



# MEMOIRE DE PROJET DE FIN D'ETUDE

Pour l'obtention du diplôme de master  
Systèmes Microélectroniques, de Télécommunications et de l'Informatique  
Industrielle

*Assistance et automatisation du processus visite à  
l'arrivée des locomotives électriques.*

*Réalisé par :*

✚ Mr Kamal BELGHAZI.  
✚ Mr Abdelkarim TAYACHE.

*Encadré par :*

✚ Pr T. LAMCHARFI  
✚ Pr F. ERRAHIMI  
✚ Mr M. EL OTMANI

*Soutenu le : 22 Juin 2012*

*Devant le jury composé de :*

✚ Pr M. LAHBABI  
✚ Pr F. ABDI  
✚ Pr T. LAMHAMDI  
✚ Pr T. LAMCHARFI  
✚ Pr F. ERRAHIMI  
✚ Mr M. EL OTMANI  
*Casablanca)*

*Pr (F.S.T.Fès)*  
*Pr (F.S.T.Fès)*  
*Pr (F.S.T.Fès)*  
*Encadrant (F.S.T.Fès)*  
*Encadrant (F.S.T.Fès)*  
*Encadrant société (ONCF*



## ANNEE UNIVERSITAIRE : 2011/2012

### REMERCIEMENT

Nous remercions Dieu de nous avoir aidés à réaliser ce travail, et nous remercions nos parents pour tout leur sacrifice et leur soutien.

Nous tenons, aussi à remercier la Direction de l'ONCF qui nous a permis d'effectuer ce stage au sein du service maintenance matériel de Casablanca, et précisément le chef de l'établissement Mr Abdelwahed ELMOUSTAFA, pour nous avoir accueilli dans son établissement et nous avoir permis de découvrir les différentes fonctions des Ressources Humaines ainsi que les mécanismes de l'entreprise .

Au terme de ce travail, nous tenons à remercier Mr.Taj-edine LAMCHARFI et Mme.Fatima ERRAHIMI d'avoir accepté de nous encadré tout au long de cette période. Nous les remercions aussi pour leur aide, leur soutien, leur disponibilité ; pour le temps et la confiance qu'ils nous ont accordé.

Nous remercions Mr Mohamed EL OTMANI, notre maître de stage pour nous avoir accueilli dans son équipe, pour son soutien et sa présence en permanence dans le déroulement du stage, nous le remercions aussi pour ses consignes et son temps accordé au long de cette période.

De plus nous adressons nos remerciements à :

- Mr. MHAYA, SAHBI et les autres membres d'équipe ONCF, pour leur accueil, leur soutien et leur confiance.
- Mr Saïd NAJAH, Mlle Fatiha EL MRABTI pour leur soutien technique.
- FST Fès, le département GE et l'ensemble des professeurs, pour ces deux années d'enseignements et de collaboration.

Nous remercions également tous nos collègues de la filière SMTII pour leur aide et leur soutien.

Que tous ceux qui nous ont aidés, de près ou de loin, trouvent ici l'expression de nos sentiments les plus distingués.

### RESUME

Ce rapport est consacré à la réalisation d'un système d'assistance et d'automatisation de la visite à l'arrivé, dans le but d'accroître les performances en terme de fiabilité, disponibilité, et d'améliorer la route d'expérience.

L'objectif de ce travail est la mise en œuvre de deux applications, une sur PC et l'autre sur PDA reliées par wifi à une base de données commune. L'automatisation du système nécessite des outils et langages permettant de créer des interfaces selon le besoin des utilisateurs.

Afin de définir les principales interfaces à réaliser, nous avons commencé par une analyse de l'existant et collecte des besoins et nous avons exploité les graphes des historiques pour améliorer notre travail et visualiser les fréquentes défaillances.



L'application réalisée permet au responsable de limiter les interventions et de modifier la base de données des intervenants dans la visite.

Nous avons proposé à la fin une étude du marché pour savoir le prix des équipements nécessaire à notre projet.

Figure 1.1 : Chiffres clés .....	13
Figure 1.2 : Vue générale du projet .....	15
Figure 2.1 : Diagramme de séquence coté PC.....	22
Figure 2.2 : Diagramme de séquence coté PDA.....	22
Figure 2.3 : Cas d'utilisation .....	23
Figure 2.4 :L'association entre la classe ETAT et MACHINE.....	21
Figure 2.5 :L'association entre la classe MACHINE et ENTREESORTIE.....	21
Figure 2.6 :L'association entre la classe MACHINE et VISITE .....	22
Figure 2.7 :L'association entre la classe TECHNICIEN et VISITE .....	22
Figure 2.8 :L'association entre la classe SOUSTÂCHE et VISITE .....	22
Figure 2.9 :L'association entre la classe SOUSTÂCHE et TÂCHE.....	22
Figure 2.10 :L'association entre la classe TECHNICIEN et CONSULTATION.....	23
Figure 2.11 :L'association entre la classe CONSULTATION et MACHINE .....	23
Figure 2.12 :L'association entre la classe TECHNICIEN et PDA.....	23



Figure 2.13 :L'association entre la classe RESPONSABLE et PDA .....	23
Figure 2.14 :L'association entre la classe TECHNICIEN et ROULEMENT .....	24
Figure 2.15 : Diagramme de classes.....	24
Figure 2.16: Le JVM selon le système d'exploitation.....	26
Figure 2.17: Les différents composants de la classe Component.....	29
Figure 2.18: Différentes comportement des évènements. ....	30
Figure 2.19 : Exemple de diagramme qu'on peut créer avec API JFreeChart.....	31
Figure 2.20 :Cycle de vie d'une MIDlet.....	33
Figure 2.21 : Classe Display.....	33
Figure 2.22: Les classes et interfaces du package java.sql .....	35
Figure 2.23 : Relation entre container Web et la requête HTTP .....	38
Figure 3.1 : Différents utilisateur de la base de données.....	41
Figure 3.2 : Utilisateur de la base de données .....	41
Figure 3.3 : Naviguer vers les tableaux de la base de données .....	42
Figure 3.4 : Les différents tableaux de la base de données .....	42
Figure 3.5 : Le tableau MACHINE en détails .....	43
Figure 3.6 : Le tableau SOUSTACHE en détails .....	43
Figure 3.7 : Le tableau TÂCHE en détails .....	43
Figure 3.8 : Le tableau TECHNICIEN en détails.....	43
Figure 3.9 : Le tableau VISITE en détails .....	44
Figure 3.10 : Le Tableau ÉTAT .....	44
Figure 3.11 : Le Tableau RESPONSABLE.....	45
Figure 3.12 : Le Tableau ENTREESORTIE .....	45
Figure 3.13 : Le Tableau PDA .....	45
Figure 3.14 : Le Tableau ROULEMENT .....	46
Figure 3.15 : Le Tableau CONSULTATION.....	46
Figure 3.16 : La barre de menu facilite l'accès aux différentes interfaces.....	47
Figure 3.17 : Les informations sur l'interface.....	47
Figure 3.18 : Identification du personnel .....	48
Figure 3.19 : Message d'erreur de saisie.....	48
Figure 3.20 : Trier selon la série de machine .....	49
Figure 3.21 : Fenêtre de confirmation de suppression. ....	49



Figure 3.22 : Les champs pour modifier mot de passe.....	49
Figure 3.23 : Erreur de saisie modifier mot de passe .....	50
Figure 3.24 : Invalidité du mot de passe.....	50
Figure 3.25 : Erreur de saisi du nouveau mot de passe .....	50
Figure 3.26 : Trier selon tâche pour interface validation .....	50
Figure 3.27 : Affichage de l'interface validation .....	51
Figure 3.28 : Message sur l'interface supervision .....	51
Figure 3.29 : Info. Supervision.....	51
Figure 3.30 : Information supplémentaire sur supervision.....	51
Figure 3.31 : Trier selon loco N°.....	51
Figure 3.32 : L'affichage de l'interface historique. ....	52
Figure 3.33 : Les informations sur une tâche (interface historique).....	52
Figure 3.34 : Interface historique PDA .....	52
Figure 3.35 : Interface liste train .....	52
Figure 3.36 : Interface consultation.....	53
Figure 3.37 : Interface liste roulement.....	53
Figure 3.38 : Interface ajout roulement .....	53
Figure 3.39 : Message d'erreur (manque de date).....	54
Figure 3.40 : Message d'erreur (rien n'est sélectionné). ....	54
Figure 3.41 : Sélectionne de date et machine pour l'interface État.....	54
Figure 3.42 : Tâches pour l'interface état. ....	54
Figure 3.43 : Interface État train.....	55
Figure 3.44 : Trier les interventions selon deux dates pour une machine. ....	55
Figure 3.45 : Label donnant des informations sur la sélection.....	55
Figure 3.46 : Graphe des interventions selon les dates. ....	55
Figure 3.47 : Spécifier date et machine pour le graphe état. ....	56
Figure 3.48 : Graphe des interventions pour une date.....	56
Figure 3.49 : Spécifier la machine et les deux dates .....	57
Figure 3.50 : Diagramme Linéique a) et Surfaique b).....	57
Figure 3.51 : Étape de la création d'une application mobile.....	58
Figure 3.52 : Interface Accueil : Responsable.....	58
Figure 3.53 : Message d'erreur .....	59



Figure 3.54 : Interface Accueil : Technicien .....	59
Figure 3.55 : Interface Welcome .....	60
Figure 3.56 : Liste Machine et ses numéros .....	61
Figure 3.57 : Liste des tâches et sous tâches .....	62
Figure 3.58 : Interface Consulter le carnet de bord .....	63
Figure 3.59 : Interface Information et texte.....	64
Figure 3.60 : État de locomotive .....	65
Figure 3.61 : Interface apparus avec la commande Partie restante.....	66
Figure 3.62 : Interface de Récupération de PDA.....	67
Figure 4.1 : Exemple de PDA .....	69
Figure 4.2 : PDA avec lecteur de données .....	72
Figure 4.3 : Exemple de router.....	74
Figure I.1 : Étape 1 d'ajout du serveur GlassFish3+ .....	78
Figure I.2 : Étape 2 d'ajout du serveur GlassFish3+ .....	79
Figure I.3 : Étape 3 d'ajout du serveur GlassFish3+ .....	79
Figure II.1 : Exécuter les commandes .....	80
Figure II.2 : Les ports libres de la machine .....	80
Figure II.3 : Changement du port .....	80
Figure II.4 : Changement d'URL .....	81
Figure III.1 : Les composants du projet MyApplication.....	82
Figure III.2 : Résultat après exécution du projet .....	84
Figure III.3 : Les tableaux essuie de la base de données .....	85

## Liste des tableaux

Tableau 1.1 : Parc Activité Voyageurs en 2010 .....	13
Tableau 1.2 : Parc Activités Marchandises en 2010.....	14
Tableau 1.3 : Parc Wagons.....	14
Tableau 2.1 : Les méthodes statiques de la Classe JOptionPane. ....	29
Tableau 4.1 : Prix de quelques routeurs selon ses caractéristiques .....	75
Tableau 4.2 : Prix de quelque PDA selon ses caractéristiques.....	76



## **Table des matières**

Remerciement.....	2
Résumé.....	2
Glossaire.....	9
Introduction générale.....	11
Chapitre 1 : Contexte généralités du stage .....	12
1.1    Présentation de l'entreprise .....	12
1.1.1    Historique de l'ONCF .....	12
1.1.2    Chiffres clés.....	12
1.1.3    Matériel de transport .....	13
1.2    Contexte du stage .....	14
1.2.1    L'analyse de l'existant.....	14



1.2.2	Problématique du système existant .....	14
1.3	Cahier de charges .....	14
1.4	Démarche de réalisation du projet.....	15
1.4.1	Coté PDA .....	15
1.4.2	Coté PC .....	15
1.4.3	Connexion avec la base de données .....	16
Chapitre 2: Conception, automatisation et assistance de VA.....		17
2.1	Notion UML.....	18
2.1.1	Diagramme de cas d'utilisation.....	18
2.1.2	Diagramme de séquences .....	18
2.1.3	Diagramme de classes .....	18
2.1.4	Réalisation des diagrammes .....	19
2.2	Base de Données .....	24
2.2.1	Utilisation d'une base de données .....	24
2.2.2	Conception de la base de données .....	25
2.3	Le langage JAVA .....	25
2.3.1	Plateformes JAVA.....	26
2.3.2	Interface graphique .....	28
2.4	Programmation mobile (Interface PDA) .....	31
2.4.1	Les configurations .....	32
2.4.2	Les profils les plus courants .....	32
2.4.3	LangageJ2ME.....	32
2.5	Connexion à la base de données.....	34
2.5.1	Package JDBC.....	34
2.5.2	Langage J2EE.....	37
Chapitre 3 : Réalisation des applications .....		40
3.1	Description de la base de données.....	41
3.2	Interface PC.....	46
3.3	Interface application mobile .....	57
Chapitre 4: Matériels et Étude technico-commercial .....		68
4.1	Les assistants numériques personnels « PDA ».....	68
4.1.1	Évolution des PDAs .....	69





4.1.2	Les « PDAs » industriels .....	70
4.1.3	Critère de choix .....	70
4.1.4	Acquisition des données .....	72
4.2	Routeur WIFI.....	72
4.3	Prix Comparatives des matériaux.....	74
Conclusion générale .....		77
Bibliographies .....		77
Annexes.....		78
Annexe I : Ajout de serveur « Glass Fish 3+ » au Netbeans.....		78
Annexe II : Changement de port du serveur .....		80
Annexe III: Exemple de la création de Servlet .....		81
Annexe IV : Outils de conception .....		85
Annexe V : Checklist.....		87

## GLOSSAIRE

<b>PDA</b>	Assistant numérique personnel : (Personal Digital Assistant) est un appareil numérique portable.
<b>J2ME</b>	(JAVA 2 Micro Edition) : est le Framework JAVA spécialisé dans les applications mobiles.
<b>AWT</b>	(Abstract Window Toolkit) : est une bibliothèque graphique pour JAVA
<b>Swing</b>	Une bibliothèque graphique pour le langage de programmation JAVA
<b>POO</b>	Programmation orientée objet
<b>JVM</b>	JAVA Virtual Machine
<b>BD</b>	Base de données
<b>JDBC</b>	(JAVA DataBase Connectivity) permet aux applications JAVA d'accéder à la BD
<b>SGBD</b>	Système de gestion de base de données
<b>SQL</b>	(Structured Query Language) est un langage informatique normalisé qui sert à effectuer des opérations sur des bases de données.



<b>URL</b>	(Uniform Resource Locator) littéralement « localisateur uniforme de ressource »
<b>ORACLE</b>	Est un système de gestion de base de données relationnel
<b>CLDC</b>	(Connected Limited Device Configuration) : est un sous-ensemble des classes bibliothèques JAVA
<b>CDC</b>	(Connected Device Configuration) est un sous-ensemble des classes bibliothèques JAVA
<b>MIDP</b>	(Mobile Information Device) : Profile est un profil J2ME utilisé par certains téléphones mobiles.
<b>KVM</b>	(Kilo Virtual Machine) : la machine virtuelle JAVA pour les périphériques mobiles
<b>MIDlet</b>	Application exclusivement réservée aux téléphones portables compatible avec J2ME.
<b>Servlet</b>	Application qui s'exécute sur un serveur, il reçoit une requête d'un client, génère une réponse et renvoie celle-ci au client respectif.
<b>http</b>	L'HyperText Transfer Protocol est un protocole de communication client-serveur.
<b>JDK</b>	(JAVA Development Kit) ensemble des bibliothèques logicielles de base du langage de programmation JAVA
<b>SDK</b>	(Software Development Kit) outils permettant aux développeurs de créer des applications de type défini.
<b>J2EE</b>	(JAVA Enterprise Edition) est une spécification pour la technique JAVA destinée aux applications d'entreprise
<b>JRE</b>	(JAVA Runtime Environmen) ensemble d'outils permettant l'exécution de programmes JAVA sur toutes les plates-formes supportées
<b>JSP</b>	(JAVA Server Pages) permet de créer dynamiquement du code HTML, XML ou tout autre type de page web
<b>MIME</b>	(Multipurpose Internet Mail Extensions) est un standard internet qui étend le format de données des courriels
<b>IDE</b>	(Integrated development environment) est un programme regroupant un ensemble d'outils pour le développement de logiciels
<b>TCP</b>	(Transmission Control Protocol) est un protocole de transport fiable, en mode connecté.
<b>JSR</b>	(JAVA Specification Requests) est un système normalisé ayant pour but de faire évoluer la plateforme JAVA.
<b>EDI</b>	Un environnement de développement intégré est un programme regroupant un ensemble d'outils pour le développement de logiciels.



## INTRODUCTION GENERALE

Ces dernières années, le marché des télécommunications sans fils connaît une énorme croissance. En parallèle les téléphones portables se développent à grand pas et envahissent tous les domaines. En effet, il y a de plus en plus des PDA ou smart phone dans le domaine industriel, qui permet d'accéder à l'information plus facilement et rapidement et aussi d'informatiser les systèmes. Dans le but de garantir une telle exigence les grands fabricants d'équipement de communication se sont mis dès la fin de l'année 1999.

Pour avoir une interopérabilité entre les différents périphériques compatibles JAVA, plusieurs fabricants et constructeurs se sont mis d'accord sur des standards de développement, afin d'offrir une plateforme JAVA compatible avec une large gamme des petits appareils connectés aux ressources systèmes très limités.

Toutes ces standardisations ont permis l'intégration de plusieurs terminaux dans le marché, et ont rendu la création des applications plus facile et la programmation des terminaux plus accessible au développeur.

Afin d'améliorer la qualité de service visite à l'arrivé ; l'ONCF a proposé l'intégration des PDA dans ce service dans le but d'automatiser et d'assister la visite à l'arrivée à l'aide des applications permettant l'interaction de manière sûre entre PDA, PC et la base de données distante.

Au terme de ce stage nous avons élaboré ce rapport qui est scindé en quatre chapitres.

- Le premier chapitre comporte une description du cadre général du projet et son cahier de charges.
- Le deuxième chapitre développe les outils utilisés au cours de ce projet.
- Le troisième chapitre porte une explication des deux applications réalisées pour l'automatisation et l'assistance du service VA (visite à l'arrivé), une application coté PC permettant la saisie et la visualisation des différents intervenants dans le service et l'autre application sur mobile permettant la consultation des tâches à faire et l'envoi des résultats par la suite.
- Le quatrième chapitre est consacré à l'étude de marché afin d'avoir une estimation sur le coût du projet.

Et nous achevons notre rapport par une conclusion générale.



# Chapitre 1 : Contexte généralités du stage

## 1.1 Présentation de l'entreprise

### 1.1.1 Historique de l'ONCF

La construction du réseau des chemins de fer au Maroc remonte au début du 20<sup>ème</sup> siècle. En effet, les premières lignes construites à voie de 0,6 m ont été établies à partir de 1916. Ce n'est qu'en 1923 que la construction des voies à écartement normal a été confiée à trois Compagnies concessionnaires privées. Ces dernières se partagent le trafic ferroviaire, en exploitant chacune la partie du réseau qui lui était concédée, jusqu'en 1963, lorsque le Gouvernement Marocain a décidé le rachat des concessions et la création de l'Office National des Chemins de Fer (ONCF).

L'ONCF, qui emploie actuellement environ 10.400 agents, gère et exploite un réseau de 1.907Km de lignes, dont 1.537Km à voie unique (80%) et 370Km à double voie (20%). Ce réseau comporte également 528Km de voie de service et 201Km de ligne d'embranchements particuliers reliant diverses entreprises au réseau ferré national. À noter que 53% de la longueur totale du réseau, soit 1003Km, est électrifié à 3000 Volts continu, alors que 904Km sont exploités en traction Diesel.

Le réseau ferroviaire marocain, qui permet des vitesses de 160Km/h sur certains tronçons, se présente sous forme d'un couloir reliant le Sud (Marrakech) à l'Est (Oujda) avec des antennes vers Tanger, Safi, Oued Zem, El Jadida et Bou Aârf. Ainsi, le train dessert les grandes villes et les principaux ports du Royaume à l'exception de ceux d'Agadir au Sud et de Nador au Nord. Il est également relié aux réseaux algérien et tunisien, avec des caractéristiques techniques similaires permettant d'assurer la circulation des trains dans de bonnes conditions d'exploitation.

Par ailleurs, il convient de souligner que le transport ferroviaire joue un rôle primordial dans le processus de développement allant de pair avec l'évolution croissante que connaît le tissu économique et la renaissance que notre pays ne cesse d'enregistrer durant les dernières décennies.

Il s'agit en fait d'un facteur essentiel de rapprochement des distances qui assure le trafic des voyageurs et des marchandises entre les grandes villes dans de meilleures conditions, mais aussi d'un moyen de transport stratégique en matière de redynamisation des activités industrielles en reliant leurs sites aux principaux ports du phosphate .

### 1.1.2 Chiffres clés

Avec un chiffre d'affaires dépassant le cap de 3 milliards de dirhams, toutes activités confondues, l'ONCF est en progression de 14,4% par rapport à 2009, dégagant ainsi une capacité d'autofinancement de 1,1 milliard de dirhams.

L'activité Voyageurs à enregistrée 31 millions de clients transportés, l'ONCF se positionne de plus en plus sur les grands axes de trafic et affine ses logiques marketing pour offrir des produits en adéquation avec les attentes et les besoins des voyageurs.



Le trafic Fret est passé globalement de 25 millions de tonnes en 2009 à 36 millions, soit une progression de 42% grâce au Port de Tanger-Med, la desserte de Nador et les terminaux conteneurs.<sup>[1]</sup>

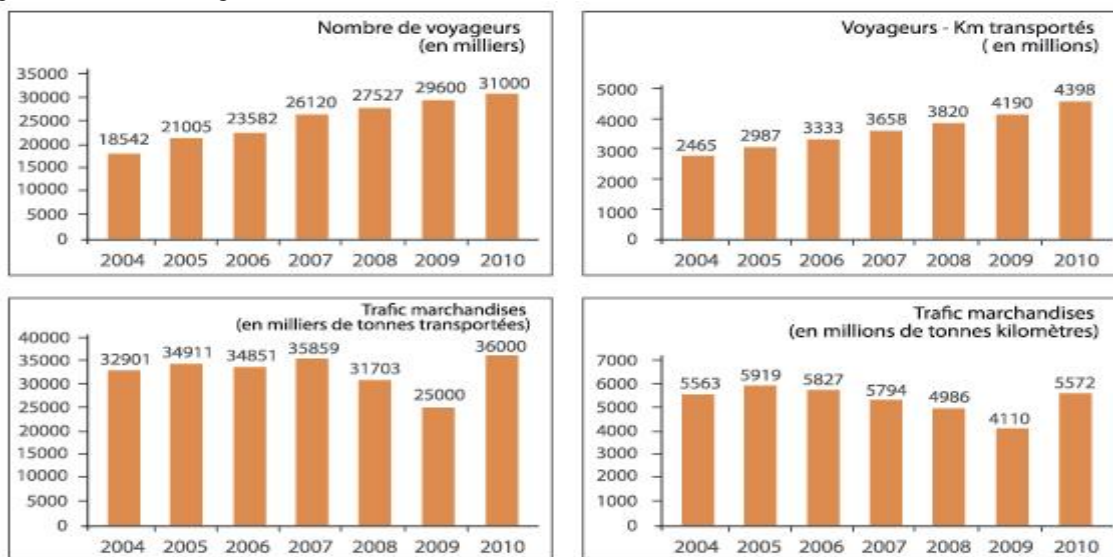


Figure 1.1 : Chiffres clés

### 1.1.3 Matériel de transport

Rames automotrices simples ou duplex<sup>[2]</sup>, locomotives électriques ou diesel, voitures places assises 1ère ou 2ème classe, voitures lits ou couchettes ..., le matériel dédié aux voyageurs évolue au rythme des clients, en capacité de transport et en confort de voyage.

Pour le fret, à chaque marchandise, son wagon. Plats, citernes, couverts ou trémies, chaque wagon est spécialisé pour transporter un produit donné : phosphates, charbon, céréales, engrais, barytine, zinc, produits pétroliers, pâtes, ciments, clinker, bois, gravettes, sucre, conserves, conteneurs, acier, fer.

Parc Activité Voyageurs en 2010

Matériel de Traction	
Automotrices	38
Locomotives Électriques	42
Locomotives Diesel	19
Locomotives de manœuvre	11
<b>Total</b>	<b>110</b>

Tableau 1.1 : Parc Activité Voyageurs en 2010

Ce qui concerne le parc des voitures il contient **40450** voiture.

Parc Activités Marchandises en 2010

<sup>1</sup><http://www.oncf.ma/>

<sup>2</sup>Voitures à deux niveaux.



Matériel de Traction	
Locomotives Électriques	47
Locomotives Diesel	19
Locomotives de manœuvre	52
<b>Total</b>	<b>118</b>

*Tableau 1.2 : Parc Activités Marchandises en 2010*

Wagons	
Wagons phosphates	1701
Autre types de Wagons	3582
<b>Total toutes séries</b>	<b>5283</b>

*Tableau 1.3 : Parc Wagons*

Les stratégies visent à satisfaire les 5 parties prenantes en positionnant l'ONCF comme :

- Le transport de référence au Maroc au service de ses clients.
- Une entreprise performante et en croissance continue.
- Un modèle en matière de gestion ressources humaines.
- Une entreprise au service de la collectivité.
- Un partenaire de ses fournisseurs.

## 1.2 Contexte du stage

### 1.2.1 L'analyse de l'existant

Vue la charge et la cadence d'utilisation des locomotives, il est nécessaire d'effectuer des visites des locomotives avant leurs départs et après leurs arrivées au terminus. Cette visite est appelée visite à l'arrivée, effectuée par les techniciens de maintenance, en électrique et mécanique, dont les tâches sont limitées dans la vérification des consommables (bande d'usure, filtre huile..) et état des différents organes.

La VA se fait en réalisant les tâches listées par des check-lists (**Voir Annexe V**), l'agent doit les remplir en même temps qu'il effectue les tâches demandées. Cette dernière est un rapport d'état de la machine et des interventions effectuées.

Le travail se fait en plusieurs étapes :

- La consultation du carnet de bord (cahier où le conducteur de train renseigne les dysfonctionnements constatés au cours de la route), s'il y'a des anomalies, l'agent spécialisé doit résoudre le problème.
- La vérification de l'état générale de la locomotive.
- La vérification de l'état des organes de sécurité et l'état des usures.
- La check-list est remplie manuellement en cochant des mentions : bon (B), intervention (I) ou à suivre (S), puis l'agent intervenant mentionne si nécessaire ses observations.

Les tâches peuvent être limitées selon le temps d'immobilisation permis de la machine au terminus, ces tâches varient d'une série à l'autre.

### 1.2.2 Problématique du système existant

En examinant l'analyse de l'existant, nous avons constaté que le système souffre des problématiques suivantes:

- Difficulté d'avoir un historique du service VA vu que le système n'est pas informatisé.
- Absence des statistiques donnant idée sur le temps moyen des tâches, les défaillances fréquentes ....
- Méthode de supervision ordinaire.
- La nécessité de la supervision en temps réel des interventions et le retour d'expérience.

## 1.3 Cahier de charges



Le choix de réaliser deux applications Java (sur PC et PDA) répond aux exigences de l'entreprise et permet de pallier au problème déjà cité.

Le cahier de charge est défini comme suit :

- Étudier le système existant pour collecter les besoins des intervenants dans la VA.
- Réaliser une application PC répondant aux exigences du responsable.
- Réaliser une application PDA permettant à l'agent de faire la VA.
- Relier les applications à une base de données via wifi.

## 1.4 Démarche de réalisation du projet

La solution proposée est de réaliser deux applications une sur PC et l'autre sur PDA reliées à une base de données via Wifi, la figure ci-dessous montre une vue générale sur le projet.

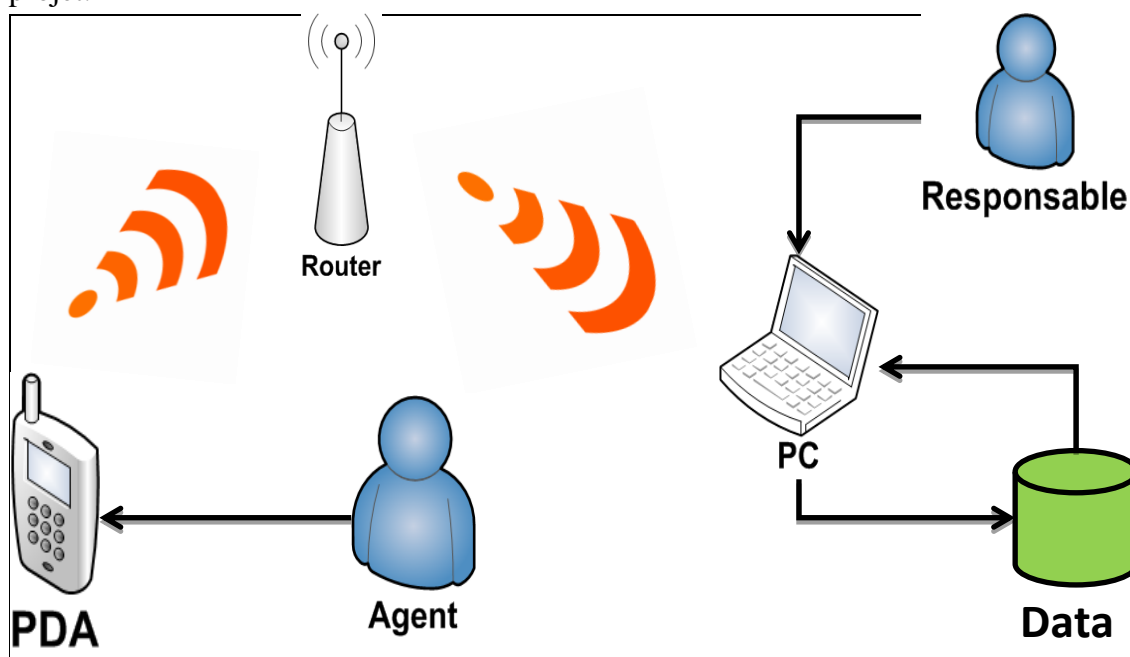


Figure 1.2 : Vue générale du projet

### 1.4.1 Coté PDA

- Faire une application JAVA permettant aux agents l'insertion des données nécessaires.
- Concevoir une interface graphique convenable au besoin. (programmée en J2ME).
- L'interface enregistre pour chaque tâche les données et l'envoi à un serveur commun entre l'interface PC et l'interface PDA.

