

République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

*Option: Système d'Information et de Connaissances (S.I.C)*

# Thème

**Système de Gestion de Tracking GPS Orienté  
Transport Routier WinRah**

Réalisé par :

- ESSEGHIER ABDERRAHMANE

*Présenté le 02 Juillet 2023 devant le jury composé de MM.*

- BENAMAR ABDELKRIM (Président)
- BENMAHDI MERIEM BOUCHRA (Encadreur)
- KAZI TANI ADILA (Examineur)
- Bendahmane Mohammed Fawzi (Examineur)

# DÉDICACE

A mes parents

A mes frères et sœurs

A Mon Encadreuse

A tous ceux qui ont participé

Abderrahmane

## REMERCIEMENTS

Avant toute chose, je remercie Dieu le tout puissant qui m'a donné la santé, la force, la volonté et la patience pour terminer ce travail.

Je remercie vivement mon encadreuse **Madame BENMAHDI BOUCHRA**, Enseignante à l'université Abou Bakr Belkaid de Tlemcen, pour son soutien, sa disponibilité et sa patience tout au long de ce travail.

Je remercie mes Professeurs au département de l'informatique de l'université de Tlemcen, pour leur patience, leurs orientations et leurs précieux conseils.

Je tiens à remercier également l'encadrement du centre I2E à l'université de Tlemcen pour leur confiance.

Je tiens à remercier également les membres de jury.

Un grand merci à mes parents, qui m'ont toujours soutenu.

Je remercie tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'études.

# TABLE DES MATIÈRES

Introduction générale : .....	6
Chapitre I Généralités sur les systèmes de suivi.....	7
I.1 Introduction.....	8
I.2 Technologies de localisation.....	8
I.2.1 Localisation Via Réseau GSM : .....	8
I.2.2 Localisation Via GPS :.....	9
I.2.3 Localisation En double-Mode.....	10
I.3 Systèmes d'informations géographiques SIG : .....	11
I.3.1 Google Earth Pro : .....	11
I.3.2 ArcGis : .....	11
I.3.3 GRASS.....	11
I.4 Les systèmes de suivi des véhicules.....	12
I.4.1 Développement des systèmes de suivi des véhicules : .....	12
I.4.2 Etude des systèmes de suivi des véhicules : .....	13
I.4.3 fonctionnalités d'un système de suivi des véhicules : .....	13
I.4.4 Principal obstacle au déploiement du système Tracking : .....	14
I.4.5 Les types des systèmes de suivi : .....	14
a. Les OBD : .....	14
b. Le Tracking Point a Point.....	15
I.4.6 Le Tracking-as-service dans le marché Algerien : .....	16
Chapitre II Technologies, Outils ET Architectures.....	17
II.1 Introduction : .....	18
II.2 Technologies base de la plate-forme.....	18
II.2.1 Servlets : .....	18
a. Généralité sur les servlets : .....	18
b. Cycle de vie d'une Servlet : .....	19
II.2.2 JSP : .....	21
a. Balises et directives.....	21
b. Portée des objets.....	22
c. Les expressions.....	23

II.2.3 JSTL :	24
a. Vue d'ensemble sur la librairie JSTL :	24
b. Bases de code JSTL :	25
II.3 Outils de développement.....	29
II.3.1 Outils Matériels.....	29
a. Arduino :	30
b. Sim808.....	31
II.3.2 Outils Logiciels.....	32
a. IDEs :	32
b. Serveurs :	33
II.4 Modèles d'Architectures.....	33
II.4.1 DAO Pattern :	33
II.4.2 EJB.....	34
a. Types des EJB :	34
b. Session Bean.....	35
c. Cycle de vie Stateless Session Bean :	35
d. Cycle de vie Stateful Session Bean :	36
e. Le packaging :	36
II.4.3 Abstract Factory Design Pattern :	37
a. Factory Design Pattern :	37
b. Factory VS Abstract Factory :	38
Chapitre III Analyse et Conception du système.....	39
III.1 Introduction.....	40
III.2 Analyse du système :	40
III.2.1 Besoin fonctionnel.....	40
a. Le module application de bureau (ERP).....	40
b. Le module application :	40
c. Le module application Mobile :	41
III.2.2 Besoins non fonctionnels.....	41
a. Performance :	41
b. Sécurité :	41
c. Portabilité :	41
d. Ergonomie :	41

e. Modularité du code :.....	41
III.2.3 Identification des cas d'utilisation :.....	41
a. Client particulier.....	42
b. Client Professionnelle.....	42
c. Administrateur.....	42
d. Diagramme des cas d'utilisation.....	43
III.2.4 Descriptions de cas d'utilisations.....	43
a. Description du cas d'utilisation « S'authentifier ».....	43
b. Description du cas d'utilisation « Consultation Parcoure Tracker ».....	44
c. Description du cas d'utilisation « Enregistrer un Trackeur ».....	45
III.3 Conception du système « WinRah ».....	46
III.3.1 Diagramme de déploiement :.....	46
III.3.2 Diagrammes de Séquences :.....	47
a. Diagrammes de séquences pour une application de bureau :.....	47
b. Diagramme de séquence Cas client Web :.....	49
c. Diagramme de séquence Cas client Application Mobile :.....	50
III.3.3 Diagrammes de classes :.....	51
a. Diagrammes de classes Au Module Web.....	52
b. Diagrammes de classes Au Module Bureau :.....	53
III.4 Exploitation des modules de la plate-forme :.....	54
III.4.1 Exploitation de l'Application WEB :.....	54
III.4.2 Exploitation de l'Application Mobile :.....	57
III.4.3 Exploitation de l'Application de bureau :.....	58
Conclusion.....	60
Conclusion générale .....	61
Références Bibliographiques .....	62
Annexes .....	63
Liste de figures .....	75
Liste des tableaux .....	76
Liste des abréviations .....	77
Résumé.....	78

## **Introduction générale :**

Lors de ces dernières années, les systèmes de suivi des points en mouvement sont devenus de plus en plus courants dans les environnements de transport de marchandise. Généralement, les systèmes de suivi de transport contiennent des appareils compatibles GPS capables de fournir des données de géolocalisation pour un grand nombre de marchandises. La gestion de ces systèmes dépend de la disponibilité de la matière première, pour ainsi réussir la réalisation des produits finis d'outils et des équipements.

Ces systèmes peuvent optimiser les processus le transport de plusieurs manières. Par exemple, pour s'assurer que les entités de transport sont utilisées efficacement et que les produits transportés sont livrés à la bonne destination et dans temps. De plus, les données GPS peuvent être utilisées pour surveiller l'entretien mécanique des entités de transport et la réparation de l'équipement et ainsi éviter les interruptions et les temps d'arrêt imprévus. Les systèmes de suivi GPS sont idéals pour les entreprises qui ont un service de transport et qui cherchent à améliorer leurs services

Lors de la préparation de notre projet, nous avons étudié deux volets. Le premier concerne les processus de collecte des données de géolocalisation, spécialement dans le cadre du transport routier. Et le deuxième volet concerne l'exploration de ces données avec des solutions logiciels développés.

Le but de notre projet de fin d'études est la mise en place d'un système de géolocalisation de transport temps réel. Ce système est dédié à n'importe quel individu ou organisme qui cherche l'outil de supervision de l'état de son réseau de transport pour une prise de décision en temps réel. Ce manuscrit est organisé en quatre chapitre

# **Chapitre I**

## **Généralités sur les systèmes de suivi**

## **I.1 Introduction**

La donnée géographique d'une entité permet de connaître sa position. Pour détecter cette donnée, il faut un protocole bien déterminé. Les données géographiques sont primordiales pour la réalisation d'un système de suivi.

Ce chapitre est divisé en trois parties. La première concerne quelques technologies de localisation, la deuxième quelques systèmes d'information. Et dans la troisième partie nous allons présenter les systèmes de suivi de véhicules.

## **I.2 Technologies de localisation**

### **I.2.1 Localisation Via Réseau GSM :**

Un réseau cellulaire se compose de plusieurs stations de base, avec des antennes respectives qui connecte les téléphones portables à proximité. Dans un réseau GSM, les stations de base sont appelées BTS (Base Transceiver Stations). Certaines BTS sont mutualisés et sont contrôlé par un BSC (contrôleur de station de base). Le niveau hiérarchique supérieure suivant est le Centre de commutation Mobile (MSC) qui achemine les appels entre les MS et entre les MS et la ligne fixe téléphones réseau. La région contrôlée par un MSC est souvent identique à la région dite zone de localisation (LA) qui joue un rôle important pour les techniques de positionnement. Le nombre de BSC contrôlés par un MSC ainsi que le nombre de BTS contrôlés par un BSC peut varier d'un fournisseur à un autre et d'une région à une autre. Les données peuvent être acquises de différentes interfaces. Entre le BTS et le BSC une interface est définie appelés Abis. Les données disponibles dans l'une de ces interfaces, sont affectés à un seul BSC ; le niveau hiérarchique suivant est « L'interface A » qui est définie entre le BSC et MSC. Toutes les données du BSC affecté à un MSC peuvent être source de l'une de ces interfaces. Une information disponible dans le réseau GSM est l'identifiant de la cellule (numéro unique attribué à une cellule respectivement à une BTS).

Lorsqu'un MS passe d'une cellule (BTS) à une autre, les deux identifiants sont disponibles. On parle alors d'une passation. Chaque LA est définie par un LA-ID unique. Si un MS change de position d'une zone de localisation à une autre, ces informations seront disponibles au sein du réseau. L'identifiant LA peut être obtenu dans le réseau GSM dans des intervalles de temps définis. Le fournisseur définit ce temps intervalle en fonction de la charge de communication générale.

Grâce aux réseaux GSM, la définition des positions est possible. En général, il existe plusieurs manières pour déterminer les positions sont avec es téléphones portables. Toutes ces méthodes reposent sur des principes géométriques. Le tableau ci-dessus **(1)** donne un aperçu sur les méthodes de positionnement mobile et leur analogie géodésique si existante ainsi que les restrictions

qui limitent la diffusion de ces méthodes.

