cas d'utilisation "Afficher les détails d'un capteur "

2. Diagramme de cas d'utilisation du Superviseur

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

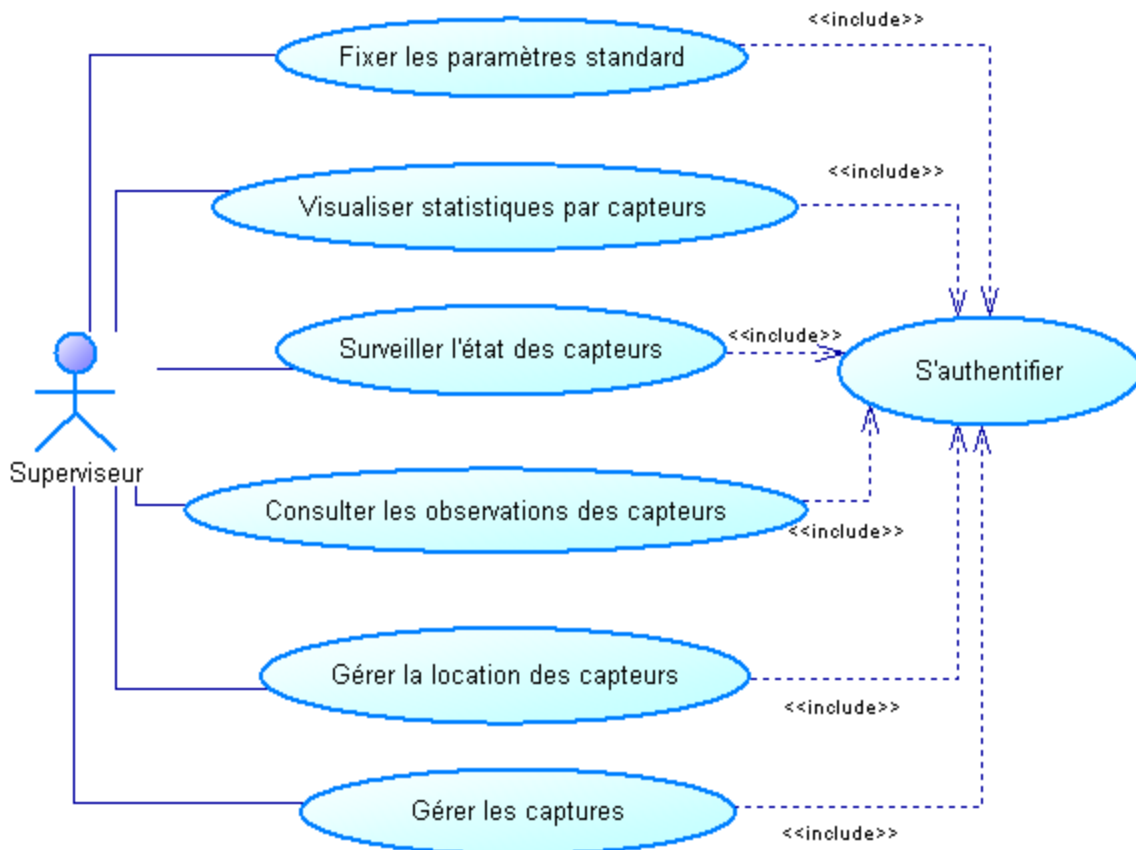


Figure 12:Diagramme de cas d'utilisation général du superviseur

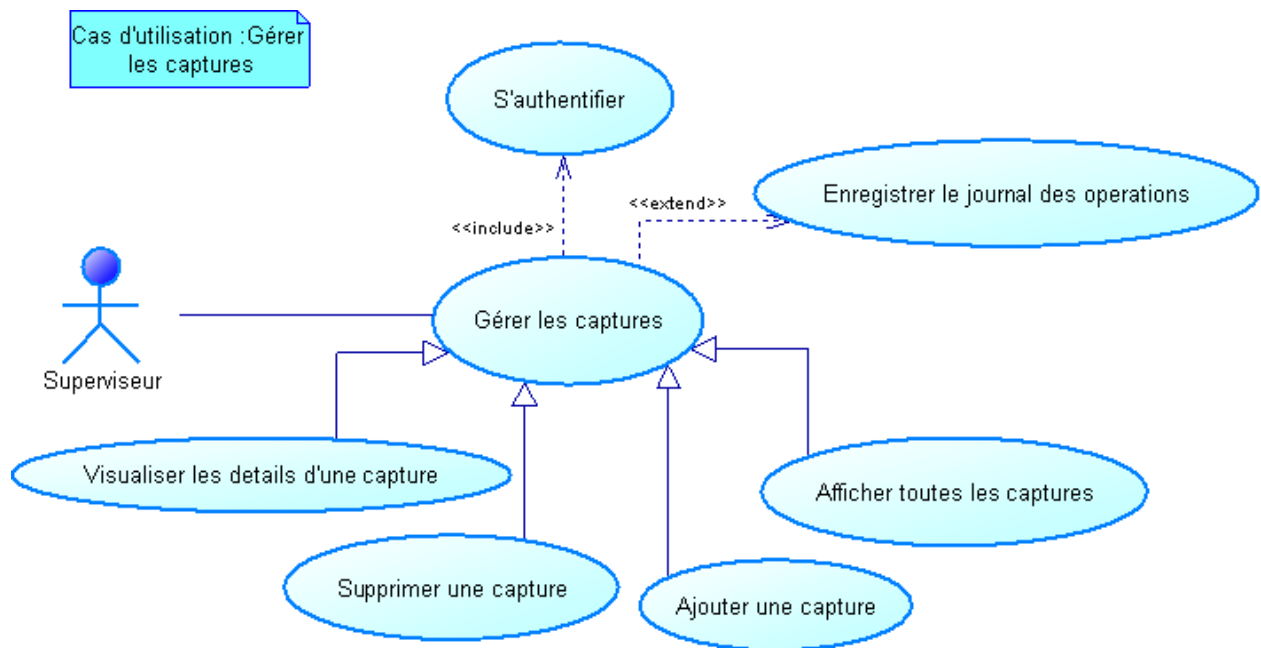


Figure 13:Diagramme de cas d'utilisation “Gérer les Captures”

Résumé de l'identification	
Nom du cas	Consultez les observations des capteurs
Objectif	Ça consiste à consulter les observations des 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'enchaînement	
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	
1. Les clics de superviseur sur le lien de simulation	

2. Le contrôleur choisit le type de simulation et / ou le dispositif de capteur, et / ou de localisation
3. Le système analyse la requête et renvoie le résultat sur la carte
4. Le superviseur peut cliquer sur un marqueur pour afficher les détails d'observation

Tableau 4:Description textuelle "Consultez les observations des capteurs"

Conclusion

Dans ce chapitre, nous avons commencé par donner une analyse de l'existant et sa critique, puis nous avons expliqué clairement les différents besoins fonctionnels et non-fonctionnels du système, à la fin, nous avons étudié le comportement par quelque cas d'utilisation.

Le chapitre suivant sera consacré à la conception pour créer une présentation virtuelle de la situation actuelle des besoins par rapport à la solution adoptée.

Chapitre 4: Conception

Introduction

La conception est une étape critique dans le cycle de vie d'une application, elle vise à développer des modèles détaillés de l'architecture du système et de réduire sa complexité.

Dans ce chapitre, nous allons étudier des aspects techniques pour assurer un meilleur résultat final afin de satisfaire les besoins nécessaires. La première partie sera dédiée pour la conception globale de la solution ainsi que le cycle de vie le plus approprié. La deuxième partie sera consacrée pour la conception détaillée en se basant sur quelques diagrammes UML statiques et dynamiques.

I. Conception Générale

1. Le cycle de vie d'un logiciel

1.1. Définition

Le cycle de vie d'un projet est la période qui s'écoule de la naissance d'une idée jusqu'à la fin de son exploitation. Il représente les étapes de développement : la planification, la conception, le codage, et le test du système. Il est fortement recommandé pour détecter les erreurs le plus tôt possible, de contrôler la qualité du logiciel, ses délais d'exécution et les coûts associés.

En général, tous les modèles de cycle de vie découpent le projet en plusieurs phases principales :

- ❏ **Définition des objectifs** : cette étape consiste à définir l'objectif du projet et son inscription dans une stratégie globale.
- ❏ **Analyse des besoins et faisabilité** : c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes.
- ❏ **La conception générale** : il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel.

- ❏ **Conception détaillée** : cette étape consiste à définir précisément chaque sous-ensemble du logiciel.
- ❏ **Codage (implémentation ou programmation)** : c'est la traduction des fonctionnalités définies lors la phase de conception dans un langage de programmation.
- ❏ **Les tests unitaires** : cette étape permet de vérifier que chaque sous-ensemble du logiciel est mis en œuvre conformément aux spécifications.
- ❏ **Intégration** : cette étape vise à s'assurer de l'interfaçage des différents éléments (modules) du logiciel.
- ❏ **Qualification (ou recette)** : cette étape vérifie la conformité du logiciel aux spécifications initiales.
- ❏ **Documentation** : elle vise à produire les informations nécessaires pour utiliser le logiciel et pour des développements ultérieurs.
- ❏ **Début de la production** : c'est le logiciel sur le site du déploiement.
- ❏ **Maintenance** : il comprend toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.

Ces activités peuvent être présentes dans le cycle de vie d'un logiciel selon le modèle choisi par le client et l'équipe de développement.

Il existe différents cycles de vie des logiciels des modèles tels que le modèle en cascade, V, spirale ...

1.2. Modèles de cycle de vie du logiciel

Afin d'être en mesure d'avoir une méthodologie commune entre le client et la société de services réalisant le développement, des modèles de cycle de vie ont été mis au point pour définir les étapes du développement ainsi que les documents nécessaires.

1.2.1. Modèle en V

Le modèle de cycle de vie en V est un modèle conceptuel qui a vu le jour suite au problème rencontré avec le modèle en cascade.

Il est appelé "**modèle en V**" en se basant sur sa représentation visuelle de deux branches obliques. Sur la barre oblique de gauche ; on représente les phases d'étude et d'analyse, de conception et de développement, sur la barre oblique remontante à droite ; on représente les phases de tests et d'acceptation finale.

Ce modèle part du principe que les procédures de vérification de la conformité du logiciel aux spécifications doivent être élaborées dès les phases de conception.

C'est un modèle éprouvé dans l'industrie, normalisé (*ISO-12207*, *MILSTD-498...*), mais il restera un modèle rigide face à une évolution du besoin.

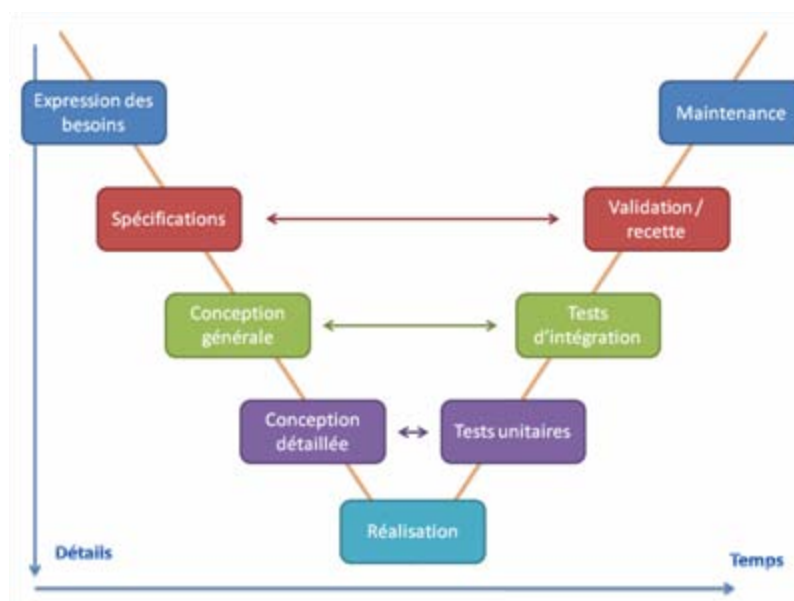


Figure 14: Le modèle en V

1.2.2. Modèle en W

Le W-modèle présente un cycle de développement standard, mais au lieu de se focaliser sur les étapes spécifiques des tests dynamiques du modèle en V, ce modèle se concentre sur les produits de développement.

Le principe est que chaque activité de développement qui produit de la matière est toujours "suivie" par une activité de test. Le but de l'activité de test est de déterminer si les objectifs d'une activité de développement ont été atteints et si le produit est conforme aux exigences.

Ce modèle illustre que l'activité de test doit impérativement se lancer dès le premier jour de l'initiation du projet. Entre autres, le produit d'une activité de développement (par exemple, les besoins d'écriture) est accompagné par une activité de test " **tester les besoins** " et ainsi de suite.

Tant que nous sommes en train de travailler sur une application pour Tunisie-Télécom donc on doit s'aligner avec leur orientation pour la planification du projet. Les cycles de vie du projet actuel sont basés sur le modèle V et le modèle W. Après cette étude, on s'est intéressé au modèle W qui convient avec nos besoins.