### 1.4.2 Coté PC

- Concevoir une interface graphique convenable au besoin.
- L'interface PC a le statut administrateur, cette interface est en JAVA AWT/Swing.



- À partir du PC l'administrateur (responsable) peut préciser les tâches que l'agent doit faire.
- L'interface PC permet l'accès à l'historique des tâches réalisées, et affiche l'avancement du travail en temps réel.
- À partir de l'interface PC on peut lister, ajouter, enlever les intervenants dans la VA selon les besoins.
- L'interface PC affiche les statistiques, les graphes ....

### 1.4.3 Connexion avec la base de données

- Les deux interfaces se communiquent via un serveur.
- Les deux interfaces échangent les données et les enregistrent dans une base de données principale après chaque validation.

### Conclusion :

La réalisation de la VA sur papier rend les données inexploitable et ne permet pas d'avoir des statistiques et retour d'expérience ; de plus il y'a un manque de système de supervision. Pour ces raisons il a fallu réaliser l'assistance et l'automatisation de la VA qui consiste à créer deux applications, une sur PC et l'autre sur PDA, la première permet la visualisation des statistiques, historiques, graphes et aussi la modification de la base de données des intervenants dans la visite ; la deuxième application sur mobile permet de consulter les tâches à faire sur une machine selon la série et la partie ciblée (électrique ou mécanique).





## Chapitre 2: Conception, automatisation et assistance de VA



Après une vue générale sur le contexte du stage et le cahier de charges, on va détailler les outils et langages de programmation utilisés pour la réalisation du projet.

Nous allons entamer par le langage UML qui permet la modélisation des applications construites autour de la notion d'objets dans le but de voir clairement les interactions entre les utilisateurs et le système en fonction du temps, ensuite nous présentons des notions sur la base de données qui est indispensable dans nos applications ainsi qu'une vue générale sur les langages utilisés.

## 2.1 Notion UML

UML (Unified Modeling Language) est une méthode de modélisation orientée objet dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets.

Cependant, UML n'est pas une méthode car il n'inclut pas la manière d'utiliser les concepts qu'il propose c'est-à-dire l'enchaînement des étapes qui mènent à la résolution des problèmes posés.

Un modèle est une représentation exacte, partielle, passée ou future, de notre réalité. Il s'agit d'un processus qui consiste à identifier les caractéristiques intéressantes d'une entité, en vue d'une utilisation précise. Chaque entité modélisable (acteur, message informationnel, événement) présente trois dimensions :

- Dimension fonctionnelle (diagramme de cas d'utilisation).
- Dimension statique (diagramme de classes).
- Dimension dynamique (diagramme de séquence).

### 2.1.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel, il représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail, dont les utilisateurs sont appelés acteurs (actors) qui interagissent avec les cas d'utilisations (use cases) pour atteindre un objectif en utilisant le système.

### 2.1.2 Diagramme de séquences

Le diagramme de séquences permet de représenter des collaborations entre objets selon un point de vue temporel, on y met l'accent sur la chronologie des envois de messages.

- Ils peuvent servir à illustrer un cas d'utilisation.
- L'ordre d'envoi d'un message est déterminé par sa position sur l'axe vertical du diagramme; le temps s'écoule "de haut en bas" de cet axe.
- Les diagrammes de séquences et les diagrammes d'état-transitions sont les vues dynamiques les plus importantes d'UML.

### 2.1.3 Diagramme de classes

Le diagramme de classes montre la structure interne, il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisations. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisations.

Il s'agit d'une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application.

Une classe est représentée par un rectangle séparé en trois parties contenant :



- le nom de la classe dans la première partie ;
- les attributs de la classe dans la seconde ;
- les méthodes de la classe dans la dernière partie.

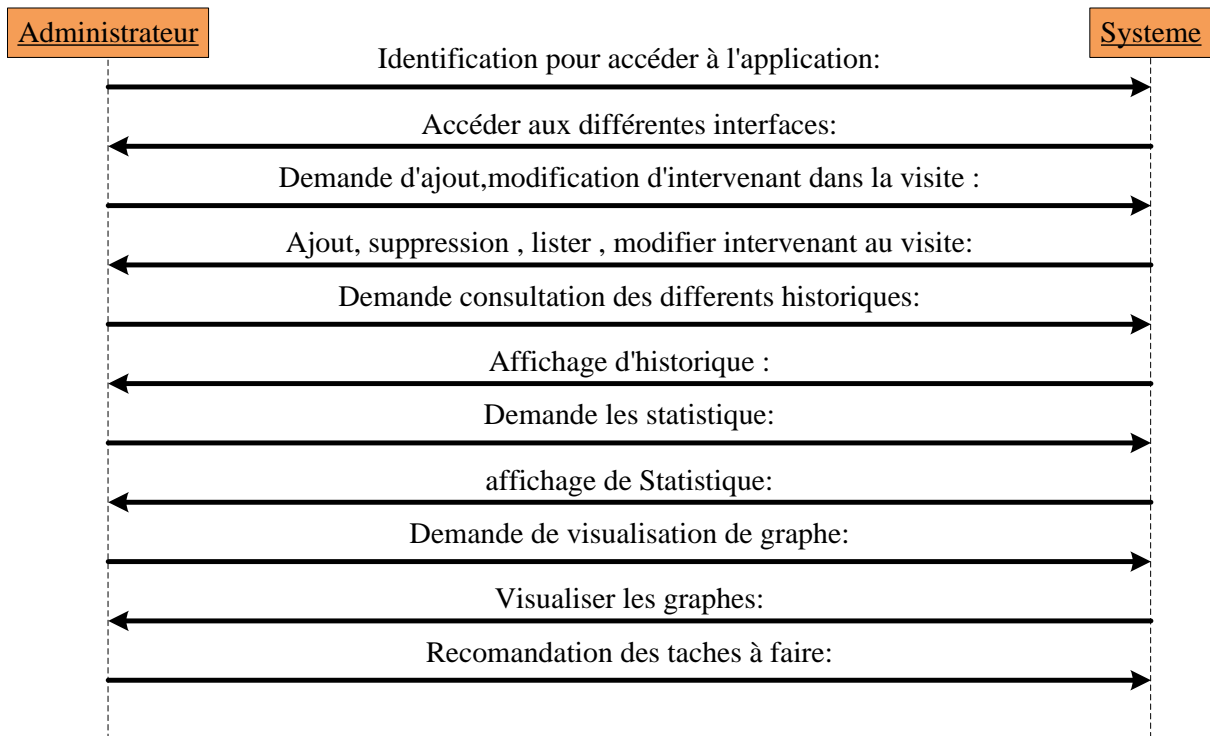
#### 2.1.4 Réalisation des diagrammes

La réalisation des diagrammes se fait selon le scénario suivant :

- Coté technicien :
  - Identification de l'agent après l'identification du responsable.
  - L'agent reçoit les tâches à faire.
  - Saisi et envoi des données des tâches effectuées.
  - Saisi et envoi des données des tâches supplémentaires.
  - Saisi et envoi de l'état de machine.
- Coté administrateur :
  - Identification du responsable.
  - Ajout, suppression des intervenants dans la visite.
  - Lister les différentes informations des intervenants.
  - Modification du mot de passe du responsable.
  - Recommandation des tâches à faire pour la prochaine visite.
  - Supervision.
  - Consulter les historiques.
  - Visualiser les graphes.
  - Voir les statistiques.

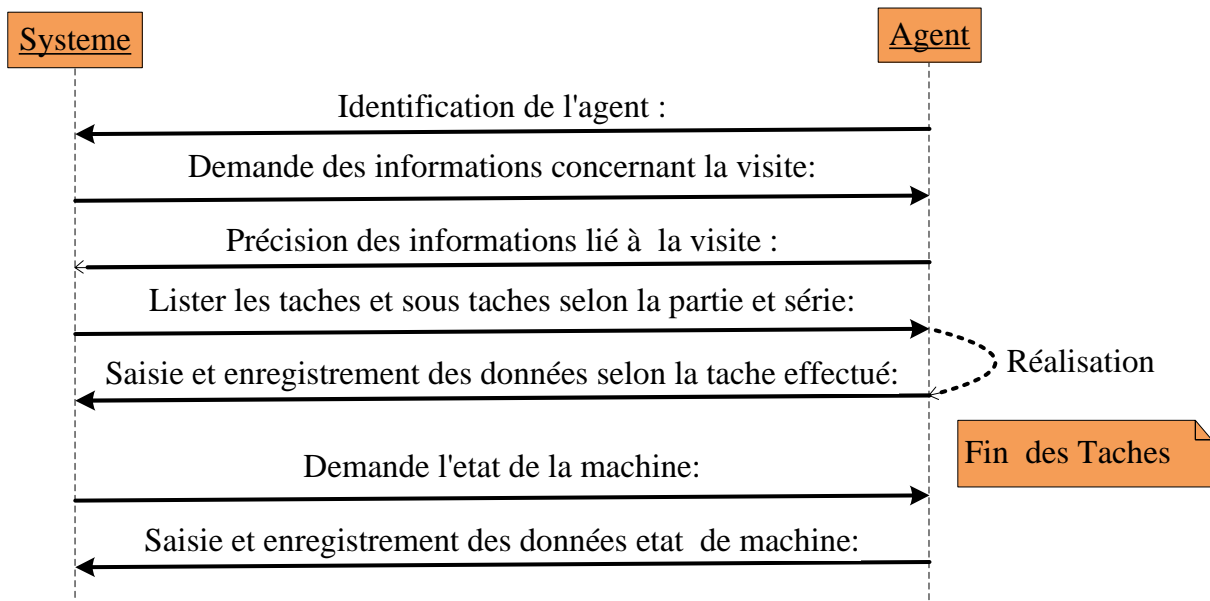
Les figures ci-dessous illustrent le Diagramme de séquence.

- Coté PC



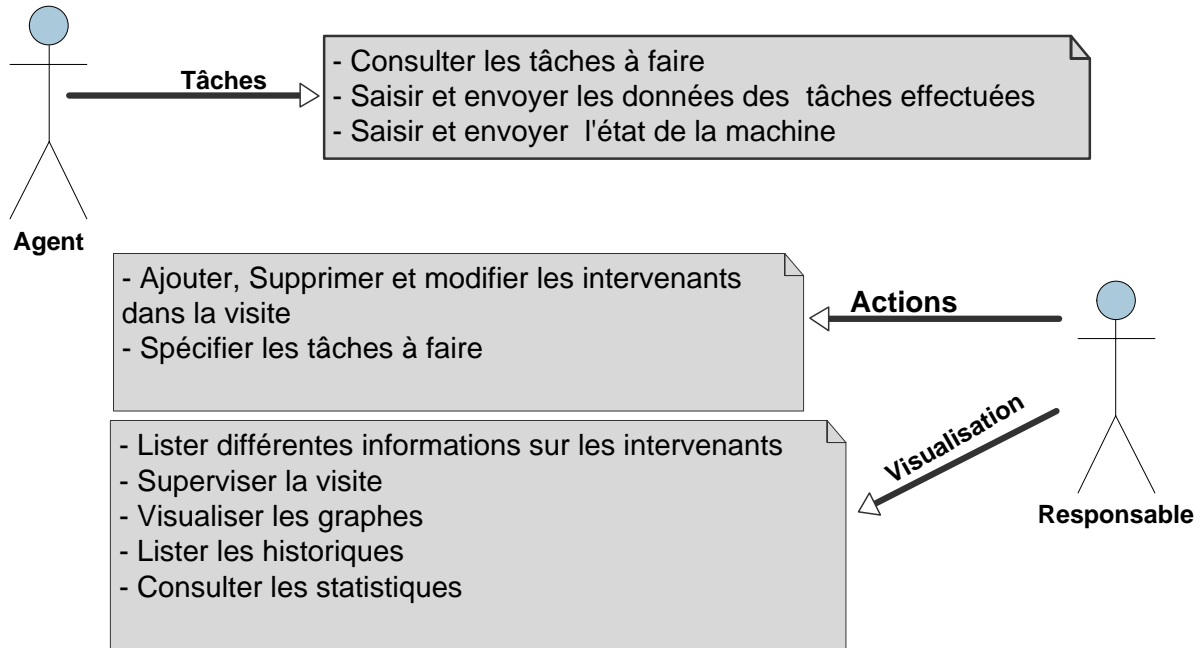
**Figure 2.1: Diagramme de séquence coté PC**

- Coté PDA



**Figure 2.2: Diagramme de séquence coté PDA**

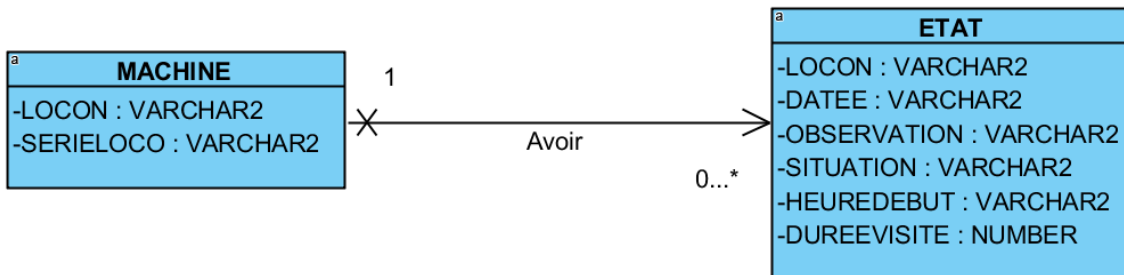
Le diagramme de cas d'utilisation ci-dessous montre les interactions entre l'agent et le responsable:



**Figure 2.3: Cas d'utilisation**

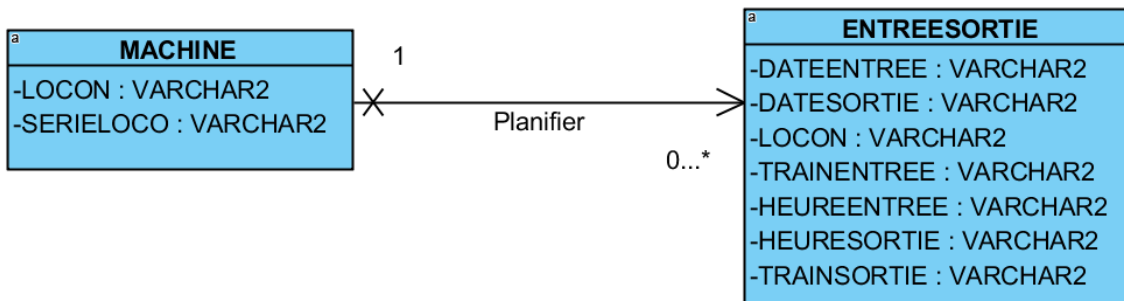
Pour exprimer la connexion sémantique bidirectionnelle entre les classes nous utilisons les associations entre différents tableaux.

La classe MACHINE, qui contient deux éléments caractérisant la machine de façon unique, est associée à la classe état puisque à la fin de chaque visite l'agent doit mentionner l'état de la machine avant son prochain départ, les informations liées à cette opération sont collectées dans la classe ÉTAT.



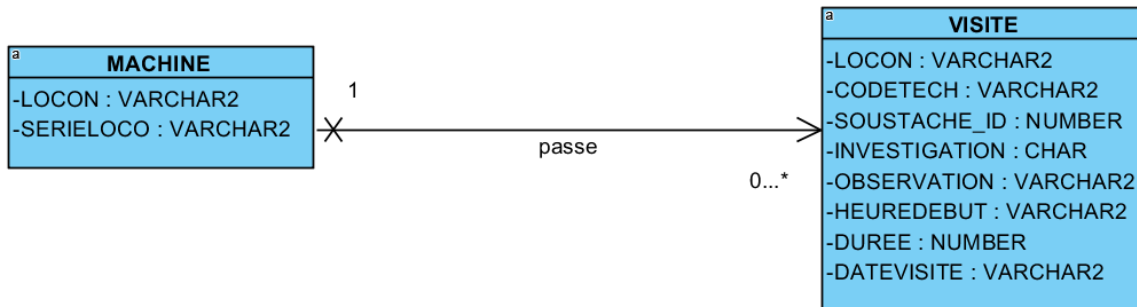
**Figure 2.4 :L'association entre la classe ETAT et MACHINE**

Pour chaque machine arrivante au pôle de maintenance, on enregistre ses données d'arrivée (DATEENTREE, HEUREENTREE, TRAINENTREE) et de son départ du service VA (DATESORTIE, HEURESORTIE, TRAINSORTIE) après quelques heures, et pour assurer le prochain voyage, il faut terminer la visite avant l'heure de sortie prévue. Les informations liées à la locomotive (LOCON, SERIELOCO) entre les deux voyages sont collectées dans la classe ENTREESORTIE



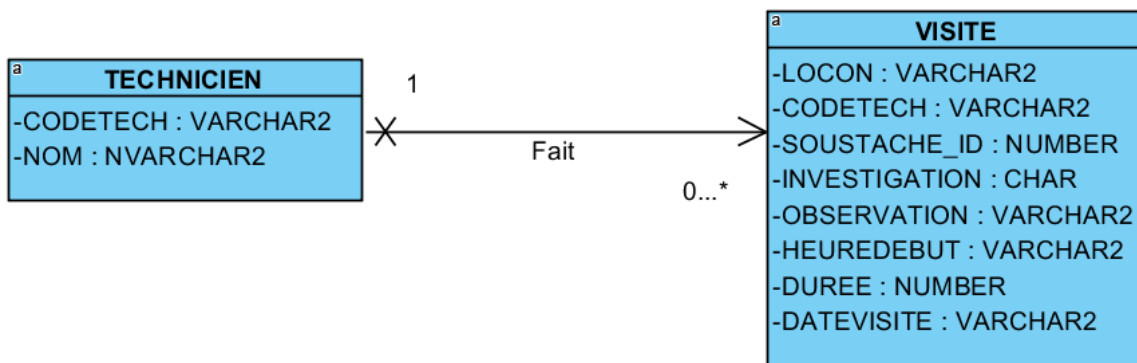
**Figure 2.5 :L'association entre la classe MACHINE et ENTREESORTIE**

Les détails de la visite du MACHINE sont enregistrés après chaque visite dans la classe VISITE



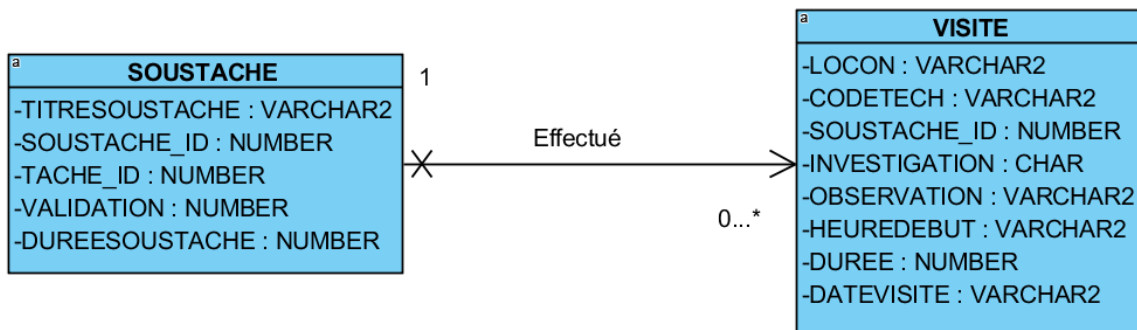
**Figure 2.6 :L'association entre la classe MACHINE et VISITE**

Chaque visite est faite par un ou plusieurs agents selon le type de visite, les informations liées à l'agent sont enregistrées dans la classe TECHNICIEN.



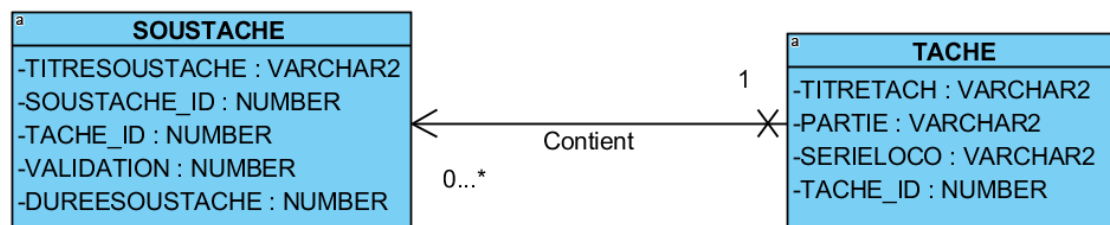
**Figure 2.7 :L'association entre la classe TECHNICIEN et VISITE**

Durant la visite l'agent doit faire plusieurs sous-tâches selon les besoins et le type de la visite et la série de la locomotive, les composants de la classe SOUSTACHE sont liés directement à la classe VISITE.



**Figure 2.8 :L'association entre la classe SOUSTACHE et VISITE**

Chaque tâche contient plusieurs sous tâche selon la série de la locomotive et aussi la partie.



**Figure 2.9 :L'association entre la classe SOUSTACHE et TACHE**

L'agent peut faire des tâches supplémentaires selon les remarques mentionnées dans le carnet de bord.

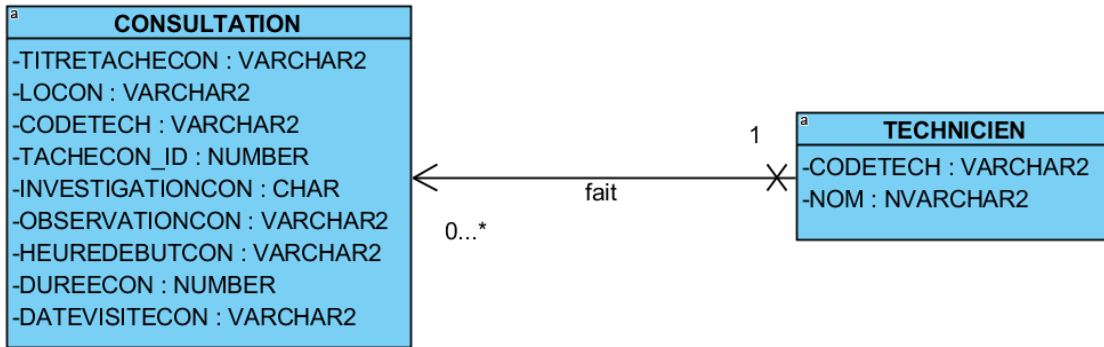


Figure 2.10 :L'association entre la classe TECHNICIEN et CONSULTATION

Chaque machine peut avoir des tâches supplémentaires selon son carnet de bord.

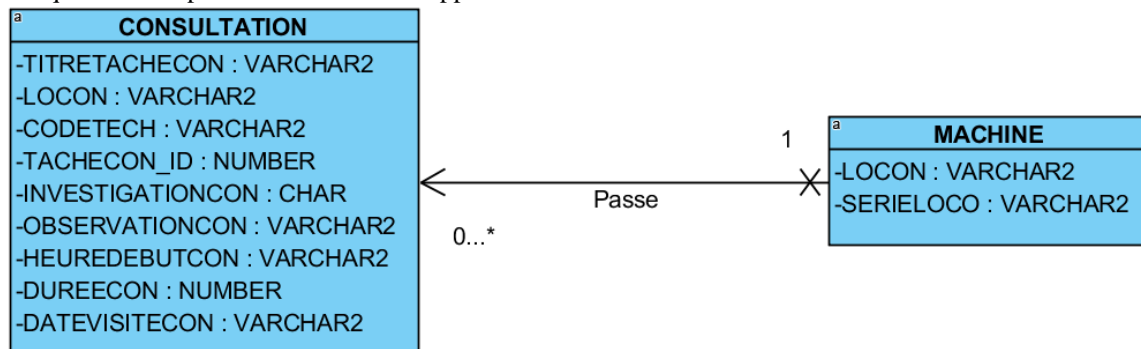


Figure 2.11 :L'association entre la classe CONSULTATION et MACHINE

La visite ne peut pas se faire sans l'utilisation de la PDA.

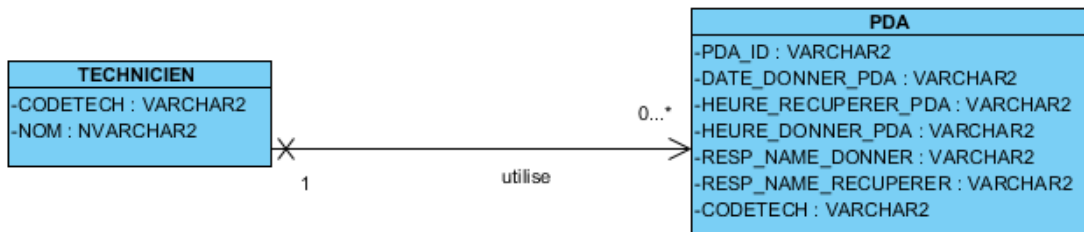


Figure 2.12 :L'association entre la classe TECHNICIEN et PDA

Le responsable du service VA peut donner ou récupérer la PDA et il assume de cette façon la responsabilité en cas de défaillance de l'appareil.

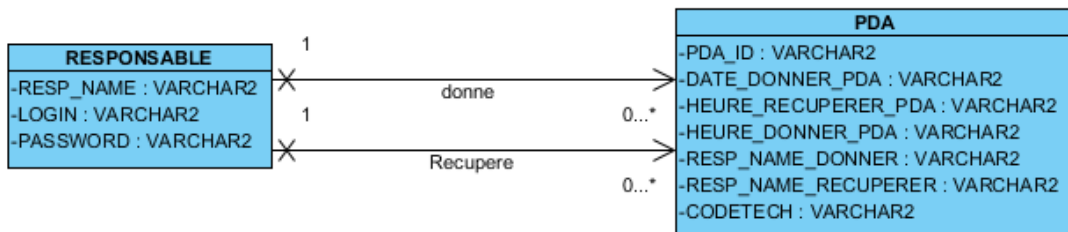


Figure 2.13 :L'association entre la classe RESPONSABLE et PDA



Chaque agent a un roulement que le responsable le prépare mensuellement.



Figure 2.14 :L'association entre la classe TECHNICIEN et ROULEMENT

### • Diagramme de classes

Le diagramme des classes rassemble toutes les classes avec ses associations.

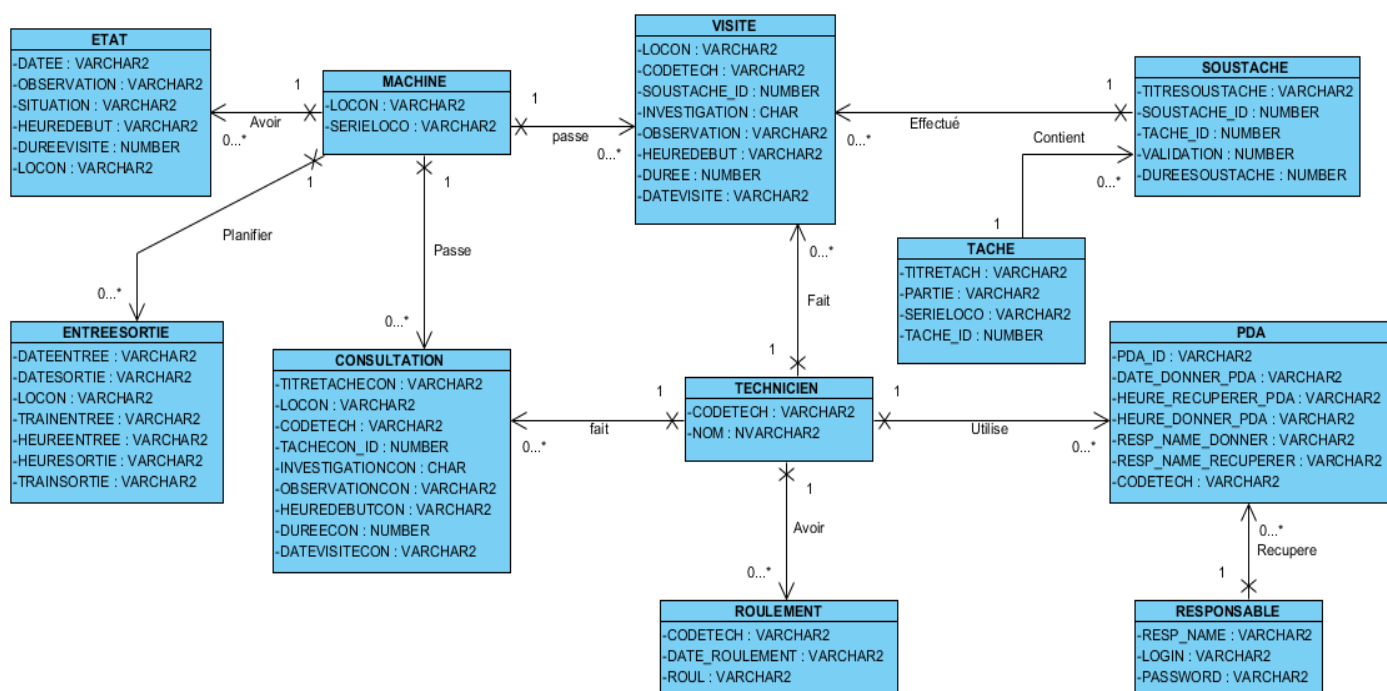


Figure 2.15 : Diagramme de classes

## 2.2 Base de Données

Dans le domaine purement informatique, pour gérer les données les développeurs d'applications s'appuient sur des bases de données externes qui sont des entités dans lesquelles il est possible de stocker des données de façon structurée et avec le moins de redondance possible. Pour manipuler les données, on utilise généralement un logiciel spécialisé appelé SGBD (Système de Gestion de Bases de Données).