Dénomination	Analogie géodésique	measurements	Restrictions
Identité de Cellule Globale (CGI)		ID cellule	Aucune
Handover /mise a jour de localisation de position		2 IDs cellule / 2 LA-IDs + 1 cellule ID	Disponible a un événement spécifié
Temps d'arrivé /TOA	Section arc	distances	Aucune
Temps d'arrivé différentiel / TDOA	Section hyperbolique	Différence de distance	BTSs doivent être synchronisés
Angle de l'arrivé / AOA	Intersection	angles	Besoin de antennes par BTS
Chevauchement de force du Signal		RXLEV (force de signal)	Donné de référence doit être disponible

**Légende :**

**CGI** :Cellule Global Identity

**LA** : location area

**TOA** :Time of arrive

**RXLEV** :Received signal level

**BTS** :Base transceiver station

**TDOA** :Time Differential of arrive

**AOA** :Angle of arriving

**Tableau I.1 : méthodes de positionnement mobile dans le reseau GSM (1)**

**I.2.2 Localisation Via GPS :**

Le GPS Global Positioning System (GPS, ou système de positionnement global), est une technologie qui permet aux récepteurs sur Terre de déterminer leur emplacement et de suivre leurs mouvements. Pour ce faire, il reçoit les signaux d'une trentaine de satellites en orbite autour de la planète. Ce système mondial de satellites fournit gratuitement des informations de géolocalisation et de temps à un récepteur GPS presque partout sur la Terre, étant donné qu'il n'y a pas d'obstacles et qu'au moins trois satellites GPS sont disponibles. La chaîne d'approvisionnement et de la logistique

sont l'une des nombreuses industries qui utilisent des récepteurs GPS. Les gestionnaires de ces domaines peuvent utiliser la technologie GPS pour améliorer l'efficacité et la précision de leurs opérations.

Le GPS fonctionne grâce à une technique appelée trilatération qui permet de calculer l'emplacement, la vitesse et l'altitude, la trilatération collecte les signaux des satellites pour produire des informations de localisation. Ce dernier triangule la position de l'appareil ou de l'utilisateur par rapport à trois satellites GPS ou plus. Chaque satellite transmet un signal qui inclut son emplacement et l'heure actuelle. En comparant les horodatages des signaux de différents satellites, le récepteur GPS peut calculer sa distance par rapport à chaque satellite. Ensuite, il utilise la trilatération pour localiser son emplacement exact sur le globe. Le GPS est impliqué dans tout, des applications de navigation aux appareils compatibles GPS comme les téléphones, les voitures et les trackers d'actifs. Grâce au GPS, nous pouvons toujours savoir où nous sommes, peu importe où nous nous trouvons dans le monde.

### **I.2.3 Localisation En double-Mode**

Avec le développement des systèmes de navigation par satellite, la conception de récepteurs de navigation multimode basé sur la radio logicielle est d'actualité. Plusieurs technologies existent, chacune repose sur des réseaux satellitaires. Le plus connu de ces types de systèmes est le Beidou/GPS en double mode.

Une des architecture utilisée pour traiter les signaux satellitaire est l'architecture FPGA+ARM(1). Cette architecture permet d'atteindre le positionnement de navigation. Avec la technologie basé sur le bi-mode Beidou/GPS, la précision de positionnement peut être améliorer en optimisant la constellation, et l'erreur de positionnement. Actuellement, le nombre de satellites dans la constellation spatiale du système Beidou est moindre à comparer avec le système GPS américain. Cependant, le nombre de satellites visibles dans tout point d'observation varie considérablement avec le temps.

Si le GPS tombe à chasse ou des interférences humaines se produisent, cela affectera sérieusement le fonctionnement normal de la navigation et des équipements de positionnement. Afin d'obtenir des fonctions de navigation et de positionnement précises et stables, il est nécessaire d'intégrer de différentes informations de navigation et de positionnement par satellite et d'utiliser les constellations spatiales GPS matures, pour ainsi compléter les défauts actuels de la constellation spatiale Beidou . L'application du système de navigation par satellite Beidou a un grand potentiel dans le marché. Le développement d'un récepteur multimode basé sur Beidou/GPS aura de bonnes perspectives.

## **I.3 Systèmes d'informations géographiques SIG :**

SIG signifie « système d'information géographique ». C'est un système informatique qui organise, stocke, analyse et cartographie les données. De cette façon, le SIG est l'intersection de l'emplacement et des données.

En termes simples, le SIG consiste à poser et à visualiser des données de localisation sur une carte, pour améliorer la prise de décision et la compréhension du monde qui nous entoure. Le SIG a été dédié aux personnes spécialistes du domaine, mais lors ces dernières décennies des logiciels SIG modernes ont vu le jour. Ces logiciels SIG modernes sont beaucoup plus accessibles et avec un coût moindre.

Le SIG a été développé au début des années 1960 avec l'objectif d'exploiter la puissance des ordinateurs afin de stocker et de traiter de grandes quantités de données géoréférencées.

### **I.3.1 Google Earth Pro :**

Google Earth Pro est un logiciel libre. bien qu'il ne soit pas un "vrai" SIG mais il est considéré comme un SIG car il permet la visualisation, l'évaluation, la superposition et la création de données géospaciales. Ce logiciel est souvent utilisé par les débutant qui veulent commencer avec des processus et des outils de base. Google Earth Pro peut également être utiliser pour visualiser des images satellite à très haute résolution, télécharger des données géospaciales, et aussi pour la localisation (par exemple pour un simple géocodage).

### **I.3.2 ArcGis :**

En 1969, l'Institut de recherche sur les systèmes environnementaux - ESRI - a été fondé. ESRI a développé de nombreuses méthodes d'analyse spatiale et de cartographie SIG utilisées aujourd'hui. Et en 1981, il a publié le premier logiciel SIG commercial.

Actuellement, ESRI fournit un logiciel de système d'information géographique (SIG) pour visualiser, gérer, créer, analyser et organiser des données géographiques appelé ArcGIS. Ce dernier permet la compréhension, la visualisation du contexte géographique des données et l'identification des relations et des modèles. Les licences des logiciels ArcGIS Desktop Student Edition et ArcGIS Pro sont fournies exclusivement à des fins éducatives et de recherche universitaire.

### **I.3.3 GRASS**

**Les logiciels SIG open source**, quant à eux, peuvent être téléchargés et utilisés gratuitement. Les options open source n'offrent généralement pas les fonctionnalités avancées des produits commerciaux et fournissent peu de ressources, mais ils sont plus que suffisant pour l'ensemble des systèmes de géolocalisation. Ce sont généralement multi-plateformes. Ils peuvent être

utilisés sous Linux, Windows et Mac OS.

Une des application SIG gratuite la plus complète disponible est GRASS. Il s'agit d'un système d'information géographique (SIG) utilisé pour la gestion et l'analyse de données géospatiales, le traitement d'images, la production de graphiques/cartes, la modélisation spatiale et la visualisation. Il est actuellement utilisé dans des environnements académiques et commerciaux du monde entier, ainsi que par de nombreuses agences gouvernementales et sociétés de conseil en environnement. GRASS peut être utilisé à la fois en ligne de commande et en mode graphique. Sa capacité à fonctionner conjointement avec l'application statistique gratuite R (au moyen du package « `spgrass6` ») est particulièrement intéressante pour les applications de recherche.

## **I.4 Les systèmes de suivi des véhicules**

Ces dernières années, le monde des systèmes de suivi ne cesse de se développer vu son importance et son besoin de notre vie quotidienne dans plusieurs applications ; par exemple pour suivre des enfants, des capteurs mobiles, des avions, des véhicules, etc. *• je l'ai écrit moi-même on n'a pas besoin de référence(1)*

### **I.4.1 Développement des systèmes de suivi des véhicules :**

Les systèmes de suivi des véhicules sont d'actualité surtout avec la tendance du transport intelligent et de la 5G. La 5G est de plus en plus impliquée dans la connectivité des véhicules. En plus, les véhicules semi-autonomes et autonomes facilitent la gestion des systèmes de suivi de véhicules avancés et standard.

Comme nous le savons, la pandémie de COVID-19 a causé d'énormes problèmes partout dans le monde et dans plusieurs domaines comme la baisse des ventes de véhicules de tourisme et des utilitaires. Cependant, la demande d'automobiles a augmenté ce qui a créé une demande de technologie de système de suivi des véhicules. Comme le commerce et la vente ne s'arrête pas, le volume de marchandises importées et exportées augmente de plus en plus.

Ce qui a mené à la nécessité des systèmes de suivi des véhicules pour ainsi suivre les véhicules et examiner minutieusement leurs systèmes dans le but de détecter tout dysfonctionnement. Même les sociétés de location de voitures ont intégré des dispositifs de suivi des voitures pour améliorer leurs services.

En outre, L'utilisation des systèmes de suivi des véhicules n'est pas seulement limitée au transport routier mais aussi au transport aérien. La diversité des domaines d'utilisation des systèmes de suivi de véhicules a permis un développement rapide et ainsi une croissance remarquable du marché.