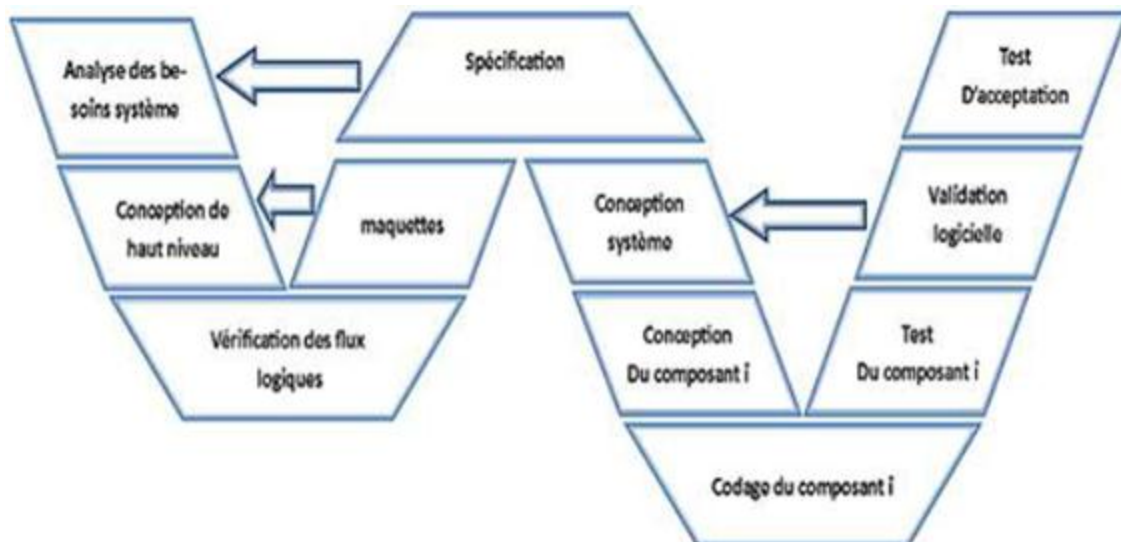


Figure 15: Le modèle en W

2. Méthodologie de conception

La conception d'un système d'informations n'est pas évidente, car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer.

La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Ce type de méthode est appelé analyse. Il existe plusieurs méthodes d'analyse telle qu'UML, MERISE ...

Dans cette partie nous allons clarifier la raison de notre orientation de conception (méthode de conception ou langage de conception).

2.1. Merise

Merise est une méthode de conception, de développement et de réalisation de projets informatiques. Son utilité est de concevoir les systèmes d'information. Cette méthode est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques en utilisant 6 niveaux : global, conceptuel, organisationnel, logique, physique et ensuite, le niveau de l'utilisateur.

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

UML (Unified Modeling Language) est un langage de modélisation graphique basée sur des pictogrammes et des diagrammes pour représenter les aspects d'un système statique ou dynamique. Il est apparu dans le monde du génie logiciel, dans le cadre de la « **conception orientée objet** ».

UML repose sur la notion orientée objet qui est un véritable atout pour cette langue. Il est couramment utilisé dans les projets logiciels, il peut être appliqué à tous les types de systèmes ne se limitent pas à la technologie de l'information.

Dans le cadre d'un processus de développement logiciel, UML permet à l'équipe de projet à comprendre certains aspects de la complexité inhérente à un logiciel, à préciser la structure et le comportement de ce système, de visualiser le système et pour construire le système, et de documenter les décisions prises le long du chemin.

En revanche, UML ne définit pas un cycle de vie d'une série de phases, parce que le processus de développement du modèle désiré est conçu d'une façon itérative est incrémentale, guidé par les cas d'usage et il est considéré comme étant un système d'architecture centrée. L'évolution dans le temps du système est maintenue grâce au fait que le processus est itératif durant toutes les phases du développement du système.

Suite à l'étude ci-dessus, nous allons adopter UML comme langage de modélisation.

3. Modèle de conception

L'application est généralement représentée par trois niveaux, la présentation, l'application (logiques métiers) et la persistance. Les modèles de conception courants pour chacun de ceux-ci sont :

- Présentation - Modèle-Vue-Contrôleur (MVC)
- Persistance - modèle Entité Accès Objet (EAO), une version du modèle de Data Access Object, mis à jour pour JPA.
- Application - Session Façade

3.1. MVC

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface. L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application. Dans l'architecture MVC, les rôles des trois entités sont les suivants :

- **Modèle** : données (accès et mise à jour).
- **Vue** : interface utilisateur (entrées et sorties).
- **Contrôleur** : gestion des événements et de synchronisation.

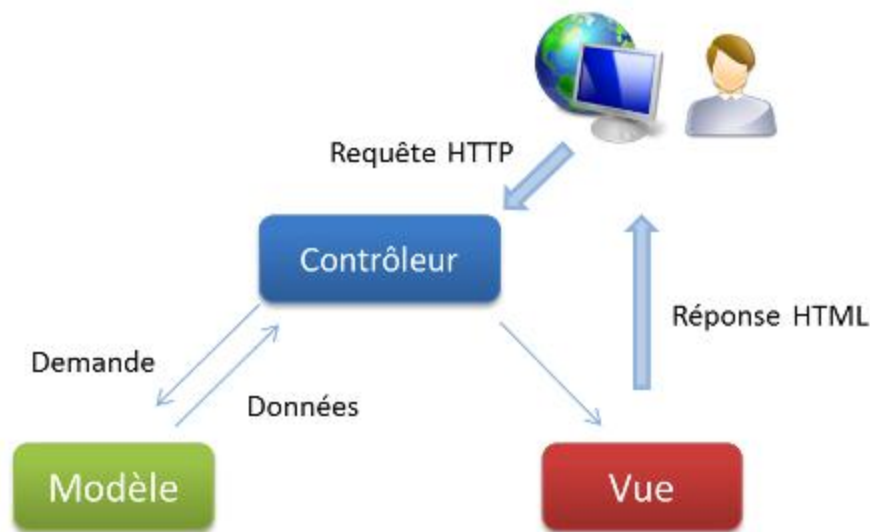


Figure 16: Le modèle MVC

3.2. EAO (Entity Access Object)

Auparavant, l'usage du JDBC du logique métier a conduit à plusieurs problèmes de maintenance dont l'invention de DAO (Data Access Object) afin de découpler le code d'accès aux données du logique métier. DAO fonctionne avec JDBC, ORM et CMP.

Le modèle Entité Accès Objet (EAO) est une nouvelle incarnation du modèle DAO que vous pouvez utiliser avec JPA. Il découple la logique d'accès de l'entité du logique métier. Le plus souvent, vous avez un objet EAO qui effectue les opérations CRUD pour les objets de domaine.

3.3. Session Façade

Session façade est l'une des couches EJB, souvent utilisée en développant des applications d'entreprise. Elle définit la façon dont on accède à un modèle d'objets persistants dans un environnement JEE.

Son but est de masquer la complexité technique d'une suite d'opérations et les rendre accessibles en un appel de méthode unique. Une façade agit donc comme une enveloppe, et expose des fonctionnalités simples clairement définies.

II. Conception Détaillé

Dans cette section, nous allons voir une vue plus profonde de l'architecture du projet en présentant quelques-uns des diagrammes UML statiques et dynamiques.

1. Diagramme Dynamique

1.1. Diagramme de séquence

Un diagramme de séquences se charge de représenter, en fonction du temps, le chaînage et l'échange d'informations entre les différents objets impliqués dans une fonctionnalité donnée du système.

On présentera ci-après les diagrammes de séquences, que nous avons jugés importants de notre application .

1.1.1. Diagramme de séquence du cas d'utilisation "Authentification"

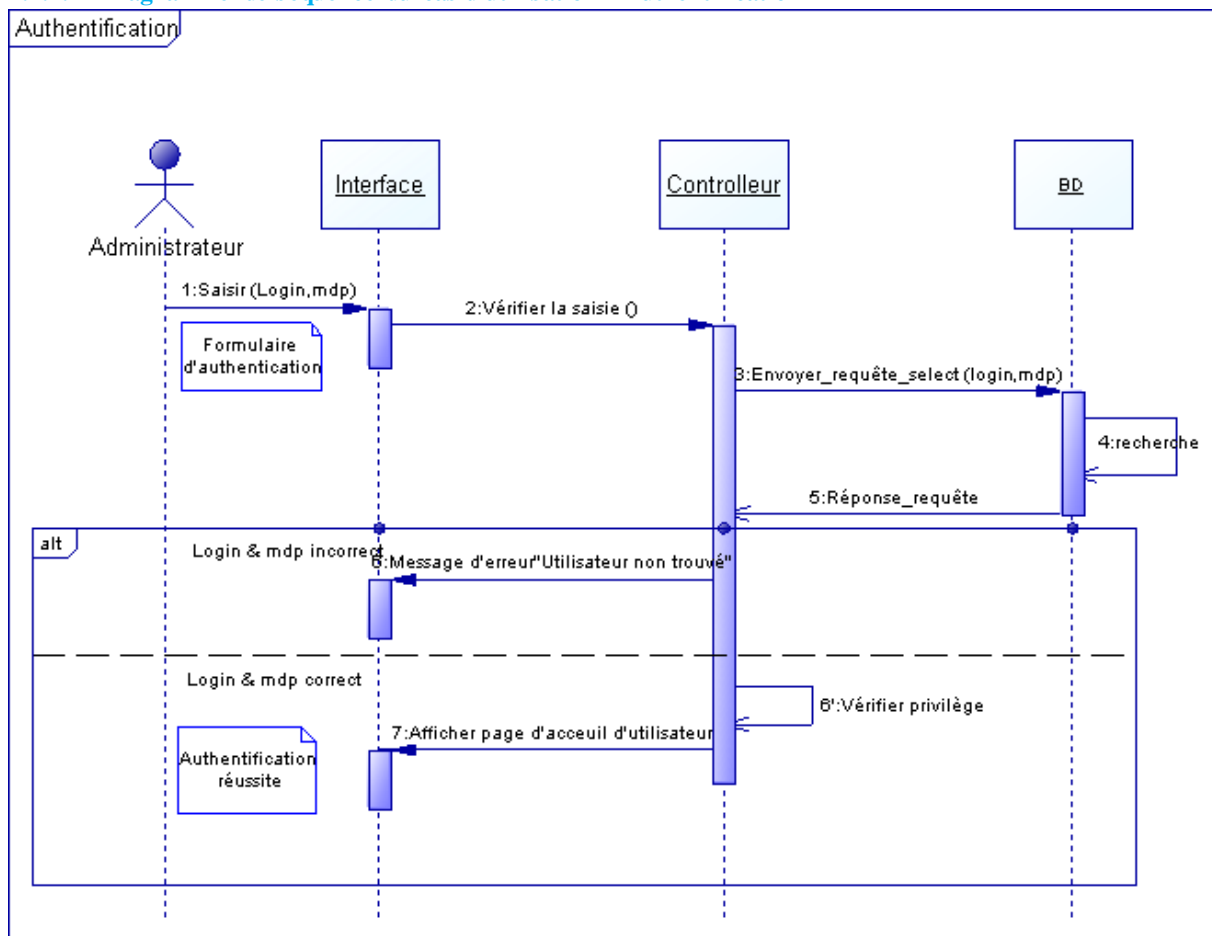


Figure 17: Diagramme de séquence du cas d'utilisation "Authentification"