### 2.2.1 Utilisation d'une base de données

La création d'une base de données recèle un but précis : elle doit permettre de retrouver de l'information par son contenu en se fondant sur des critères de recherche. On désire, par exemple, retrouver toutes les informations liées à une machine quelconque.

La grande différence avec un programme écrit dans un langage de programmation est qu'une base de données doit pouvoir répondre à des questions pour lesquelles elle n'a pas forcément été prévue à la conception.

Une autre différence est que les données sont susceptibles d'être utilisées par des applications différentes. Dans un programme classique, la structuration des données est décrite directement dans le code, ce qui rend leur utilisation difficile par d'autres programmes, en particulier lorsque l'on modifie cette structure.





Ce que l'on recherche en utilisant une base de données est d'assurer l'indépendance entre le traitement et les données. C'est pourquoi, il est nécessaire que l'application obtienne des informations sur la structure des données (nom, type, taille, etc.).

L'idée générale est que l'utilisateur ou l'application utilisatrice des données ne doit pas être dépendante de leur représentation interne, ce qui constitue une abstraction des données. C'est la raison pour laquelle on utilise une description des données sous la forme d'un modèle pour permettre la restitution la plus efficace possible de l'information.

Comme on l'a évoqué précédemment, l'un des objectifs de création d'une base de données est de pouvoir retrouver les données par leur contenu. Dans cette optique, il faut s'assurer que les données contenues dans la base sont de « bonne qualité »<sup>[3]</sup>.

De nombreux critères peuvent être pris en compte ; on peut citer parmi les principaux : la cohérence des données contenues dans la base et l'absence de redondance.

## 2.2.2 Conception de la base de données

On peut décomposer le processus de conception d'une base de données en plusieurs étapes :

- l'analyse du système du monde réel à modéliser ;
- la mise en forme du modèle pour l'intégrer dans un SGBD ;
- la création effective dans le SGBD des structures et leur remplissage.

### • Langage SQL

Le langage SQL (Structured Query Language) peut être considéré comme le langage d'accès normalisé aux bases de données. Il est aujourd'hui supporté par la plupart des produits commerciaux que ce soit par les systèmes de gestion de bases de données micro tel que « Access » ou par les produits plus professionnels tels que « ORACLE ». Il a fait l'objet de plusieurs normes ANSI/ISO dont la plus répandue aujourd'hui est la norme SQL2 qui a été définie en 1992.

Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma conceptuel<sup>[4]</sup> pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution. Le langage SQL propose un langage de requêtes ensembliste. Néanmoins, le langage SQL ne possède pas la puissance d'un langage de programmation : entrées/sorties, instructions conditionnelles, boucles et affectations. Pour certains traitements il est donc nécessaire de coupler le langage SQL avec un langage de programmation plus complet (JAVA par exemple).

### • Catégories d'instructions

Les instructions SQL sont regroupées en catégories en fonction de leur utilité et des entités manipulées. Nous pouvons distinguer cinq catégories, qui permettent :

- La définition des éléments d'une base de données (tables, colonnes, clefs, index, contraintes . . .).
- la manipulation des données (insertion, suppression, modification, extraction . . .)
- la gestion des droits d'accès aux données (acquisition et révocation des droits).
- la gestion des transactions.
- le SQL intégré.

## 2.3 Le langage JAVA

JAVA est un langage de programmation récent (les premières versions datent de 1995) développé par Sun Microsystems. Il est fortement inspiré des langages C et C++.

<sup>3</sup> Création de bases de données, N.Larrousse, E.Innocenti, Page 1, Edition 2006

<sup>4</sup> Un schéma conceptuel est une représentation d'un ensemble de concepts reliés sémantiquement entre eux.



Comme C++, JAVA fait partie de la « grande famille » des langages orientés objets. Il répond donc aux trois principes fondamentaux de l'approche orientée objet (POO) : L'encapsulation, le polymorphisme et l'héritage.

### 2.3.1 Plateformes JAVA

Sun définit trois plateformes d'exécution (ou éditions) pour JAVA pour des cibles distinctes selon les besoins des applications à développer :

- JAVA Standard Edition (J2SE / JAVA SE) : environnement d'exécution et ensemble complet d'API pour des applications de type desktop. Cette plate-forme sert de base en tout ou en partie aux autres plateformes
- JAVA Enterprise Edition (J2EE / JAVA EE) : environnement d'exécution reposant intégralement sur JAVA SE pour le développement d'applications d'entreprises
- JAVA Micro Edition (J2ME / JAVA ME): environnement d'exécution et API pour le développement d'applications sur appareils mobiles et embarqués dont les capacités ne permettent pas la mise en œuvre de JAVA SE.

D'autre côté pour que le byte code soit exécutable sur n'importe quel appareil, on a recourt à une machine virtuelle JAVA.

#### • JAVA virtuelle machine

La machine virtuelle JAVA (anglais JAVA Virtual Machine abr. JVM) est un appareil informatique fictif qui exécute des programmes compilés sous forme de byte code JAVA.

L'appareil est simulé par un logiciel spécifique à chaque plate-forme ou couple (machine/système d'exploitation) et permet aux applications JAVA compilées en byte code de produire les mêmes résultats quelle que soit la plate-forme, tant que celle-ci est pourvue de la machine virtuelle JAVA adéquate.

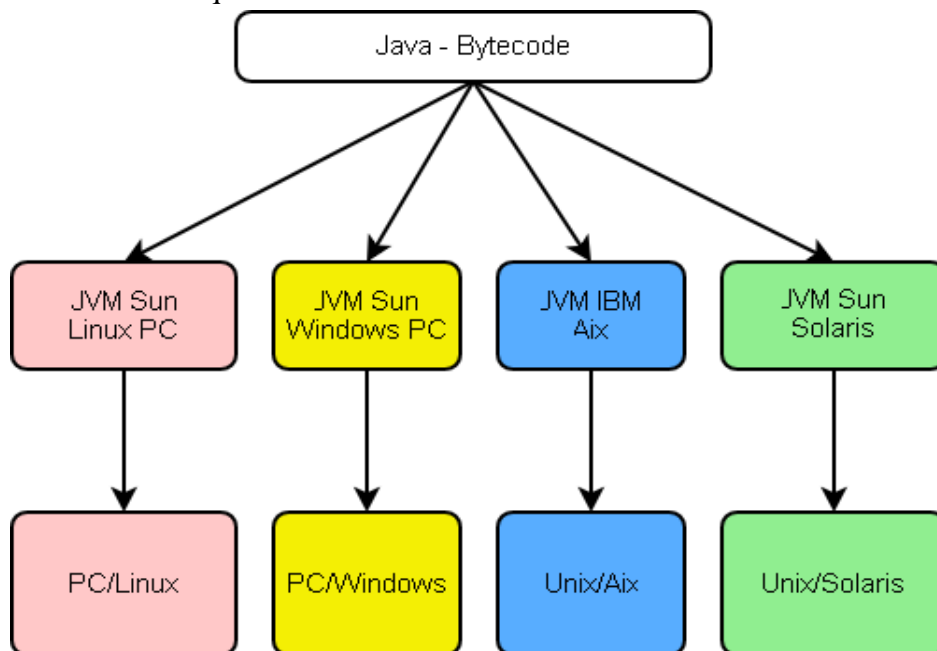


Figure 2.16: Le JVM selon le système d'exploitation



Après la compilation de byte code JAVA plusieurs fichier JAVA se créé parmi ses fichiers : src, bin et classes.

### • Les fichiers du répertoire JAVA

Le projet est composé de plusieurs répertoires comme:

- **src** : rassemble les fichiers .JAVA contenant le code source de l'application
- **bin** : contient le cœur de l'application (fichier jad, fichier jar et Manifest.MF)
- **classes** : regroupe les fichiers .class résultant de la compilation des fichiers .JAVA
- **lib** : inclus les librairies nécessaires au projet créé (permet de les importer sur n'importe quelle machine)
- **res** : contient les fichiers nécessaires à l'internationalisation.

### • Outil et vocabulaires de programmation.

#### Classes et objets

Le langage JAVA est "orienté objet", il est entièrement basé sur la notion d'encapsulation, c'est-à-dire qu'il permet de définir des classes qui contiennent des membres. Les membres sont soit des champs soit des méthodes. L'instance de la classe (appelé objet) est alors implicitement passée en paramètre à la méthode. Les champs (parfois appelé attribut) sont des variables qui définissent des caractéristiques propres à chaque objet (variables d'instance) ou à un groupe d'objets de la classe (variables de classe). Ces variables ont une portée différente suivant l'endroit où elles sont déclarées.

Parmi les méthodes on trouve des méthodes particuliers appelés les constructeurs.

#### Les constructeurs

Le constructeur est une méthode spéciale d'une classe, permettant d'initialiser les membres de cette classe et également, d'exécuter différentes opérations initiales définies par le programmeur.

L'instanciation d'une classe (ou la création d'un objet) nécessite l'utilisation de l'opérateur new appelant un constructeur, et pour encapsuler tous ses classe ou méthode on a recourt à la notion package.

#### Package

Les packages permettent d'organiser des groupes de classes. Un package contient un nombre quelconque de classes reliées entre elles par leur objectif, leur portée ou leur héritage. Les packages deviennent intéressants à utiliser dès que l'application contient beaucoup de classes. Il y a plusieurs raisons d'utiliser les packages:

- Ils permettent d'organiser les classes en unité (de la même façon qu'on a des dossiers ou des répertoires).
- Ils réduisent les conflits de noms.



- Ils permettent de protéger des classes, des méthodes ou des variables.
- Ils peuvent servir à identifier les classes. Il est possible de nommer un package avec un identificateur unique propre à une organisation donnée.

### **Modificateurs**

Les modificateurs sont des mots-clés qui permettent de définir l'accessibilité de méthodes, variables ou classes dans le programme dans lequel ils sont utilisés ou dans d'autres programmes ayant une interaction avec le programme dans lequel ils sont utilisés.

Il existe deux catégories de modificateurs :

- les modificateurs d'accès : déterminent la visibilité des classes, méthodes ou variables par rapport aux autres membres et aux autres classes, ainsi qu'entre les paquetages (public, private et protected).
- les autres modificateurs : indiquant certaines particularités pour des éléments.

### **Méthodes**

Les méthodes JAVA, équivalentes aux fonctions ou aux procédures présentes dans d'autres langages, sont utilisées dans les classes et permettent de définir des opérations à accomplir sur un objet ou dans une classe.

Une méthode est un sous-programme destiné à exécuter un bloc d'instructions suite à son appel dans un programme principal.

## **2.3.2 Interface graphique**

### **• Interface graphique JAVA**

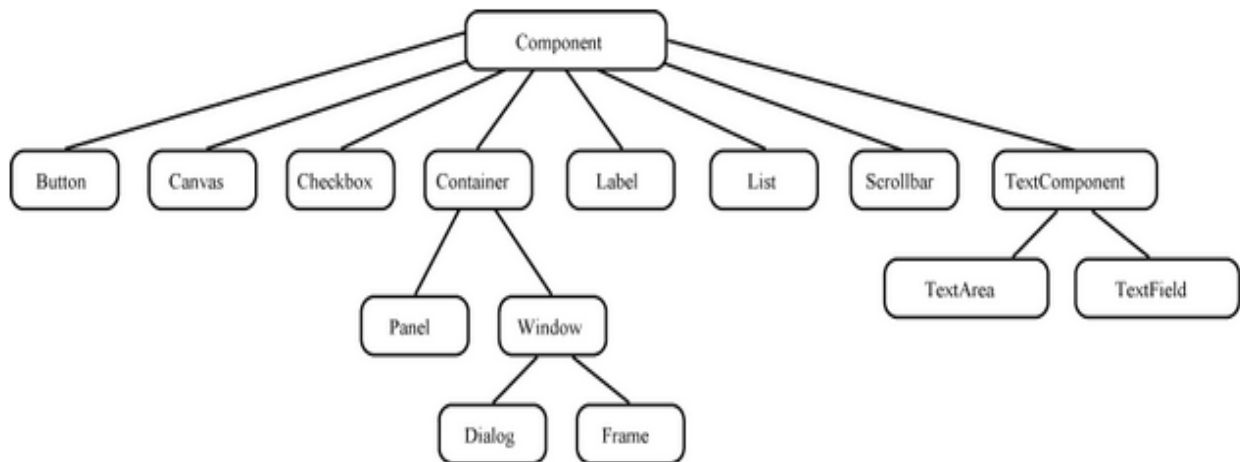
Une interface graphique est un ensemble de composant permettant à un utilisateur de communiquer avec un logiciel. Les éléments graphiques de l'interface JAVA sont définis dans le paquetage (package) AWT (abstract Windows Toolkit). AWT est une librairie de classes graphiques portables s'exécutant sans recompilation sur tout ordinateur disposant d'un interpréteur de code JAVA (machine virtuelle JAVA). Cependant, la machine virtuelle s'appuie en partie sur les ressources graphiques du système d'exploitation.

L'apparence des fenêtres et boutons diffère d'un système d'exploitation à autre avec AWT. Il existe une autre librairie d'interfaces graphiques appelée Swing entièrement autonome, qui ne dépend pas du système d'exploitation, mais qui peut prendre l'aspect de tel ou tel système d'exploitation à la demande.

### **• Classe component**



Les composants graphiques d'AWT constituent une hiérarchie de classes utilisant le principe de l'héritage. La classe Component dérive de la classe d'AWT dérive de la classe Objet.



**Figure 2.17: Les différents composants de la classe Component**

La classe component est une classe abstraite définissant les caractéristiques communes aux objets pouvant être représentés graphiquement à l'écran. Ces objets ont des propriétés communes (taille, emplacement sur l'écran ...) et des méthodes pour gérer les événements communs (fournir la taille de l'objet, afficher ou masquer l'objet, gérer les événements concernant ...).

De plus, nous avons utilisé des classes héritant de la classe Swing

– Classe **JOptionPane** :

C'est une classe de JAVA qui permet de créer facilement des boîtes de dialogue, on a utilisé ses méthodes afin de simulé des messages d'alerte en cas de confirmation, de suppression et manque de données dans le cas d'ajout.

L'utilisation typique de cette classe consiste à des appels à une des méthodes statiques qui suivent:

<i>Méthodes</i>	<i>Description</i>	<i>Retourne</i>
showConfirmDialog	Demande une question qui se répond par oui/non/annulé.	Option
showInputDialog	Demande de taper une réponse.	String(Objet)
showMessageDialog	Transcrit un message à l'utilisateur.	Rien
showOptionDialog	Une méthode générale réunissant les 3 précédentes.	Option

**Tableau 2.1 : Les méthodes statiques de la Classe JOptionPane.**

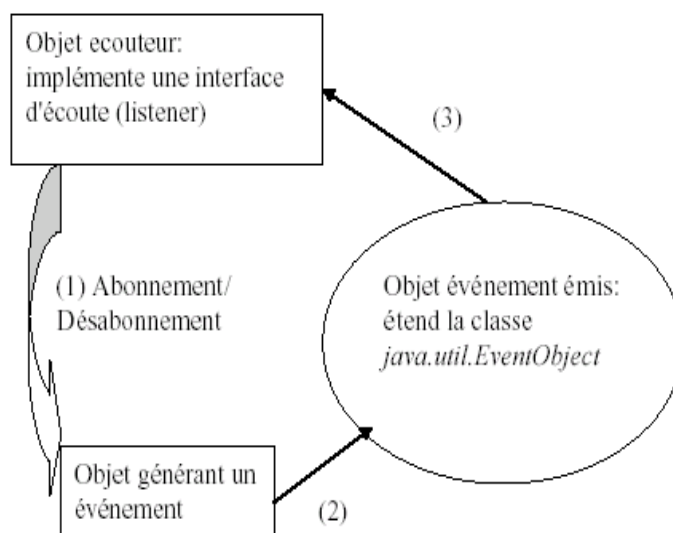
### • Événements

Les interfaces **EventListener** Permet de définir un comportement en réponse à une action d'un utilisateur ou du système, les interfaces **EventListener** dépendent du type de composants graphiques utilisés, Une classe doit contenir une interface auditeur pour chaque type de composant :

– **Interface ActionListener** : cette interface permet d'effectuer une action spécifique sur un composant.



- **Interface `TextListener`** : Cette interface permet de réagir à la modification de zone de saisie ou de texte.
- **Interface `ItemListener`** : Cette interface permet de réagir à la sélection de cases à cocher et de liste d'options.



**Figure 2.18: Différents comportement des événements.**

### • Exceptions

Pour traiter les erreurs, Java propose un mécanisme qualifié d'exception, consistant à effectuer les instructions dans un bloc d'essai (le bloc try) qui surveille les instructions. Lors de l'apparition d'une erreur, celle-ci est lancée dans un bloc de traitement d'erreur (le bloc catch, appelé handler d'exception) sous forme d'un objet appelé Exception.

Le bloc de traitement d'erreur va lever l'exception. Le handler d'exception peut alors traiter l'erreur puis lancer à nouveau l'exception vers un bloc de plus haut niveau.

### • JFreeChart

JFreeChart est une bibliothèque JAVA totalement gratuite. Elle permet de rendre facilement accessible aux développeurs des graphiques de qualité professionnelle pour leurs applications.

C'est un projet fondé en février 2000 par David Gilbert. Il est utilisé approximativement par 40 000 à 50 000 développeurs. Il est maintenu essentiellement par M. Gilbert avec quelques autres contributeurs. Les fonds sont fournis par la société Object Refinery Limited. Elle s'appuie sur la bibliothèque JCommon qui est dans les mêmes mains. JFreeChart inclut :

- Une très large et consistante API qui supporte de nombreux types de graphiques.
- Une grande documentation propre à cette API qui est fortement utile et bien écrite.
- Plusieurs types d'exports (PDF, EPS, SVG, PNG, JPEG...).
- Une licence open-source GNU Lesser General Public Licence (LGPL).
- Exemples de graphiques disponibles sous Jfreechart :
- Graphiques
- Diagramme camembert
- Diagramme de Gantt



- Histogrammes
- Thermomètres, compas, compteur de vitesse, etc...

Voici un petit aperçu de ce que l'on peut réaliser avec cette API :



Figure 2.19 : Exemple de diagramme qu'on peut créer avec API JFreeChart

- Pour utiliser ce plug-in, il faut télécharger le fichier jfreechart-0.9.18.zip ou plus et le décompresser. Son utilisation nécessite l'ajout dans le classpath<sup>[5]</sup> des fichiers jfreechart-0.9.18.zip<sup>[6]</sup> et des fichiers .jar présents dans le répertoire lib décompressé.
- Les données utilisées dans le graphique sont encapsulées dans un objet de type Dataset. Il existe plusieurs sous-types de cette classe en fonction du type de graphique souhaité.
- Un objet de type JFreechart encapsule le graphique. Une instance d'un tel objet est obtenue en utilisant une des méthodes de la classe ChartFactory<sup>[7]</sup>.

## 2.4 Programmation mobile (Interface PDA)

Pour concevoir l'interface PDA il y'a deux principaux langages à utiliser : C# ou JAVA ME, dans ce projet on utilise JAVA ME (J2ME).

J2ME (JAVA 2 Micro Edition) ou JAVA ME est le Framework<sup>[8]</sup> JAVA spécialisé dans les applications mobiles. Des plates-formes JAVA compatibles avec J2ME sont embarquées dans de nombreux téléphones portables et PDA.

Une plate-forme J2ME est composée de :

- une JVM (JAVA Virtual Machine), une machine virtuelle capable d'exécuter une application JAVA;
- une « configuration », une API donnant accès aux fonctions de base du système;

<sup>5</sup>Classpath est un paramètre passé à une JVM qui définit le chemin d'accès au répertoire où se trouvent les classes et les packages.

<sup>6</sup><http://www.jfree.org/jfreechart/download.html>

<sup>7</sup><http://www.mti.epita.fr/blogs/2008/09/16/jfreechart-et-comment-refondre-le-panel-doptions-des-graphiques/>





- un « profil », une API donnant accès aux fonctions spécifiques de la plate-forme.  
**Rappel:** JAVA Virtuelle Machine (déjà détaillé) : est un appareil informatique fictif qui exécute des programmes compilés sous forme de byte code JAVA.  
Il existe principalement 3 machines virtuelles JAVA intéressantes sous Pocket PC :
  - Crème de Nico.
  - Super Aba.
  - J9(Ive) d'IBM.

### 2.4.1 Les configurations

Les configurations définissent les caractéristiques de bases d'un environnement d'exécution, pour certain type de machine possédant un ensemble de caractéristiques et de ressources similaires, elles se composent d'une machine virtuelle et d'un ensemble d'API de base, deux configurations sont actuellement définies :

- La CLDC 1.0 est spécifiée dans la JSR 030<sup>[9]</sup>, elle concerne les appareils possédant des ressources faibles (moins de 512 Kb de RAM, faible vitesse du processeur, connexion réseau limitée et intermittente) et une interface utilisateur réduite (par exemple un téléphone mobile ou un PDA "entrée de gamme"). Elle s'utilise sur une machine virtuelle KVM.  
Il est à signaler que la version 1.1 est le résultat des spécifications de la JSR 139, une des améliorations les plus importantes et le support des nombres flottants.
- La CDC est spécifiée dans la JSR 036, elle concerne des appareils possédant des ressources plus importantes (au moins 2Mb de RAM, un processeur 32 bits, une meilleure connexion au réseau), On l'utilise sur une machine virtuelle CVM.

### 2.4.2 Les profils les plus courants

Le profil est une spécification détaillant une API. Il est basé sur et utilise une configuration. Le profil est nécessaire pour fournir un environnement de fonctionnement complet pour un type d'appareil donné.

- Le MIDP (Mobile Information Device Profile) est une API JAVA permettant la gestion de l'interface utilisateur, la gestion de l'interface réseau et la gestion d'une base de données sur le mobile. Cette API est à ce jour utilisée sur les mobiles "compatibles J2ME".
- Le Dja (Docomo JAVA) est une version Japonaise du MIDP. C'est une API optimisée pour les terminaux i-mode de NttDocomo. On peut noter une gestion améliorée de l'interface graphique par rapport au MIDP tel qu'il est spécifié par Sun Microsystem.

### 2.4.3 LangageJ2ME

La MIDlet possède trois méthodes gérant le cycle de vie de l'application en trois états possibles (active, suspendu ou détruite) :

- ✓ **StartApp ()** : méthode appelée à chaque démarrage ou redémarrage de l'application.
- ✓ **PauseApp ()** : cette méthode est appelée lors de la mise en pause de l'application.
- ✓ **destroyApp ()** : cette méthode est appelée lors de la destruction de l'application.

---

<sup>8</sup><http://fr.wikipedia.org/wiki/Framework>



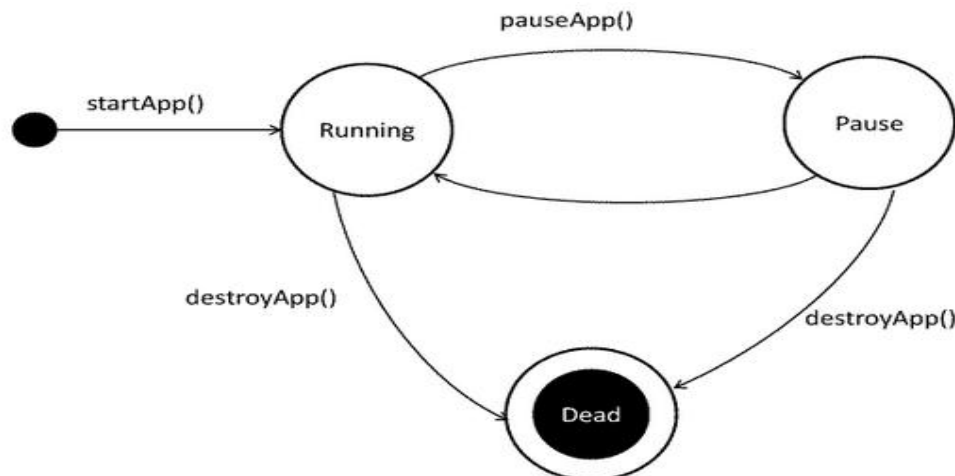


Figure 2.20 : Cycle de vie d'une MIDlet.

### – La Classe **Display** :

Toute application MIDlet doit obligatoirement instancier un objet de type Display. Cet objet assure le dialogue entre l'application et le système graphique du système d'exploitation utilisé.

- Pour utiliser les éléments graphiques, il faut instancier un objet de type **Screen**.
- L'objet de type Display possède des méthodes pour afficher l'élément graphiques.
- La méthode statique `getDisplay ()` renvoie une instance de la classe **Display** qui encapsule l'écran associé à la MIDlet fournie en paramètre de la méthode.

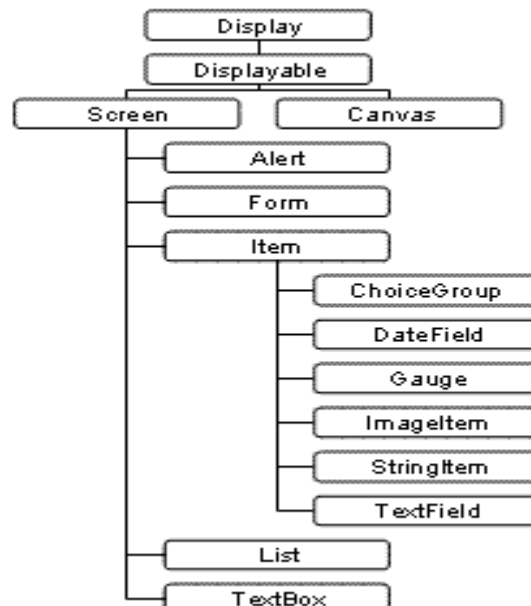


Figure 2.21 : Classe Display

Tout élément graphique mis dans un **Display**, hérite de **Displayable**, un seul objet de ce dernier peut être affiché, soit des éléments graphiques de haut niveau héritant de la classe mère **Screen** ou des éléments graphiques de bas niveau fils de la classe mère **Canvas**.

Parmi les classes utilisées on cite :

<sup>9</sup>[http://fr.wikipedia.org/wiki/JAVA\\_Specification\\_Requests](http://fr.wikipedia.org/wiki/JAVA_Specification_Requests)



### – Les « Ticker »

Les éléments Ticker peuvent être attachés à n'importe quel élément dérivant de Displayable. Les Tickers permettent l'affichage d'un texte défilant en bas de l'écran.

### – Les classes Timer et TimerTask

Ces deux classes du paquetage `java.util` permettent de lancer l'exécution des tâches à des intervalles donnés, `TimerTask` a une méthode `run()` qui détermine la tâche à accomplir. `Timer` détermine le moment d'exécution des tâches qu'on lui associe.

### – Classe Command :

C'est un « Bouton MIDP » que l'utilisateur actionne à l'aide des touches du clavier, le bouton s'ajoute dans l'interface graphique du Display en fonction du nombre de boutons, type de l'écran ou du téléphone mobile. La programmation des traitements des événements est similaire en principe à celle de l'application PC.

## 2.5 Connexion à la base de données

L'application permet la communication à une base de données, c'est pour cela la création d'une base de données est nécessaire, celle-ci est totalement séparée de notre application et doit se communiquer de façon flexible, dans cette optique nous avons pensés à des méthodes pour lier l'application avec la base de données. Cette connexion se fait selon deux façons, une connexion directe en utilisant le Package JDBC et une connexion indirecte en utilisant le langage J2EE plus précisément les servlets comme moyen intermédiaire entre l'application MIDlet et la base de données. Par la suite nous allons aborder ces deux types de connexions.

### 2.5.1 Package JDBC

JDBC est un package contenant des classes et des interfaces permettant l'accès à une base de données depuis un programme JAVA.

L'API JDBC est contenu dans le package `java.sql` et permet une :

- connexion à une base de données ;
- expression de requêtes relationnelles en SQL ;
- encapsulation des résultats des requêtes.