### I.4.2 Étude des systèmes de suivi des véhicules :

Selon le fournisseur d'études de marché et de veille concurrentielle **Fact.MR** - le marché des systèmes de suivi des véhicules a reflété un TCAC de valeur de 13,1 % au cours de la période 2017-2021. Et ce marché devrait croître régulièrement à un TCAC de 13,7 %, au cours de la période 2022-2032. En 2022, la capitalisation boursière a été estimée à 21,7 milliards de dollars américains comme mentionner dans le **tableau I.2.**

Observant la montée en flèche de la demande sur le marché mondial, le marché des systèmes de suivi des véhicules devrait atteindre 78,35 milliards de dollars américains d'ici 2032 comme mentionner dans le **tableau I.2.**

### I.4.3 fonctionnalités d'un système de suivi des véhicules :

Les systèmes de suivi des véhicules peuvent offrir plusieurs fonctionnalités entre autres :

- Le suivi temps réel : essentiellement la géolocalisation
- Les notifications d'entretien du véhicule : une identification de la distance traversé a une période donnée mène a une connaissance de l'état de véhicule et de sont mécanique par rapport au indication du constructeur
- L'analyse du comportement du conducteur : une connaissance des horaires de conduite le circuit traversé ,l'état de santé et l'age du conducteurs et son expérience sont des facteurs important pour une gestion de planning
- L'historique des trajets : cet aspect est l'atout pour le choix des circuit dans le future par rapport a l'état de la route,embouteillage,la disponibilité des services ...
- La gestion du carburant des véhicules : pour chaque circuit ,chauffeur ,véhicule et cargaison un été de consommation de carburant peut être étudié puis estimé
- L'assistance à la répartition : un localisation d'un été d'urgence donne la possibilité a une intervention rapide et efficace
- Les notifications de protection des véhicules : une projection d'un circuit/véhicule sur les intervenant comme la météo par exemple aide a la décision et la planification d'une transportation donnée.

Attributs du rapport	Détails
Taux de croissance mondial (2022-2032)	TCAC de 13,7%
Valeur marchande attendue 2022	21,7 milliards de dollars américains

Valeur prévisionnelle projetée (2032)	78,35 milliards de dollars américains
Entreprises clés profilées	<ul style="list-style-type: none"> <li>• Verizon</li> <li>• Cartrack</li> <li>• Géotab Inc.</li> <li>• CalAmp</li> <li>• Tom Tom International BV</li> <li>• Teltonika</li> <li>• ORBCOMM inc.</li> <li>• Xirgo Technologies inc.</li> </ul>

**Tableau I.2 : Attribut du marché Tracking-as-service (2)**

#### **I.4.4 Principal obstacle au déploiement du système Tracking :**

Généralement les données et les informations stockées dans un système de suivi des véhicules, contiennent des informations sensibles telles que les détails d'achat du véhicule, les détails du propriétaire et d'autres données importantes stockées par le propriétaire, comme le carnet d'adresses par exemple. La non sécurité de ces informations peut mener à des opérations criminelles telles que le vols des véhicules, l'usurpation d'identité, La violation de la confidentialité des données et d'autre crime.

En outre, les événements tels que l'accès illégal ou le piratage du système de connectivité des véhicules peuvent diminuer la demande des systèmes de suivi des véhicules.

Le principal obstacle dans les systèmes de suivi de véhicule est la sécurité. Cependant le critère de sécurité doit être pris en considération lors de la réalisation d'un tel système.

#### **I.4.5 Les types des systèmes de suivi :**

##### **a. Les OBD :**

Le traqueur GPS de diagnostic embarqué (OBD) est un appareil qui a la capacité de surveiller l'emplacement et l'état du véhicule. Les véhicules comme la voiture, le camion, l'automobile peuvent mieux s'adapter à ce tracker GPS OBD. Ce tracker vous donne des alertes instantanées dans le cas où la voiture soit égarée ou ait des problèmes de moteur. OBD permet de détecter les informations suivantes : la température, le niveau de carburant, la falsification, l'accélération rapide, l'état d'allumage, la tension, les rappels d'entretien, etc. Actuellement le traqueur GPS OBD-II est la technologie la plus récente. IL diffère des technologies antérieures dans le fonctionnent et dans la

précision des informations de localisation. Ce tracker est juste un appareil plug & play similaire à une clé USB.

### **b. Le Tracking Point a Point**

Les systèmes de suivi des colis de courrier fonctionnent en mettant à jour l'emplacement d'un colis de courrier à intervalles réguliers, puis en transmettant ces informations aux clients.

Ils sont généralement plus chers que les services postaux standard, mais offrent une plus grande flexibilité et de nombreuses fonctionnalités supplémentaires. Vous pouvez facilement suivre les colis en temps réel via leur application ou leur site Web en ligne et vous avertir également s'ils sont retardés ou doivent être récupérés à un point de destination. Pour offrir un service aussi fiable, certains étapes sont suivis.

### **Génération AWB**

Lorsqu'une commande est passée et qu'une sélection d'un agent de messagerie est effectuée, une AWB (facture aérienne) est générée. Ce numéro sert d'identifiant unique à des fins de suivi.

### **Numérisation de codes-barres**

Lorsque la commande est prête à commencer son voyage, elle est scannée par code-barres. Il sert de première étape dans le processus de livraison.

### **Stockage d'informations**

Une fois qu'un colis est scanné, toutes les informations souhaitées sont stockées dans le logiciel du partenaire de messagerie. Les informations incluent le suivi des commandes, l'entrepôt d'origine et la destination.

### **Stockage en transit**

La commande doit parcourir un long trajet après avoir été récupérée à l'entrepôt. Les sociétés de logistique disposent de différentes installations de stockage.

### **Nouvelle numérisation**

Après avoir atteint un point de concentration, le colis est à nouveau scanné, puis il part pour le prochain stockage en transit. Le transporteur partenaire recueille tous les détails liés à la commande.

### **En cours de livraison**

Vient ensuite le dernier kilomètre vers l'achèvement de la livraison de la commande. Les agents de livraison quittent le point de concentration pour se rendre à l'adresse du client.

### **Livraison réussie**

C'est la dernière étape de livraison. L'agent de livraison remet la commande au client et les logiciels de suivi du courrier sont mis à jour avec le statut "livré". L'heure de livraison et le nom du destinataire sont également mis à jour dans le système pour référence future.

#### **I.4.6 Le Tracking-as-service dans le marché Algérien :**

De nombreuses solutions Tracking-as-service existent dans le marché algérien qui se focalise sur l'aspect Mono-Tracking. Dans l'aspect Mono-Tracking en ce qui concerne l'utilisation des traceurs à base GPS. On peut citer la solution **Traxy** qui se base sur une application Mobile qui récupère les données GPS depuis le mobile. Cette méthode est utilisée par les systèmes de location de voitures ou par des systèmes de courses comme **Yassir** par exemple. Autre logique de tracking ce repose sur la signalisation point-a-point, qui est utilisée par les fournisseurs de service de livraison.

Éventuellement des produits d'importation existent basés sur l'obtention du trackeur, et son inscription sur des plateformes dans le web pour pouvoir être tracer, mais cette solution n'est pas adaptée au marché public vis-à-vis la réglementation.

### **Conclusion**

Dans ce chapitre, nous avons fait le point sur les systèmes de suivi de véhicules, dans le prochain chapitre nous allons voir les détails des technologies et des outils nécessaires à la réalisation de notre contribution.

## **Chapitre II**

# **Technologies, Outils ET Architectures**

## II.1 Introduction :

Multiple Technologies , framework ou autres , sont utilisés dans le cycle de développement des différent modules de notre système de tracking : web ,mobile et application de bureau ,notre approche est l'utilisation de ce qu'on peut classé comme indépendants des différents facteurs comme les systèmes d'exploitations et système de gestion de base de données ,ou même les spécifications des composants électronique des module de tracking ,la chose qui a une influence économique plus que purement technique .

## II.2 Technologies base de la plate-forme

### II.2.1 Servlets :

#### a. Généralité sur les servlets :

Les servlets Java sont une partie de Java Entreprise Edition (Java EE). C'était la première technologie pour Java et beaucoup d'autres technologies web avaient arrivées depuis.

Les Servlets Java sont des programmes qui hérite les capacités de serveur et peuvent répondre à n'importe quel type de requêtes. Cependant, les servlets implémentent les applications sur des serveurs web. Les servlet en Java sont équivalentes aux technologies comme PHP et ASP.NET.

Les servlets sont utilisées pour créer une application web et pour implémenter les classes Java qui répondent aux requêtes des utilisateurs. Elles peuvent communiquer à travers n'importe quel protocole client-serveur, mais elles sont parfois utilisées avec le protocole HTTP. Un développeur web peut utiliser une servlet pour ajouter un contenu dynamique au serveur web en utilisant la plateforme Java. Le contenu généré est un code HTML, mais peut être aussi un autre comme le XML. Plusieurs servlets constituent une application web.

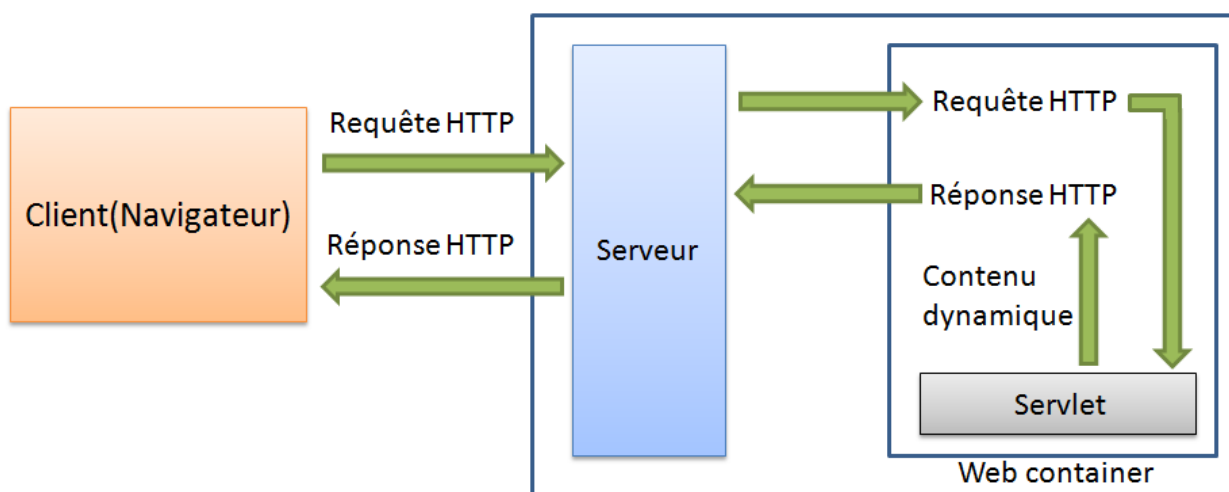


Figure II.1: Architecture des Servlets. (4)

L'API servlet se trouve dans le package `javax.servlet`. Ce package définit l'interaction entre une servlet et le web container. L'objet servlet reçoit une requête et génère une réponse basée sur cette requête.