1.1.2. Diagramme de séquence du cas d'utilisation "Ajouter un nouvel utilisateur "

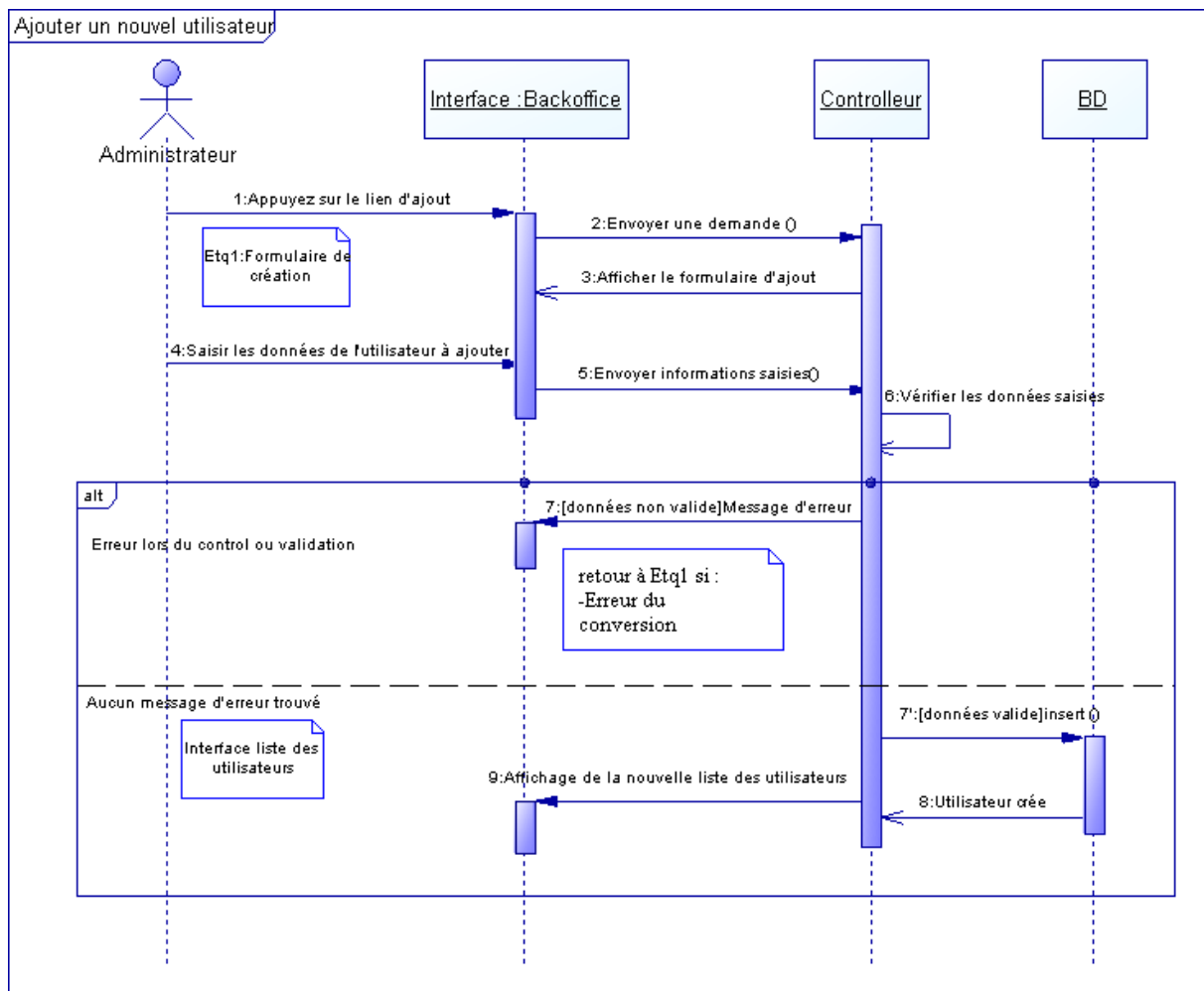


Figure 18:Diagramme de séquence du cas d'utilisation "Ajouter un nouvel utilisateur "

1.1.3. Diagramme de séquence du cas d'utilisation "Visualiser statistiques par capteurs "

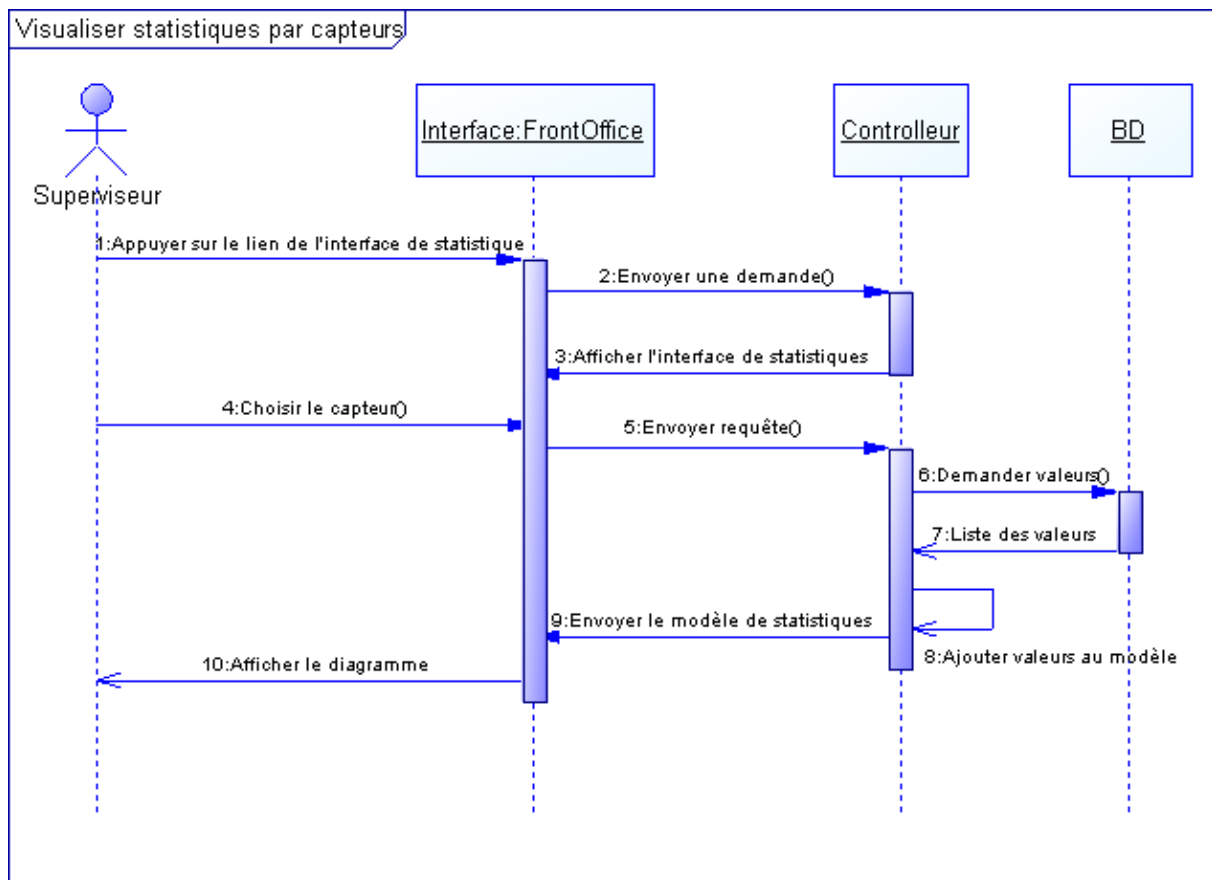


Figure 19:Diagramme de séquence du cas d'utilisation "Visualiser statistiques par capteurs "

1.2. Diagramme d'activité

Un diagramme d'activités permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Il est recommandable pour exprimer une dimension temporelle sur une partie du modèle, à partir de diagrammes de classes ou de cas d'utilisation, par exemple. En d'autres termes, le diagramme d'activités est une représentation proche de l'organigramme. Il nous permet de voir le comportement interne du système.

Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis, le passage d'une activité à une autre se fait par une flèche appelée « **transition** », liant ainsi les états d'activité entre eux séquentiellement.

Voici un schéma descriptif pour expliquer la structure d'un diagramme d'activité

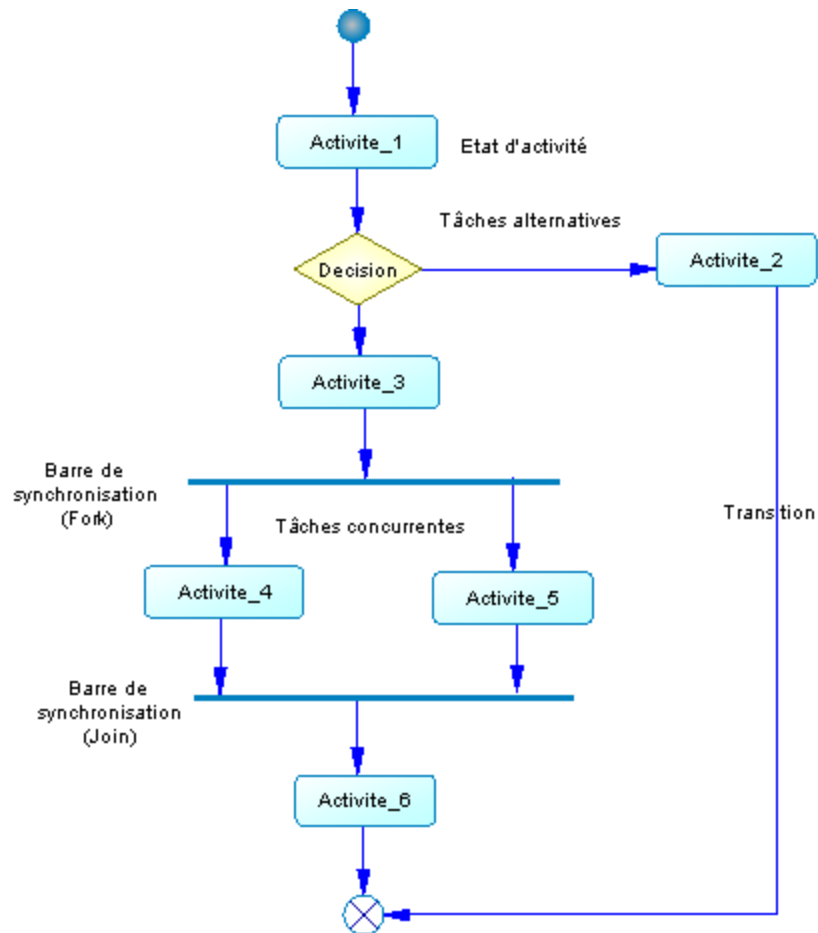


Figure 20:La structure du diagramme d'activité

Dans ce qui suit, nous présentons les diagrammes d'activités pour quelques cas d'utilisation dans notre système.

1.2.1. Diagramme d'activité "Authentification"

Le diagramme d'activité d'authentification décrit le processus d'authentification afin d'accéder à l'écran d'accueil selon le privilège d'utilisateur.

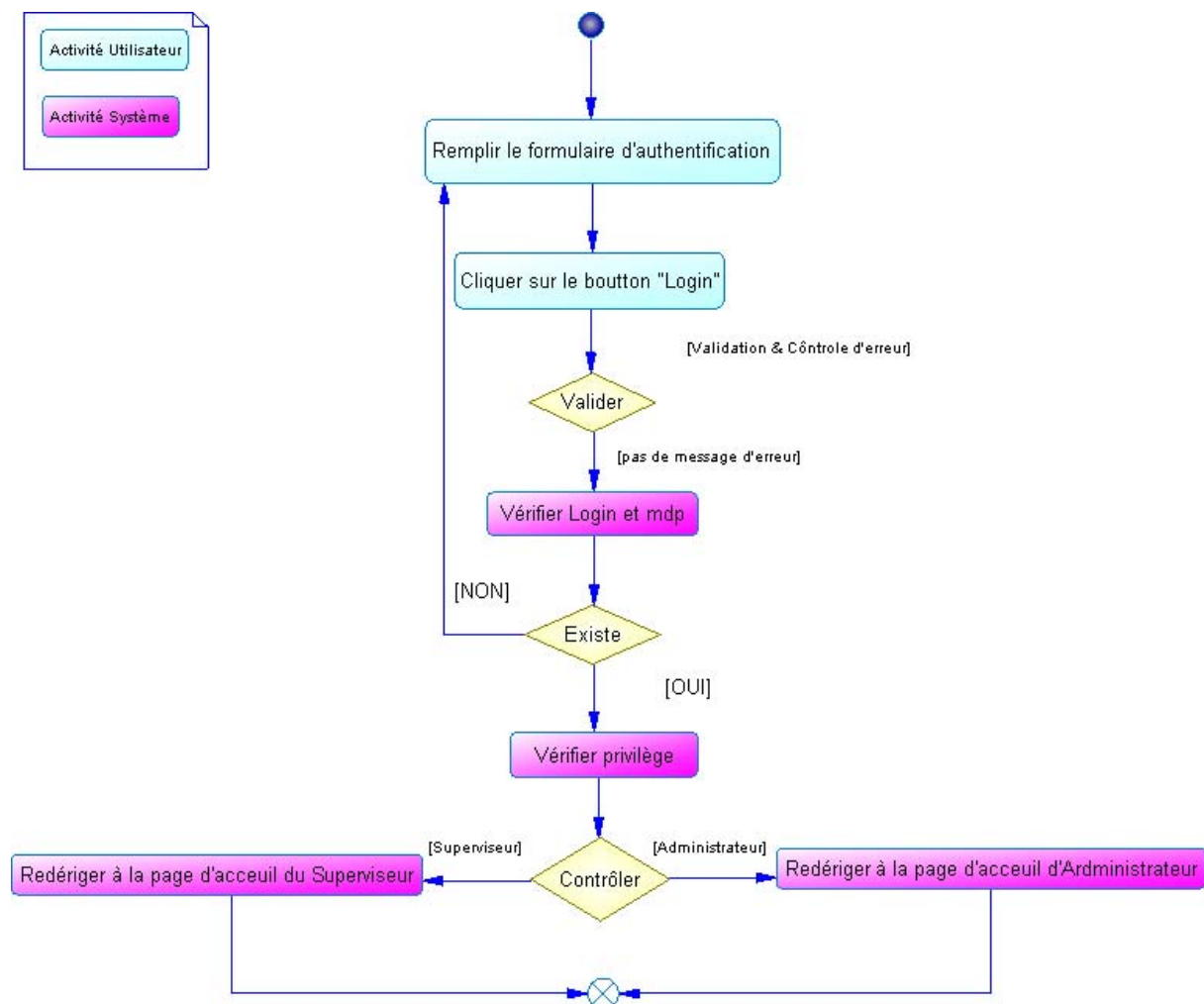


Figure 21:Diagramme d'activité "Authentification"

1.2.2. Diagramme d'activité "Ajouter capture "

Le diagramme d'activité "**Ajouter capture**" décrit comment créer une nouvelle capture de lecture manuellement dans la carte.