Ses principales caractéristiques sont :

- interface uniforme permettant un accès homogène aux SGBD ;
- simple à mettre en œuvre ;
- indépendant du SGBD ;
- supportant les fonctionnalités de base du langage SQL.

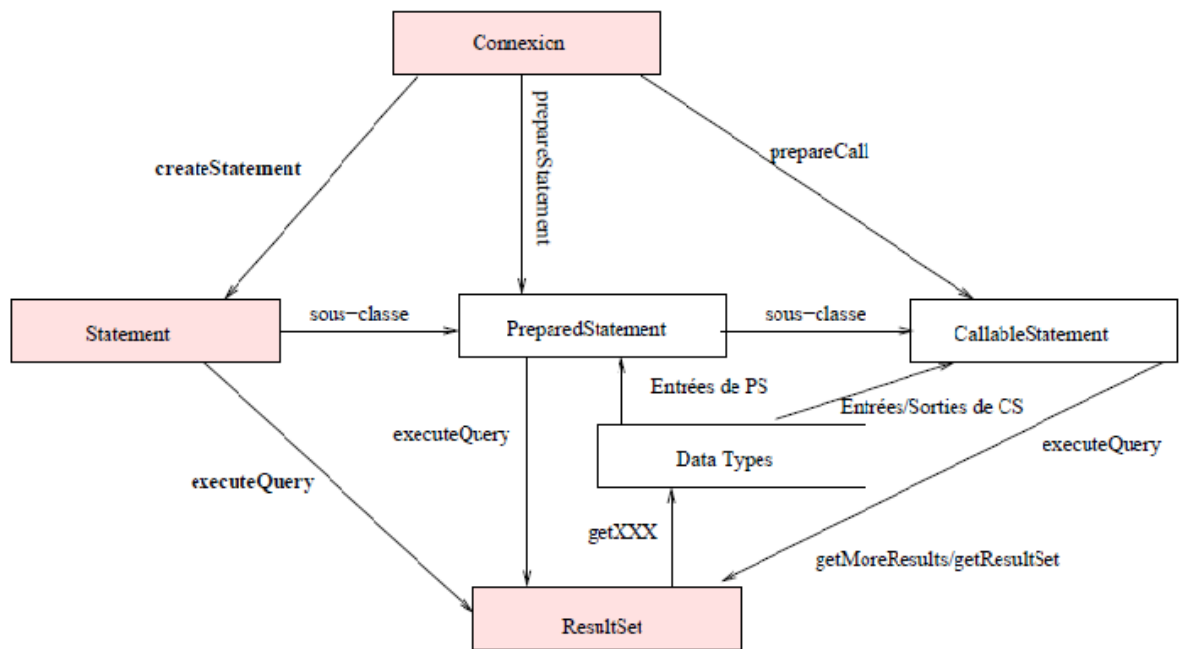


Figure 2.22: Les classes et interfaces du package java.sql

Les étapes d'un programme utilisant JDBC sont :

- mettre en place le pilote ou driver ;
- établir une connexion avec une source de données ;
- effectuer les requêtes ;
- utiliser les données obtenues pour des affichages, des traitements statistiques, etc.
- mettre à jour les informations de la source des données ;
- terminer la connexion.

Les mots clé utilisés sont :

- **DriverManager**: charge et configure le driver de la base de données.
- **Connection** : réalise la connexion et l'authentification à la base de données.
- **Statement** : contient la requête SQL et la transmet à la base de données.
- **ResultSet** : permet de parcourir les informations retournées par la base de données dans le cas d'une sélection de données.

### Les drivers JDBC :

L'hétérogénéité des bases de données empêche une connexion directe entre une base de données et une application JAVA.

Il faut utiliser un pilote qui "traduit" des commandes JAVA pour une base de données :

- Chaque pilote est propre à une base de données.
- Le pilote implémente l'interface `java.sql.Driver`.

Pour utiliser le driver d'ORACLE on insère le code suivant :

```
Class.forName("ORACLE.jdbc.driver.ORACLEDriver");
```

Pour l'ouverture d'une connexion JDBC, les méthodes JAVA qui s'exécutent à l'extérieur de la base de données doivent spécifier :



- le nom du pilote JDBC (url) ;
- le nom d'utilisateur (user) ;
- le mot de passe de cet utilisateur (password);
- les paramètres pour localiser la base de données.

Les paramètres nécessaires pour l'ouverture de connexion sont :

```
String url = "jdbc:ORACLE:thin:@acer-77dcce6f82:1521:XE";
String user = "oncf";
String passwd = "oncf";
conn = DriverManager.getConnection(url,user,passwd);
```

Avec

- acer-77dcce6f82 nom du PC.
- 1521 port utilisé.
- XE nom de la base de données.

- Le driver JDBC effectue d'abord un premier accès à la base pour découvrir les types des colonnes impliquées dans la requête puis un deuxième pour exécuter le code SQL.

#### Problème rencontré :

Il s'est avéré de déclarer plusieurs pools de connexions, des sources de données associées et de JDBC Persistence Store. Au bout d'un (trop court) moment, le serveur weblogic annonce l'erreur suivante:

```
BEA-001129 Received exception while creating connection for pool "cgPool":
Listener refused the connection with the following error:
ORA-12519, TNS: no appropriate service handler found
The Connection descriptor used by the client was:127.0.0.1:1521: xe
```

Le problème est que le listener effectue un comptage des connexions. Lorsque le nombre de connexions dépassent la valeur indiquée par le paramètre PROCESSES il refuse les connexions suivantes. Il faut donc ajuster ce paramètre pour que WebLogic puisse utiliser ses pools de connexions de manière optimale. Cependant il existe un bug: le listener ne décrémente jamais ce compteur. Donc l'erreur peut aussi se produire après plusieurs arrêts-relances sur le serveur d'application.

Pour remédier à ce problème :

Il est nécessaire d'augmenter le paramètre PROCESSES de l'instance ORACLE XE comme suit :

1. Ouvrir une session SQL (type SQLPLUS)
2. `SQL> SHOW PARAMETER PROCESSES` donne la valeur du paramètre. Par défaut sous ORACLE XE, la valeur est 40
3. `SQL> ALTER system SET processes=100 scope=spfile;` cette commande met à jour la valeur du paramètre PROCESSES dans le fichier de configuration d'ORACLE. En effet, sous ORACLE XE, il existe dans le fichier qui est ini.ORA un lien vers un fichier binaire contenant les paramètres de configuration de la base, un SPFILE<sup>[10]</sup>.
4. `SQL> COMMIT;` pour valider les modifications.
5. Arrêter la base.
6. Redémarrer la base: les paramètres sont pris en compte.

Après ces modifications, nous avons pallié à ce problème.

<sup>10</sup>SPFILE (Fichier de paramètres persistant) est un fichier binaire recherché automatiquement au démarrage de l'instance. Son nom par défaut est spfile <SID>.ora. Le fichier SPFILE est recommandé du fait que les paramètres d'initialisation sont gérés dynamiquement dans un fichier persistant côté serveur. il est modifié par le serveur ORACLE.



## 2.5.2 Langage J2EE

### •Présentation

J2EE est une plate-forme fortement orientée serveur pour le développement et l'exécution d'applications distribuées. Elle est composée de deux parties essentielles :

- Un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en JAVA : un tel environnement se nomme serveur d'application.

- Un ensemble d'API qui peut être obtenu et utilisé séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers.

J2EE permet une grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants. Ce choix dépend des besoins auxquels doit répondre l'application mais aussi des compétences dans les différentes API de J2EE. L'architecture d'une application se découpe idéalement en au moins trois tiers :

- La partie cliente : c'est la partie qui permet le dialogue avec l'utilisateur. Elle peut être composée d'une application web ou d'applets

- La partie métier : c'est la partie qui encapsule les traitements (dans des EJB ou des JAVA Beans).

- La partie donnée : c'est la partie qui stocke les données

J2EE regroupe un ensemble d'API pour le développement d'applications d'entreprise. Ces API peuvent être regroupées en trois grandes catégories :

- Les composants : Servlet, JSP, EJB
- Les services : JDBC, JTA/JTS, JNDI, JCA, JAAS
- La communication : RMI-IIOP, JMS, JAVA Mail

Dont on s'intéressera aux composants Servlet et au service JDBC.

### • Les outils nécessaires pour développer des servlets

Initialement, pour développer des servlets, il faut utiliser le JAVA Server Development Kit (JSDK) qui est une extension du JDK. Pour réaliser les tests, le JSDK fournit, dans sa dernière version, un outil nommé servlet-Runner qui fournit un serveur http allégé.

Actuellement, pour exécuter des applications web, il faut utiliser un conteneur web ou serveur d'application : il existe de nombreuses versions commerciales tel que IBM WebSphere ou BEA WebLogic mais aussi des versions libres tel que Tomcat du projet GNU Jakarta ou encore « GlassFish 3+ ».

Ce serveur d'application ou ce conteneur web doit utiliser ou inclure un serveur http dont le plus utilisé est Apache.

Le choix d'un serveur d'application ou d'un conteneur web doit tenir compte de la version du JSDK qu'il supporte pour être compatible avec celle utilisée pour le développement des servlets. Le choix entre un serveur commercial et un libre doit tenir compte principalement du support technique, des produits annexes fournis et des outils d'installation et de configuration.

Pour simplement développer des Servlets, le choix d'un serveur libre se justifie pleinement de leur gratuité et de leur « légèreté ».

Pour cela nous avons choisis le JAVA EE 6 SDK qui intègre déjà un serveur « GlassFish 3+ », pour l'intégrer avec l'IDE Netbeans il suffit de l'installer sous Windows et puis l'ajouter d'après l'IDE Netbeans. (Voir Annexe I)

### Problème rencontré :

Durant l'installation de ce serveur, un problème de configuration de port est survenu, dont le serveur « Glass Fish 3+ » n'accepte pas le port 8080 puisqu'il est déjà réservé à ORACLE XE 10g, c'est pour cela on a changé le port d'ORACLE et conserver le port 8080 au serveur. (Voir Annexe II)

### • Mode de communication

Le mode utilisé est un mode de programmation client-serveur impliquant qu'il y ait au moins deux acteurs en jeu, un client et un serveur communiquant entre eux souvent à travers un réseau :

- Un serveur est un programme ou, par extension, une machine programmée pour rendre un service donné, en réponse à une requête qui lui est adressée.
- Un client est un programme ou, par extension, une machine qui demande à un serveur un service, en lui adressant une requête.

L'architecture client-serveur sépare le serveur qui, généralement, gère des données, du client qui cherche à accéder à ces données

### • Choisir un protocole pour communiquer

La clé de voûte qui permet à une architecture client-serveur de fonctionner, sans que ni le client ni le serveur ne sachent comment ils fonctionnent l'un l'autre, est l'utilisation d'un protocole commun. Un protocole établit la norme que doit respecter un client pour communiquer avec un serveur.

Le protocole utilisé pour communiquer est le protocole HTTP qui a été inventé par Tim Berners-Lee avec les adresses Web et le langage HTML pour créer le World Wide Web, HTTP est un protocole de la couche application. Il peut fonctionner sur n'importe quelle connexion fiable, dans les faits on utilise le protocole TCP<sup>[11]</sup> comme couche de transport.

Les clients HTTP les plus connus sont les navigateurs Web permettant à un utilisateur d'accéder à un serveur contenant les données.

### • Présentation des servlets

Une servlet est un composant qui étend les fonctionnalités d'un serveur web de manière portable et efficace. Un serveur web héberge des classes Java servlets qui sont exécutées à l'intérieur du container web. Le serveur web associe une ou plusieurs URLs à chaque servlet qui est invoquée lorsque des requêtes HTTP utilisateur sont soumises au serveur.

*Remarque :*

- Conteneur web : décode les requêtes et formate les réponses dans le format approprié.
- Tous les conteneurs de servlets doivent supporter le protocole HTTP et peuvent aussi supporter le protocole HTTPS.

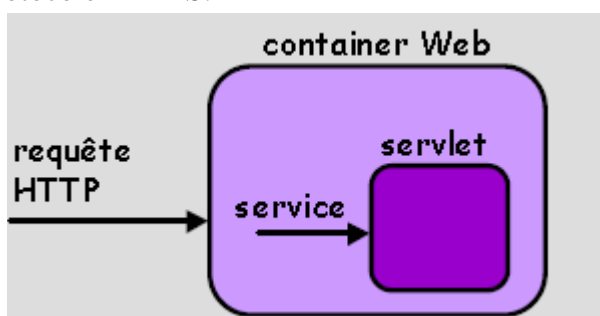


Figure 2.23 : Relation entre container Web et la requête HTTP

Les servlets sont conçues pour agir selon un modèle de requête/réponse. Tous les protocoles utilisant ce modèle peuvent être utilisés tel que http, ftp, etc. ...

Un développeur de servlet utilise l'API servlet pour :

- Initialiser et finaliser la servlet.
- Accéder à l'environnement de la servlet.

<sup>11</sup>TCP est un protocole orienté connexion, c'est-à-dire qu'il permet à deux machines qui communiquent de contrôler l'état de la transmission



- Recevoir ou rediriger les requêtes et envoyer les réponses.
- Interagir avec d'autres servlets ou composants.
- Maintenir les informations de sessions du client.
- Filtrer avant ou après traitement les requêtes et les réponses.
- Implémenter la sécurité sur le tiers web.

L'API servlet regroupe un ensemble de classes dans deux packages :

- `Javax.servlet` : contient les classes pour développer des servlets génériques indépendantes d'un protocole
- `Javax.servlet.http` : contient les classes pour développer des servlets qui reposent sur le protocole http utilisé par les serveurs web.

Exemple d'une Servlet: (voir [Annexe III](#))

### • Le choix entre JSP et Servlets

Les JSP (JAVA Server Pages) sont une technologie JAVA qui permettent la génération de pages web dynamiques.

La technologie JSP permet de séparer la présentation sous forme de code HTML et les traitements sous formes de classes JAVA définissant une servlet. Ceci est d'autant plus facile que les JSP définissent une syntaxe particulière permettant d'appeler une servlet et d'insérer le résultat de son traitement dans la page HTML dynamiquement.

Les servlets et les JSP ont de nombreux points communs puisque qu'une JSP est finalement convertie en une servlet. Le choix d'utiliser l'une ou l'autre de ces technologies ou les deux doit être fait pour tirer le meilleur parti de leurs avantages.

Dans une servlet, les traitements et la présentation sont regroupés. L'aspect présentation est dans ce cas pénible à développer et à maintenir à cause de l'utilisation répétitive de méthodes pour insérer le code HTML dans le flux de sortie.

De plus, une simple petite modification dans le code HTML nécessite la recompilation de la servlet. Avec un JSP, la séparation des traitements et de la présentation rend ceci très facile et automatique.

L'usage des servlets est obligatoire si celles-ci doivent communiquer directement avec une applet ou une application et non plus avec un serveur web.

Dans notre cas on n'a pas besoin d'avoir un affichage sous forme d'une page HTML, mais seulement d'avoir une connexion à notre base de données, pour cela le choix du servlet est le meilleur puisqu'on peut interroger la base de données à travers cette servlet.

### Conclusion :

Ce chapitre présente toutes les notions qui facilitent la compréhension de notre projet en utilisant la notion UML qui modélise les relations entre les utilisateurs et les différentes applications, ainsi que les langages de programmation : J2SE, J2ME, J2EE pour programmer les deux applications coté PDA et coté PC sans oublier le langage SQL nécessaire pour créer la base de données convenable aux besoins.



## Chapitre 3 : Réalisation des applications





Après avoir détaillé les outils nécessaires pour la réalisation de notre projet, nous développons dans ce chapitre les différentes interfaces réalisées, la méthode d'utilisation des deux applications (coté PC et PDA) et la création de la base de données.

### 3.1 Description de la base de données

Pour faire les deux applications, il était nécessaire de collecter les informations autour de VA et de les mettre dans une base de données.

Nous avons créé une base de données composée de onze tableaux, chacun de ces tableaux conserve des informations liées à la tâche faite par les agents du service.

L'interface graphique d'ORACLE propose des outils pour la création d'un compte utilisateur des bases de données et aussi la gestion de tous les utilisateurs de la base de données.

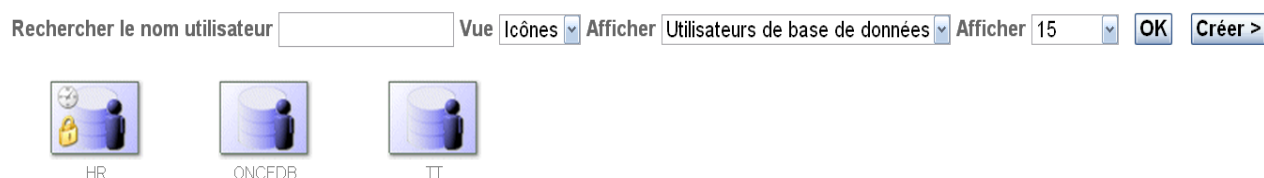


Figure 3.1 : Différents utilisateur de la base de données

Ce compte nous permet d'entamer les différents outils d'ORACLE et aussi de créer une base de données, de la modifier ou de la supprimer ...

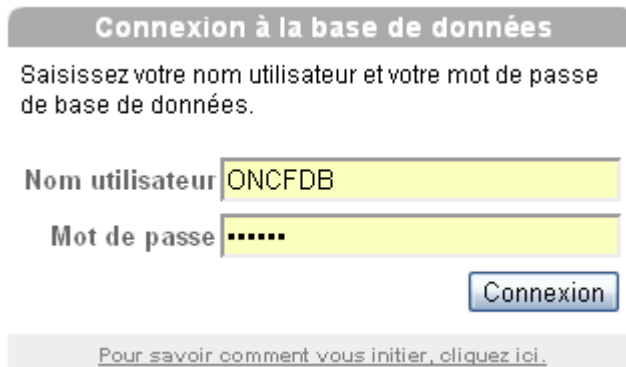


Figure 3.2 : Utilisateur de la base de données

L'exploration et les modifications de la base de données se fait via le Navigateur d'objet lié à ORACLE, dont on peut créer, modifier ou supprimer des tableaux de la base de données.

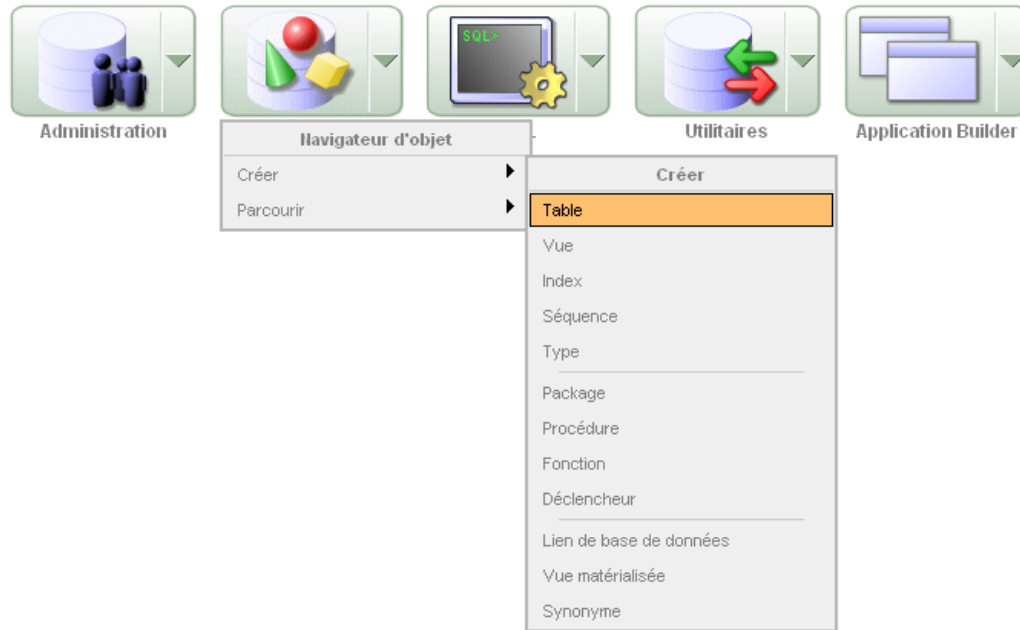


Figure 3.3 : Naviguer vers les tableaux de la base de données

Nous avons créé onze tableaux en relation avec nos besoins : tableau MACHINE, SOUSTÂCHE, TÂCHE, TECHNICIEN, VISITE, ENTREESORTIE, ÉTAT, PDA, RESPONSABLE, ROULEMENT et CONSULTATION comme affiché ci-dessous, ce tableau rassemble toutes les informations concernant le service VA.

Chaque tableau doit contenir une clé primaire et aussi il doit préciser le type des données de chaque colonne.

À l'aide d'interface ORACLE, on a créé un compte utilisateur dont on peut créer une base de données composée de plusieurs tableaux différents de leur contenu, liés entre eux avec des contraintes précises.

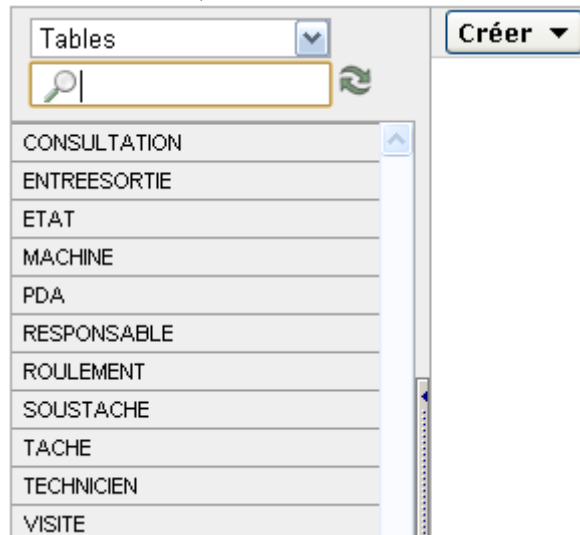


Figure 3.4 : Les différents tableaux de la base de données

### • Tableau MACHINE

Ce tableau contient deux colonnes : LOCON de type VARCHAR2 qui est le numéro spécifique à chaque locomotive, SERIELOCO aussi de type VARCHAR2 qui est une série de train de différents numéros LOCON.

Exemple : les machines de la série 1300 contient 18 machines de 1300 à 1318.

LOCON est la clé primaire de ce tableau puisqu'il est unique pour chaque machine.



Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
LOCON	VARCHAR2(20)	No	-	1
SERIELOCO	VARCHAR2(20)	No	-	-
				1 - 2

**Figure 3.5 : Le tableau MACHINE en détails**

#### • Tableau SOUSTACHE

SOUSTACHE contient tous les sous-tâches que l'agent doit les faire à chaque arrivée de machine selon son type et aussi sa série et suivant le type de la visite.

Ce tableau contient six champs dont SOUSTACHE\_ID est une clé primaire et TACHE\_ID est une clé secondaire.

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
SOUSTACHE_ID	NUMBER(6,0)	No	-	1
TITRESOUSTACHE	VARCHAR2(1000)	No	-	-
TACHE_ID	NUMBER(6,0)	No	-	-
VALIDATION	NUMBER(1,0)	No	-	-
DUREESOUSTACHE	NUMBER(6,0)	No	-	-
				1 - 5

**Figure 3.6 : Le tableau SOUSTACHE en détails**

#### • Tableau TACHE

Le tableau TACHE contient quatre champs dont TACHE\_ID est une clé primaire, il contient en plus toutes les tâches que l'agent doit passer selon le type de la visite et la série de la locomotive.

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
TACHE_ID	NUMBER(6,0)	No	-	1
TITRETACH	VARCHAR2(300)	No	-	-
PARTIE	VARCHAR2(60)	No	-	-
SERIELOCO	VARCHAR2(20)	No	-	-
				1 - 4

**Figure 3.7 : Le tableau TACHE en détails**

#### • Tableau TECHNICIEN

Chaque visite est faite par plusieurs agents dont un d'entre eux est l'agent principal, dont ce tableau rassemble toute les informations concernant cet agent principal comme le code « CODETECH » qui est une clé primaire et son nom « NOM ».

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
CODETECH	VARCHAR2(20)	No	-	1
NOM	NVARCHAR2(30)	No	-	-
				1 - 2

**Figure 3.8 : Le tableau TECHNICIEN en détails**



### • Tableau VISITE

Le besoin d'avoir un archivage de tous les opérations faites dans le service VA nécessite la création du tableau VISITE qui collecte toute les informations concernant la visite comme la date « DATEVISITE », la durée « DUREE » de la visite, le numéro de la machine « LOCON » ainsi le code d'agent « CODETECH » sans oublier les titres des tâches « TITRETACH » et sous-tâches « TITRESOUSTACHE » faites durant toute la visite.

DATEVISITE, HEUREDEBUT et SOUSTACHE\_ID sont des clés primaires ainsi que LOCON une clé secondaire (ou étrangère).

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
DATEVISITE	VARCHAR2(10)	No	-	1
HEUREDEBUT	VARCHAR2(6)	No	-	2
SOUSTACHE_ID	NUMBER(6,0)	No	-	3
LOCON	VARCHAR2(20)	No	-	-
CODETECH	VARCHAR2(20)	No	-	-
INVESTIGATION	CHAR(1)	No	-	-
OBSERVATION	VARCHAR2(600)	Yes	-	-
DUREE	NUMBER(6,0)	Yes	-	-
				1 - 8

Figure 3.9 : Le tableau VISITE en détails

### • Tableau ÉTAT

L'agent qui fait la visite doit à la fin de chaque visite préciser l'état générale de la machine, que ce soit en état bonne (aucune observation n'est exigée), bonne avec perspective ou encore indisponible (une observation est obligatoire dans ces deux derniers états).

Plusieurs informations sont récupérées automatiquement de la PDA comme la date et l'heure de la visite, le numéro de la locomotive et la durée de cette visite.

C'est pour ce besoin on a créé le tableau ÉTAT, dont DATEE, HEUREDEBUT et LOCON représente une clé primaire.

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
DATEE	VARCHAR2(20)	No	-	1
HEUREDEBUT	VARCHAR2(10)	No	-	2
LOCON	VARCHAR2(10)	No	-	3
OBSERVATION	VARCHAR2(400)	Yes	-	-
SITUATION	VARCHAR2(40)	No	-	-
DUREEVISITE	NUMBER(6,0)	No	-	-
				1 - 6

Figure 3.10 : Le Tableau ÉTAT

### • Tableau RESPONSABLE

Tout agent du service VA a un ou plusieurs responsables, ces responsables assume totalement la responsabilité sur la PDA et aussi ont le rôle d'administrateur dans l'interface graphique coté PC.

Le tableau RESPONSABLE contient les noms des responsables (RESP\_NAME clé primaire), leurs login et mot de passe qui peuvent le changer à l'aide d'interface PC.



Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
RESP_NAME	VARCHAR2(40)	No	-	1
LOGIN	VARCHAR2(10)	Yes	-	-
PASSWORD	VARCHAR2(10)	Yes	-	-
				1 - 3

**Figure 3.11 : Le Tableau RESPONSABLE**

### • Tableau ENTREESORTIE

Chaque locomotive arrivant au service VA est accompagnée de plusieurs informations nécessaires comme l'heure (HEUREENTREE) et date (DATEENTREE) de son entrée au gare du train et quelle heure (HEURESORTIE) et date (DATESORTIE) de son prochain voyage et avec quel train la locomotive était liée (TRAINENTREE) ou serait liée (TRAINSORTIE), c'est une sorte de planning que les différentes machines doivent assimiler.

Toutes ces informations sont collectées dans le tableau ENTREESORTIE qui a comme clé primaire les deux champs (TRAINENTREE, DATEENTREE).

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
TRAINENTREE	VARCHAR2(20)	No	-	1
DATEENTREE	VARCHAR2(20)	No	-	2
DATESORTIE	VARCHAR2(20)	Yes	-	-
LOCON	VARCHAR2(10)	Yes	-	-
HEUREENTREE	VARCHAR2(10)	Yes	-	-
HEURESORTIE	VARCHAR2(10)	Yes	-	-
TRAINSORTIE	VARCHAR2(20)	Yes	-	-
				1 - 7

**Figure 3.12 : Le Tableau ENTREESORTIE**

### • Tableau PDA

Pour contrôler l'utilisation de la PDA plusieurs données doivent être présentes après chaque utilisation. Ce tableau PDA collecte les informations concernant la personne qui a donné (RESP\_NAME\_DONNER) ou récupéré (RESP\_NAME\_RECUPERER) la PDA en quelles date (DATE\_DONNER\_PDA) et heure (HEURE\_DONNER\_PDA, HEURE\_RECUPERER\_PDA) ainsi que la personne qui l'a utilisée (CODETECH).

Les trois premiers champs jouent le rôle d'une clé primaire et CODETECH une clé étrangère (secondaire).

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
PDA_ID	VARCHAR2(20)	No	-	1
DATE_DONNER_PDA	VARCHAR2(10)	No	-	2
HEURE_DONNER_PDA	VARCHAR2(10)	No	-	3
HEURE_RECUPERER_PDA	VARCHAR2(10)	No	-	-
RESP_NAME_DONNER	VARCHAR2(40)	No	-	-
RESP_NAME_RECUPERER	VARCHAR2(40)	No	-	-
CODETECH	VARCHAR2(40)	No	-	-
				1 - 7

**Figure 3.13 : Le Tableau PDA**



### • Tableau ROULEMENT

Le roulement consiste à attribuer des journées de repos hebdomadaire différentes à des salariés ou des équipes de salariés exerçant la même activité, c'est le cas des agents du service VA.