Les servlets peuvent être générées automatiquement à partir de Java Server Page (JSP) par le compilateur JavaServer Page. La différence entre les servlets et les JSPs est que les servlets sont intégrées dans le code Java, tandis que les JSPs sont intégrées dans le code HTML.

Un conteneur de servlets peut exécuter plusieurs applications web en même temps, chacune a de multiple servlets qui sont entrains de s'exécuter à l'intérieur.

## **b. Cycle de vie d'une Servlet :**

Une requête dans un conteneur web est prise en charge par le conteneur; si une requête arrive pour la première fois dans le serveur Web (conteneur), on a le conteneur qui effectue les actions suivantes:

### **Chargement de la classe Servlet**

Pour qu'une classe java arrive dans la mode exécution, elle doit être chargée et c'est la même chose pour une Servlet. Lorsqu'une Servlet est chargée, il n'existe qu'une seule instance de la Servlet durant son cycle de vie.

### **Création de l'instance d'une Servlet**

Une fois que la Servlet est chargée, le conteneur crée une seule instance de la Servlet. Le container exécute cette étape un seule et une seule fois pour une Servlet donnée.

### **Initialisation de la Servlet (le Container appelle la méthode `init()`)**

L'initialisation est réalisée une fois lorsque la Servlet est créée. Dans la méthode `init()`, on peut avoir les paramètres d'initialisation dans le fichier de configuration `web.xml`; voir un exemple de `web.xml` .

### **Appel de la méthode `service()`**

C'est la méthode principale qui exécute la logique métier de l'application et c'est elle qui est aussi responsable du traitement des requêtes des clients. Quand une nouvelle requête d' une Servlet arrive, le conteneur génère une nouvelle Thread; la nouvelle thread exécute la méthode `service()` puis la logique métier de l'application ou la tâche prévue de la Servlet.

La définition de la méthode `service()` est:

```
public void service(ServletRequest request, ServletResponse response)  
throws ServletException, IOException {  
}
```

La méthode `service()` prend deux arguments "request" de type `ServletRequest` et "response" de `ServletResponse`.

Une requête HTTP peut être de différents type comme GET, POST, PUT, DELETE, etc; le rôle de la méthode `service()` de voir la requête HTTP requise et de faire appel à la méthode adéquate qui peut être `doGet`, `doPost`, `doPut`, `doDelete`, etc .

- **`doGet(ServletRequest req, ServletResponse res)`**: le type par défaut de toute requête http est "GET"; cela veut dire que si on ne spécifie pas un type de requête c'est le type "GET" qui prend en charge la requête donc la méthode `doGet()` au sein d'une Servlet. Comme son nom l'indique la méthode `doGet()` récupère des données ou des informations du serveur via une ou plusieurs requêtes. Toutefois on peut aussi envoyer des données ou des informations au serveur via une requête. La méthode `doGet(ServletRequest req, ServletResponse res)` a comme signature:

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
}
```

- **`doPost(ServletRequest req, ServletResponse res)`**: Cette méthode est utilisée pour envoyer des données au Serveur; pour faire des requêtes HTTP de type POST on le spécifie dans la balise "form" par l'attribut POST; La méthode POST est une bonne alternative à la méthode GET. Cette méthode code les informations de la même façon que la méthode GET (encodage URL et paires nom/valeur) mais elle envoie les données à la suite des en-têtes HTTP, dans un champ appelé ***corps de la requête***. De cette façon la quantité de données envoyées n'est plus limitée, et est connue du serveur grâce à l'en-tête permettant de connaître la taille du ***corps de la requête***. La signature de la méthode `doPost(ServletRequest req, ServletResponse res)` est:

```
public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
}
```

### Appel de la méthode `destroy()`

La méthode **`destroy()`** est appelée à la fin du cycle de vie de la Servlet; elle permet aussi, si elle est invoquée, de fermer tous les fichiers ou bases de données ouverts par la Servlet lorsqu'elle était chargée par le conteneur.

Une Servlet est déchargée par le conteneur dans les deux cas suivants:

- 1.Si le conteneur s'arrête,
- 2.Si le conteneur recharge l'environnement d'exécution de l'application web

Après l'appel de la méthode **`destroy()`**, la Servlet es envoyée au ramasse-miette. La signature de la méthode **`destroy()`** est:

```
public void destroy(){
}
```

Comme mentionné ci-dessus, la Servlet est chargée dès qu'elle sollicitée pour la première fois pour traiter une requête. Il existe des cas où on veut qu'elle le soit avant la sollicitation d'une requête; pour cela on le définit dans le fichier de configuration "web.xml" en se servant de la propriété "<load-on-startup>" . La valeur qu'on donne à cette propriété permet de forcer le conteneur de charger la Servlet dès que le conteneur démarre.

```
<servlet>
  <servlet-name>MyFirstServlet</servlet-name>
  <servlet-class>MyFirstServlet</servlet-class>
  <init-param>
    <param-name>param-name</param-name>
    <param-value>param-value</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

Le nombre entre les deux balise ci-après "<load-on-startup>1</load-on-startup>" indique au conteneur de charger la Servlet dès que le que le conteneur est démarré. Si le nombre est négatif, la Servlet le conteneur peut charger la Servlet à n'importe quel moment.

## II.2.2 JSP :

Les JSP sont une approche ancienne qui a beaucoup bénéficié de l'introduction de constructions syntaxiques plus élégantes que l'introduction de code Java, et d'un codage basé sur les conventions qui simplifie considérablement l'écriture. Les langages de vues proposés par les frameworks sont pour l'essentiel assez proches des JSP. Nous allons passer en revue des concepts généraux à connaître.

### a. Balises et directives

L'idée initiale des JSP était d'étendre HTML avec des « balises » spéciales interprétées comme des instructions Java, avec l'espoir d'aboutir à une syntaxe à peu près uniforme intégrant le HTML « statique » et les parties dynamiques. De nombreuses mises au point ont été nécessaires pour aboutir à un résultat satisfaisant.

Cette approche repose clairement sur l'idée de simplifier l'écriture d'une servlet en évitant de placer des **out.println** devant chaque ligne HTML, une ambition louable mais limitée, ouvrant surtout la porte à des dérives vers du code illisible et non maintenable.

Le conteneur compile une page JSP sous la forme d'une servlet dans laquelle les éléments HTML sont transcrits en instructions **out.println()**. Il s'agit donc vraiment, au moins initialement, d'un moyen simplifié de créer des servlets produisant du HTML

La norme JSP comprend également des directives qui se présentent sous la forme de balises

<%@ %>. Toujours dans une optique de programmation, on peut par exemple importer des packages Java:

```
<%@ page import="java.util.Date" %>
```

le mot-clé page indique le type de la directive. Reprenez les JSP créées précédemment: elles contiennent une directive initiale indiquant l'encodage de la page.

Une des améliorations des JSP a consisté à les rendre extensibles en ajoutant des bibliothèques de balises (tag libraries) associées à des fonctionnalités externes. Ces bibliothèques doivent être déclarées avec une directive:

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

Le prefix désigne un espace de nommage (namespace) qui permet d'identifier une balise comme appartenant à cette bibliothèque. Par exemple, une balise <c:import /> sera identifiée comme appartenant à la bibliothèque **core** dans **http://java.sun.com/jsp/jstl**, et le code associé sera exécuté.

Un autre directive est **include** qui permet de composer des pages HTML à partir de plusieurs fragments:

```
<%@ include file="header.jsp" %>
```

Comme les balises de code, les directives sont anciennes et peuvent dans la grande majorité des cas être remplacées par une approche plus moderne et plus élégante.

## **b. Portée des objets**

Un objet instancié dans la servlet-contrôleur peuvent être mis à disposition de la vue en l'affectant à l'objet-requête request. Il existe plus généralement quatre portées de visibilité des objets dans une application. Un objet placé dans une portée est accessible par tous les composants de l'application (par exemple les JSP) qui ont accès à cette portée :

**la portée page**: tout objet instancié dans une JSP est accessible dans cette même JSP; c'est la portée la plus limitée;