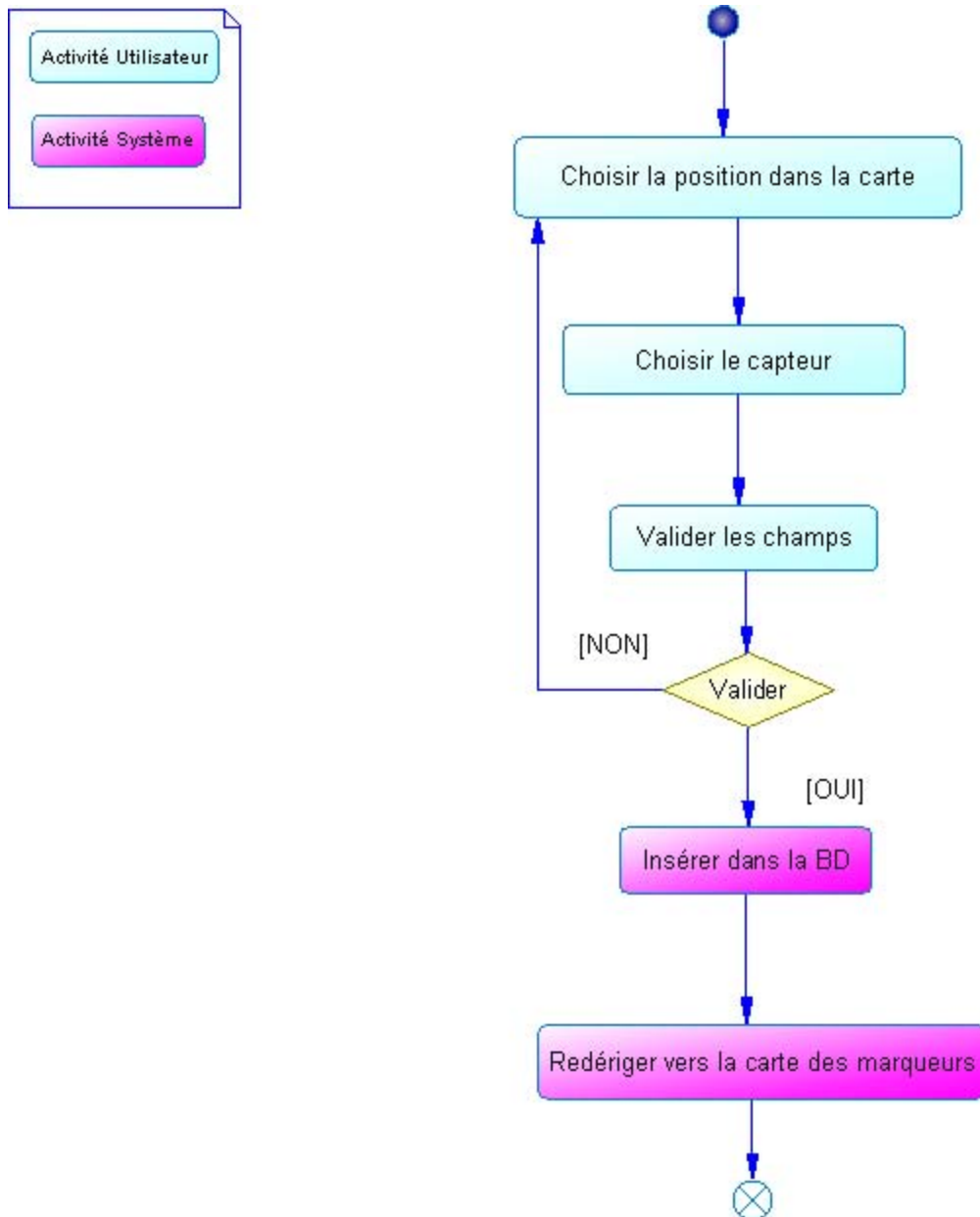


Figure 22:Diagramme d'activité "Ajouter capture "

2. Diagramme Statique

2.1. Diagramme de classe

La modélisation des concepts (ou des domaines) permet d'identifier les objets importants dans une application. Ce processus nous permettra d'identifier les futurs problèmes et de mieux comprendre le fonctionnement de l'application.

Ces concepts sont représentés dans le diagramme de classes. Le diagramme de classes est la clé de la conception orientée objet. Ce diagramme représente la structure du code à développer. Certaines applications UML permettent même d'exporter du code à partir de diagrammes de classes. Cela permet d'unifier le travail de plusieurs programmeurs au sein d'une même équipe, en plus de sauver du temps.

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** : 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.

Les diagrammes ci-dessous nous donnent une idée sur l'implémentation de différentes classes constituant notre application.

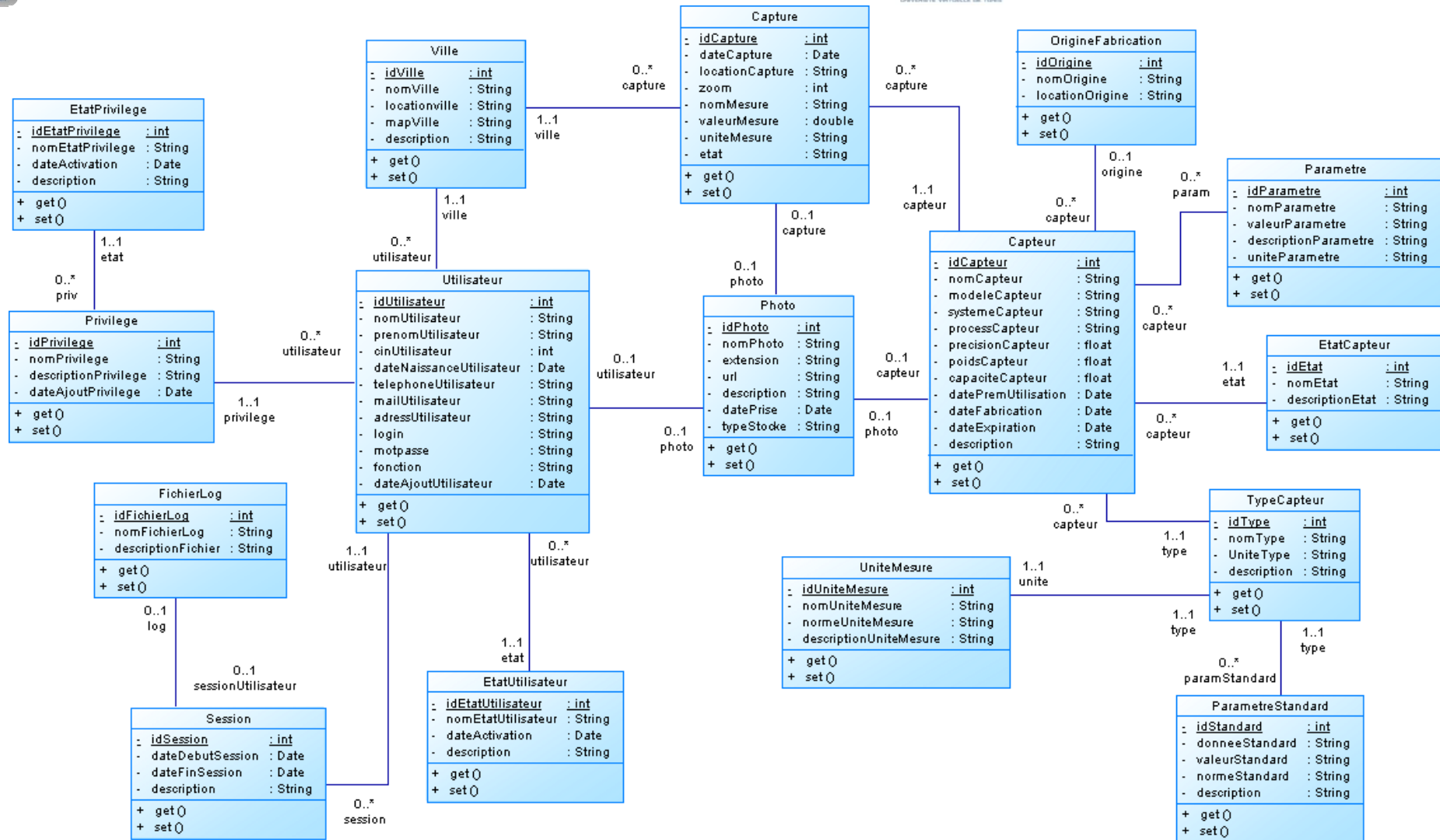


Figure 23:Diagramme de Classe

2.2. Diagramme de déploiement

Un diagramme de déploiement décrit, la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds.

La figure suivante présente le diagramme de déploiement de notre application :

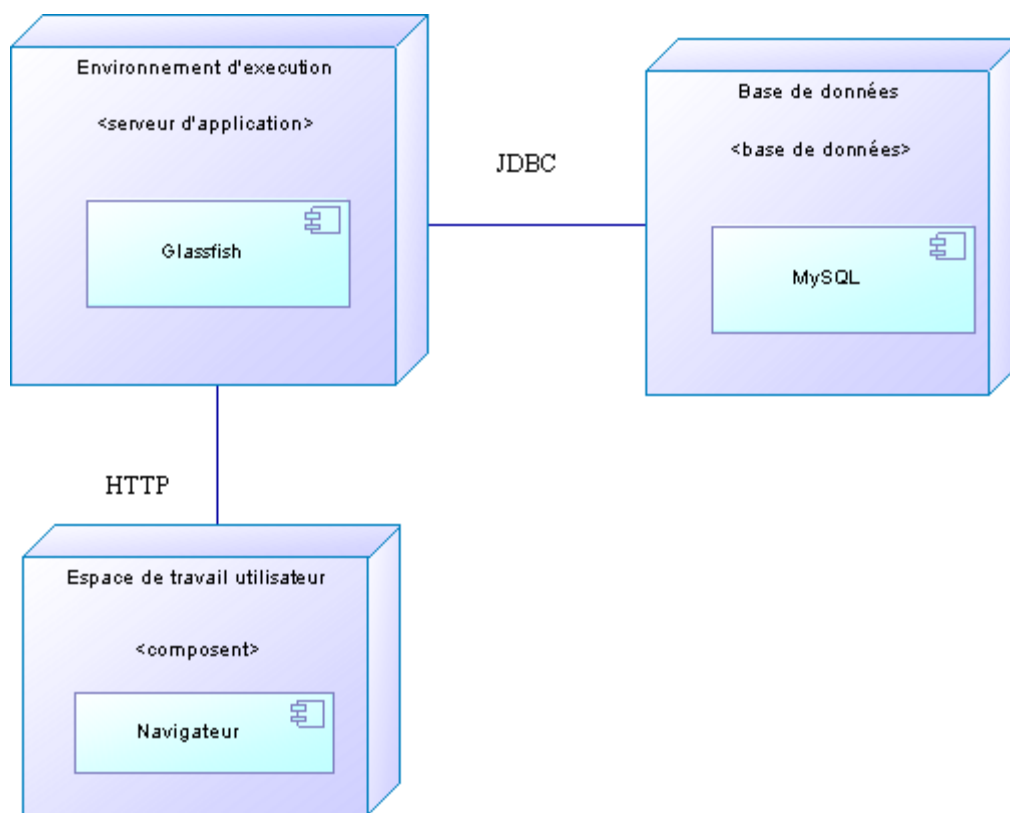


Figure 24:Diagramme de déploiement

Conclusion

Dans ce chapitre, nous avons conçu notre approche en adoptant la meilleure méthodologie, le choix du cycle de développement approprié et en indiquant certains modèles de conception. Après que nous approchions de la conception détaillée en donnant d'abord une vue sur le déroulement des événements à travers deux diagrammes d'activités à mettre en évidence deux séquences et enfin une vue globale du diagramme de classe, nous avons achevé le chapitre de conception avec un diagramme de déploiement afin de décrire la disposition matérielle. Le chapitre suivant sera consacré à la phase de réalisation du projet.

Chapitre 5 : Développement et Réalisation

Introduction

Ce chapitre donne un aperçu des conditions de travail réelles du projet. Il sera divisé en 3 sections. Tout d'abord, nous allons présenter l'architecture sur laquelle nous avons développé notre application. Ensuite, nous allons décrire les logiciels et l'environnement matériel pour atteindre la phase de développement. Enfin, nous exposons le travail accompli en détaillant quelques captures d'écran représentant les fonctionnalités réalisées, et justifiant les choix techniques des technologies adoptées.