Le tableau ROULEMENT collecte les informations nécessaires pour ce roulement, toutes ces informations sont précisées par le responsable du service.

Ce tableau contient le code d'agent: CODETECH qui une clé primaire et étrangère en même temps, la date de roulement : DATE\_ROULEMENT qui est aussi une clé primaire et contient aussi : ROUL la période hebdomadaire du travail c'est-à-dire la nuit « N », matin « M » ou soirée « S » ;

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
CODETECH	VARCHAR2(20)	No	-	1
DATE_ROULEMENT	VARCHAR2(10)	No	-	2
ROUL	VARCHAR2(1)	Yes	-	-
				1 - 3

Figure 3.14 : Le Tableau ROULEMENT

### • Tableau CONSULTATION

Après la consultation du carnet de bord l'agent est invité à faire des tâches supplémentaires hors de la liste des tâches listées dans la Checklist, c'est pour cela on a créé un tableau qui est semblable au tableau visite pour enregistrer toute tâche supplémentaire faite par l'agent.

Le tableau contient une seule clé primaire qui est TACHECON\_ID et deux clé étrangères : LOCON et CODETECH.

Nom De Colonne	Type De Données	Valeur Nullable	Valeur Par Défaut	Clé Primaire
TACHECON_ID	NUMBER(6,0)	No	-	1
TITRETACHECON	VARCHAR2(100)	No	-	-
LOCON	VARCHAR2(20)	No	-	-
CODETECH	VARCHAR2(20)	No	-	-
INVESTIGATIONCON	CHAR(1)	Yes	-	-
OBSERVATIONCON	VARCHAR2(400)	Yes	-	-
HEUREDEBUTCON	VARCHAR2(6)	Yes	-	-
DUREECON	NUMBER(6,0)	Yes	-	-
DATEVISITECON	VARCHAR2(10)	Yes	-	-
				1 - 9

Figure 3.15 : Le Tableau CONSULTATION

## 3.2 Interface PC

L'interface de la supervision (application PC) permet d'effectuer plusieurs opérations comme la recherche, la suppression et l'enregistrement dans la base de données...etc.



Dans ce but, nous avons réalisé plusieurs conteneurs graphiques de type Panel, pour traiter les cas d'utilisations possibles en simplicité et intelligibilité sans oublier l'élégance de l'application.

Nous avons réalisé un gestionnaire de passage entre conteneurs (Panel) à l'aide d'une barre de menu attachée au conteneur principale de type Frame, ensuite nous avons ajouté toutes les autres conteneurs de type Panel au conteneur principale, ce qui permet à chaque fois de basculer entre les différentes interfaces de gestion.

La barre de menu existe dans tous les interfaces du programme et permet d'accéder à n'importe quelle interface à partir depuis une autre.

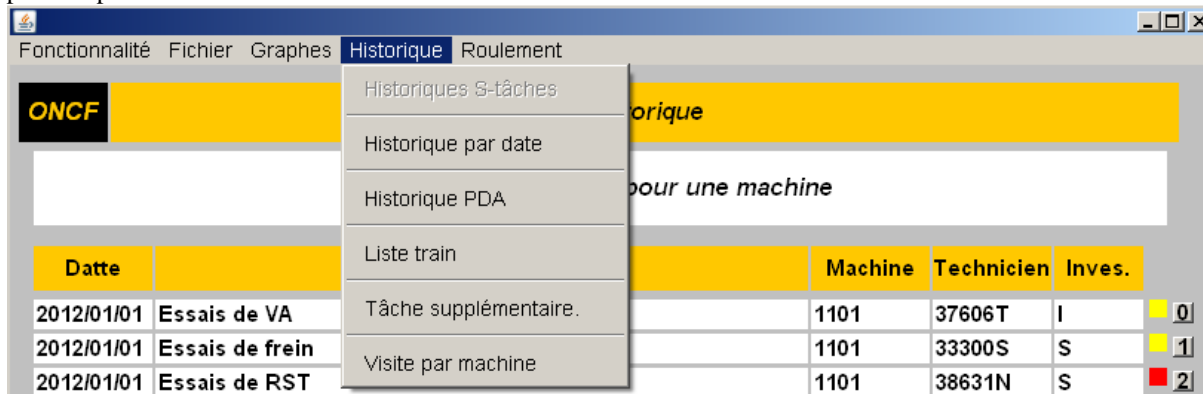


Figure 3.16 : La barre de menu facilite l'accès aux différentes interfaces.

Lorsqu'une interface est sélectionnée dans la barre de menu, son menu-item associé est désactivé (comme le cas de l'historiques S-tâches dans la figure ci-dessous) et cela se fait en donnant le paramètre false a la méthode `setEnabled()`;

Pour quitter l'application on fait appel à la méthode `System.exit()`.

Dans tous les panels nous avons ajoutés des labels pour donner des informations sur l'interface.

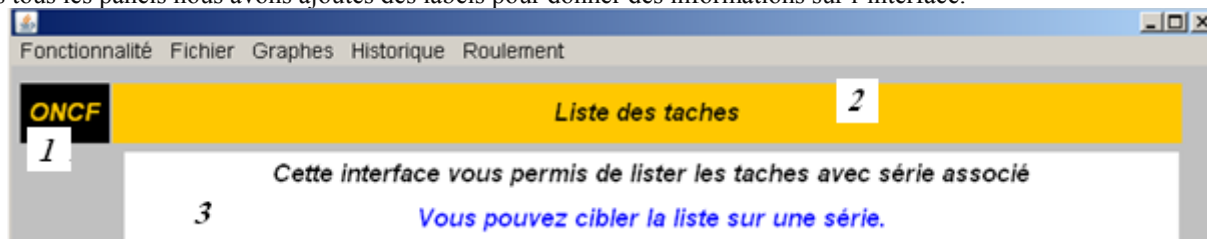


Figure 3.17 : Les informations sur l'interface.

- 1 : Label affichant nom de la société.
- 2 : Label affichant le titre de l'interface.
- 3 : Label affichant les informations sur l'interface (en noir) ou donne des recommandations sur l'utilisation de l'interface (en bleu).

La barre de menu contient 5 menus regroupant un ensemble d'interface :

**Fonctionnalité** : donnant la main pour visualiser la supervision, l'état de la machine et aussi recommander les tâches à faire ou à ne pas faire.

**Fichier** : permet de parcourir les tableaux des différents intervenants dans la visite (tâche, sous-tâche, agent, responsable).

**Graphe** : visualise différents graphes qui exploitent l'historique de visite.

**Historique** : archive des interventions pendant les visites passées.

**Roulement** : planning des horaires de travail des agents, vu qu'ils travaillent 8h par jour soit matinée, soirée ou nuit.

### • Interface accueil

Cette interface (Panel) contient des composantes graphiques de type étiquette (Label) qui indique les noms de champs et donnant des informations générales sur l'application.



**Figure 3.18 : Identification du personnel**

Afin de sécuriser le système juste un nombre limité de personnel (responsable) peuvent y accéder, pour cela le personnel est censé d'entrer un mot de passe et login qui va -une fois actionner le bouton « OK » -être identifié a une liste déjà définie dans la base de données (tableau responsable).

Si le mot de passe et login sont identifiés, l'utilisateur pourra exploiter l'application sinon les zones texte vont s'initialiser.

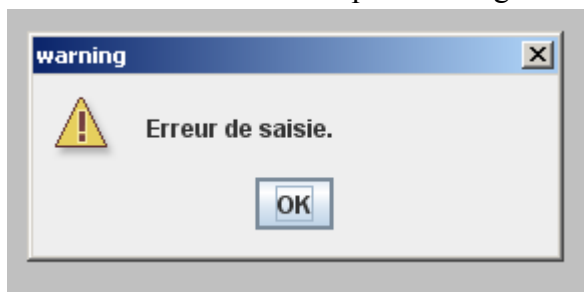
Le bouton « in » sert à initialiser le champ de saisie des mots de passe.

- **Interface d'addition de machine, agent, responsable, tâche et sous-tâche**

De même, cette interface est créée par des étiquettes, des zones de texte et deux boutons « Ok » et « in » pour initialiser et enregistrer les champs.

Il se peut que l'interface contienne des listes d'option « Choice » si l'un des champs est déjà dans la base de données.

En cas de validation avant l'insertion d'un champ un message d'erreur est affiché.



**Figure 3.19 : Message d'erreur de saisie**

Après avoir rempli tous les champs on fait appel à l'instruction d'SQL : `INSERT INTO table (champ1, champ2,..) VALUES (valeur1, valeur2,..)` pour insérer les données

Les éléments sont ajoutés dans les tableaux déjà définis dans la base de données avec les restrictions exigées.

- **Interface des listes de machine, agent, responsable, tâche et sous-tâche**

Sous forme de tableau de zones de texte avec des listes d'option permettant de cibler la recherche des « Choice » sont ajoutées, et au-dessous de chaque interface nous avons ajouté deux boutons « Suivant » et « Précédent » pour basculer entre les différentes pages de la liste.



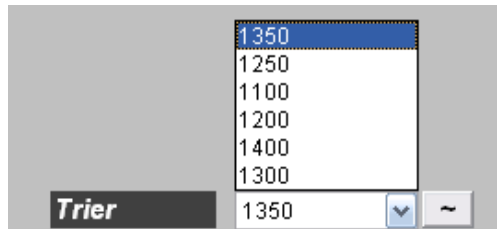


Figure 3.20 : Trier selon la série de machine

Ces interfaces donnent une vue générale sur les différents tableaux de la base de données.

• **Interface de suppression de machine, agent, responsable, tâche et sous-tâche:**

Sous forme de tableau de zones de textes, avec des listes d'options permettant de cibler la recherche avec des cases à cocher pour spécifier les éléments à supprimer, dès la validité de la suppression un message de confirmation apparaît pour valider une deuxième fois l'action.

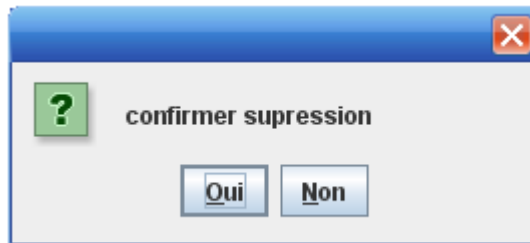


Figure 3.21 : Fenêtre de confirmation de suppression.

Au-dessous de chaque interface nous avons ajouté deux boutons « Suivant » et « Précédent » pour basculer entre les différentes pages de la liste.

Ses interfaces suppriment des éléments de la base de données dans le cas d'erreur de saisie ou en cas de leur inutilité, on donne une souplesse au donneur enregistreur.

• **Interface modification mot de passe du responsable**

Cette interface permet au responsable de modifier son mot de passe, l'interface est constituée d'un « Choice » qui implémente les noms des responsables déjà dans le tableau responsable, on a ajouté des labels indiquant les noms des champs de texte.

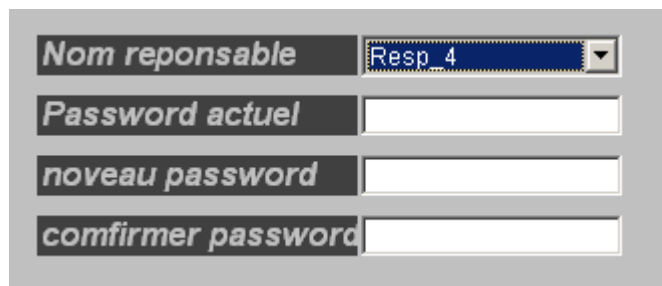
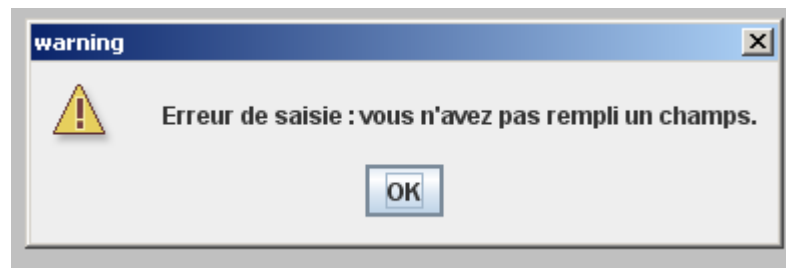


Figure 3.22 : Les champs pour modifier mot de passe

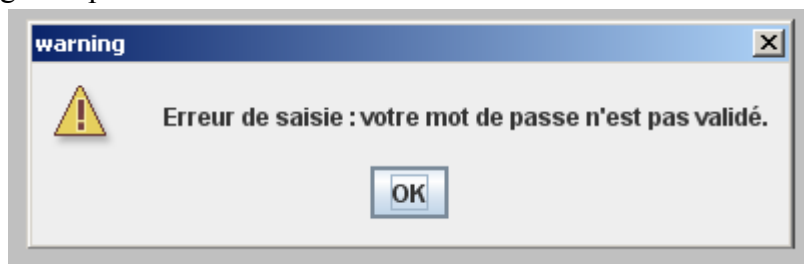
Après avoir choisi le nom convenable, le responsable doit insérer son mot de passe actuel, le nouveau mot de passe et le confirmer dans le dernier champ, après la validation quatre cas sont possibles :

- L'agent n'a pas rempli tous les champs, un message va lui faire la remarque et lui donne la main pour un éventuel essai.



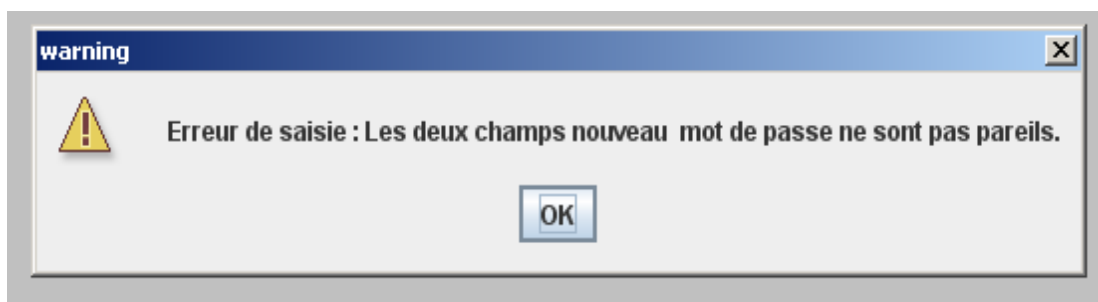
**Figure 3.23 : Erreur de saisie modifier mot de passe**

- Le deuxième cas si le mot de passe n'est pas celui du responsable choisi, un message va spécifier l'erreur.



**Figure 3.24 : Invalidité du mot de passe**

- Après vérification du mot de passe actuel si les deux champs du nouveau mot de passe ne sont pas pareils un message d'erreur s'affichera.



**Figure 3.25 : Erreur de saisi du nouveau mot de passe**

- Si le mot de passe actuel est vérifié et les deux champs des nouveaux mots de passe sont pareils, le mot de passe sera modifié et la liste des responsables s'affichera avec leurs informations.

Pour tous les champs texte nous avons utilisé la méthode `.setEchoChar ('*')` pour cacher le texte écrit et ainsi renforcer la sécurité.

### • Interface validation

Sous forme de tableau de zones de textes avec des listes d'option permettant de cibler la recherche selon une tâche, les listes de sous-tâches s'afficheront.



**Figure 3.26 : Trier selon tâche pour interface validation**



Après valider une tâche des cases à cocher à côté des sous tâches permettent de spécifier les éléments à valider pour la prochaine visite c'est comme des ordres que l'agent va exécuter.

Titre sous-tache	durée sous-tache	Série loco	validation
Compresseurs auxiliaires	10	1200	<input checked="" type="checkbox"/>
Fixation des pantographes	10	1200	<input type="checkbox"/>

Figure 3.27 : Affichage de l'interface validation

### • Interface supervision

Cette interface permet de visualiser les tâches effectuées durant le jour courant avec un ordre décroissant de l'heure, l'interface contient des boutons « suivant » et « précédent » pour basculer entre les pages.

Bouton « Actualiser » pour rafraichir l'interface et si des nouvelles tâches sont ajoutées les visualiser.

Deux cas sont possibles :

- Aucune tâche à visualiser: un message est affiché.



Figure 3.28 : Message sur l'interface supervision

- Il y'a des tâches à visualiser : les infos tâches sont visualisés comme suit.

Heure	Titre tache	Machine	Technicien	Inves.
11:04	Etat et fixation des marches-pieds	1101	30361	S <span style="color: green;">■</span> 0

Figure 3.29 : Info. Supervision

Pour accéder à plus d'interface on actionne les boutons à droite (un message s'affichera).

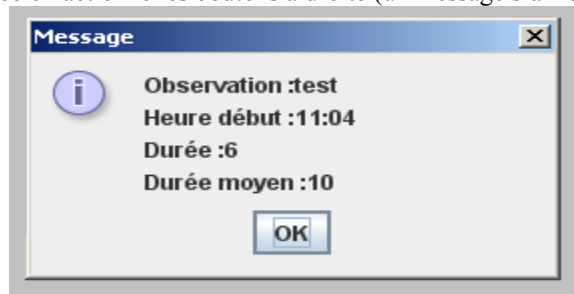


Figure 3.30 : Information supplémentaire sur supervision

### • Interface historique S-tâches

De même, sous forme de tableau de zones de textes avec des listes d'option permettant de cibler la recherche selon le N° de locomotive.

Série Loco	1350	▼	~
Loco N°		▼	OK

Figure 3.31 : Trier selon loco N°



L'interface historique permet de visualiser les tâches déjà effectuées sur différentes machines, c'est un archive des visites.

Date	Titre tache	Machine	Technicien	Inves.	
2012/03/01	Compresseurs auxiliaires	1118	38206	B	0
2012/03/02	Exfoliation	1118	33105K	I	1
2012/03/02	Ensemble des paliers essieux (fixation, chauffage)	1118	42747N	S	2

Figure 3.32 : L'affichage de l'interface historique.

L'interface donne des informations sur la date titre tâche, numéro de machine et numéro technicien les couleurs à gauche permettant de donner l'information si le travail a été effectué dans la durée exigée ou pas- vert si la durée de la tâche est inférieure aux durées exigées, jaune si les deux durées sont proche et rouge si l'agent a dépassé la durée exigée- pour les buttons numérotés ils permettent d'afficher plus d'information sur la tâche.

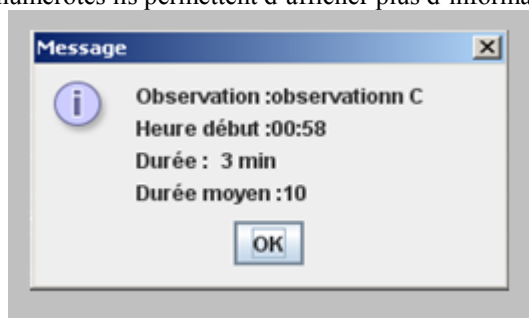


Figure 3.33 : Les informations sur une tâche (interface historique)

Si on clique sur le premier bouton par exemple plus d'information s'affiche sous forme de label (on voit que la durée est inférieure au durée moyen ou durée exigée pour sa la couleur verte), pour quitter label on clique sur OK.

#### • Interface historique PDA

De même, sous forme de tableau de zones de textes avec des listes d'option permettant de cibler la recherche selon la date, cette interface permet de visualiser le passage de la PDA entre les agents afin qu'il assume sa responsabilité.

Date	PDA_ID	Heur donner	Heur récupérer	Resp.donner	Resp.récupérer	Nom agent
2012/03/04	1	08:00	14:00	RESP_1	RESP_2	29393D
2012/03/06	3	09:00	17:00	RESP_1	RESP_1	19778W

Figure 3.34 : Interface historique PDA

Ses informations sont récupérées du tableau « PDA ».

#### • Interface liste train

Cette interface permet de lister le planning des trains, sous forme de tableau de zones de textes avec des listes d'option permettant de cibler la recherche selon la date.

L'interface affiche des informations récupérées du tableau « entrée-sortie ».

Machine	Date entrer	heur entrer	N°train entrer	Date sortie	heur sortie	N°train sortie
1101	2012/02/01	09:00	1	2012/02/02	11:00	23
1102	2012/02/02	10:00	2	2012/02/03	12:00	34
1103	2012/02/03	11:00	3	2012/02/04	13:00	23

Figure 3.35 : Interface liste train



### • Interface Tâche supplémentaire

Afin que l'application garde une grande souplesse, Dorénavant l'agent peut ajouter des tâches (via PDA), ces tâches peuvent être inscrites sur le carnet de bords.

L'interface consultation permet d'afficher l'historique de ses tâches ajoutées selon une machine (à l'aide de « Choice »).

Date	Titre tâche	Machine	Technicien	Inves.	
2012/05/18	KIKU	1416	38206	B	<input type="checkbox"/> 0
2012/05/18	KIKU	1108	38206	B	<input type="checkbox"/> 1

Figure 3.36 : Interface consultation

Les informations sont récupérées du tableau « Consultation » avec présence de carré coloré pour savoir si le temps exigé est respecté, bouton (à droite) pour plus d'information et des boutons (Suivants et précédents) en bas pour visualiser les différentes pages.

### • Interface Liste roulement

Sous forme de tableau de zones de texte cette interface affiche le planning des horaires de travail pour les agents, avec M : matinée ; S : soirée ; N : nuit.

Date	Nom agent	Horaire
2020/05/12	KAOUTI	M
2020/05/12	MOURABI	M
2020/05/12	RIHY	N

Figure 3.37 : Interface liste roulement

Ses informations sont récupérées du tableau « Roulement ».

### • Interface Ajout roulement

Aussi sous forme de tableau de zones de texte avec des labels précisant le nom des champs et des cases à cocher pour préciser le roulement, on a ajouté une zone de texte pour ajouter la date.

		Nom agent	code agent	M	N	S
		39469L	MARDHI	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		38206	KAOUTI	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Date YYYY/MM/DD <input type="text"/>		42747N	BADIR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		29393D	BOUHYA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 3.38 : Interface ajout roulement

Label date montre comment écrire date pour avoir une uniformité de données, Il y'a aussi trois bouton, deux (précédant et suivant) pour basculer entre les pages et Valider afin d'enregistrer le roulement, dès que le bouton valider est actionné il y'a trois cas :

- Date n'est pas inséré alors un message s'affiche pour le signaler.

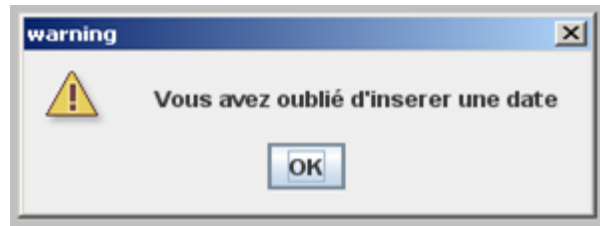


Figure 3.39 : Message d'erreur (manque de date).

- Le deuxième cas si la date est insérée mais aucun roulement n'est ajouté, ici encore un message signale l'erreur.

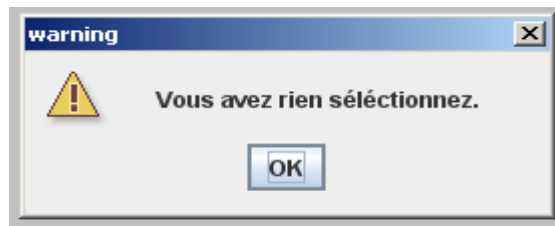


Figure 3.40 : Message d'erreur (rien n'est sélectionné).

- Le dernier cas si la date est insérée le roulement est insérer, les données sont insérées dans le tableau « Roulement » à l'aide de la requête d'SQL : `insert into roulement (codetech, date_roulement, roul) values (Ex, Ex2 ,Ex3);` Après la validation le passage à l'interface liste roulement s'effectue.

#### • Interface historique par date

Cette interface est semblable à celui de « historique S-tâches » permet d'afficher les tâches réalisées pendant une visite la sélectionne cette fois ne se fait pas juste sur une machine mais aussi sur la date afin de visualiser l'état des différents tâches effectuées sur la machine pendant une journée.

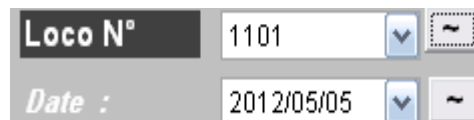


Figure 3.41 : Sélectionne de date et machine pour l'interface État

Pour l'affichage des informations, cette interface est semblable à celui de visite avec des couleurs à droite donnant idée si le travail à respecter le temps exigé, pour plus d'information il y'a des boutons affichant des informations supplémentaires sur la tâches.

2012/01/01	Cheminees	1101	38631N	B	<span style="color: green;">■</span> 0
2012/01/01	Contacteurs electromagnetiques et electropneumatiques	1101	33860J	I	<span style="color: green;">■</span> 1

Figure 3.42 : Tâches pour l'interface état.

#### • Interface Visite par machine

Sous forme de tableau de zones de textes avec des listes d'option permettant de cibler la recherche selon le N° de locomotive, l'interface permet de visualiser l'état général de train.



À la fin de visite l'agent est prié de préciser l'état générale de la machine cela est enregistré dans le tableau « État ».

Date	Heur	Machine	Etat	Observation	Durré
2012/02/01	08:00	1101	BON	BSVSQDV	30 min
2012/02/02	09:00	1102	BON	BSVSQDV	31 min

Figure 3.43 : Interface État train

Comme montre la figure l'interface affiche des informations sur état avec des couleurs à droite précise si le travail a été effectué dans le temps exigé.

On a ajouté des boutons pour passage entre les pages et « Choice » pour cibler la recherche.

### • Interface VA selon machine

Cette interface permet de spécifier une sous tâche afin de faire la visualisation des différentes interventions sur une machine entre deux dates.

Date début : 2012/05/15  
 Date fin : 2012/05/15

Partie : Partie électrique  
 Série Loco : 1100  
 Loco N° : 1101

Titre tâche : Consulter le carnet de bord et le cahier de réparation  
 Titre S-tâche : Consulter le carnet de bord et le cahier de réparation

Figure 3.44 : Trier les interventions selon deux dates pour une machine.

Après avoir validé les deux dates ses information de trie vont être le titre d'un label.

Partie : Partie électrique    Série loco : 1118  
 Titre tâche : Consulter le carnet de bord et le cahier de réparation  
 Titre sous tâche : Consulter le carnet de bord et le cahier de réparation

Figure 3.45 : Label donnant des informations sur la sélection.

Nous avons utilisé la classe Graphics afin de tracer la courbe des différentes interventions selon les dates et durés, on compare aussi la durée moyenne des interventions avec la durée exigée.

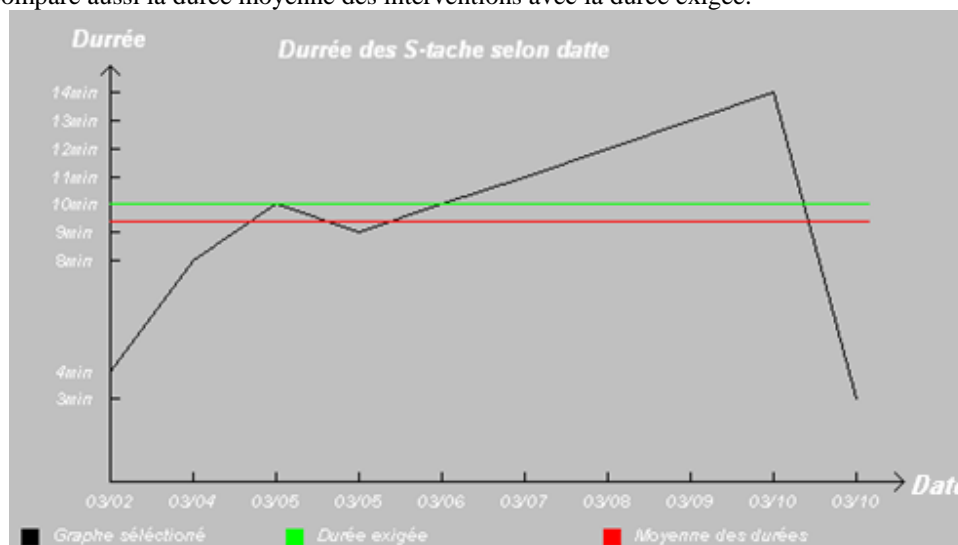


Figure 3.46 : Graphe des interventions selon les dates.



La signification des couleurs de graphe est indiquée en bas, l'axe X indique la date moi/jour et l'axe Y donne la durée en minute ou heure, minute.

La différence entre durée exigée et moyenne des durées donne des idées si le travail dans cette période a atteint l'objectif ou pas en terme durée.

### • Interface durée des s-tâches

Cette interface permet de visualiser les différentes tâches effectuées sur une machine pendant une date, pour spécifier de quelle date et machine il s'agit on utilise des Choix.

Série Loco	1100	~
Loco N°	1101	~
Date :	2012/05/05	~

Figure 3.47 : Spécifier date et machine pour le graphe état.

Après avoir validé la date les informations de tri vont être le titre d'un label, et une courbe des différentes tâches effectuées pendant la date s'affichera en comparant la durée de la tâche à celle exigée.