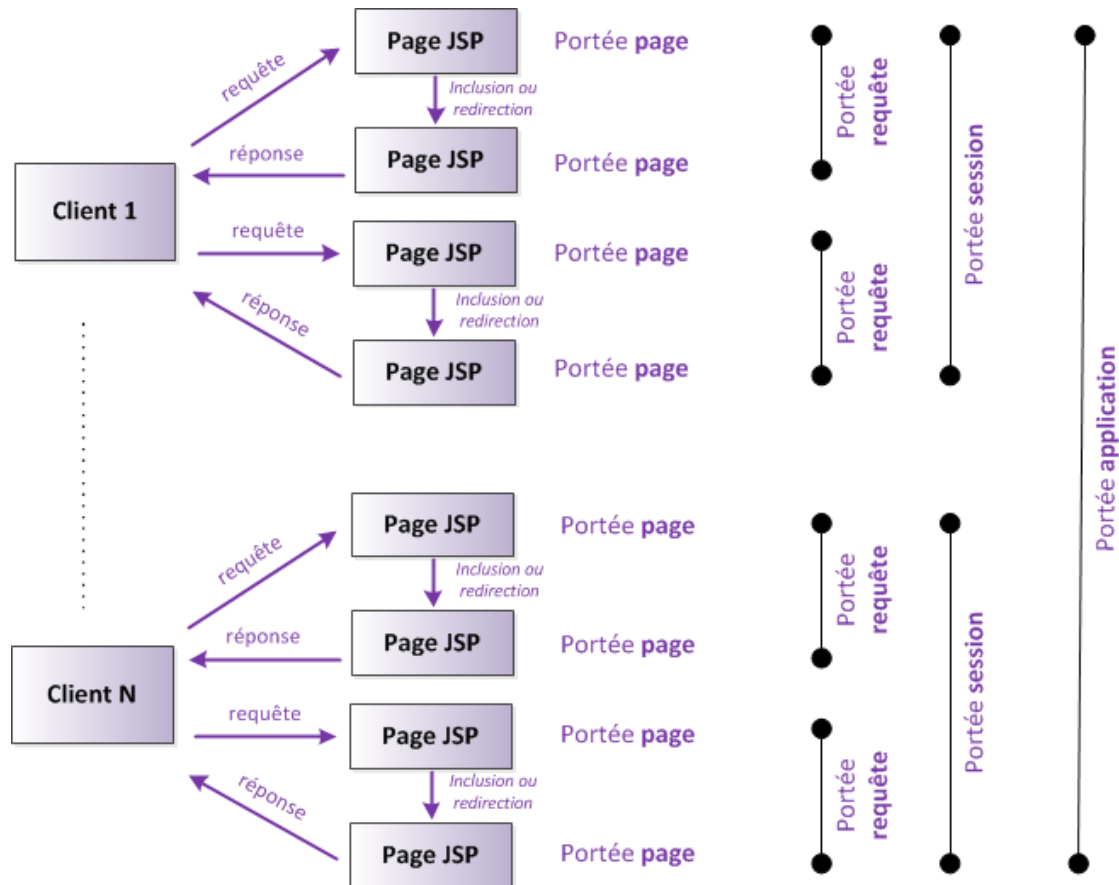
**la portée request**: les objets placés dans cette portée restent accessibles durant toute l'exécution d'une requête HTTP par le serveur, même si l'exécution de cette requête implique plusieurs contrôleurs (par des mécanismes de forward) ou plusieurs JSP;

**la portée session**, une session étant constituée par un ensemble d'interactions client/serveur, un objet placé dans la session (par exemple le fameux panier des achats dans un site de commerce) reste accessible sur la période couverte par ces interactions;

**la portée application** est la plus générale de toute: un objet placé dans cette portée reste accessible tant que l'application est active dans le conteneur de servlets.

Le protocole HTTP ne mémorise pas les échanges entre un client et un serveur. Pour effectuer un suivi, on doit donc simuler des sessions, le plus souvent en se basant sur les cookies.

Au sein d'une JSP, on peut accéder à une variable en indiquant la portée dans laquelle elle a été placée: `requestScope.compte` par exemple est l'objet 'Compte' placé au préalable comme attribut dans l'objet request.



**Figure II.2: Les portées des variables en JEE. (5)**

Si la portée n'est pas indiquée, le conteneur va explorer les portées pour chercher un objet portant le nom indiqué. Cette pratique est potentiellement source d'ambiguïté et d'erreurs difficiles à détecter

### c. Les expressions

Les Objets dans un JSP sont déclarés pour accéder à leurs propriétés (pour les afficher dans la vue) avec une syntaxe minimisée basée sur un langage d'expressions. Une expression désigne, en

programmation, toute instruction ou ensemble d'instructions renvoyant une valeur (fonction, opération arithmétique, etc.). En JSP on garde la même définition. Les expressions sont présentées sous la forme:

```
${expression}
```

Quand la JSP est exécutée, la valeur de l'expression (après évaluation par le moteur de servlet) est insérée dans la page HTML produite, en substitution de l'expression. C'est une simplification syntaxique de la balise ancienne des JSP `<%= %>` :

```
${ 11 +15 }
```

**Cette expression sera remplacé par 26 à l'exécution.**

la principale utilisation des expressions est l'accès aux propriétés des objets. Pour reprendre l'exemple plus haut:

```
${requestScope.Compte.pseuso}
```

Est une expression qui a pour effet d'afficher la propriété **pseuso** de l'objet **Compte** placé dans la portée request. Si cette propriété n'est pas publique (ce qui est toujours recommandé), le conteneur de servlet va chercher une méthode « getter ».

L'objet requestScope que nous utilisons dans notre code qui précèdent est un objet implicite accessible par les expressions. Il en existe d'autres, l'ensemble constituant un contexte de communication entre le contrôleur (la servlet) et la vue (la page JSP). Notez que la plupart des objets sont des tables d'association (Map).

### **II.2.3 JSTL :**

#### **a. Vue d'ensemble sur la librairie JSTL :**

Une évolution majeure des JSP a consisté à les rendre extensibles en permettant la définition de balises fonctionnelles associées à une librairie Java. Le placement d'une de ces balises dans une page JSP correspond à l'exécution d'une fonctionnalité dans la librairie. C'est un moyen élégant de faire appel à des éléments dynamiques (programmés) dans une vue, sans avoir à intégrer du code Java.

Ces bibliothèques de balise peuvent être définies par tout programmeur, mais certaines sont destinées à résoudre des problèmes récurrents, et à ce titre elles sont fournies sous la forme d'une librairie standard, la Java Standard Tag Library ou JSTL. Nous allons brièvement l'explorer dans cette section, en nous basant sur la version 1.2.

Pour que les balises de la JSTL soit reconnues par Tomcat, il faut installer les deux fichiers :

- [jsp-api-2.1-6.0.2.jar](#)
- [jstl-1.2.jar](#)

Pour la mise en service du JSTL on doit récupérer ces fichiers et copiez-les sous **WEB-INF/lib**. Il peut être nécessaire de redémarrer Tomcat pour que les librairies soient prises en compte.

Une fois les librairies mises en place, on peut les utiliser dans des JSP. Voici un exemple de code.

```
<%--
  Document   : cat_serv_form
  Created on : 13 avr. 2022 23:55:12
  Author    : user
--%>

<%@page language="java" contentType="text/html; charset=utf-8"
pageEncoding="UTF-8" %>
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>
<c:set var="now" value="<%=new java.util.Date()%>" />
<!DOCTYPE html>

<div id="cat_space" class="cat">

  <div class="icon_type">
  </div>
  <div id="map">
    <c:forEach items="${trackers_now}" var="tracker" varStatus="i">
      <div class="tracker">
        <span class="trackerCryId">${tracker.getIdTracker}</span>
        <span class="tracker_titre">${tracker.getTitretracker()}</span>
        <c:forEach items="${tracker.getTrackingList()}" var="tracking"
varStatus="i">
          <span class="tracker_position">${tracking.getLatitudeString()},${
{tracking.getLongitudeString()}</span>
        </c:forEach>
      </div>
    </c:forEach>
  </div></div>
```

## b. Bases de code JSTL :

### Variables et expressions :

Deux choses sont à noter. Tout d'abord la librairie est incluse avec la directive taglib:

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

L'URI (Universal Resource Identifier) identifie la librairie, que Tomcat va donc chercher dans son environnement. Si le jar est bien installé comme indiqué ci-dessus, il devrait le trouver dans **WEB-INF/lib**.

Le préfixe **c**: va servir, lui, à reconnaître les balises de la librairie. C'est par exemple:

```
<c:out value="Compte.pseudo" />
```

Le préfixe agit comme un espace de nommage au sens de XML. Toutes les balises dont le nom est préfixé par **c:** sont considérées comme appartenant à la JSTL, et cette dernière leur associe une fonctionnalité particulière. Dans notre code, il n'est pas trop difficile d'imaginer que **c:out** permet d'afficher une valeur dans le document HTML produit.

La librairie JSTL est étroitement associée aux expressions. On peut donc combiner une expression, et l'affichage de son résultat, comme dans:

```
<c:out value="{15-3}" />
```

Le JSTL effectue un pré-traitement de la chaîne de caractères à afficher, pour la rendre compatible avec XHTML. Les caractères réservés comme **<**, **>**, **&**, sont remplacé par des références à des entités, comme **&lt;**, **&gt;**, **&amp;**, etc.

La protection des caractères interdits est particulièrement importante quand on veut afficher du texte provenant d'une source externe (formulaire, base de données) pour éviter les attaques d'injection (de code javascript par exemple);

### Conditions et boucles

Pour affichage d'un message ou un fragment HTML en fonction d'une certaine condition. Par exemple, vous voulez affichage du message de bienvenue si l'utilisateur est connecté, un formulaire de connexion sinon. On a besoin de tester la valeur d'une variable. Avec les JSTL, c'est la balise **<c:if>**, avec un attribut **test** contenant une expression Booléenne. Voici un exemple :

```
<c:if test="{pseudo.lenght > 0 }">
  pseudo n'est pas vide</c:if>
```

Pour des tests un peu plus étendus, on peut recourir à une variante du **case/switch**:

```
<c:choose>
  <c:when test="{uneVariable == 0}">Variable null</c:when>
  <c:when test="{uneVariable > 0 && uneVariable < 200}">Valeur</c:when>
  ...
<c:otherwise>Sinon...</c:otherwise></c:choose>
```

La possibilité d'exprimer des boucles est également indispensable dans une vue pour traiter des variables de type liste ou ensemble. Premier cas, le plus simple : on dispose d'une variable de type tableau indicé. Nous avons donc besoin d'un moyen d'énumérer toutes les valeurs de l'indice. Le voici :

```
<table>
  <c:forEach var="i" begin="0" end="5" step="1">
```

```

        <tr>
          <td><c:out value="{i}" /></td>
        </tr>
      </c:forEach>
    </table>

```

il s'agit d'une boucle for d'un langage de programmation classique. Dans le cas où la variable correspond à une *collection* dont on ne connaît pas à priori la taille (*ArrayList*, *HashMap*, *Collection*, typiquement), la boucle JSTL prend une forme légèrement différente. Supposons que la collection soit dans une variable *trackers*. On la parcourt de la manière suivante:

```

<c:forEach items="{trackers}" var="trkr">
  <c:out value="{trkr.id}" />
</c:forEach>

```

Une variable locale à la boucle, de nom **trkr**, est ici définie et successivement associée à toutes les valeurs de la collection. Dans la boucle, on se retrouve donc en situation d'avoir une variable simple, objet, à laquelle on peut appliquer les opérations déjà vues d'accès aux propriétés et d'affichage.