I. Architecture de développement

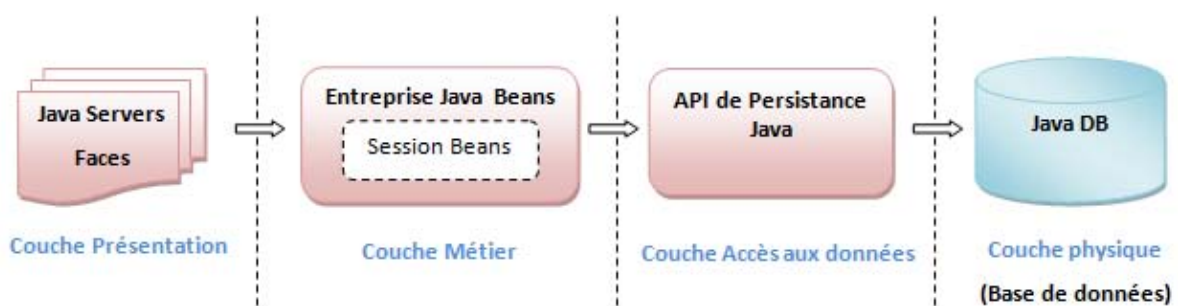


Figure 25:Architecture de développement

Pour notre projet, nous avons utilisé l'architecture Java Enterprise Edition ou Java EE (anciennement, J2EE), qui est une spécification Sun Java, spécialement conçu pour les applications d'entreprises. La figure ci-dessus a présenté l'architecture de développement adoptée, où il est organisé en différents niveaux : couche présentation, couche métier, couche d'accès aux données et de données.

Afin de faciliter la création d'applications distribuées utilisant les API Java, cette technologie fournit une plate-forme qui a été conçue pour faciliter la construction d'applications distribuées. Elle est connue par, sa sécurité, la disponibilité, l'évolutivité, portabilité, etc.

II. Environnement de développement

1. Environnement matériel

Pendant les processus de développement et de test, j'ai utilisé un ordinateur portable avec les spécifications suivantes :

- ❏ Fabricant : Dell.
- ❏ Modèle : Inspiron **n5110**.
- ❏ Processeur : Intel® Core i7-2670 QM.
- ❏ Mémoire : 8 Go de RAM.
- ❏ Type de système : Windows 7, 64bits.



Figure 26:PC DELL

2. Environnement logiciel

2.1. Environnement de développement intégré

Le projet NetBeans est constitué d'un IDE open-source et une plateforme d'applications qui permettent aux développeurs de créer rapidement des applications web, entreprise, mobiles en utilisant la plateforme Java, ainsi que PHP, JavaScript et Ajax, Groovy et Grails, et C / C ++.

L'IDE NetBeans est un environnement de développement intégré disponible pour Windows, Mac, Linux, et Solaris. La caractéristique la plus intéressante de l'utilisation de NetBeans pour notre application est qu'elle peut facilement créer des applications en utilisant les frameworks web avancées telles que JSF, Struts, Spring, Hibernate, et un ensemble complet d'outils pour Java EE 6, GlassFish 3, EJB, et les services Web développement.

En outre, il supporte les CSS, JavaScript, XHTML, Ajax et autres bibliothèques de composants comme PrimeFaces, RichFaces ...

Au cours de la phase de développement, nous avons utilisé NetBeans, il a aussi été utilisé pour mettre en œuvre et déployer les services sur le serveur Web GlassFish.

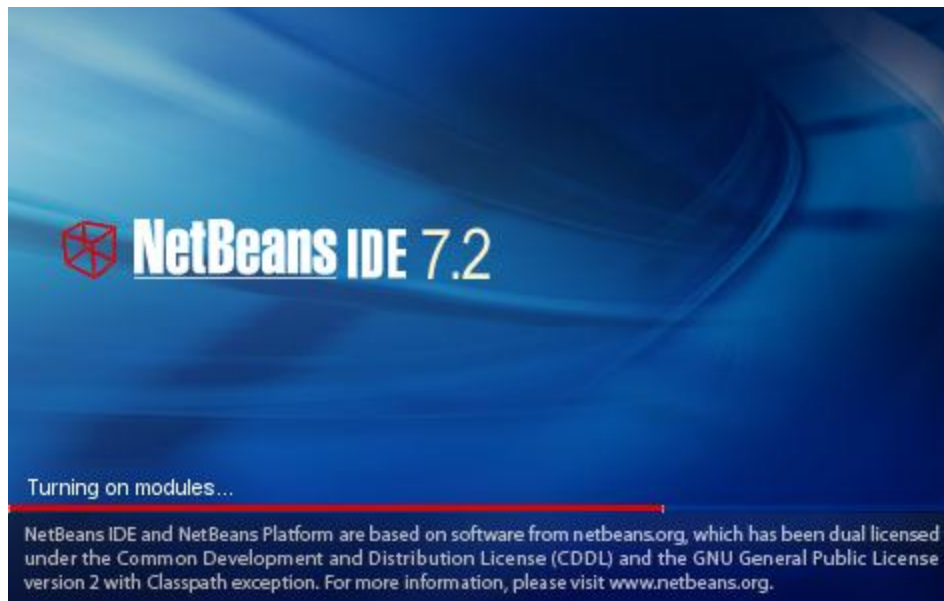


Figure 27:l'IDE Netbeans

2.2. Serveurs

2.2.1. GlassFish Open Source Edition 3.1.2

GlassFish est le nom du serveur d'applications Open source Java EE5 et désormais projet Java EE6 lancé par Sun Microsystems pour le Java EE plate-forme et maintenant parrainée par Oracle Corporation . La version prise en charge est appelée "Oracle GlassFish server".

GlassFish est l'implémentation de référence de Java EE et elle prend en charge les Enterprise JavaBeans , JPA , JavaServer Faces , JMS, RMI , JavaServer Pages , servlets, etc.

Cela permet aux développeurs de créer des applications d'entreprise qui sont portables et évolutives, et qui intègrent des technologies existantes. Les composants optionnels peuvent également être installés pour des services supplémentaires. GlassFish Server est aussi le plus rapide serveur d'applications open source offrant des fonctionnalités avancées.

✍ *Justification du choix : comparaison entre Glassfish et Apache Tomcat:*

GlassFish et Tomcat sont tous les deux des cadres de déploiement d'applications Web pour Java, nous avons choisi Glassfish selon les critères suivants.

Déploiement :

Les deux GlassFish et Tomcat sont écrits en Java pour l'environnement d'exécution Java. Ils sont conçus pour déployer des applications Web Java.

- ✓ GlassFish a pratiquement un support complet pour Java EE 5 et 6, tandis que Tomcat ne supporte que les normes de servlets et JSP.
- ✓ Pratiquement tous les services web conçus pour fonctionner sur Tomcat peuvent également être exécutés sur GlassFish.

Administration :

- ✓ GlassFish fournit plus d'outils pour l'administration mieux que Tomcat.

Développement / maintenance :

Glassfish est maintenant possédé par Oracle et est maintenu par Oracle et sa communauté de développement. Tomcat est un projet de la fondation Apache et est maintenu par la communauté de développement.

Performance :

Tomcat est plus rapide à charger, mais GlassFish est plus rapide à recharger. Outre les temps de chargement, Tomcat et GlassFish sont similaires dans la performance.

Dépendances :

- ✓ GlassFish est livré avec sa propre interface HTTP nommé Grizzly.
- ✓ Tomcat est souvent installé avec Apache httpd pour équilibrage de charge. Elle ajoute grande souplesse dans la configuration.

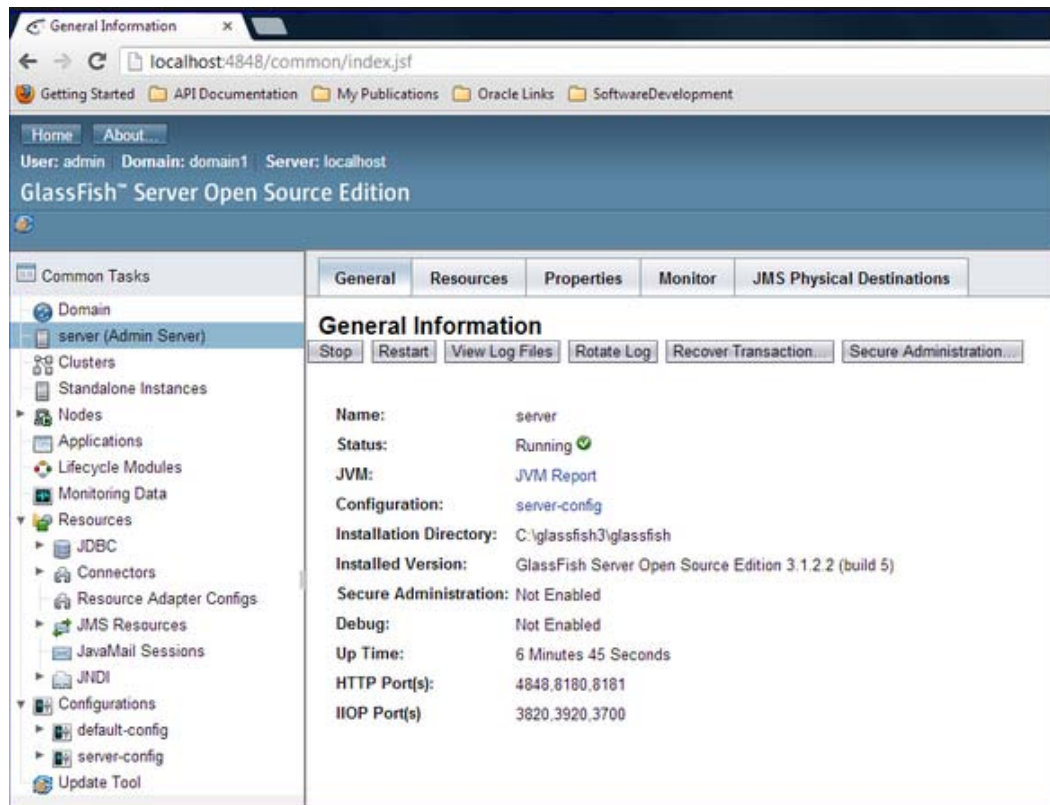
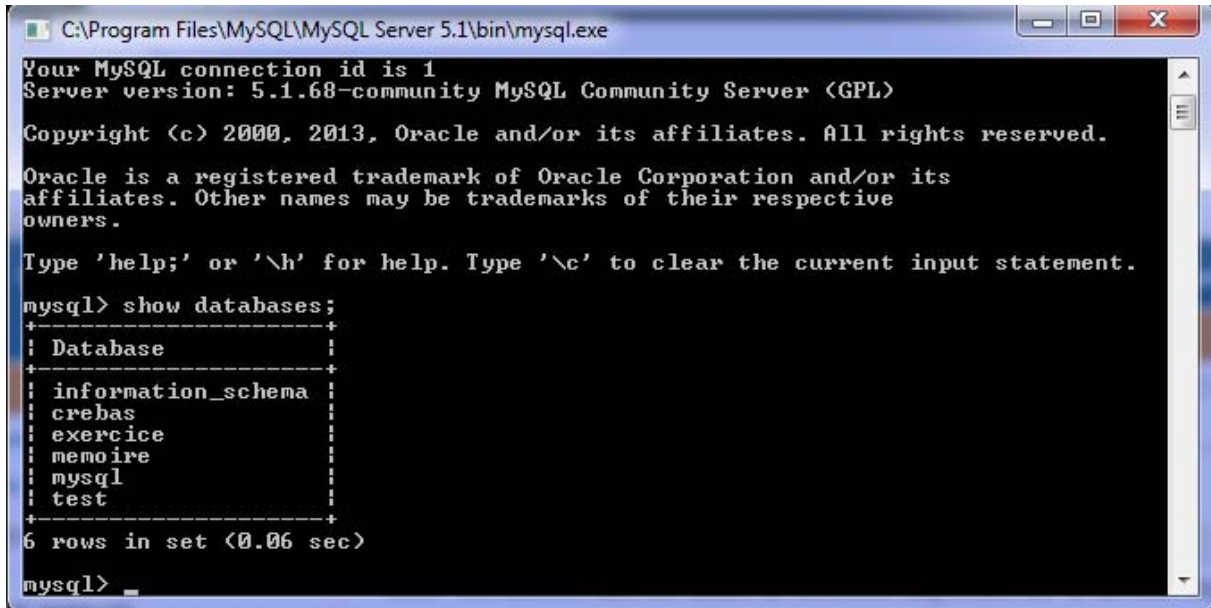


Figure 28: GlassFish Open Source Edition

2.2.1. MySQL 5.1

MySQL est un système de gestion de base de données (SGBD) open source et le plus populaire au monde. Il est distribué sous une double licence GPL et propriétaire. Il est le plus utilisé par le grand public (applications web) que par des professionnels, il fonctionne sur plusieurs plates-formes (Windows, Mac OSX, Linux, Open Solaris...).

MySQL supporte deux langages informatiques, le langage de requête SQL et le SQL/PSM (Persistent Stored Modules) qui permettent de combiner des requêtes SQL et des instructions procédurales (boucles, conditions...), dans le but de créer des traitements complexes destinés à être stockés sur le serveur de base de données.



```

C:\Program Files\MySQL\MySQL Server 5.1\bin\mysql.exe
Your MySQL connection id is 1
Server version: 5.1.68-community MySQL Community Server (GPL)
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| crebas      |
| exercice    |
| memoire     |
| mysql       |
| test        |
+-----+
6 rows in set (0.06 sec)
mysql>

```

Figure 29: L'espace de travail de MySQL 5.1

2.3. Outils de conception (Power AMC)

PowerAMC est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées. Créé par SDP sous le nom AMC Designer, racheté par Powersoft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque PowerDesigner.