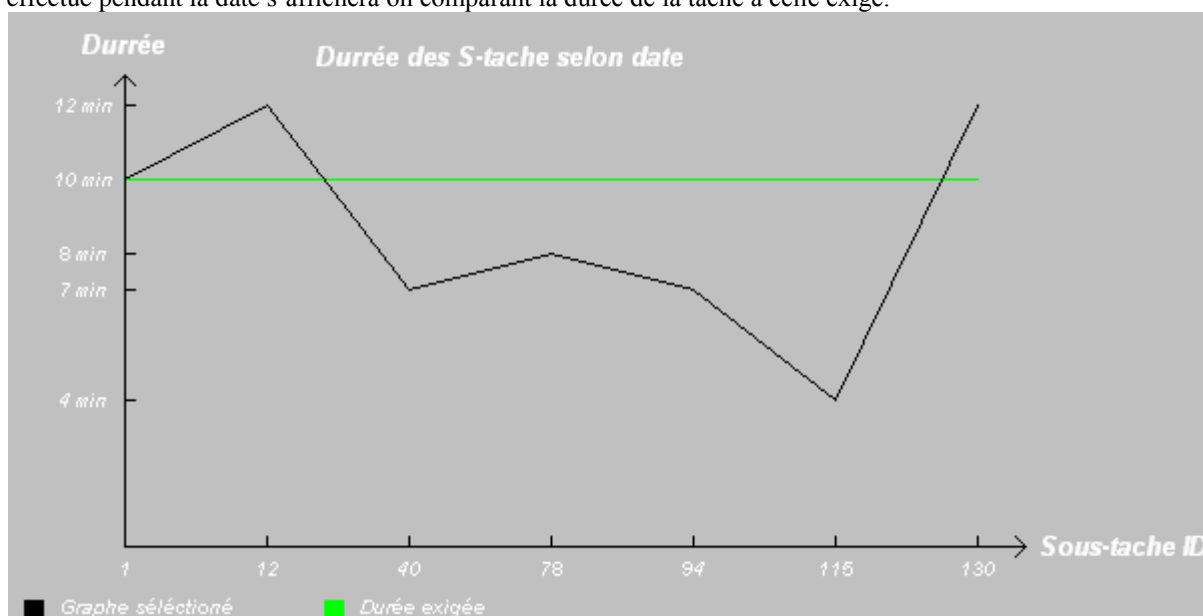


Figure 3.48 : Graphe des interventions pour une date

La signification des couleurs de graphe est indiquée en bas, l'axe X indique le Id de sous tâche et l'axe Y donne la durée en minute ou heure, minute.

La différence entre courbe des durées exigées et courbe des durées donne des idées si le travail dans cette journée a atteint l'objectif ou pas en terme durée.

### • Graphe durée visite

Cette interface permet de visualiser l'évolution de la durée de visite en fonction de la date pour chaque machine.

Série Loco	1100	~
Loco N°		~



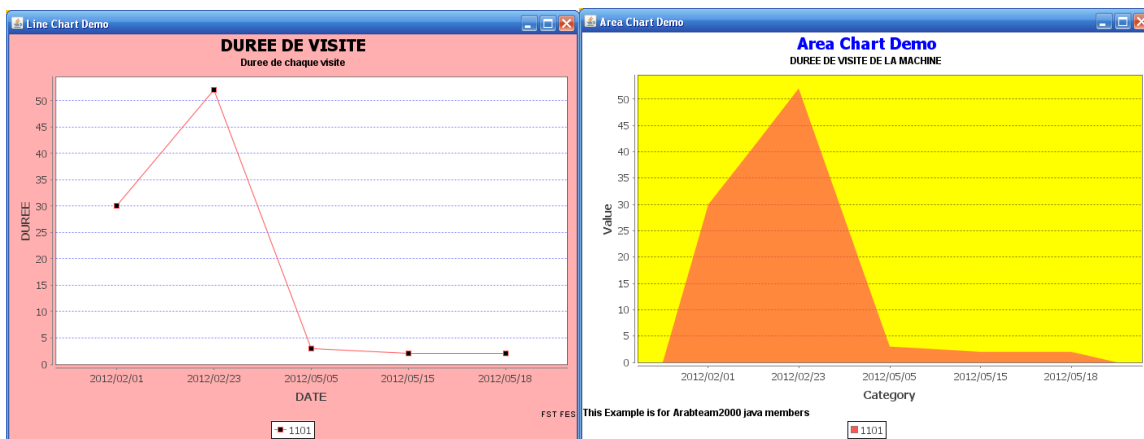


Date 1 : 2012/02/01

Date 2 : 2012/05/05

Figure 3.49 : Spécifier la machine et les deux dates

Dans cette interface il y a le choix entre deux types de graphes, le diagramme linéique : Bouton Line et le diagramme surfacique : Bouton Area ces deux graphes utilisent les bibliothèques JFreeChart figuré au-dessus, dont on peut créer ces diagrammes selon seulement le numéro de locomotive avec tout date existe, ou encore selon le numéro de locomotives et deux dates différentes : Date 1 et Date 2



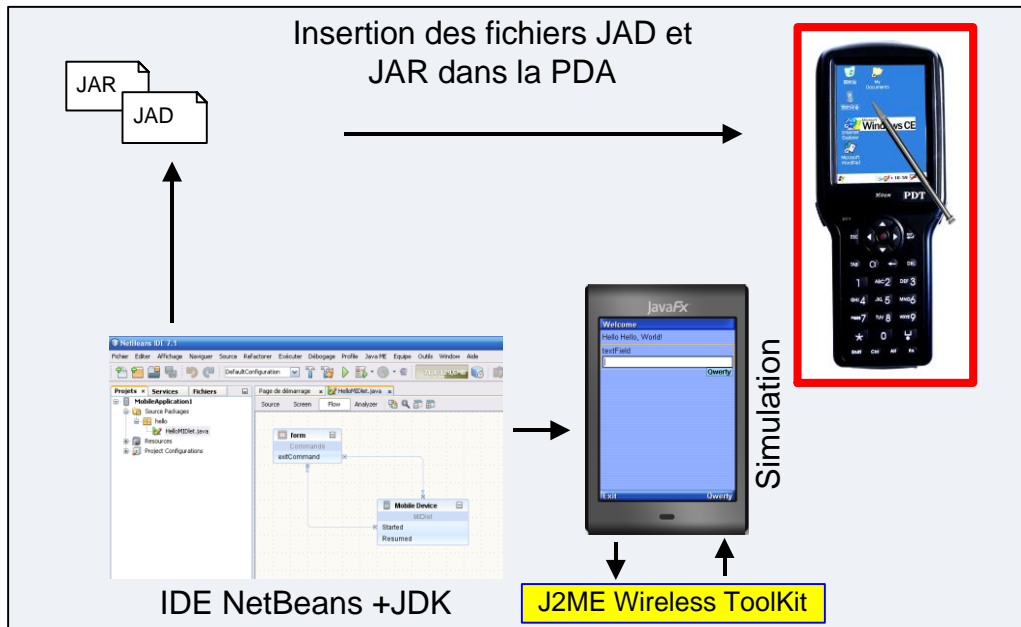
a) b)  
Figure 3.50 : Diagramme Linéique a) et Surfaccique b)

### 3.3 Interface application mobile

L'Application MIDlet représente le coté client de notre projet, dont l'agent peut l'utiliser pour insérer les données dans la base de données pendant chaque visite.

La MIDlet est écrit en J2ME, ce qui impose la création d'une application en JAVA qui sert à la connexion, c'est pour cela on a choisit d'utiliser le langage J2EE dont la « Servlet » serait un intermédiaire entre l'application mobile et la base de données.

La figure suivante illustre les différentes étapes de la création à l'implantation de notre application dans le terminal spécifique (phone, PDA ...)



**Figure 3.51 : Étape de la création d'une application mobile**

L'étape qui suit la création de la MIDlet est l'établissement de la connexion avec la base de données, cette base qui est installée dans une machine, dont on ne peut plus la transmettre à notre PDA, dans ce cas on a pensé à faire une connexion entre le PC et le mobile via une connexion (câblé ou non) à l'aide du protocole http. Cette application est constituée de plusieurs interfaces, chaque interface à sa propre fonctionnalité. Presque toutes les interfaces contiennent en bas de page une ligne de texte animé de droit à gauche de façon continue : « ONCF CasaBlanca ---- FST Fes ». Pour améliorer la vue des interfaces on a intégré des images et icônes reflétant l'esprit de l'entreprise.

### • Interface « Accueil : Responsable »

Accueil : Responsable est réservé à l'identification du responsable pour assumer la responsabilité concernant la PDA. Le responsable est invité à remplir les champs mot de passe et login pour les comparer avec les données de la base de données.



**Figure 3.52 : Interface Accueil : Responsable**



Cette interface (Figure 3.52) contient la date et l'heure de visite et encore deux champ à remplir par les données réservés aux responsables (Login et Password).

Le responsable a le choix soit de quitter l'application en cliquant sur la commande Quitter, soit de valider les champs qu'il a remplis, s'ils sont les mêmes que celles enregistrés dans la base de données l'interface passe automatiquement à l'interface : « Accueil : Technicien ».

Lors de la vérification de login et mot de passe avec celle de la base de données (dans le tableau RESPONSABLE), l'application tente de se connecter au serveur où se trouve la base de données, au cas où la connexion échoue un message d'erreur apparait nous informons qu'il y a une erreur dans la connexion, dont on doit vérifier où se trouve le problème d'après le code donné dans le message (Figure 3.53).

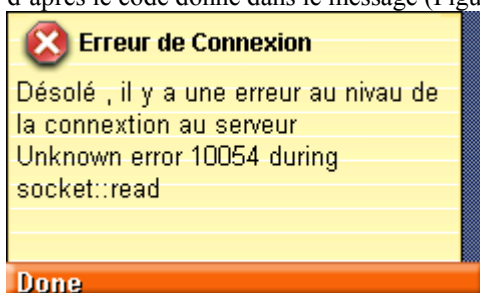


Figure 3.53 : Message d'erreur

Ce message va être affiché dans toutes étapes d'utilisation du PDA s'il y a une erreur au niveau de la connexion. Dans le cas où les champs remplis par le responsable ne sont pas identiques à celle de la base de données les champs s'initialise, et donnant au responsable la main pour essaye à nouveau de s'identifier.

#### • Interface « Accueil : Technicien »

La nouvelle interface prend automatiquement la date et l'heure et l'enregistre comme étant la référence temporaire de la visite (afin de les transmettre par la suite dans les données des tâches effectués), l'interface Technicien contient aussi un champ code dont l'agent doit le remplir par son propre code pour pouvoir poursuivre l'insertion des données, ce champs permet en plus de limiter l'accès à l'application, d'enregistrer tous les tâches au nom de cet agent.

L'interface (Figure 3.54) contient une commande:

- Valider : pour valider le mot de passe.



Figure 3.54 : Interface Accueil : Technicien

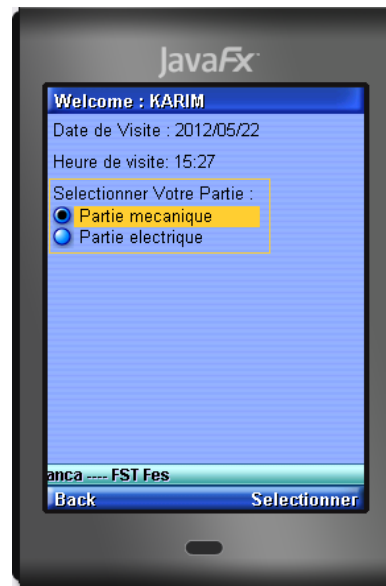


Cette interface (Figure 3.55) contient une liste de choix dont l'agent doit en choisir un pour passer à l'étape suivante. Cette liste est créée en sélectionnant les différents parties enregistrer dans la base de données. Il doit préciser la partie qu'il doit entamer durant la visite en choisissant soit la partie électrique ou la partie mécanique (ou autre partie).

D'après le code insérer dans le champ Code (dans l'interface Technicien), le nom de technicien est tiré et s'ajoute automatiquement au titre de l'interface « Welcome ».

Cette interface contient deux commandes:

- Sélectionner: pour passer à l'interface suivante qui contient une liste de machines.
- Back : pour revenir à l'interface précédant au cas d'un changement de l'agent.



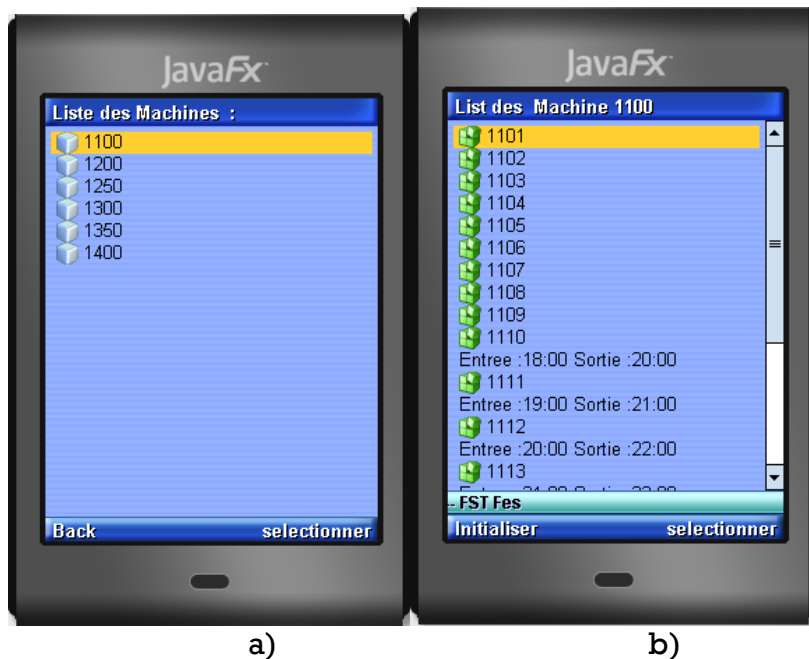
**Figure 3.55 : Interface Welcome**

#### • Interface « Liste des Machine »

Après la sélection de la partie de la visite une nouvelle interface apparus, elle contient une liste de série de machines « Liste des Machines » (Figure 3.56-a) dont chaque élément contient encore une liste de numéro réservé à chaque machine, dont la sélection d'un élément (série) provoque l'affichage de la liste de numéro machine (Figure 3.56-b).

Cette interface contient deux commandes :

- Back : pour revenir en arrière.
- Sélectionner : pour sélectionner un élément de la List.



**Figure 3.56 : Liste Machine et ses numéros**

La liste de numéro machine contient deux commande : Sélectionner pour la sélection d'un élément de la liste, Initialiser pour initialiser la liste Machine au cas d'une faute de sélection de série machine.

De plus l'agent doit savoir quelle machine à entrer au service et en quelle heure et date elle doit sortir, pour cela l'affichage des liste numéros de machine donne idées sur le planning des machines enregistrer dans le tableau «ENTREESORTIE » c'est-à-dire l'application va sélectionner les liste machine avec une information supplémentaire (planning de train) conditionné par l'heur de sortie des machine du service VA et aussi l'heure de l'entrée.

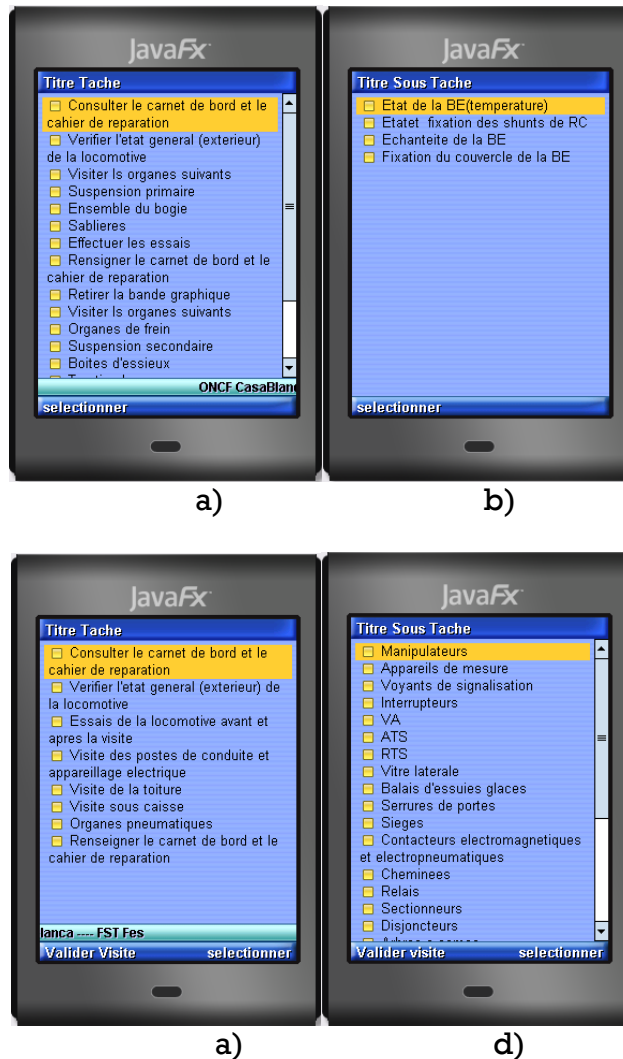
Une fois un élément de la liste des numéros de train est sélectionné, un chrono démarre automatiquement pour préciser la durée que l'agent passe dans chaque sous tâche, la durée sera enregistré en minute.

#### • Interface « Titre Tâche » et « Titre Sous Tâche »

Après la sélection du numéro de la locomotive, il apparaît deux autres fenêtres lié entre eux : la première est une fenêtre qui contient tous les titres des tâches lié au numéro du locomotive et au partie sélectionnée (Figure 3.57-a), chaque élément de cette liste est lui associé une liste de sous tâche à faire (Figure 3.57-b), les deux fenêtres contient deux commande

- Valider Visite : pour passer directement à l'interface « État ».
- Sélectionner : pour sélectionner la tâche ou sous tâche voulue et pour passer à l'interface « Information ».

Remarque Importante : La commande Valider Visite s'affichera dans ces deux interfaces, mais juste après l'insertion des données au tableau « VISITE » au moins une fois (Figure 3.57-c, d).

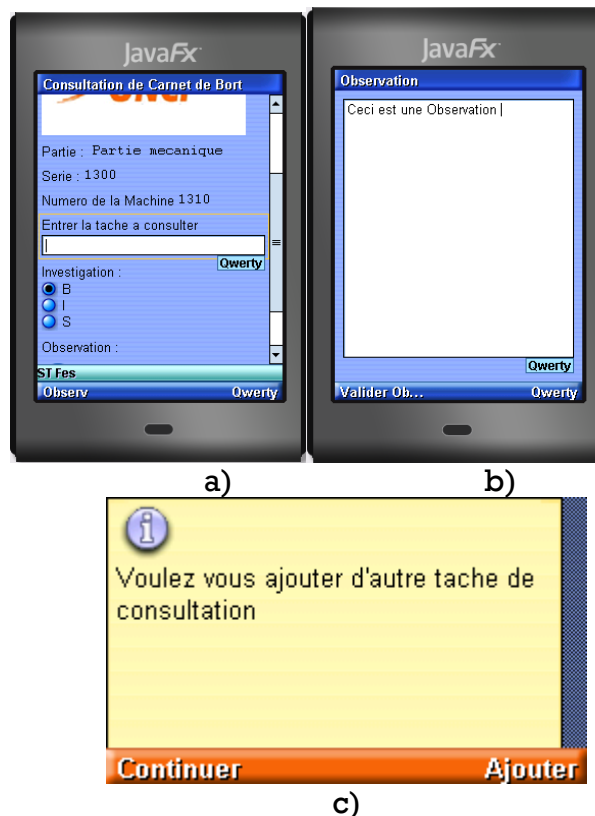


**Figure 3.57 : Liste des tâches et sous tâches**

Dans le cas où le titre tâche sélectionné est l'un des deux titres suivants :

- Consulter le carnet de bord et le cahier de réparation
- Consulter le carnet de bord

Une nouvelle interface apparaît de titre « Consulter le carnet de bord » (Figure 3.58-a), avec cette interface l'agent peut insérer des tâches supplémentaires hors de la liste des tâches et sous tâches qu'il a comme standard, ces tâches de consultation sont en relation avec le carnet de bord qui collecte toutes indications de chauffeur.



**Figure 3.58 : Interface Consulter le carnet de bord**

Par défaut l'interface contient une commande « Observ » qui permet de passer à une fenêtre pour écrire l'observation convenable (Figure 3.58-b), on valide l'observation en cliquant sur « Valider observ » dans l'interface « Observation », automatiquement une commande s'ajoute à l'interface « Consulter le carnet de bord » et en même temps elle apparaît.

La nouvelle commande « Valider » permet l'insertion des données remplit dans le tableau « CONSULTATION » et aussi l'affichage d'une fenêtre (Figure 3.58-c) qui demande si l'agent veut continuer la visite en cliquant sur la commande « Continuer » d'où on revient directement à l'interface « Titre Tâche » ou ajouter d'autre tâche supplémentaire en cliquant sur la commande « Ajouter » d'où on revient à l'interface « Consulter le carnet de bord ».

#### • Interface « Information »

L'interface « Information » recueille toutes les informations saisies au début et aussi des informations importées de la base de données comme le titre de tâche et sous-tâche.

Cette interface contient des champs que l'agent doit les remplir en effectuant la tâche demandée (Figure 3.59-a).

Si l'agent choisit l'un des champs d'investigation (B, S ou I), il doit ajouter des observations dans une nouvelle fenêtre « text », cette fenêtre contient une commande « Valider » pour valider l'observation et l'insérer dans l'interface Information de façon automatique (Figure 3.59-b).

L'interface information contient dans un premier temps deux commandes :

- OK : pour valider le choix d'Investigation et obtenir l'identificateur du sous-tâche sélectionné (SOUSTÂCHE\_ID).
- Valider : pour insérer les données dans la base de données et puis supprimer l'élément sélectionné de la liste des sous-tâches et s'il ne reste plus de sous-tâche dans l'interface « Titres sous-tâche » il passe automatiquement à l'interface Titre

Tâche et supprime l'élément sélectionné de cette interface et ainsi de suite . Dans le cas où il ne reste aucun élément dans les deux interfaces il arrête le chrono et passe directement à l'interface État, et pour assurer l'interaction de l'application avec la base de données, il faut obligatoirement insérer les données d'au moins une sous-tâche pour valider la visite. Dans le cas où l'agent insère les données avec succès à la base de données une commande «Valider Visite » s'ajoute automatiquement à l'interface « Titre Tâche » et « Titre Sous Tâche ».

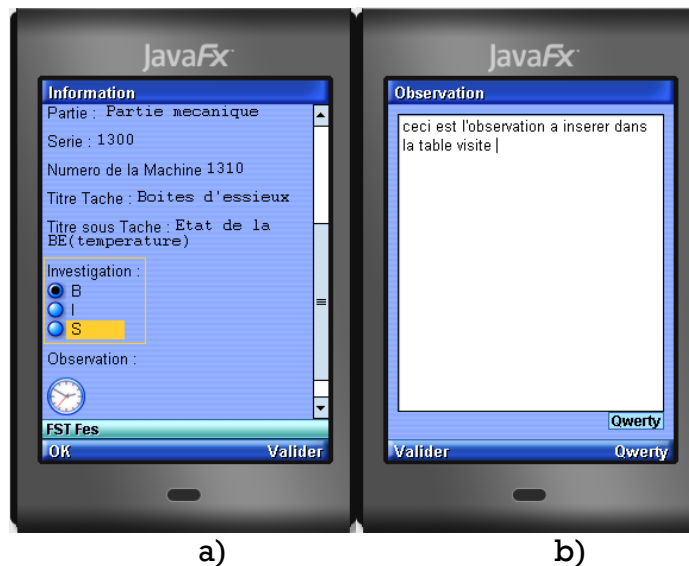


Figure 3.59 : Interface Information et texte

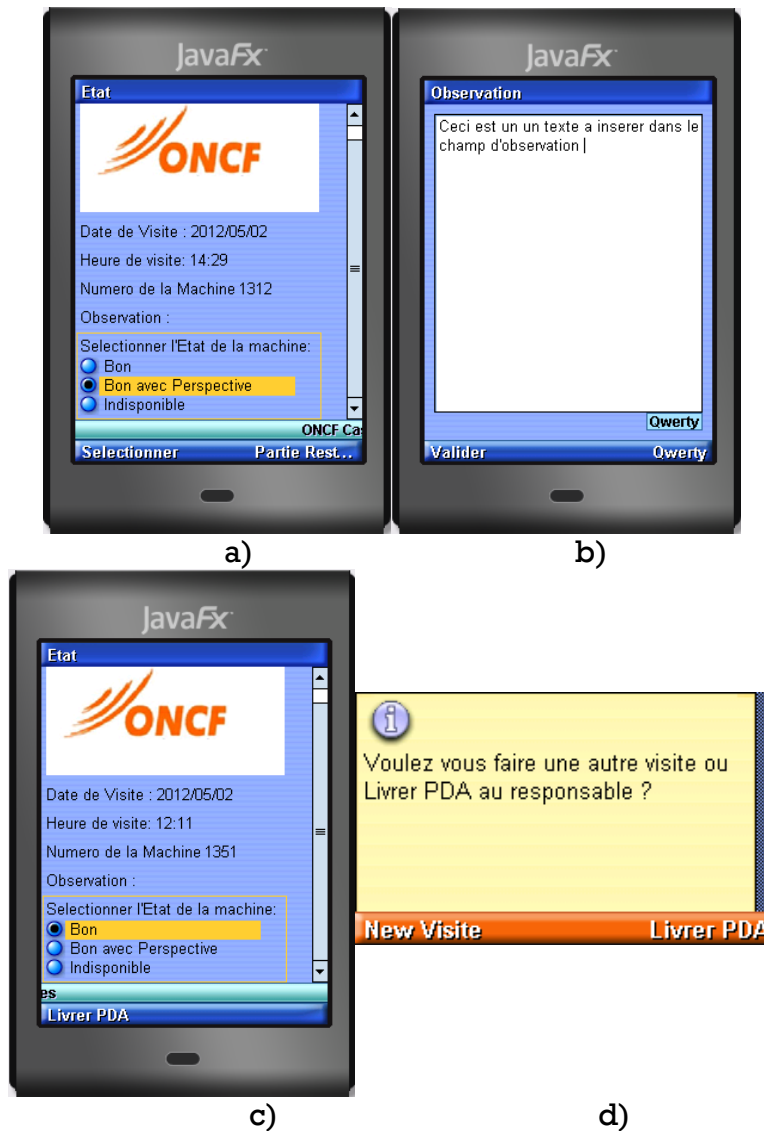
#### • Interface « État »

État est une interface qui sert à identifier l'état générale du locomotive après remplissage des données pour les tâche effectuée, ces données sont insérer dans le tableau ETAT, l'agent a le choix entre trois État : Bon (qui ne nécessite plus d'observation) et Bon avec Perspective et indisponible (l'observation est obligatoire) (Figure 3.60-a).

Cette fenêtre contient deux commandes:

- Sélectionner pour valider le choix de l'état et l'insertion des données dans le tableau ÉTAT, si l'élément sélectionné est autre que « BON » une nouvelle fenêtre apparait et l'agent peut écrire ces remarques pour les ajouter au champs observation en cliquant sur la commande Valider (Figure 3.60-b).





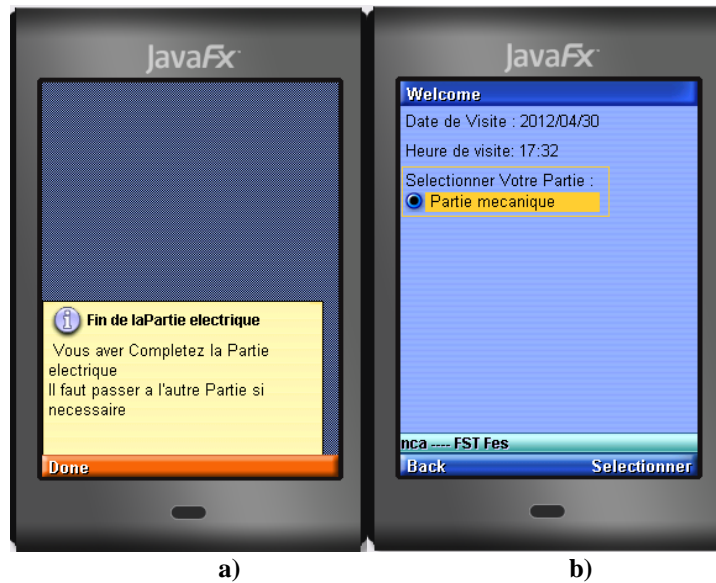
**Figure 3.60 : État de locomotive**

Livrer PDA : En cliquant sur cette commande, une fenêtre apparut (Figure 3.60-d) en demandant si l'agent veut faire une autre visite ou Livrer la PDA au responsable, cette commande n'est apparue que lorsqu'on clique la commande Valider de l'interface « Observation » (Figure 3.60-b-c).

L'agent peut soit cliquer sur :

New Visite : cette commande permet de réinitialiser tous les champs remplis dans le dernière visite et aussi d'aller directement à l'interface « Accueil Technicien » pour faire une nouvelle visite.

Livrer PDA : permet d'aller à l'interface Responsable.