L'objet implicite *pageContext* est un peu différent. Ce n'est pas une Map mais un objet donnant accès au contexte de l'exécution d'une JSP par l'ensemble des attributs suivants:

### Attributs de l'objet *pageContext*

Nom	Type	Description
request	ServletRequest	L'objet request
response	ServletResponse	L'objet response
servletConfig	ServletConfig	Configuration de la servlet
servletContext	ServletContext	Contexte de la servlet
session	HttpSession	La session

### Liens :

À priori un lien peut être directement inclus comme du HTML, sous la forme standard d'une ancre `<a/>`. Par exemple:

```

http://winrahpfeweb.domaine.com

```

Dans de nombreux des éléments dynamiques interviennent dans la création de l'URL du lien.

Voici les principaux cas:

- le lien mène à une autre page du site: on peut indiquer une adresse relative ou une adresse absolue, ça dépend de la situation
- le lien comprend des paramètres: leur valeur doit être encodée selon la norme HTTP pour survivre au transfert réseau;
- Le lien rentre dans le cadre de la gestion des sessions.

La balise **JSTL** `<c:url>` fournit de manière transparente un support pour gérer les besoins ci-dessus.

Voici une brève présentation, en commençant par la gestion du contexte.

```
${pageContext.request.contextPath}/convertisseur
```

L'URL interne est `/convertisseur`. Pour ne pas dépendre, dans l'URL complète, du contexte qui peut varier (dev pour le développeur, testapp pour la machine d'intégration, appname, etc.), nous avons introduit un contexte dynamique représenté par `${pageContext.request.contextPath}`. Il s'agit d'une expression, mais nous n'avons pas encore la possibilité d'utiliser les extensions JSTL. Avec ces dernières, le lien devient simplement:

```
<c:url value="/localisation"/>
```

Notez bien que cette adresse est absolue. Elle est automatiquement étendue par le conteneur de servlet en ajoutant le chemin de l'application. Pour nous, cela donne:

```
/TRK1/localisation
```

Faut-il préciser qu'il est extrêmement important de ne pas coder « en dur » des valeurs dépendants de la configuration dans du code? Le tag `c:url` est intégrable tel-quel dans une balise HTML `a`, ce qui donne une syntaxe un peu étrange:

```
<a href="<c:url value="/convertisseur"/>">Lien vers mon convertisseur</a>
```

Le second avantage de `c:url` est la prise en charge automatique de l'encodage des valeurs de paramètre. La syntaxe pour les paramètres est la suivante:

```
<c:url value="/localisation">  
  <c:param name="param1" value="${compte.pseudo}"/>  
  <c:param name="param2" value="${tracker.latitude}"/>  
</c:url>
```

la balise `<c:param>` sera encodée selon les règles décrites à:

[http://www.w3schools.com/TAGs/ref\\_urlencode.asp](http://www.w3schools.com/TAGs/ref_urlencode.asp).

En retrouve d'autres balises : '`<c:redirect>`' (envoie une directive de redirection au navigateur) et

'<c:import>' pour importer des fragments HTML.

```
</head> <body class="index_body" id="index_body">
<c:import url="/WEB-INF/Palettes/head.jsp"/>
```

Notez également qu'il existe quelques autres bibliothèques que la bibliothèque core d'un intérêt moins central: xml (traitement de documents XML), fmt (formatage, par exemple des dates), fn (chaînes de caractères).

### II.1.5 JNDI :

JNDI est l'acronyme de Java Naming and Directory Interface. Cette API fournit une interface unique pour utiliser différents services de nomenclatures ou d'annuaires et définit une API standard pour permettre l'accès à ces services.

Il existe plusieurs types de services de nommage parmi lesquels :DNS, LDAP ,NIS ,COS , etc, ...

Un service de nommage permet d'associer un nom unique à un objet et de faciliter ainsi l'obtention de cet objet.

Un annuaire est un service de nommage qui possède en plus une représentation hiérarchique des objets qu'il contient et un mécanisme de recherche.

**JNDI** propose donc une abstraction pour permettre l'accès à ces différents services de manière standard. Ceci est possible grâce à l'implémentation de pilotes qui mettent en oeuvre la partie SPI (Service Provider Interface) de l'API **JNDI**. Cette implémentation se charge d'assurer le dialogue entre l'API et le service utilisé.

**JNDI** possède un rôle particulier dans les architectures applicatives développées en Java car elle est utilisée dans les spécifications de plusieurs API majeures : JDBC, EJB, JMS, ...

De plus, la centralisation de données dans une source unique pour une ou plusieurs applications facilite l'administration de ces données et leur accès.

Pour utiliser **JNDI**, il faut un service de nommage correctement installé et configuré et un pilote dédié à ce service.

On a utilisé le **JNDI** comme moyen de consultation d'objets hébergés dans un serveur distant ,sous une Architecture **EJB**, par l'application de bureau ERP .

## II.3 Outils de développement

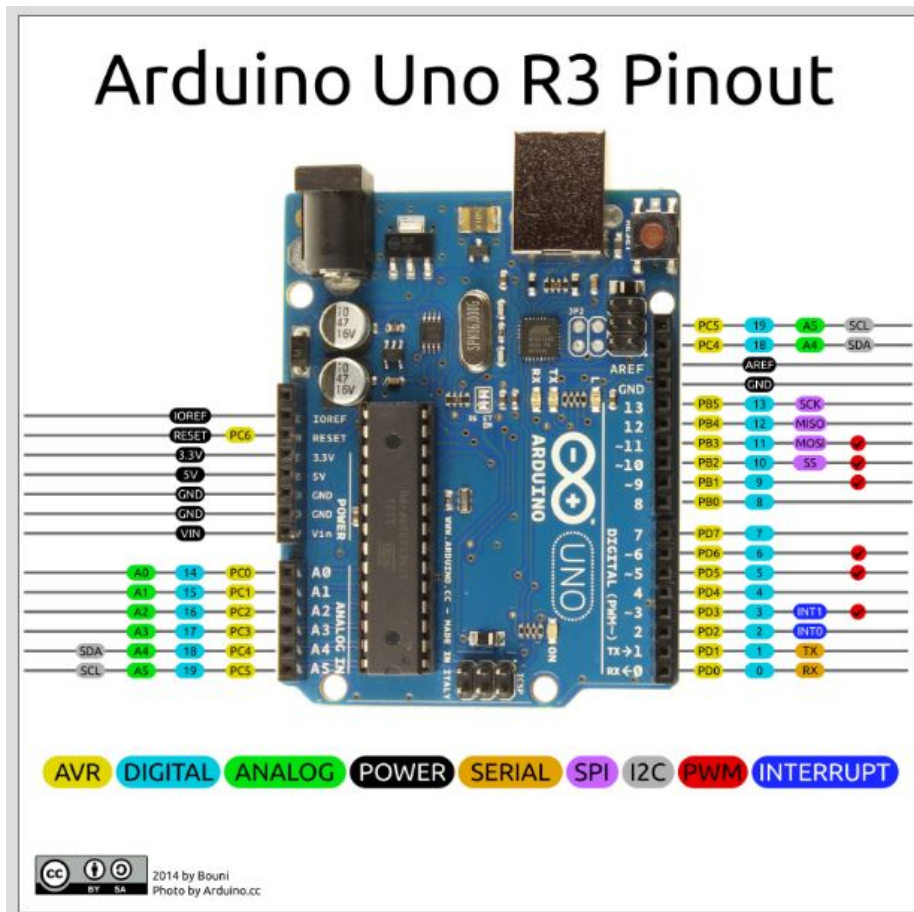
### II.3.1 Outils Matériels

Les principaux composants de nos trackers sont une carte de développement Arduino et un module de localisation et d'émission de requête qu'on trouve plusieurs possibilités dans le marché algérien, notre choix a été pour la carte arduino UNO et le module de localisation/communication

avec le système SIM808 .

a. **Arduino :**

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation.



**Figure II.3 – Schéma pin de Arduino UNO R3 Board (6)**

Cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles sans le besoin d'une connaissance approfondi en électronique .

Ce qui est une abstraction des aspects fondamentale de la semi-conduction en construisant un pont tendu entre le monde réel et le monde numérique pour étendre les capacités de relations homme/machine ou environnement/machine.

Le projet Arduino est né à petite ville du nord de l'Itali et qui, de manière totalement inattendue, est en train de révolutionner le domaine de l'électronique à l'échelle mondiale.