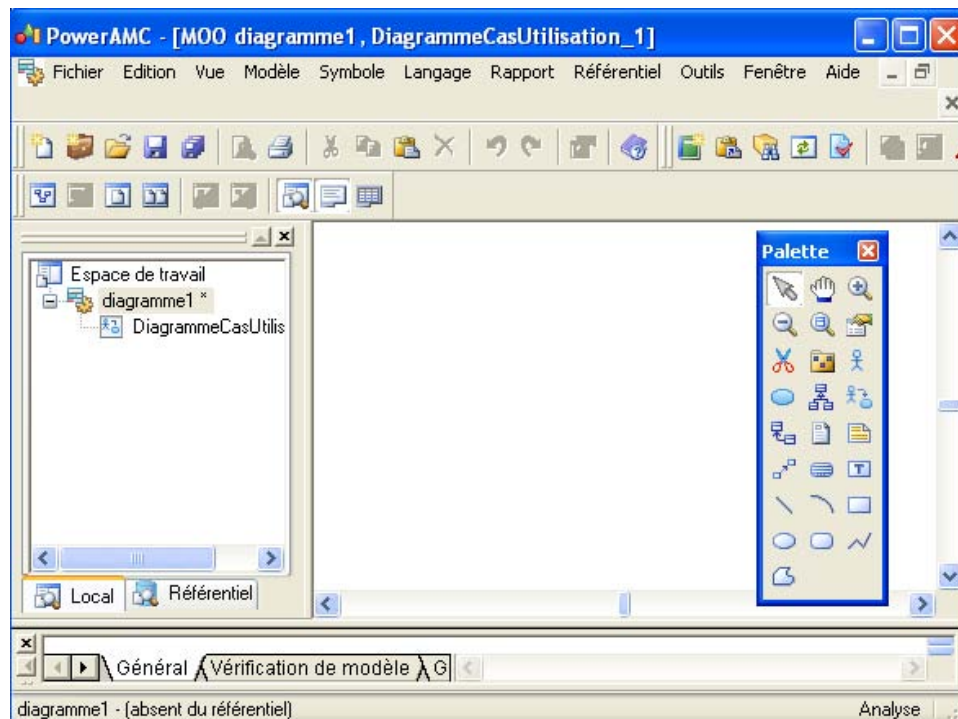


Figure 30: L'espace de travail de PowerAMC

2.4. Framework et choix techniques

2.4.1. JSF 2.1

Java Server Faces (abrégé en JSF) est un Framework Java, pour le développement d'Web. À l'inverse des autres Framework MVC traditionnels à base d'actions, il est basé sur la notion de composants, comparable à celle de Swing ou SWT, où l'état d'un composant est enregistré lors du rendu de la page, pour être ensuite restauré au retour de la requête.

Justification du Choix : JSF vs. Struts :

Le point commun de ces deux Framework, c'est que tous les deux respectent le pattern MVC "Modèle Vue Contrôleur". Le Framework Struts est implémenté par apache, alors que JSF est mis en œuvre par Sun Microsystems.

Les Points forts de Struts :

- ✓ Paramétrage par des fichiers XML (emplacement des beans, mapping, ...);
- ✓ Pas de servlet mais des Action et des Action Form (formulaire facilité avec vérification des champs);
- ✓ Paramétrage des langues (messageRessource.properties);

Les Points faibles de Struts :

- ✗ Installation et paramétrage difficile ;
- ✗ Erreur difficile à trouver (car séparation des fichiers et des Action à effectuer) ;

Les Points forts de JSF :

- ✓ Faciliter l'écriture d'interface à partir d'une bibliothèque de contrôles ;
- ✓ Gérer automatiquement l'état HTTP entre client et serveur (en cas de Postback par exemple);
- ✓ Fournir un modèle simple pour la gestion des événements côté client et coté serveur ;
- ✓ Autoriser les créations ou l'enrichissement de composants graphiques utilisateur (custom) ;
- ✓ Utilise les managed bean ;
- ✓ Facilité de contrôle de formulaire ;
- ✓ Nombreux composants graphiques réutilisables ;
- ✓ Mapping HTML/Objet ;
- ✓ Configuration de la navigation entre les pages.

Les Points faibles de JSF :

- ✗ Maturité : technologie récente qui nécessite l'écriture de beaucoup de code ;
- ✗ Manque de composants évolués en standard : la plupart ont une correspondance directe avec l'HTML.

2.4.2. EJB 3.1

Enterprise JavaBeans (EJB) est une architecture de composants logiciels côté serveur pour la Plateforme Java Enterprise Edition (Java EE). La technologie EJB permet le développement rapide et simplifié des applications distribuées, transactionnelles, sécurisées et portables basés sur la technologie Java.

EJB sont des composants cotés serveur qui simplifient le développement d'applications distribuées en Java. Ils peuvent faire des applications évolutives et sécurisées et sans la nécessité des Framework complexe.

EJB permettent le développement rapide des composants cotés serveur. Ils sont conçus pour être portables et réutilisables par l'utilisation des Sessions Beans. La plus commune utilisation d'une session bean est d'implémenter session façade d'un modèle de conception J2EE.

Une session façade est une session bean qui regroupe les données et les présente à la demande à travers la couche de modèle. Session façades a des méthodes qui accèdent à des entités ainsi que les méthodes qui exposent des services aux clients. Les Sessions Beans ont un contexte transactionnel via le conteneur, ils prennent automatiquement en charge la fonctionnalité de base CRUD.

2.4.3. JPA 2

L'API Java Persistence (JPA), est un Framework qui permet de gérer les données relationnelles dans une application utilisant la plateforme java. JPA a deux aspects, le premier est la capacité d'associer des objets à une base de données relationnelle, la deuxième est que la configuration par exception permet aux fournisseurs de persister plus gros du travail sans ajouter beaucoup de code, mais la richesse de JPA a également la possibilité d'adapter ces associations à l'aide des annotations ou des descriptions XML. Que ce soit une simple modification (changer le nom d'une colonne, par exemple) ou une adaptation plus complexe (pour traduire l'héritage), JPA offre une large grande possibilité.

Dans le monde Java, où on manipule des objets, un langage conçu pour les tables (SQL) devrait être légèrement déformé en fonction d'un langage orienté objet (Java). C'est la JPQL (Java Persistence Query Language) entre en jeu.

JPQL est la langue utilisée par JPA pour interroger les entités stockées dans une base de données relationnelle. Sa syntaxe est similaire à SQL, mais fonctionne sur les entités objets au lieu d'agir directement sur les tables.

2.4.4. PrimeFaces 3.3

PrimeFaces est une bibliothèque de composants open source légers pour Java Server Faces. Nous avons utilisé PrimeFaces pour les raisons suivantes :

- Ensemble riche de composants d'interface utilisateur.
- Aucune configuration XML supplémentaire n'est nécessaire

- Pas de dépendances nécessaires.
- Basé sur la norme JSF 2.0 API Ajax.
- Des thèmes et des apparences multiples sont disponibles.

La version que nous avons utilisée est 3.3.

✍ *Justification du choix: Primefaces vs. Richfaces vs ICEfaces:*



Figure 31:L'utilisation au cours des dernières années

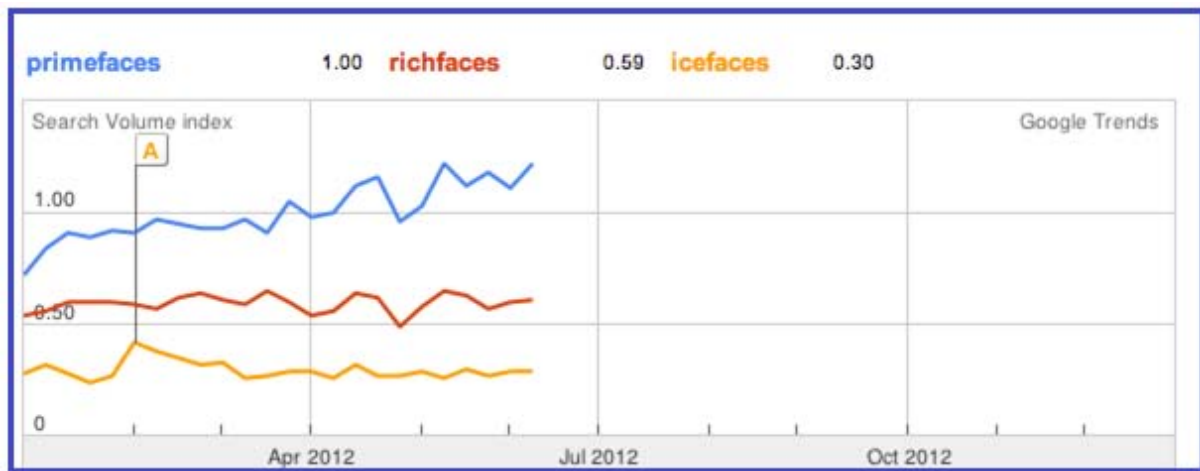


Figure 32:L'utilisation au cours des derniers mois

2.4.5. Log4J

Dans la plupart des tâches de serveur, l'exploitation est cruciale, car l'administrateur n'est pas généralement présent quand les échecs se produisent, mais il peut vérifier après. Dans le cas d'une erreur, la pile de traces peut dire l'état du programme au point exact de l'échec. Une autre utilisation de la journalisation, qui pourrait ne pas être pratique pour tous les programmes, est l'analyse des statistiques.

Par exemple, la connexion à chaque demande qui entre dans le serveur Web, puis analyse le fichier journal peut aider à produire des graphiques des heures de pointe, les pages les plus populaires et ainsi de suite.

Nous avons opté pour Log4j qui est un cadre simple et flexible de journalisation Apache Software Foundation. Il définit un ensemble de niveaux de journalisation et les messages, par ordre de gravité décroissante (FATAL ERROR, WARN, INFO ...)

III. Travail réalisé

Le travail réalisé est présenté par les applications Front-office et Back-office, dans cette section, nous allons présenter quelques captures d'écran des interfaces utilisateur.