**Figure 3.61 : Interface appareil avec la commande Partie restante**

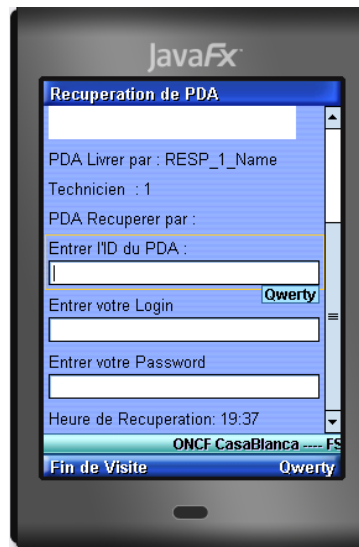
- **Partie Restante** : En cliquant sur cette commande , un Message (Figure 3.61-a) s'affiche et nous informe qu'on a terminé la partie sélectionnée au début de la visite et qu'on peut entamer l'autre partie, la partie sélectionnée est supprimée automatiquement de la liste des choix de partie dans l'interface « Welcome » et il reste seulement l'autre partie non entamée (Figure 3.61-b).

- **Interface « Récupération de PDA »**

Pour avoir des données concernant la PDA, on a pensé à ajouter une interface qui collecte l'information autour de la PDA (Figure 3.62).

Cette interface nous permettra de savoir qui a donné ou récupéré la PDA en quelle heure et date, qui la utilise, et pour identifier la PDA on a pensé à donner un identificateur unique à chaque PDA(PDA\_ID).

La seule commande contenue dans cette interface « Fin de visite » sert à l'insertion de ces détails dans le tableau PDA.



**Figure 3.62 : Interface de Récupération de PDA**

Durant la création des deux applications du projet, le problème le plus fréquent était le refus de la base de données d'enregistrer certaines données qui contiennent des apostrophes comme la phrase suivante : L'état de la machine....

La solution proposée était de remplacer l'apostrophe « ' » par une double apostrophe « '' » et la phrase précédente sera : L''état de la machine..., la base de données accepte cette phrase sans donner une erreur.

## Conclusion

Nous avons développé, dans ce chapitre consacré à la réalisation du projet, les différentes interfaces de l'application côté PC ainsi que leur rôle et leur interaction avec la base de données, ensuite nous avons traité la description de l'application côté PDA en détaillant chaque interface reliée avec la base de données. Ces deux applications nécessitent une base de données que nous avons détaillées en listant ses différents tableaux.



## Chapitre 4: Matériels et Étude technico-commercial

L'étude de marché est un élément indispensable dans notre projet, nous allons détailler une vue général au niveau de prix et performances concernant les terminaux qui seront utilisés (PDA et router Wifi) pour matérialiser les applications programmées.

### 4.1 Les assistants numériques personnels « PDA »

Dans notre projet nous allons utiliser les PDA comme un outil d'enregistrement et d'envoi et commande à distance. Cet outil doit être portable, d'un poids faible et rigide sans oublier qu'il doit supporter des environnements explorable par l'utilisateur.



**Figure 4.1 : Exemple de PDA**

Les PDAs (pour Personal Digital Assistant) sont des dispositifs mobiles légers conçus autour d'un écran tactile, cette technologie a récemment connu un bond important en termes de puissance de traitement et de capacité de télécommunications. Sa capacité sans cesse croissante de traitement et de stockage, ses fonctionnalités conviviales de télécommunications sans fil et sa petite taille font du PDA un dispositif très attrayant pour les entreprises.

Les PDA sont maintenant en mesure de gérer les fichiers et les applications que l'on trouve normalement sur les ordinateurs personnels (les PC) et les portatifs. On peut aisément transférer de l'information et des données du réseau de l'organisation aux PDA, grâce à divers supports. Les PDAs proposent des fonctionnalités liées à un usage en situation de mobilité :

- Un agenda, pour l'organisation de l'emploi du temps ;
- Un gestionnaire de tâches faisant office d'aide-mémoire pour les tâches à effectuer ;
- Un carnet d'adresses (gestionnaires de contacts), permettant d'avoir à tout moment sous la main les coordonnées de ses contacts ;
- Un logiciel de messagerie ;
- Synchronisation avec un ordinateur fixe.

En plus de ces fonctions de base, de plus en plus les PDA proposent des outils multimédias avancés permettant de lire des vidéos (dans les différents formats, y compris le format DivX), des musiques (notamment au format mp3) ou des animations Flash.

Certains disposent en plus de fonctionnalités téléphoniques (GPRS), de communications sans-fil (Bluetooth, Wifi, infrarouge) ou de positionnement satellite (GPS).

Les principaux fabricants de PDAs sont Compaq – HP, Sony, Acer, Toshiba, Fujitsu et Palm. Les PDAs fabriqués par ce dernier utilisent un système d'exploitation propre appelé Palm OS ; alors que les autres utilisent pour la plupart un système d'exploitation basé sur Microsoft Windows CE. La version actuelle de Windows CE est 4.2, appelée Windows CE .Net.

Les téléphones portables de dernière génération (dit Smartphones) proposent des fonctionnalités les rapprochant de plus en plus des PDAs (gestion des rendez-vous, synchronisation avec un ordinateur fixe...). Le fait qu'ils utilisent de plus un système d'exploitation dérivé de Windows CE tend à faire disparaître la frontière existante entre les PDAs et ces dispositifs.

### 4.1.1 Évolution des PDAs

Originellement appuyés sur le principe d'une calculatrice évoluée, les assistants personnels servent d'agenda, de carnet d'adresses et de bloc-notes. On les dote de clavier, avec des petites touches, ou d'écran tactile, associé alors à un stylet.



Selon la définition qu'on leur donne, les premiers ordinateurs de poche (qui n'en portaient donc pas encore le nom) sont :

- Le Wizard OZ-7000 de Sharp(1988)
- Le Portfolio d'Atari (1989)
- Le Refalo de Kyocera (1990)
- Le Series 3 de Psion (1991)

La première utilisation publique du terme PDA remonte au 7 janvier 1992, lors du Consumer Electronics Show à Las Vegas où John Sculley présenta le Newton d'Apple le pionnier des assistants personnels. Puis deux modèles se sont imposés grâce aux fonctions de synchronisation avec un ordinateur de bureau: le Palm Pilot, commercialisé dès 1996 par US Robotics, et le Psion. Aujourd'hui de nombreuses marques proposent des assistants personnels très sophistiqués, pouvant allier multimédia et accès Internet sans fil : Compaq, Hewlett-Packard, Toshiba, Sagem, etc.

Progressivement les avancées technologiques permettent aux PDA de combiner, dans un volume réduit, les principales fonctions de la bureautique, du multimédia, de l'Internet, de la géo-localisation et de la téléphonie. Assez vite, les utilisateurs ont pu synchroniser leurs données avec des ordinateurs personnels via des câbles. Puis les capacités sans fil, au début limitées à l'infrarouge, ont été étendues pour accéder à différents types de réseaux via les technologies sans fil ou la téléphonie mobile numérique.

Comme pour les autres ordinateurs, toutes ces applications s'appuient sur un système d'exploitation qui permet la standardisation de leur fonctionnement et de leur développement.

#### 4.1.2 Les « PDA » industriels

L'usage des PDA (Personal Digital Assistants, ou assistants numériques personnels) s'est démocratisé. On les voit aujourd'hui dans tous les domaines. La baisse de prix des composants électroniques, l'augmentation des performances, et la généralisation des outils informatiques mobiles ont élargi le spectre de ses utilisations. Mais les PDA ne sont pas pour autant l'outil "universel". Il faut considérer les PDA comme étant l'"une" des solutions mobiles disponibles sur le marché. En effet, dans le domaine des équipements industriels mobiles, les PC portables durcis (également appelés Notebooks durcis), les tablettes PC et autres terminaux mobiles ont tous leur place. Dans la plupart des cas, on optera pour l'un ou l'autre de ces formats selon le nombre de déplacements à effectuer. Mais aussi en fonction du risque de chutes et de détériorations.

Les collaborateurs ont l'habitude d'utiliser des ordinateurs portables, et ils savent en prendre soin, ce qui n'est pas forcément le cas avec les terminaux mobiles et les PDA. Les petits équipements sont donc davantage exposés aux chocs, mais dans le même temps il est plus facile pour les constructeurs de "durcir" un petit produit. Les PDA figurent donc parmi les équipements d'informatique industrielle les plus robustes. Mais ce n'est pas là leur seul point fort. D'abord parce que ni les PC portables ni les terminaux mobiles ne peuvent proposer des démarrages instantanés comme les PDA. Ensuite, parce qu'ils offrent des caractéristiques d'autonomie supérieures.

Enfin, parce que d'importants efforts de conception sont mis en œuvre par les constructeurs afin que les PDA industriels ne génèrent pas de perturbations électromagnétiques (contrairement aux autres équipements communicants). Les PDA modernes présentent en effet d'intéressantes caractéristiques de compatibilité électromagnétique.

L'autre aspect primordial de la démocratisation de ces PDA industriels est la connectivité. Qu'il s'agisse du contrôle de production, de la maintenance ou de la logistique, la communication en temps réel est de mise, par leurs possibilités de connectivité sans fil multi protocole, les PDA industriels se démarquent des autres outils informatiques mobiles.

#### 4.1.3 Critère de choix

Les critères du choix du PDA dépendent de l'application et l'environnement d'utilisation on peut citer :

- La robustesse des équipements c'est en fonction du risque de chutes et de détériorations et l'exposition aux chocs.
- L'autonomie du PDA est fonction des caractéristiques de la batterie.



- Fonctionnalités de connectivité (GPRS<sup>[12]</sup>, Bluetooth, Wifi, infrarouge) on parle aussi de synchronisation de données avec d'autres terminaux.
- Le choix du système d'exploitation dépend des logiciels utilisés dans l'application ainsi le langage de programmation.
- poids et dimensions : Le PDA est destiné à être emporté partout et doit donc tenir dans la main ou la poche. Ses dimensions et son poids doivent donc être choisis les plus petits possible, en gardant à l'esprit le besoin d'ergonomie et de surface d'affichage.

### • Choix de l'écran

L'écran est souvent l'élément le plus délicat du PDA. Il peut s'avérer fragile, même sur des modèles dits "durcis". Si les conditions d'utilisations sont difficiles, il est judicieux d'opter pour un appareil avec une dalle rigide, en verre plutôt que des dalles souples, en plastique, qui équipent la plupart des PDA. La dalle en verre résiste aux chocs et est moins sujette à l'usure (qui rend la surface de l'écran de moins en moins transparente au fil du temps).

Les dimensions d'écrans s'échelonnent entre 3,5 et 4''. En termes de résolution, la majorité des PDA du marché disposent d'écrans QVGA, c'est-à-dire 320 x 240 (ou plutôt 240 x 320, puisqu'ils sont en mode "portrait"). Les modèles ayant une résolution VGA (640 x 480) se font encore rares, mais quelques constructeurs s'efforcent tout de même d'en proposer. Ils sont utiles pour toutes les applications qui nécessitent l'affichage de courbes ou de graphiques.

Enfin, concernant l'éclairage et la lisibilité de l'écran, différentes technologies sont disponibles : écrans LCD avec rétro-éclairage par néon (technologie classique) ou encore un rétro-éclairage par LED. Cette nouvelle technologie présente de nombreux avantages : une luminosité fortement augmentée (avec un réglage plus fin), une consommation considérablement réduite, la suppression des perturbations électromagnétiques au démarrage, et enfin la possibilité de désactiver certaines zones afin d'obtenir des noirs très profonds. Ces deux technologies néon et LED peuvent être combinées à des écrans translatifs. Il s'agit de dalles spéciales qui utilisent les rayons du soleil pour augmenter l'intensité du rétroéclairage et évitent les reflets rencontrés sur un écran classique.

### • Système d'exploitation

Les PDA possèdent des systèmes d'exploitation dont la définition est adaptée à la résolution d'affichage de l'écran et dont les fonctionnalités correspondent aux caractéristiques de ce type d'appareil.

Il existe plusieurs systèmes d'exploitation pour PDA, correspondant la plupart du temps à des types de PDA différents et portés par des constructeurs différents, au même titre qu'il existe des ordinateurs Mac et PC.

Les deux principaux systèmes sont:

- PalmOS, promu par la société Palm.
- Windows Mobile ou Pocket PC (anciennement Windows CE), promu par la société Microsoft.

Ces deux systèmes possèdent à peu près les mêmes caractéristiques et les mêmes fonctionnalités avec une prise en main différentes mais surtout des applications incompatibles entre les deux systèmes.

Il est à noter qu'il existe des systèmes d'exploitation Linux développés spécifiquement pour les deux types de machines.

### • Le processeur

Comme un ordinateur, un PDA nécessite un processeur pour fonctionner. La plupart des modèles actuels en possèdent un cadencé à 300 Mhz voire 400 Mhz ce qui va permettre à l'agent d'utiliser son appareil sans que celui-ci ne manifeste de ralentissement. Plus la vitesse du processeur sera élevée, plus le PDA sera réactif. Cette caractéristique est importante à prendre en compte dans l'utilisation d'un réseau de connexion.

### • La mémoire embarquée : ROM et RAM

---

<sup>12</sup>GPRS : General Packet Radio Service est une norme pour la téléphonie mobile dérivée du GSM permettant un débit de données plus élevé





Autre point important à prendre en compte dans le choix de son PDA : la mémoire. On parle de mémoire fixe ou ROM (Read Only Memory) pour de la mémoire qui peut seulement être lue. Sa taille est en moyenne de 64 Mo et elle supporte le système d'exploitation de l'assistant personnel. La RAM ou mémoire vive peut quant à elle être lue et modifiée. Sa taille varie en moyenne selon les modèles de 32 à 128 Mo et c'est elle qui détermine réellement les capacités de l'assistant personnel. Plus il disposera de mémoire, plus nous pourrons y entrer ou recevoir des données (adresses, agenda, notes, textes, livres électroniques), mais aussi y installer des applications supplémentaires, et surtout nous pourrons utiliser plusieurs programmes en même temps.

#### • **Emplacement pour carte mémoire**

Les PDA disposent également de ports pour cartes mémoire. La plus courante est la carte de type SD (Secure Digital) dont la capacité peut aller jusqu'à 4 Go. Les PDA récents sont également compatibles avec la norme MMC (MultiMedia Card) qui est de format identique.

#### • **Connectivité**

Infrarouge (IrDA), Bluetooth, Wifi, de nombreux assistants personnels actuels offrent une connectivité complète. Les premiers prix offrent généralement l'infrarouge et le Bluetooth, le Wifi étant proposé sur des appareils plus "haut-de-gamme".

Cependant si notre PDA n'est pas Wifi, il existe une solution pour le rendre compatible à moindre coût : la carte mémoire Wifi. Légèrement plus grande qu'une carte SD classique, celle-ci fait office d'antenne Wifi (802.11b/g). Certains modèles offrent même en plus d'une compatibilité Wifi, un espace de stockage (en moyenne de 512 Mo).

### 4.1.4 Acquisition des données

Certains PDA proposent des lectures de code-barres, en une ou deux dimensions, d'autres l'identification radiofréquence (RFID). Et quelques modèles supportent les deux techniques. On trouve même des caméras CCD pour réaliser des clichés, pour par exemple mémoriser l'état d'un organe défectueux de la machine.

La lecture des données portées par le code-barres ou les étiquettes RFID permet de faire de la gestion rapide de stocks, de suivre un produit tout au long d'une chaîne de production, les PDA servent à scanner les codes-barres en deux dimensions présentes sur les machines.



**Figure 4.2 : PDA avec lecteur de données**

## 4.2 Routeur WIFI

Un routeur est un élément intermédiaire dans un réseau informatique assurant le routage des paquets entre réseaux indépendants. Ce routage est réalisé selon un ensemble de règles formant la table de routage.





Le routage traite les adresses IP en fonction de leur adresse réseau définie par le masque de sous-réseaux et les redirige selon l'algorithme de routage et sa table associée.

Le Wifi est une technologie permettant de créer des réseaux informatiques sans fil (Wireless), sa portée varie d'un appareil à l'autre entre quelques dizaines de mètres à plusieurs centaines de mètres, ce qui en fait une technologie de premier choix pour le réseau domestique avec connexion internet. Le wifi est de plus en plus utilisé par divers matériels informatiques, ordinateurs, organiseurs (PDA), consoles de jeux portables voire des imprimantes utilisent elles aussi le Wifi pour simplifier leur connexion.

### • Sécurité

C'est le point le plus important, Il est facile de monter un réseau, mais il ne faut pas oublier de le sécuriser, il y'a diverses possibilités par exemple : WEP, WPA, MAC, etc.

Le SSID ou "nom du réseau" identifie le réseau, donne un nom pour le différencier des autres. On sera les seuls à le connaître et c'est tout de suite plus difficile de se connecter à notre réseau.

Le WEP/WPA, ce sont deux possibilités d'encrypter les données qui circulent sur le réseau. Le problème du WIFI est qu'on n'a aucun contrôle de médium sur lequel circulent les données contrairement aux réseaux filaires. Donc on ne sait pas qui est à l'écoute. Encrypter les données permet d'en assurer la confidentialité. Cela se fait à l'aide de ce que l'on appelle une clef. Cette clef permet également de sécuriser l'accès au réseau car, si on ne la connaît pas, impossible de communiquer, donc incapable de lire les trames et/ou d'en envoyer au bon format.

WEP est mis par défaut sur la LiveBox par exemple, mais ce type de cryptage consomme plus de ressources et est très facilement craquable (notamment sous Linux).

WPA est plus performant et beaucoup moins facilement craquable. Pour plus de sécurité, il est conseillé de changer les codes tous les mois.

Le filtrage d'adresse MAC, sur chaque carte réseau possède un identifiant unique: l'adresse MAC, Le routeur wifi permet généralement de créer une liste des adresses MAC des cartes réseau qu'on autorise à se connecter à notre réseau. C'est un filtre efficace mais qui là aussi peut être contourné.

Le DHCP (Dynamic Host Configuration Protocole) est un mécanisme qui permet d'affecter automatiquement les valeurs nécessaires à la communication sur le réseau (adresse IP, masque de sous-réseau, passerelle, DNS). C'est pratique mais c'est aussi très pratique pour un pirate, qui n'aura pas à deviner la configuration de notre sous-réseau.

*Il est important de remarquer que chacun des points peut être contourné d'une façon ou d'une autre. En fait, c'est la combinaison de tous ces points qui va faire de notre réseau un réseau bien sécurisé. Il ne faut pas baser la sécurité de notre réseau sur un seul de ces éléments. Le minimum à conseiller étant le WEP et un filtrage par adresse MAC.*

Les fonctionnalités WIFI d'un routeur peuvent aller du très simple au beaucoup plus complexe. Le minimum syndical est actuellement une compatibilité avec la norme 802.11n et une protection par clé WPA. La présence d'une fonction permettant la création rapide d'un réseau invité ou un chiffrement plus abouti est réservée à des modèles plus onéreux.





**Figure 4.3 : Exemple de router**

Enfin N'oublions pas de prendre en compte l'épaisseur des murs ou la surface à couvrir avant d'équiper et si le WI-FI ne convient pas au bâtiment de l'entreprise.

*La compatibilité avec la dernière norme WIFI 802.11n doit être assurée*

### 4.3 Prix Comparatives des matériaux

**Prix Routeur Wifi** <sup>[13]</sup>:

Nom du Router	Caractéristiques	Prix(DH)
Cisco Small Business PRO SA520 	<ul style="list-style-type: none"> <li>• Débit du firewall : 200 Mbps</li> <li>• Débit du VPN : 65 Mbps</li> <li>• Connexions VPN : 50</li> <li>• Commutateur Ethernet 4 ports 10/100/1000</li> <li>• Possibilité de créer 16 VLAN</li> <li>• Accès SSL simultanés : 2 (extensible à 25)</li> <li>• 2 ports WAN 10/100/1000 pour répartition de charge entre 2 accès Internet.</li> <li>• 1 port WAN utilisable en DMZ</li> </ul>	À partir de 3500 DH
Dlink Routeur DWR-113 	D-Link DWR-113 3G routeur Wi-Fi est un routeur Wifi 150N poche, il prend en charge dongle 3G jusqu'à HSPA +. Les caractéristiques du routeur NAT comprend, Routage, Firewall, VPN pass-through, WPS, Auto-3G-Dial-Up Connection de sauvegarde, serveur DHCP et ainsi de suite. Il est géré facilement par interface Web et est compatible UPnP. Pare-feu SPI, filtrage des paquets et de contrôle d'accès est intégré dans le produit. Il soutient des approches différentes de connexion à l'ISP tels que l'IP statique, IP dynamique (client DHCP), PPTP, L2TP, PPPoE et 3G. Il est également conforme à la norme IEEE 802.11n/b/g standard pour WiFi.	590.00
SMCWBR14S-3GN	Conçu pour les particuliers et les petites entreprises, le routeur sans-fil Barricade™ N (SMCWBR14-3GN) est une solution réseau tout-en-un pour le raccordement et le partage des	1 144.00

<sup>13</sup><http://shopping.cherchons.com/dossier/routeur-wi-fi-cisco.html>



	<p>connexions Internet mobile 3G, afin de permettre un accès Internet dans les zones blanches sans connexion DSL/câble ou en déplacement. Il prend en charge un modem USB sans-fil 3G (non fourni) pour la connectivité WAN primaire ou secondaire, avec commutation automatique pour garantir une connexion permanente en cas de défaillance des services Internet DSL/câble. Doté d'un point d'accès sans-fil 802.11n-draft 2.0 intégré, le routeur est capable de transférer des données à une vitesse jusqu'à 300 Mbps, soit jusqu'à 25 fois le débit des périphériques 802.11b standard, tout en restant compatible avec les équipements 802.11 b/g classiques. La sécurité devenant un critère de choix déterminant, le routeur prend en charge les toutes dernières normes de cryptage sans-fil WPA2, ainsi que la technologie WPS (Wi-Fi Protected Setup™) grâce à un bouton-poussoir ou une configuration par cavaliers.</p>	
<p>Cisco Linksys 4200 v2</p> 	<p>Le « Linksys 4200 v2 Maximum Performance Dual-Band N900 » bénéficie effectivement d'une bande passante sans fil totale portée à 900 Mbps, en réponse à TRENDnet à qui l'on doit le premier routeur du genre. Le point d'accès Wi-Fi N de ce routeur repose pour ce faire sur des antennes dites « 3x3 spatial stream » à la fois sur la bande des 2,4 GHz et sur celle des 5,0 GHz, soit un total de 6 antennes, ce qui permet d'atteindre 450 Mbps par bande, et donc 900 Mbps avec un équipement prenant en charge le double bande en simultanée.</p> <p>En plus du concentrateur Ethernet Gigabit à quatre ports et du port pour modem, on retrouve le port USB 2.0 qui permet de partager une imprimante et/ou le contenu d'un disque dur externe avec les ordinateurs et/ou les équipements multimédias (DLNA) connectés, mais le SoC cadencé à seulement 480 MHz qui bridait considérablement son prédécesseur a judicieusement été remplacé par un nouveau cadencé à 1,2 GHz !</p>	<p>À partir de 2000 DH</p>

Tableau 4.1 : Prix de quelques routeurs selon ses caractéristiques

Prix PDA<sup>[14]</sup> :

Nom du PDA	Caractéristiques	Prix(DH)
Intermec CK3	• Type de lecteur: <b>Imageur linéaire, Imageur</b>	À partie de

<sup>14</sup><http://www.directindustry.com/industrial-manufacturer/PDA-76404.html>

	<p><b>2D</b>Mémoire: <b>128 MB RAM + 256 MB flash</b></p> <ul style="list-style-type: none"> <li>• Vitesse de lecture: <b>100 scans/sec</b></li> <li>• Système d'exploitation: <b>Windows CE 5.0 ou Windows mobile 6.1</b></li> <li>• Connectivité : <b>802.11a/b/g, Bluetooth 2.0 EDR</b></li> <li>• Dimensions: <b>61 x 84 x 21mm</b></li> <li>• inclus: <b>Styler, Dragonne</b></li> </ul>	<p>6000 DH jusqu'à 19200 DH</p>
<p>Nautiz X7 Rugged PDA</p> 	<ul style="list-style-type: none"> <li>• Type de lecteur: <b>Laser 1D ou 2D</b></li> <li>• Mémoire: <b>128 MB RAM + 256 MB flash</b></li> <li>• Vitesse de lecture: <b>100 scans/sec</b></li> <li>• Système d'exploitation : <b>Windows mobile 6.1</b></li> <li>• Connectivité : <b>Wifi 802.11g/b, Bluetooth</b></li> <li>• Dimensions: <b>179 x 97 x 37mm</b></li> <li>• inclus: <b>sangle, styler, batterie, câble USB, RFID, Lecteur code barre</b></li> </ul>	<p>À partie de 13000 DH jusqu'à 20000 DH</p>
<p>Terminal PDA Memor DATALOGIC</p> 	<ul style="list-style-type: none"> <li>• Type de lecteur: <b>Laser 1D ou 2D</b></li> <li>• Mémoire: <b>128 MB RAM + 256 MB flash</b></li> <li>• Vitesse de lecture: <b>100 scans/sec</b></li> <li>• Système d'exploitation: <b>Windows CE 5.0 ou Windows mobile 6.1</b></li> <li>• Connectivité : <b>Wifi 802.11g/b, Bluetooth</b></li> <li>• Dimensions: <b>152 x 55 x 40 mm</b></li> <li>• inclus: <b>sangle, styler, batterie, câble USB</b></li> </ul>	<p>À partie de 7000 DH jusqu'à 10200 DH</p>

*Tableau 4.2 : Prix de quelque PDA selon ses caractéristiques*

### Conclusion :

Nous avons présenté dans ce chapitre quelques caractéristiques de PDA, leur évolution et les critères de choix utilisés de façon générale. Ensuite, nous avons abordé des notions sur la connexion sans fil Wifi, ses caractéristiques et les méthodes de sécurisation possible pour ce type de réseaux.



## CONCLUSION GENERALE

Ce stage a parfaitement répondu à nos attentes, il nous a permis de découvrir le domaine du travail au sein d'une entreprise. De s'intégrer à un univers que nous ne connaissons finalement que très peu mais pour lequel nous portons un immense intérêt.

Les cours de Réseaux et programmation informatique qui nous ont été enseignés ces deux années de formation ont eu une importance capitale au bon déroulement du stage. On a revu plusieurs notions vues en cours mais de manière plus approfondie comme le langage J2ME, J2EE et J2SE. Nos connaissances en base de données nous ont été également très utiles car nous avons été souvent confrontés à enregistrer et consulter des données et aussi lier la BD avec les interfaces graphiques coté PC et coté PDA.

Son principal atout a été sa richesse ; il nous a permis d'aborder plusieurs thèmes : création d'une MIDlet, d'une interface graphique et d'une base de données. Nous avons en effet essayé quelques moments de doute, notamment lors du portage de l'application de l'émulateur au PDA, de l'envoi des données du mobile à la Servlet, de la configuration du port du serveur « GlassFish3+ » et la limitation du nombre de session possible à ouvrir entre le logiciel ORACLE et les deux applications.

L'implémentation de l'application réalisée va faciliter le service VA, on pourra par la suite généraliser ce système aux différents types de visite, cette généralisation est faisable car nous avons laissé la possibilité d'ajouter différent intervenant au visite. D'autre part nous pouvons nationaliser le projet en utilisant les technologies Web et aussi la technologie de troisième génération comme la GPRS, de façon qu'on lie les différents pôles de maintenance de Maroc en partageant l'information en même temps dans un réseau national bien sécurisé.

Les locomotives peuvent être identifiés par un lecteur de code barre intégré dans le PDA.

Dans ce stage nous avons appris à être plus autonome, à configurer et mettre en place notre environnement de travail. Nous avons utilisé de nouvelles technologies, qui suscitent un grand engouement, nous avons appris à ne pas nous décourager et à contourner les difficultés.

## BIBLIOGRAPHIES

### • Livre

- Michel Divay , JAVA et la programmation objet. Dunod, Edition 2002,
- James Keogh , J2ME: The Complete Reference.McGraw-Hill, Edition 2003.
- SingLi ,Jonathan Knudsen, Beginning J2ME: From Novice to Professional.Apress, Edition 2005.
- [Pierre-Yves Saumont](#) , [Antoine Mirecourt](#) , Le guide du développeur JAVA 2.Eyrolles, Edition 2003
- [Christian Soutou](#) , UML 2 pour les bases de données. Eyrolles, Edition 2007.
- [Christian Soutou](#) , SQL pour ORACLE Applications avec JAVA, PHP et XML.Eyrolles, Edition 2008.
- Emmanuel Puybaret, Les cahiers du programmeur : JAVA 1.4 et 5.0.Eyrolles, Edition 2004.
- Saïd Najah, Cours Développement des applications mobiles sous J2ME, 2010.



- Chris Zeis, Chris Ruel, Michael Wessler, ORACLE 11g ForDummies.For Dummies, Edition 2009.