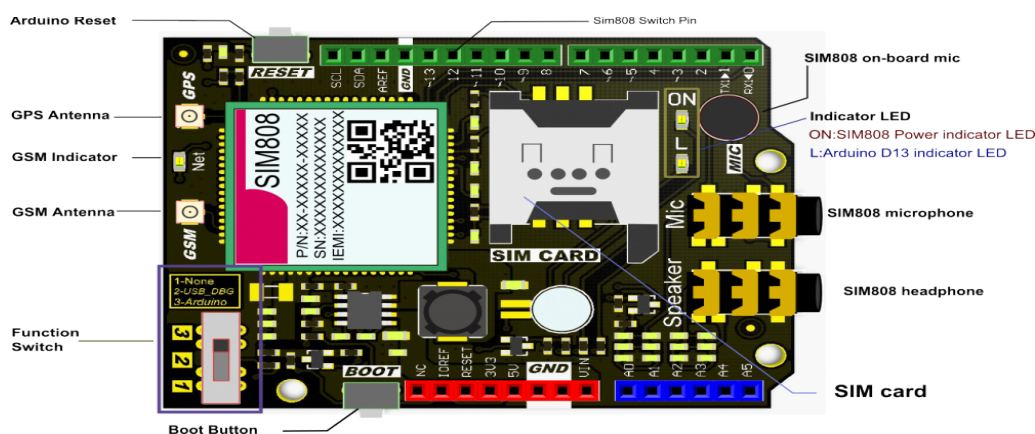
Le choix initial des licences libres qui a conditionné non seulement son bas prix et sa massive diffusion mais également son approche et son état d'esprit. Hiver 2005, Massimo Banzi enseigne dans une école de Design à Ivrea en Italie, et souvent ses étudiants se plaignent de ne pas

avoir accès à des solutions bas prix pour accomplir leurs projets de robotique. Banzi en discute avec David Cuartielles, un ingénieur Espagnol spécialisé sur les micro-contrôleurs et Ils décident de créer leur propre carte en embarquant

Dans leur histoire, David Mellis l'un des étudiants de Banzi, été le chargé de créer le langage de programmation allant avec la carte en deux jours.

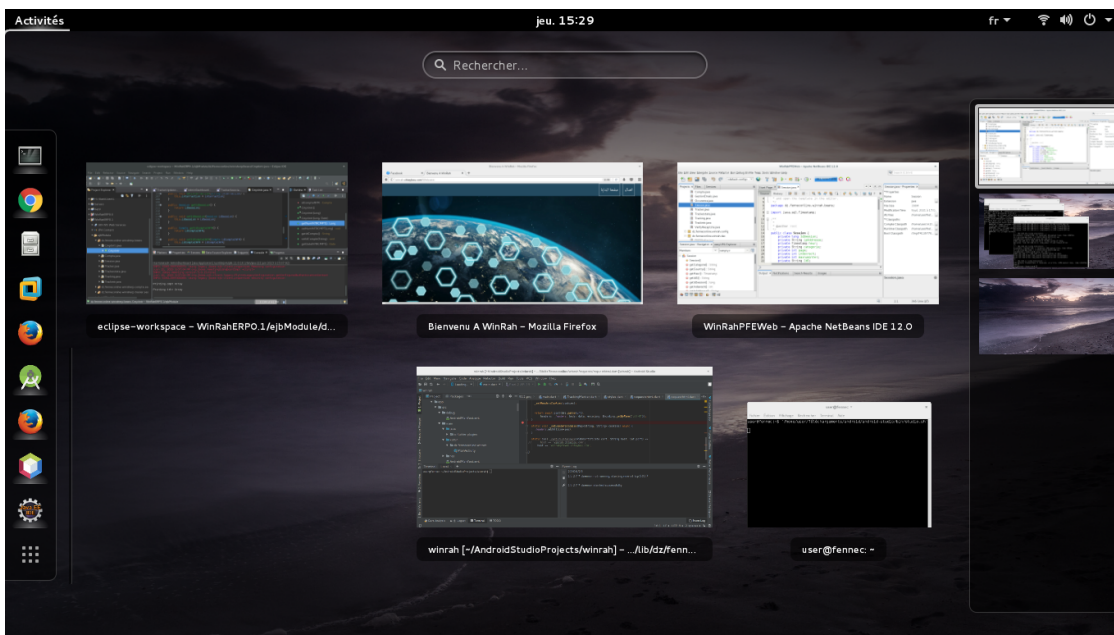
## b. Sim808

Le module arduino SIM808 GPS/GPRS/GSM est un module d'extension Arduino GSM/GPRS et technologie de navigation GPS quadri-bande intégré. A des dimension d'une carte de crédit, selon le package standard des pins Arduino, compatible avec Arduino UNO, arduino Leonardo, arduino Mega et d'autres cartes arduino. Les performances et la stabilité ont été amélioré par rapport aux génération précédente SIM908, SIM808 . En plus des fonctions SMS et de la téléphonie , le module prend également en charge les fonctions MMS, DTMF, FTP et autres. Vous pouvez réaliser l'acquisition de données, l'émission-réception de données sans fil, l'application IoT et la localisation GPS. Ce module comporte un microphone intégré et une connexion pour casque, il peut également se connecter directement à l'antenne GSM et GPS par un connecteur d'antenne externe. SIM808 GPS/GPRS/GSM Arduino Shield V1.0 utilise la dernière version du module Simcom SIM808, par rapport au premier module SIM808 disponible sur le marché, le nouveau module a une meilleure stabilité. Mais la partie GPS des commandes AT n'est pas compatible avec l'ancienne



**Figure II.4 -Sim808 Board (7)**

## II.3.2 Outils Logiciels



**Figure II.5: Outils Logiciel de développement**

### a. IDEs :

- **Netbeans** : NetBeans IDE est un environnement de développement intégré gratuit et à code source ouvert destiné au développement d'applications sous Windows, Mac, Linux et Solaris. L'environnement IDE simplifie le développement d'applications Web, d'entreprise, de bureau et mobiles en utilisant les plates-formes Java et HTML5 , On choisie netbeans pour le développement de la plate forme web du au outil qu'il fournie pour la gestion des BDD et du serveur Web .
- **Eclipse** : **Eclipse** est une plate-forme de développement libre, écrite en Java et composée d'une multitudes de projets et sous-projets. Le projet fondateur, Eclipse Platform constitue le noyau et les composants de base d'**Eclipse**. Il s'avère extensible via l'ajout de plugins supplémentaires,l'environnement eclipse été notre choix pour le développement du module ERP de bureau pour ses modules adaptés au développement des services **Java EE**
- **Android Studio** : Android Studio est l'environnement de développement intégré (IDE) officiel des applications Android. Basé sur le puissant outil de développement et d'édition de code d'**IntelliJ IDEA**, Android Studio offre encore plus de fonctionnalités qui améliorent la productivité pour crée des applications Android. Grâce au module pour le développement **Flutter** on réalisé l'application mobile de la plate-forme .

## b. Serveurs :

- **Tomcat** : Apache Tomcat est un **logiciel de serveur d'applications web** open source conçu pour la programmation en Java et développé et maintenu par Jakarta, le groupe de projets open source Java de la fondation Apache. On a utilisé Tomcat pour **héberger et déployer les servlets Java** .
- **Wildfly** : WildFly est un serveur d'applications Java Enterprise Edition complet qui fournit toutes les fonctionnalités nécessaires pour exécuter une application Web Java. WildFly est conçu et maintenu par Red Hat et était officiellement connu sous le nom de JBoss AS. Notre choix été pour le serveur **Wildfly** comme serveur pour le module ERP

## II.4 Modèles d'Architectures

### II.4.1 DAO Pattern :

L'un des aspects importants de la couche métier est la couche d'accès aux données qui connecte les services aux bases de données. L'accès aux données dépend de la source de données, du type de stockage comme la base de données, le fichier texte, le fichier xml, le fichier json, etc. Même il est différent de son implémentation, par exemple : la syntaxe de requête SQL diffère entre MySQL, SQL Server, Oracle, etc. Avec lui En s'attendant à ce qu'il n'y ait aucune différence lors de l'accès à une base de données relationnelle, à l'analyse d'un fichier xml ou à toute autre source de données, nous pouvons appliquer le modèle de conception à **Data Access Object DAO Pattern** ) .

Le modèle d'objet d'accès aux données (DAO) est l'un des modèles appartenant au groupe de modèles structurels. Le modèle de conception DAO est utilisé pour séparer la logique de stockage des données dans une couche distincte. De cette façon, les services sont masqués sur la façon dont les opérations de bas niveau pour accéder à la base de données sont effectuées.

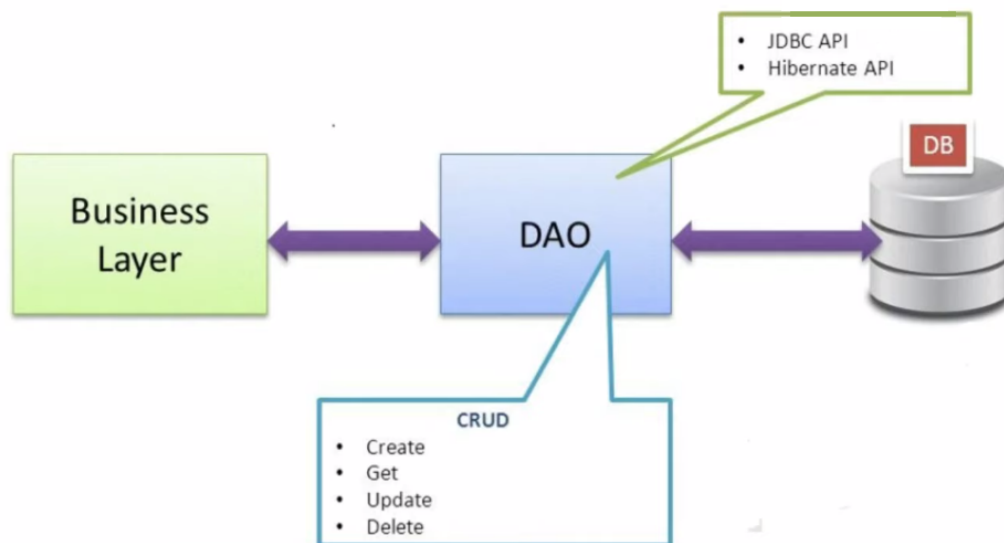


Figure II.6: Le Déploiement du pattern DAO (8)

Il est également connu sous le nom de principe de séparation de la logique .