1. Page d'authentification

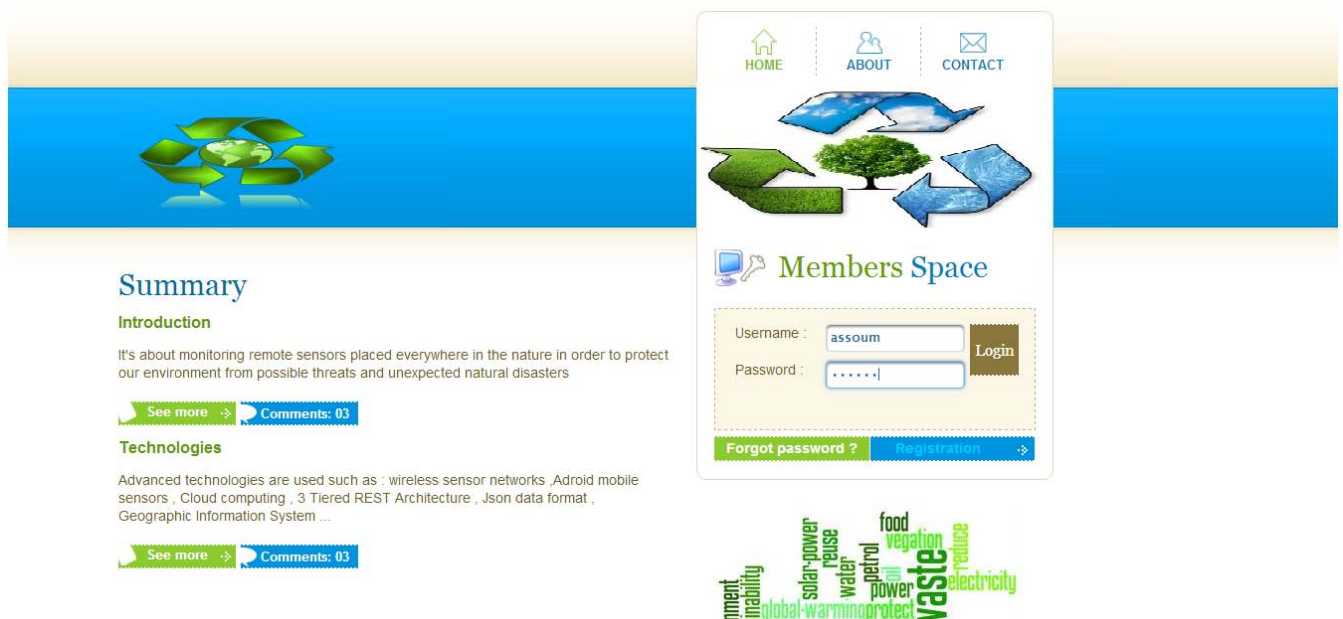


Figure 33:Page d'authentification

2. Application Back-office

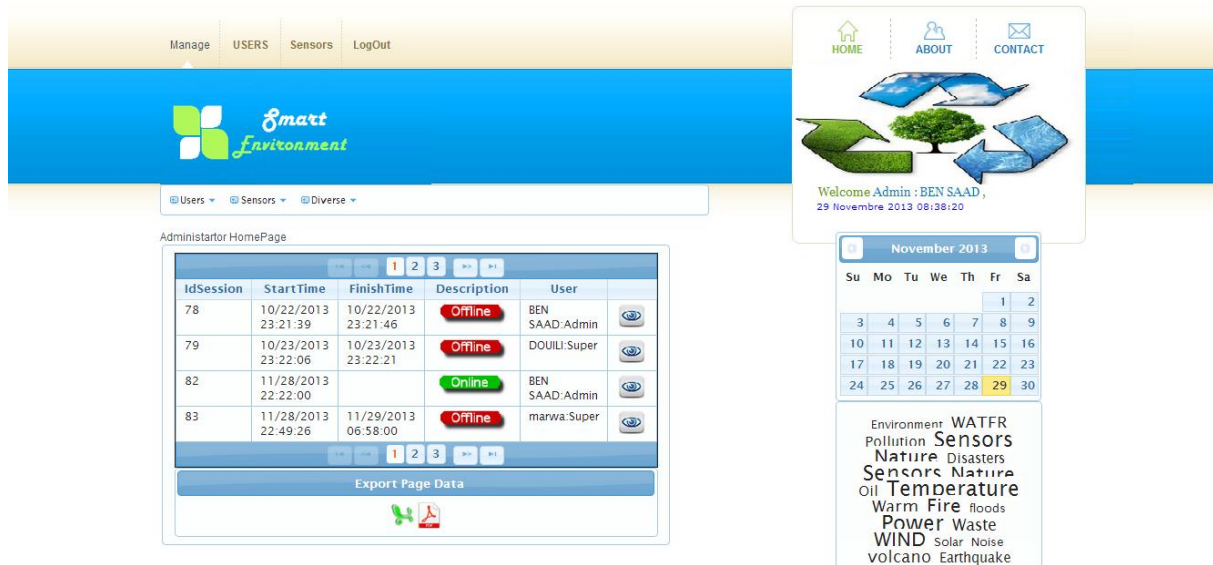


Figure 34:Interface d'accueil Administrateur

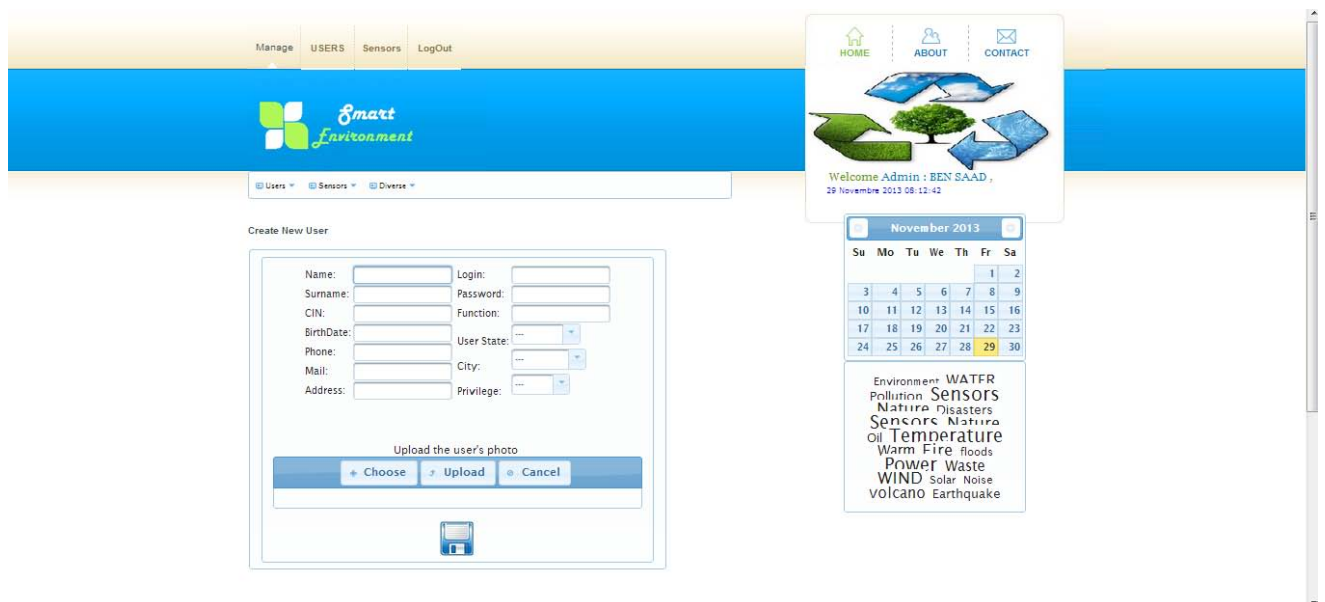


Figure 35:Interface Ajout d'un nouvel utilisateur

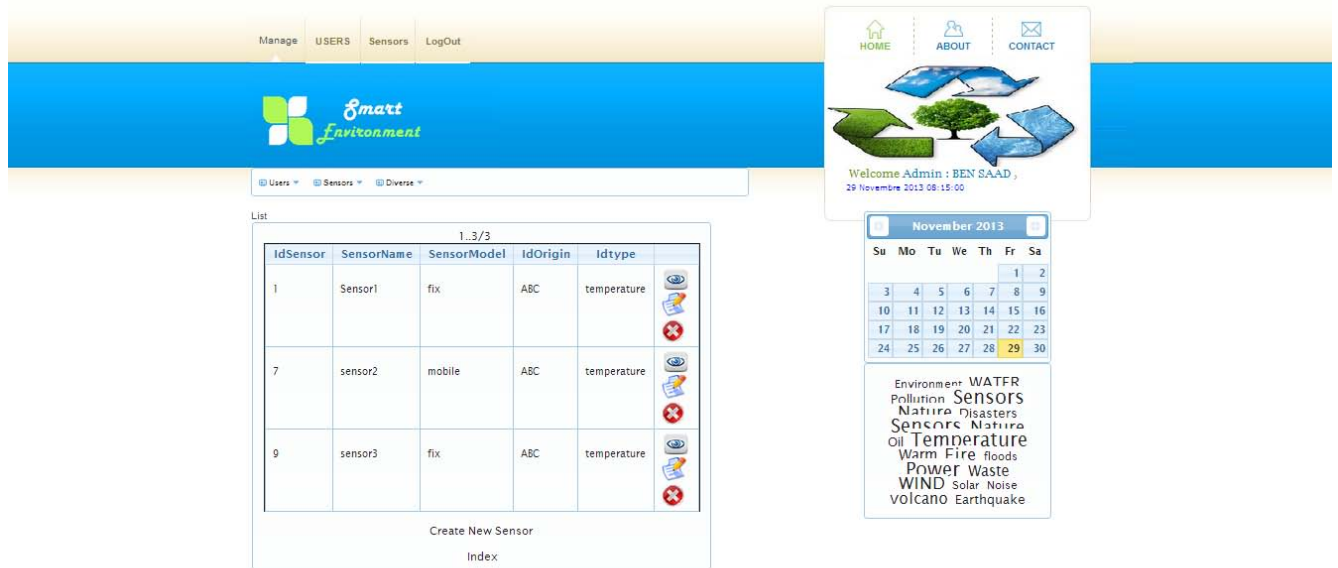


Figure 36:Interface Afficher tous les capteurs

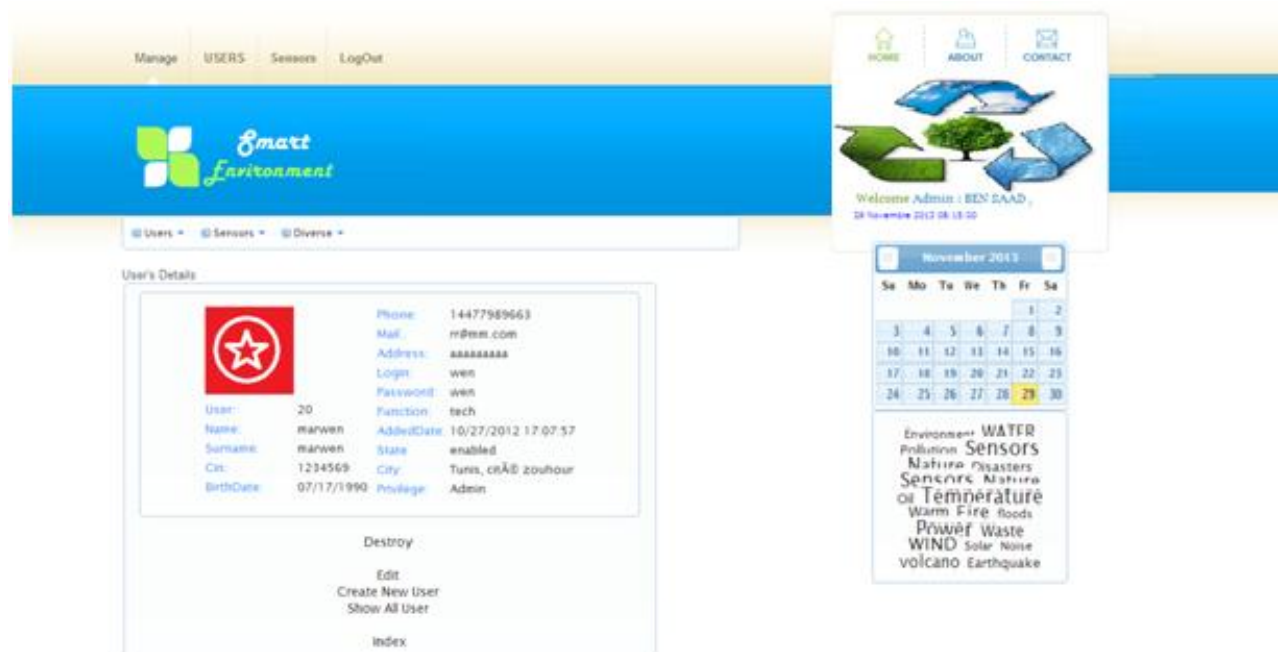


Figure 37:Interface Afficher les détails d'un utilisateur

3. Application Front-office

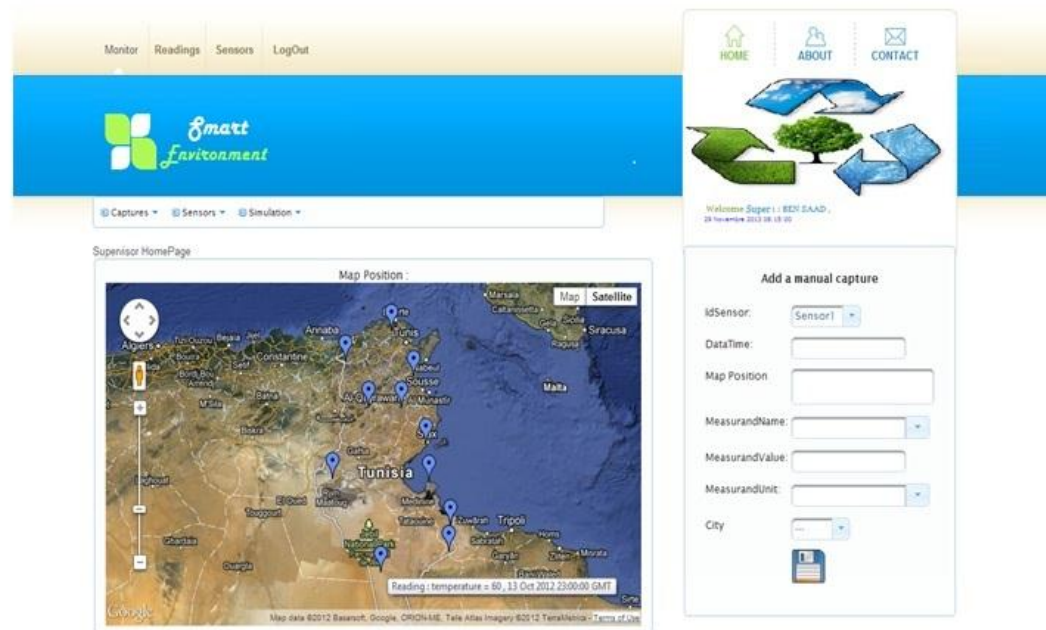


Figure 38:Interface d'accueil Superviseur

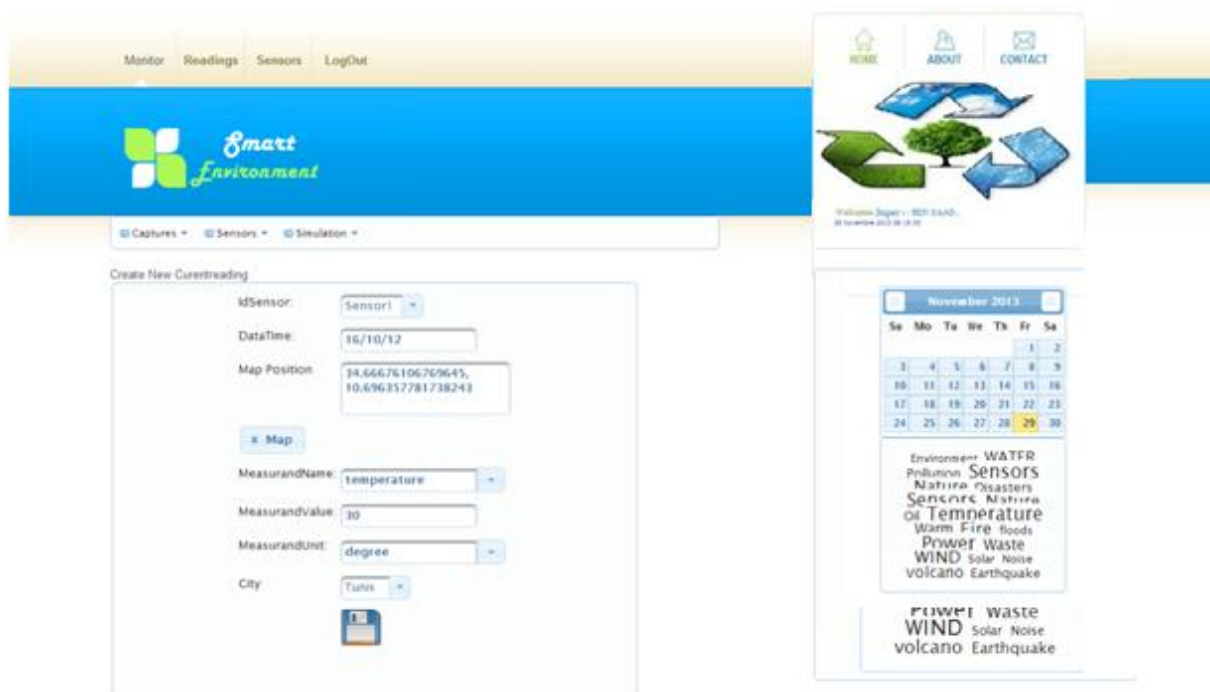


Figure 39:Interface Ajouter nouvelle capture



Figure 40:Interface Simuler captures du température

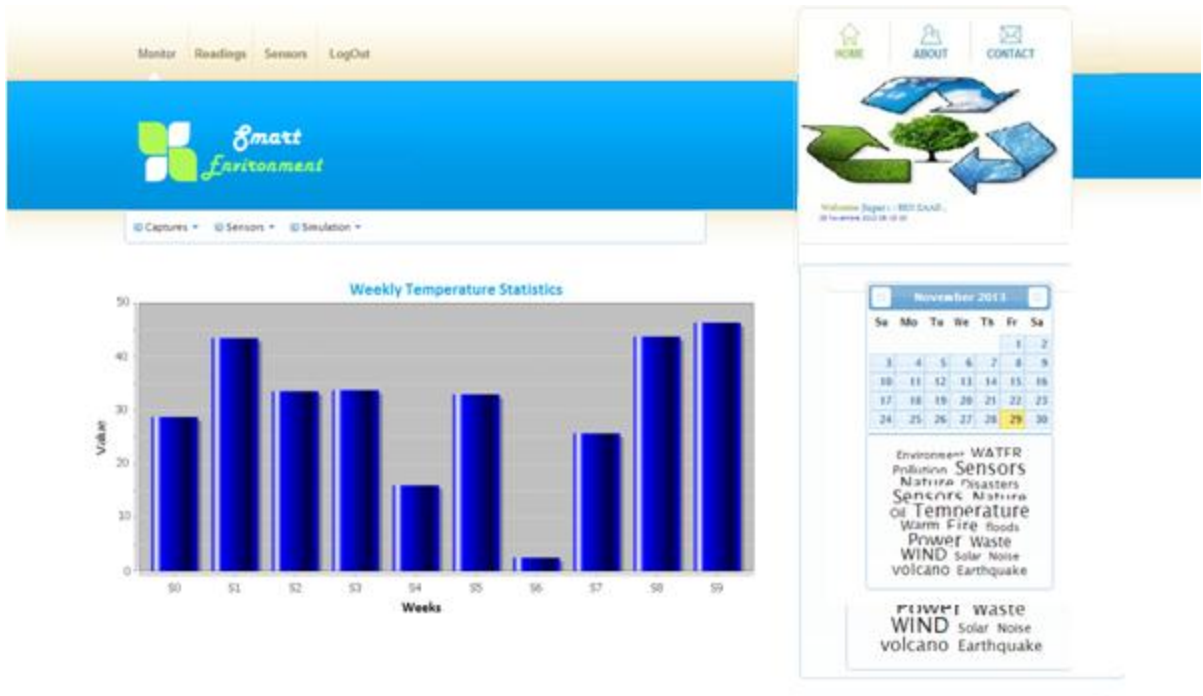


Figure 41:Interface Générer les statistiques de température

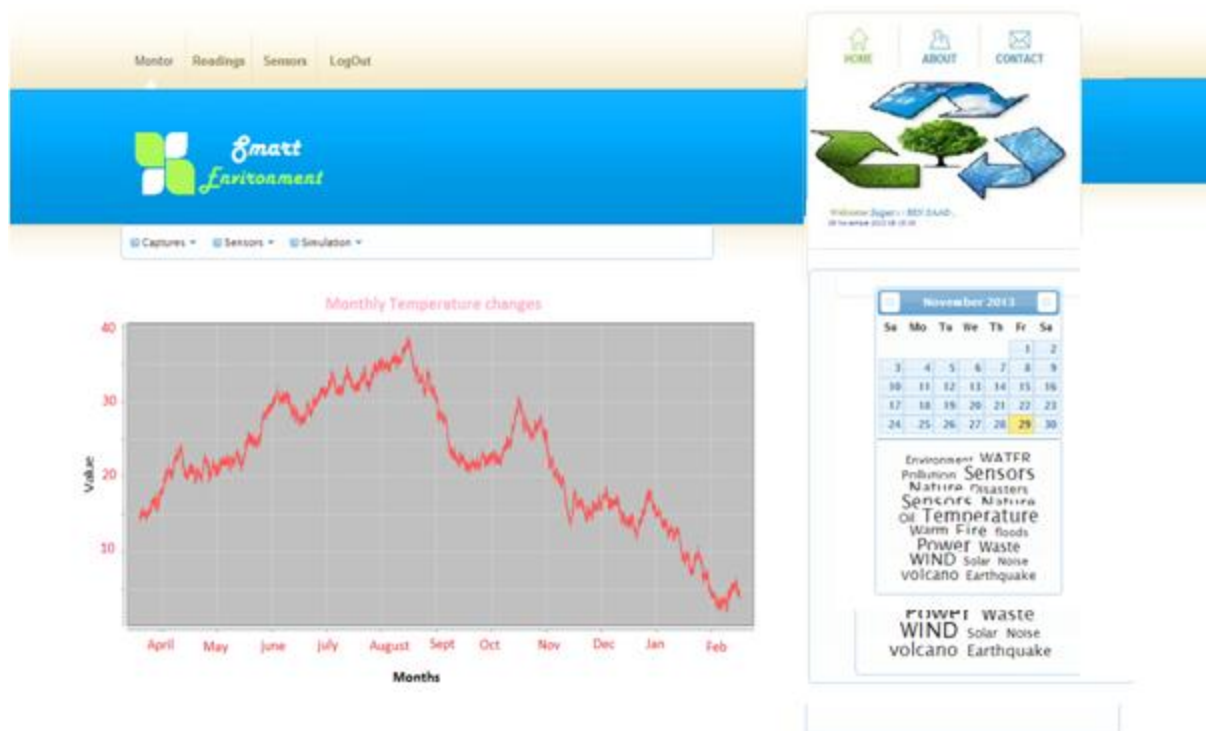


Figure 42:Interface Générer les statistiques de température

Conclusion

Dans ce chapitre, nous avons parlé de l'environnement de travail et les outils utilisés ensuite exposer les différents résultats obtenus.

Conclusion et perspectives

Le travail élaboré dans ce projet a été réalisé durant le stage d'un projet d'obtention de diplôme chez l'UVT en collaboration avec le centre de développement de Tunisie-Télécom. L'objectif principal pour notre projet de fin d'études est la conception ainsi que l'implémentation d'une application de gestion du trafic de données d'un réseau mobile de capteurs.

Nous avons ainsi divisé notre travail en 3 parties principales :

Pour la première partie, nous avons commencé par la description de l'organisation hôte dont on a eu l'opportunité pour exécuter ce projet et nous nous sommes approchés de fond du projet en définissant quelques concepts basiques des capteurs et des applications et généralement en passant par les problèmes des architectures existantes.

Dans la deuxième partie, on a mis en transparence les différentes spécifications des analyses et des exigences.

La troisième partie, a été consacrée pour la réalisation de la conception de l'application ainsi que la présentation d'une conception globale de la solution puis accédée à la conception détaillée de l'application avec les différents diagrammes UML.

Finalement, nous avons donné un petit aperçu de la réalisation du projet par un listing des environnements matériels et logiciels et par la suite, les différents Framework et les technologies exploités durant les processus de développement ensuite l'exposition des différents résultats obtenus ...

L'usage de toutes ces technologies m'a donné l'opportunité de mettre en pratique certaines connaissances obtenues tout au long de mon cursus d'études de Master. L'évolution de mon projet m'a appris diverses choses telles que le traitement des délais, la résolution des problèmes en suivant les méthodologies, correction des bugs, la recherche ciblée

C'était une grande expérience en termes de travail d'équipe, la communication et la synchronisation avec d'autres équipes : c'était une occasion unique pour avoir l'opportunité à expérimenter la participation au monde professionnel et à mon optique personnelle, j'ai vraiment apprécié le travail sur ce projet.