- Site web

<http://www.tuto-dba-ORACLE.com/gerer-instance-ORACLE/page-3>

<http://www.digilife.be/quickreferences/pt/J2ME%20step%20by%20step.pdf>

[http://www.siteduzero.com/tuto\\_pdf.php?vrsid=10601](http://www.siteduzero.com/tuto_pdf.php?vrsid=10601)

<http://ivmad.free.fr/cours/IC4-AI-JAVA.pdf>

[http://jmdoudoux.ftp-developpez.com/cours/developpons/JAVA/dej\\_1\\_00.pdf](http://jmdoudoux.ftp-developpez.com/cours/developpons/JAVA/dej_1_00.pdf)

<http://notes.corewebprogramming.com/student/JAVA-Basic-Syntax.pdf>

<http://www.atpeaz.com/index.php/2010/fixing-the-ora-12519-tnsno-appropriate-service-handler-found-error/#comment-22>

[http://deptinfo.unice.fr/twiki/pub/Minfo05/AdministrationBasesDeDonnees/AdminBDORACLE\\_x4.pdf](http://deptinfo.unice.fr/twiki/pub/Minfo05/AdministrationBasesDeDonnees/AdminBDORACLE_x4.pdf)

<http://jmvanel.free.fr/JAVA/formation-ird3/jfreechart.pdf>

<http://www2.lifl.fr/~seinturi/middleware/jsp.pdf>

<http://pagesperso-systeme.lip6.fr/Gael.Thomas/cours/mdoc/cours/mdoc2-servlet-jsp.pdf>

<http://deptinfo.unice.fr/twiki/pub/Minfo03/ServletEtXml/50-JAVA-servlet-jsp.pdf>

<http://www.ncottin.net/download/ServletsJSP.pdf>

[http://www.siteduzero.com/tuto\\_pdf.php?vrsid=112219](http://www.siteduzero.com/tuto_pdf.php?vrsid=112219)

<ftp://87.98.130.175/grisha/JAVAsched.pdf>

## ANNEXES

### Annexe I : Ajout de serveur « Glass Fish 3+ » au Netbeans

Pour ajouter le serveur libre « Glass Fish 3+ », dans l'IDE Netbeans, On clique Outils/serveurs puis on suit les instructions selon les figures ci-dessous :

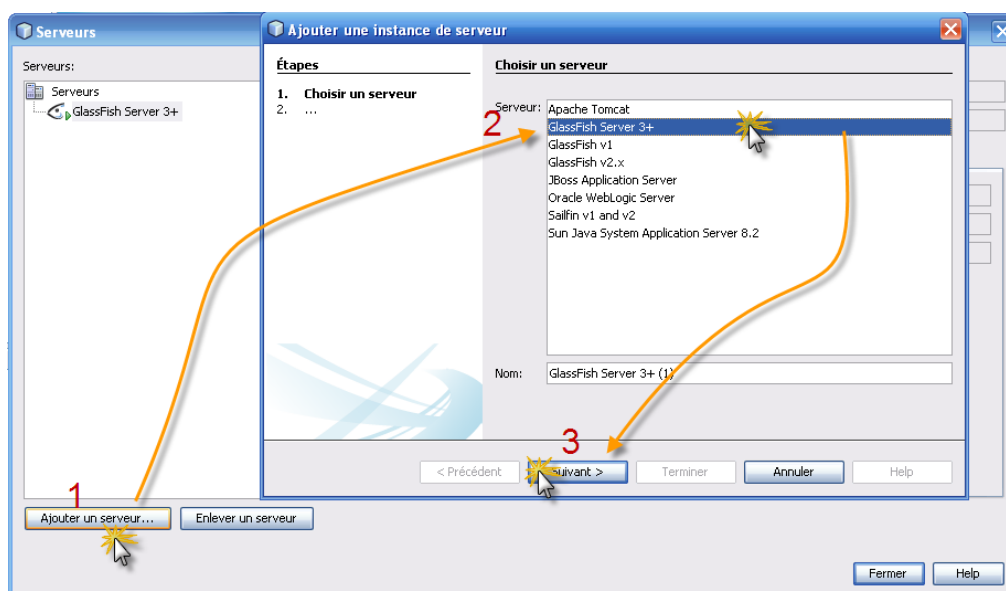


Figure I.1 : Étape 1 d'ajout du serveur GlassFish3+

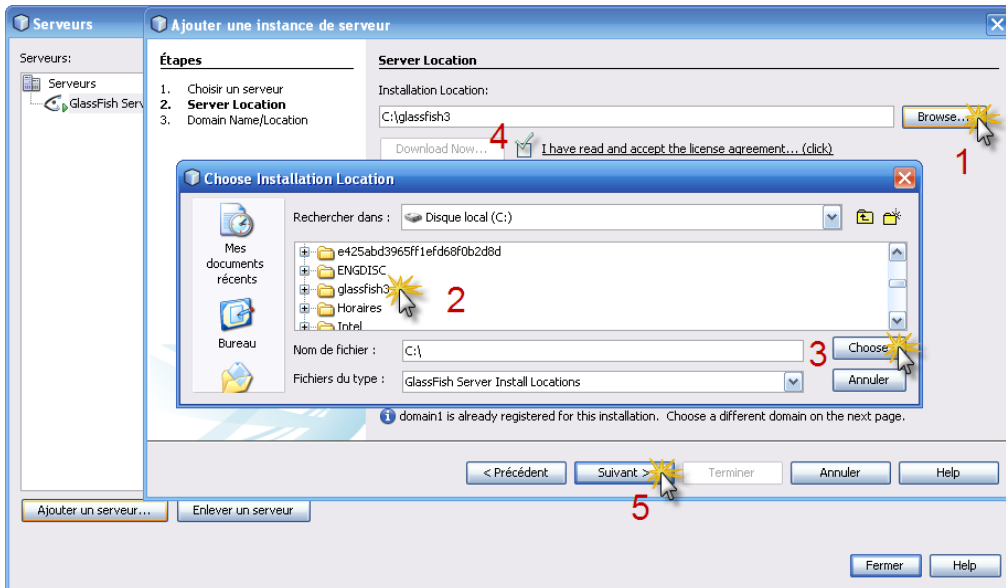


Figure I.2 : Étape 2 d'ajout du serveur GlassFish3+

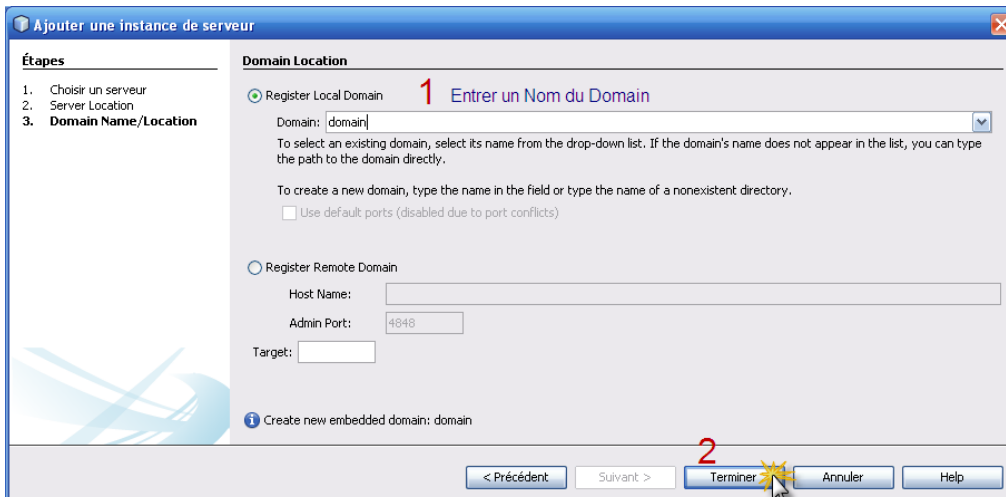


Figure I.3 : Étape 3 d'ajout du serveur GlassFish3+





## Annexe II : Changement de port du serveur

Lors de l'installation du JAVA EE 6 SDK et précisément lors de l'installation du « GlassFish 3+ », elle se bloque dans la configuration de port du serveur disant que le port par défaut 8080 est déjà utilisé par un autre programme, en effet ORACLE Xe réserve le port 8080 par défaut lors de son installation au début, pour cela nous avons pensé au changement du port d'ORACLE pour que l'installation se poursuive normalement et que le serveur « GlassFish 3+ » prend son port usuel

En premier on doit vérifier les ports libres de notre machine, en lançant la MsDos.

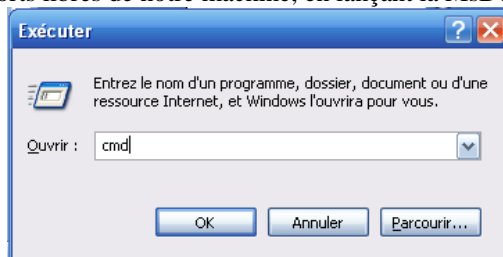


Figure II.1 : Exécuter les commandes

En utilisant la commande « netstat -anvo » ou tout simplement « netstat -a » si nous ne voulons pas plus de détails pour afficher tous les ports actifs dans notre machine.

UDP	0.0.0.0:445	*:*	4
UDP	0.0.0.0:5000	*:*	1568
UDP	0.0.0.0:1030	*:*	1856
UDP	0.0.0.0:4500	*:*	1568
UDP	127.0.0.1:123	*:*	1856
UDP	127.0.0.1:1031	*:*	1856
UDP	127.0.0.1:1144	*:*	4756
UDP	127.0.0.1:1900	*:*	180

Figure II.2 : Les ports libres de la machine

Ensuite on lance « SQL Command Line » d'ORACLE, on se connecte avec notre session d'ORACLE et puis on exécute la commande suivante :

`exec dbms_xdb.sethttpport('port libre')` en remplaçant 'port libre' par l'un des ports libres affiché dans la figure II.2.

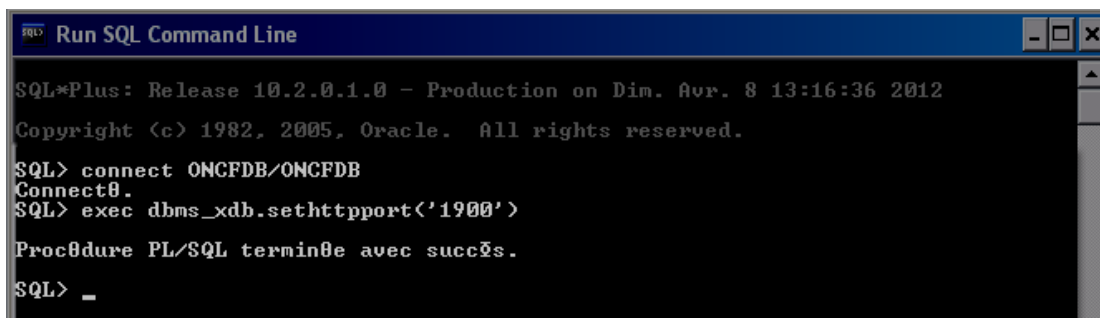
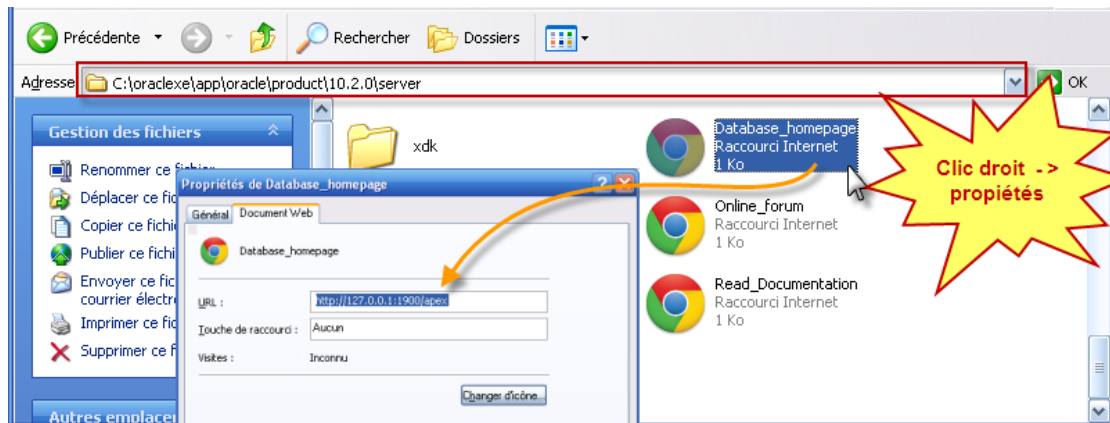


Figure II.3 : Changement du port

Redémarrant notre machine ; le port d'ORACLE est finalement changé au nouveau port (dans notre cas c'est le port 1900).





**Figure II.4 : Changement d'URL**

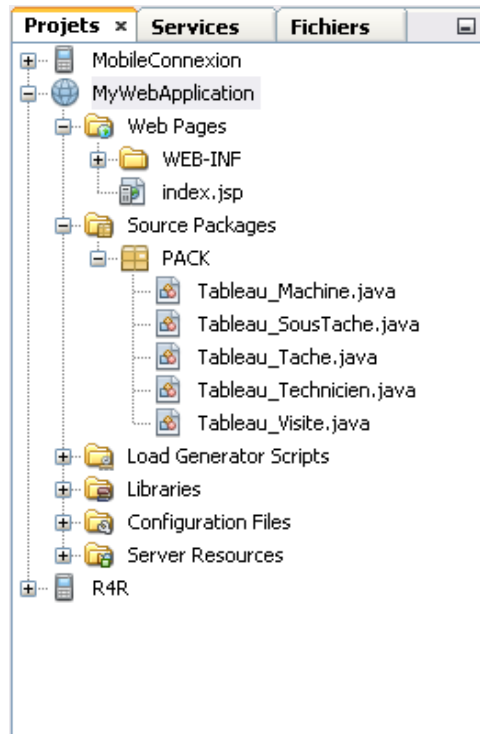
Remarque : On doit modifier l'URL de la page d'accueil d'ORACLE à l'URL suivant <http://127.0.0.1:1900/apex>.

Après on peut poursuivre l'installation du JAVA EE 6 SDK normalement et le serveur prend le port 8080 par défaut.

## Annexe III: Exemple de la création de Servlet

Cet exemple permet d'afficher les données d'une base de données dans une page HTML après une assure de connexion entre la servlet et la base de données à l'aide du driver JDBC d'ORACLE.

Cette figure ci-dessous montre les fichiers d'une application web, dont on trouve principalement les fichiers Servlet et JSP.



**Figure III.1 : Les composants du projet MyApplication**

Code Servlet du classe Tableau\_Machine.JAVA:



```
package PACK;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;
import java.util.ArrayList;
import java.util.*;
import java.io.*;

@WebServlet(name = "Tableau_Machine", urlPatterns = {"/Tableau_Machine"})
public class Tableau_Machine extends HttpServlet {
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    Connection connexion=null;
    Statement requete=null;
    ResultSet resultat=null;

try {
    out.println("<html><body>");
    out.println("<FORM ACTION=http://localhost:8080/MyWebApplication METHOD=GET>");
    out.println("<INPUT TYPE=SUBMIT VALUE=Retourner >");
    out.println("</FORM>");
    Class.forName("oracle.jdbc.driver.OracleDriver");
    connexion =DriverManager.getConnection("jdbc:oracle:thin:@th-
d3a2629a531d:1521:XE", "ONCFDB", "ONCFDB");
    requete = connexion.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPDATABLE);
    resultat = requete.executeQuery("SELECT * FROM MACHINE ORDER BY SERIELOCO ASC ");
    out.println("<table width=30% border=1><tr><td colspan=3 align=center bgcolor=#B0E0E6> TABLEAU
MACHINE </td></tr></table>");
    out.println("<table width=30% border=1><tr><td width=50% colspan=3 align=center
bgcolor=#FFA500>NUMERO LOCOMOTIVE </td><td width=50% colspan=3 align=center bgcolor=#FFA500> SERIE LOCOMOTIVE
</td></tr></table>");
        while (resultat.next()) {

            out.println("<table width=30% border=1><tr>");
            out.println("<td width=50% colspan=3 >");
            out.println(resultat.getString("LOCON"));
            out.println("</td>");
            out.println("<td width=50% colspan=3 >");
            out.println(resultat.getString("SERIELOCO"));
            out.println("</td>");

            out.println("</tr></table>");

            // Etc...
        }
    }
    catch (ClassNotFoundException e) {
        System.out.println("Erreur du chargement du Driver");
    }
    catch (SQLException e) {
        response.sendError(500, "Exception accés SGBD " + e);
    }
    finally {if (resultat != null) {
        try {
            resultat.close();
        }
        catch (SQLException e) {}
    }
    if (requete != null) {
        try {
            requete.close();
        }
        catch (SQLException e) {}
    }

    if (connexion != null) {
        try {
            connexion.close();
        } catch (SQLException e) {}
    }
    out.println("</body></html>");

    out.close();
}
}
```



Le code des autres class est similaire avec des petites modifications, le code JSP fait appel à tous ces servlets :

```
<!--
  Document   : index
  Created on : 2 avr. 2012, 12:30:03
  Author    : Tayache
-->

<%@page import="javax.persistence.Table"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>

    Choisir la table a Afficher:<p>
    <table >
      <tr><td><FORM ACTION="http://localhost:8080/MyWebApplication/Tableau_Tache"
        METHOD=GET>
        TACHE :      </td><td><INPUT TYPE=SUBMIT VALUE="TACHE"></td></tr><p>
      </FORM>
      <tr><td><FORM
        ACTION="http://localhost:8080/MyWebApplication/Tableau_SousTache" METHOD=GET>
        SOUS TACHE :</td><td><INPUT TYPE=SUBMIT VALUE="SOUS TACHE "></td></tr><p>
      </FORM>
      <tr><td><FORM
        ACTION="http://localhost:8080/MyWebApplication/Tableau_Technicien" METHOD=GET>
        TECHNICIEN :</td><td><INPUT TYPE=SUBMIT VALUE="TECHNICIEN "></td></tr><p>
      </FORM>
      <tr><td><FORM
        ACTION="http://localhost:8080/MyWebApplication/Tableau_Machine" METHOD=GET>
        MACHINE :   </td><td><INPUT TYPE=SUBMIT VALUE="MACHINE "></td></tr><p>
      </FORM>
      <tr><td><FORM
        ACTION="http://localhost:8080/MyWebApplication/Tableau_Visite" METHOD=GET>
        VISITE :    </td><td><INPUT TYPE=SUBMIT VALUE="VISITE
        "></td></tr><p>
      </FORM></table>
    </body>
  </html>
```

L'exécution avec le serveur « GlassFish3+ » nous donne :

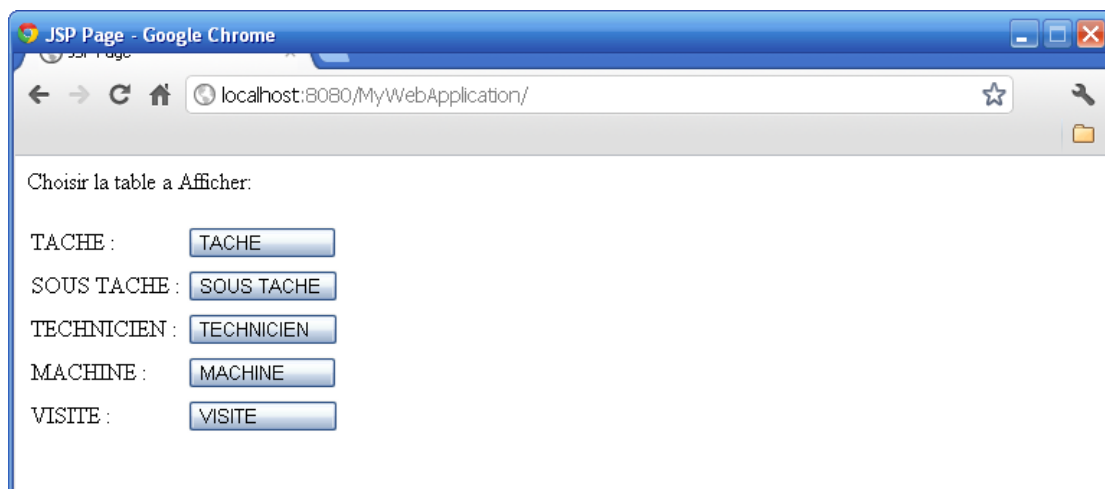


Figure III.2 : Résultat après exécution du projet

Les servlets après l'exécution :

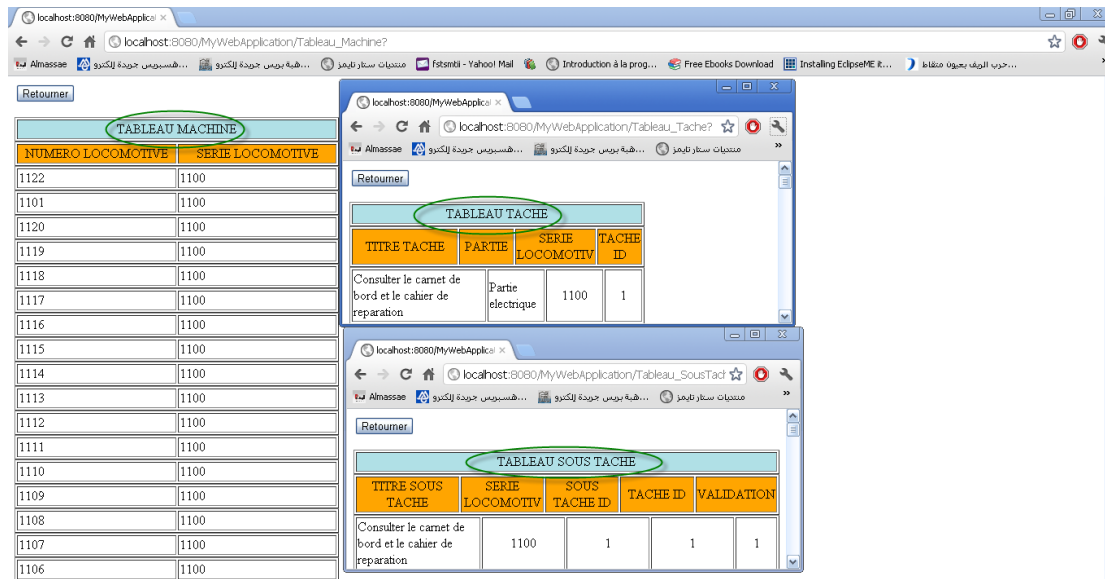


Figure III.3 : Les tableaux essuie de la base de données

## Annexe IV : Outils de conception

### IV.1 ORACLE

Il sera très difficile, pour ne pas dire impossible, à un autre éditeur de logiciels de trouver un nom mieux adapté à la gestion des données que celui d'« ORACLE ». Ce nom semble prédestiné à cet usage ; citons Le Petit Larousse :

ORACLE n.m. (lat. oraculum) ANTIQ. Réponse d'une divinité au fidèle qui la consultait ; divinité qui rendait cette réponse ; sanctuaire où cette réponse était rendue. LITT. Décision jugée infaillible et émanant d'une personne de grande autorité ; personne considérée comme infaillible.

ORACLE représenterait ainsi à la fois une réponse infaillible, un lieu où seraient rendues cette réponse et une divinité. Rien que ça ! Tout cela peut être en partie vérifié si votre conception est bien faite, vos données insérées cohérentes, vos requêtes et programmes bien écrits.

ORACLE Corporation, société américaine située en Californie, développe et commercialise un SGBD et un ensemble de produits de développement. ORACLE a des filiales dans un grand nombre de pays. La filiale française (ORACLE France) a été créée en 1986, elle est composée de cinq départements (marketing, commercial, avant-vente, conseil et formation).

Comme tout logiciel, ORACLE contient plusieurs interfaces graphiques aidant de façon remarquable le programmeur à créer une base de données plus sûrement et plus rapidement, avec d'autres choix liés à cette base de données comme l'affichage des tableaux et aussi la création des jointures sans se référer au code SQL ....etc.

### Composants ORACLE



- Base de données
- Tables
- Index
- Cluster
- Tablespace
- Schéma
- Séquence
- Procédure
- Fonction
- Package
- Profil
- View
- Répertoire
- Trigger
- Bibliothèque
- Utilisateur
- Synonyme
- etc.



## IV.2 NETBEANS

NetBeans, créé à l'initiative de Sun Microsystems (Noyau de Forte4J/SunOne), présente toutes les caractéristiques indispensables à un EDI de qualité, que ce soit pour développer en JAVA, Ruby, C/C++ ou même PHP.

De licence OpenSource, NetBeans permet de développer et déployer rapidement et gratuitement des applications graphiques Swing, des Applets, des JSP/Servlets, des architectures J2EE, dans un environnement fortement personnalisable.

L'EDI NetBeans repose sur un noyau robuste, la plateforme NetBeans, que vous pouvez également utiliser pour développer vos propres applications JAVA, et un système de plugins performant, qui permet vraiment d'avoir un EDI sur mesure.

Ainsi, si vous ne téléchargez que la déclinaison concernant le développement JAVA SE, votre EDI ne sera pas livré avec le support JAVA EE, EJB, Jsp, Tomcat, Glassfish, ...

À côté de la version complète de l'EDI NetBeans, vous avez différentes déclinaisons (6 au total, sans compter la version complète) qui se concentre sur une plateforme ou langage bien précis :

- JAVA ME
- JAVA (SE + ME + EE)
- Ruby
- C/C++
- PHP

NetBeans rajoute, à côté du support pour CVS et SubVersion, un support pour ClearCase, mais aussi pour Mercurial.

Il vous permet également de déployer les applications Web, non seulement vers Tomcat et Glassfish qui sont livrés avec le "Pack Web", mais aussi vers JBoss, WebSphere 6.1, WebLogic 9.

En plus d'avoir :

- un système de projets basé sur Ant<sup>[15]</sup>, permettant une meilleure indépendance et portabilité de vos travaux ;
- un refactoring avec des fonctionnalités comme le renommage, le déplacement et la recherche des déclarations pour classes, méthodes et packages ;
- le support de JAVA SE 5.0 (Tiger) avec les nouveaux mots clefs du langage, les annotations, les generics, etc. ;
- le support de JAVA SE 6.0 (Mustang) ;
- le support de développement d'applications Web avec des améliorations pour l'édition des JSP, la gestion serveur, le support des dernières versions de Tomcat, etc. ;
- le support pour JAVA ME et les standards MIDP 2.0 et CLDC 1.1, possibilités d'intégrations d'émulateurs extérieurs, etc. ;
- un débogueur de grande qualité ;
- une interface graphique améliorée.

La dernière version propose, entre autres :

- un éditeur grandement amélioré quant au support JAVA script, qu'il soit non seulement dans un fichier JAVA script, mais aussi dans une page html, jsp, rhtml...
- un éditeur visuel pour les fichiers CSS ;
- la gestion de bases de données grandement revue, avec un support direct pour JAVADB/Derby, MySql, PostgreSQL ;
- une fonctionnalité "deploy on save" pour les applications Web ;
- un support de GlassFish V3 Prelude ;
- un support pour importer des projets Eclipse.

Avec les Pack Mobility (JAVA ME), Web et JAVA EE, UML, SOA, Ruby, PHP, ..., vous vous retrouvez directement pleinement opérationnel une fois l'installation terminée<sup>[16]</sup>.

## Annexe V : Checklist

<sup>15</sup>Ant : "Another Neat Tool" il est principalement utilisé pour automatiser la construction de projets en langage JAVA, mais il peut être utilisé pour tout autre type d'automatisation dans n'importe quel langage.

<sup>16</sup><http://mac.developpez.com/telecharger/detail/id/156/NetBeans>



**Université Sidi Mohammed Ben Abdellah  
Faculté Des Sciences et Techniques Fès  
Département de Génie Electrique**

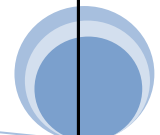


Exemple de checklist utilisé dans le service VA :

Journée du : ...../...../.....  
Train N° : .....

.Loco N° : .....

Désignation organe	B	I	S	Organes remplacés	Obse
1. <u>Consulter le carnet de bord et cahier de réparation</u> .....					
2. <u>Vérifier l'état général de la locomotive</u> (extérieur).....					
3. <u>Retirer la bande graphique</u> .....					
4. <u>Essais avant et après visite</u> .....					
• Eclairage intérieur et extérieur.....					
• Signalisation.....					
4-1.Montée et descente des pantos.....					
4-2.Essais des auxiliaires.....					
4-3.Essais à blanc ( la grille des contacteurs ).....					
4-4.Essais de traction ( AV- AR).....					
4-5.Essais de frein.....					
4-6.VA, ATS et RST .....					
4-7.Essais des essuies glaces.....					
4-8.Essais des avertisseurs sonores .....					
4-9 Remplissage réservoir de secours des pantos.....					
4-10 Débit des CP1 et CP2.....					
5. <u>visite des postes de conduite et appareillage électrique</u>					
• Manipulateurs.....					
• Appareils de mesure.....					
• Voyants de signalisation.....					
• Interrupteurs .....					
• VA, ATS et RST .....					
• Vitres latérales.....					
• Balais d'essuie-glaces.....					
• Serrures des portes.....					
• Sièges .....					
• Contacteurs électromagnétiques et électropneumatiques					
• Cheminées .....					
• Relais.....					
• Sectionneurs.....					
• Disjoncteurs.....					
• Groupe MG 1200, CV E1000, CVS E1100.....					
• Soufflets de ventilation.....					
• Convertisseurs.....					
• Arbres à cames.....					



• Coupleurs d'appareillage.....  
• Batterie (aspect, tension, niveau d'électrolyte)