Comme il s'agit d'un projet ouvert qui dépasse et surpasse le cadre de ce projet de fin d'études plusieurs améliorations sont encore à faire et à appliquer. On pouvait distinguer des améliorations qui sont d'ordre immédiat et d'autres à long terme. Les améliorations immédiates sont souvent relatives au back-end office. Certaines modifications peuvent apporter plus de fonctionnalités, parmi eux, je pourrais citer et mentionner la partie mobilité du sujet, l'optimisation de la structure de données pour une meilleure gestion des ressources du CPU. En outre, un autre objectif en futur proche à citer est le développement d'une interface graphique mobile améliorée avec un aperçu en temps réel.

En ce qui concerne les perspectives à long terme, je peux envisager que la continuité et la réussite de ce projet dépendent impérativement de la mise à jour et l'enrichissement de certaines de ses caractéristiques. Néanmoins, l'extension de la compatibilité de la solution de mobilité afin d'équilibrer et compenser les terminaux mobiles tels qu'iPhone et BlackBerry. En effet, ces 2 OS ainsi que l'Android sont les plus répandus et exploités de nos jours.

À cause des contraintes relatives au temps et aux ressources, d'autres améliorations auraient pu être appliquées sur ce travail, particulièrement, les aspects concernant la sécurité du tout le trafic réseau et l'utilisation de l'interface du réseau POP (point of présence) afin de gérer les systèmes de notification et de filtrage.

Bibliographie & Netographie

- [1] Liljana Gavrilovska, Srdjan Krco, "Application and multidisciplinary aspects of wireless sensor networks", Springer, December 2009, pages 267-269.
- [2] Cristian Hesselman, "Mobilware 2009", Springer, 2009, page 149.
- [3] Charith Pereray, Arkady Zaslavskyy, Peter Christen, Ali Salehiy and Dimitrios Georgakopoulos, Article: "Capturing Sensor Data from Mobile Phones using Global Sensor Network Middleware", 1 Feb 2013.
- [4] Xiang Sheng, Xuejie Xiao, Jian Tang and Guoliang Xue, Article: "Sensing as a Service: A Cloud Computing System for Mobile Phone Sensing" Published in Sensors, 2012 IEEE
- [5] Czech Science Foundation. <http://agrisensor.geogr.muni.cz>.
- [6] PrimeFaces :<http://www.primefaces.org/showcase-labs/ui/home.jsf>
- [7] TP: <http://netbeans.dzone.com/articles/develop-java-ee-6-app-jsf2>
- [8] Cycle en V: http://fr.wikipedia.org/wiki/Cycle_en_V
- [9] "Deployment Diagram and Object Diagrams" from exforsys.com:
<http://www.exforsys.com/tutorials/uml/uml-elements-class-diagram/1.html>
- [10] PowerAMC de la Wikipedia: <http://en.wikipedia.org/wiki/PowerDesigner>
- [11] NetBeans: <https://netbeans.org/community/releases/70/>
- [12] Glassfish de la Wikipedia : <http://en.wikipedia.org/wiki/GlassFish>
- [13] Comparaison entre Glassfish et Apache Tomcat :
<http://stackoverflow.com/questions/3821640/what-is-difference-between-tomcat-and-jboss-and-glassfish>
- [14] JavaServer_Faces de la wikipedia : http://fr.wikipedia.org/wiki/JavaServer_Faces
- [15] JSF vs. Struts: http://gardeux-vincent.eu/Documents/ProjetJEE/AAHFCL_iBatis_JSF/jsf-5.html