Le modèle de conception DAO est utilisé en Java pour séparer la logique métier et la logique d'accès aux données dans une application. Il fournit une couche d'abstraction entre l'application et la source de données, telle qu'une base de données, un système de fichiers ou un service Web externe. Le modèle DAO implique la définition d'un ensemble d'interfaces et de classes qui encapsulent la logique d'accès aux données. Les interfaces définissent le contact entre l'application et la couche DAO, tandis que les classes implémentent les interfaces et fournissent l'implémentation réelle des opérations d'accès aux données.

#### **II.4.2 EJB**

Enterprise JavaBeans (EJB) est une architecture distribuée offrant des composants logiciels déployés côté serveur. • Cette architecture propose un cadre pour créer des composants distribués et facilement réutilisables.

Les EJB sont hébergés au sein d'un conteneur EJB évoluant dans un contexte transactionnel.

• Ils permettent de :

1. représenter des données (EJB Entity),
2. de proposer des services avec ou sans conservation d'état entre les appels (EJB session),
3. ou encore d'accomplir des tâches de manière asynchrone (EJB dit message).

##### **a. Types des EJB :**

Il existe trois types d'EJB :

- les beans de session (session beans)
- les beans entité (les entity beans)
- Depuis la version 2.0 des EJB, il existe un troisième type de bean : les beans orientés message (message driven beans).

Les session beans peuvent être de deux types :

- Sans état (stateless) ou avec état (stateful).
- Les beans de session sans état peuvent être utilisés pour traiter les requêtes de plusieurs clients.
- Les beans de session avec état ne sont accessibles que lors d'un ou plusieurs échanges avec le même client.

Les beans entités assurent la persistance des données.

Il existe deux types d'entity bean :

- persistance gérée par le conteneur (CMP : Container Managed Persistence). Avec un bean entité CMP (container-managed persistence), c'est le conteneur d'EJB qui assure la persistance des

données.

- persistance gérée par le bean (BMP : Bean Managed Persistence). Un bean entité BMP (bean-managed persistence), assure lui-même la persistance des données grâce à du code inclus dans le bean.

Message-driven Bean

- Les beans orientés messages (message driven beans) peuvent être utilisés dans un contexte d'appels asynchrones.

Des avantages sont acquis :

- Les services (system level) sont assurés par le conteneur EJB
- Encapsulation de la logique métier
- Réutilisation de la logique métier

## **b. Session Bean**

Un bean session est un type d'EJB qui :

- implémente l'interface SessionBean
- représente l'état d'un seul client dans un serveur.

Il existe deux types de beans session :

- Stateless. (sans état)
- Stateful (avec état)

### **Stateless**

Utiliser un bean session sans état (stateless) si :

- Le bean n'enregistre pas des données d'un client
- le bean doit faire un appel de méthode.
- le bean est utilisé pour extraire des données d'une base de données sans persistance dans la session.
- Pour consulter en lecture seule des données persistantes

### **Stateful**

Utiliser un bean session avec état (stateful) si :

- Le bean enregistre des données d'un seul client particulier pendant la durée d'appel de plusieurs invocations de méthodes (comme une session d'un navigateur Web).
- Le même Bean est utilisé pour servir tous les appels du même client

## **c. Cycle de vie Stateless Session Bean :**

Un java bean stateless a deux états possibles ; soit il n'existe pas ( "état inexistant" ) soit il est dans

un "état prêt"; il ne peut pas être dans un "état passif »

1. si le besoin de se servir d'un java bean session stateless se fait sentir, le conteneur d'EJB crée et maintient un pool de java beans session stateless, exécute toutes les injections de dépendance et invoque la méthode annotée avec `@PostConstruct` si elle existe.
2. Le java bean est, à partir de ce moment, dans l' "état prêt " à l'emploi (i.e qu' un client peut faire appel à ce java bean).
3. A la fin du cycle de vie, le conteneur d'EJB appelle la méthode annotée avec `@PreDestroy` (s'il en existe une) et ainsi l'instance du java bean est mise à la disposition du ramasse miette (garbage collection) .

### Interface Local ou Remote

```
@javax.ejb.Local
@javax.ejb.Remote
import javax.ejb.Remote;
@Remote
public interface    CompteSessionRemote
{
public Compte add(Compte compte);
public boolean sub(String idCompte);
}
```

Pour un Client local typiquement une servlet ou une JSP colocalisée est sur le même serveur que le bean et la demande de service et l'appelle ce fait par un mécanisme dit "injection" de dépendance ,et l'attribut est du type de l'interface annoté `@EJB`.

Pour un client distant d'abord la récupération de la référence vers l'annuaire JNDI,ensuite la recherche du bean dans l'annuaire puis l'appel des méthodes du bean .

#### d. Cycle de vie Stateful Session Bean :

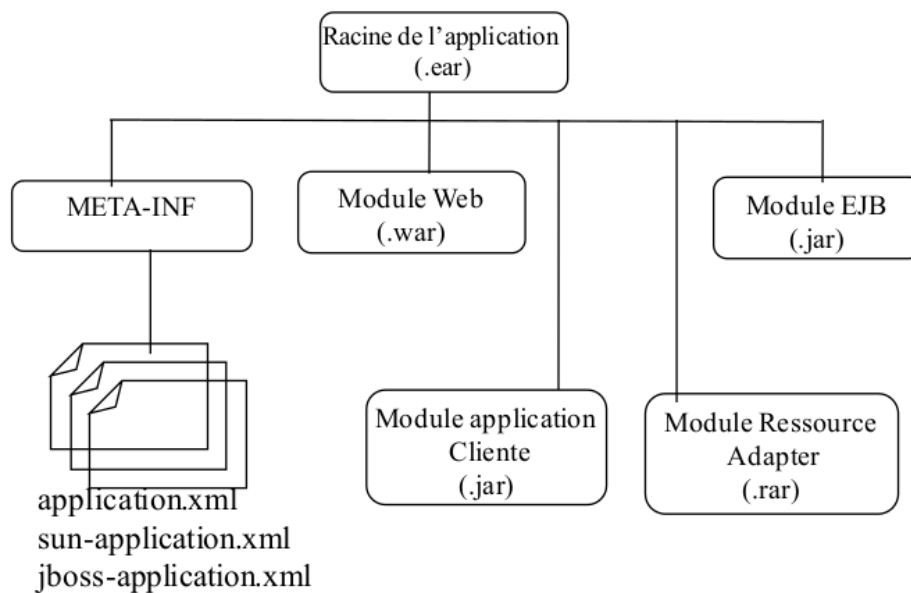
- 1.le client initie le cycle de vie d'un bean session stateful en obtenant sa référence. Ainsi le container exécute toutes les Injections de Dépendance, invoque la méthode ayant l'annotation `@PostConstruct` s' il en existe une;
- 2.à partir de là le bean session stateful passe à "l'état prêt" i.e qu'il peut-être utilisé par un client.
- 3.Pendant le temps où le java bean est dans "l'état prêt", le conteneur d'EJB peut le faire passer dans un état passif.
- 4.A la fin du cycle de vie du java bean, le client invoque la méthode avec l'annotation `@Remove`, et le conteneur d'EJB quant à lui fait appel à celle ayant l'annotation `@PreDestroy`;

#### e. Le packaging :

Un fichier d'archive d'entreprise (EAR) se compose essentiellement d'un Module EJB (.jar) ,Module

web (.war) Application cliente (.jar) Module 'Resource Adapter' (.rar)

- Chaque fichier d'archive (ear, war, rar ou jar) contient un descripteur de déploiement (fichier xml)



**Figure II.7: Le Packaging EAR**

Le descripteur de déploiement sert à définir les paramètres de déploiement d'une application (ear) ou d'un module , Par Exemples :

- **application.xml** : déclarer l'ensemble des modules intégrés dans un fichier (.ear)
- **ejb-jar.xml** : configurer les EJB dans un module d'un fichier (.jar)
- **web.xml** : configurer les paramètres du module web (servlets, JSP, TagLib..) d'un fichier (.war)

#### **II.4.3 Abstract Factory Design Pattern :**

Une Abstract Factory est une interface permettant de créer des familles d'objets sans avoir besoin de spécifier la classe concrète qui est utilisée au moment de leur création.

##### **a. Factory Design Pattern :**

Une Factory (fabrique en français), sans la notion d'Abstract, est en soi un Design Pattern. Ainsi, elle a pour but de créer des objets. Encore plus intéressant, elle va produire des objets qui vont dépendre de l'environnement dans lequel ils sont créés : elle s'adapte au client qui a commandé l'objet. Pour ce faire, l'objet est créé à partir d'une classe mère abstraite qui va s'adapter au contexte de la création par polymorphisme.

Le client utilise la Factory plutôt que l'instanciation directe avec le mot d'appel *new*! Ainsi, la Factory nous donne une grande flexibilité car nous ne sommes pas obligés d'effectuer des modifications dans tout notre code à chaque fois que l'on rajoute une nouvelle solution à notre logiciel : on a juste à changer la Factory.

#### **b. Factory VS Abstract Factory :**

Une Abstract Factory n'est autre qu'une super-factory, autrement dit, une Factory de Factory. Ainsi, cette super-factory va créer non pas un objet mais une famille d'objets liés entre eux. Pour cela, on dispose d'une super-classe abstraite dont les éléments interdépendants sont instanciés par une des sous-classes. Autrement dit, l'Abstract Factory dispose d'une sous-factory pour chaque variante de ses objets.

On imagine une usine fabriquant des comptes de deux types : des comptes particulier et des comptes pro. Cependant, un compte, qu'il soit particulier ou pro, possède à peu près les mêmes attributs : un id, pseudo, une date de création ect. Chacun de ces composants possède des pattern de base : toutes les pseudo ont un nombre de caractères par exemple. Ainsi, on voit bien que si les comptes possèdent les mêmes composants, la compatibilité des composants entre eux est très importante. On pourrait ainsi parler de familles de composants : les composants compte particulier d'un côté et ceux de compte pro de l'autre : polymorphisme d'une classe abstraite.

#### **Conclusion :**

Comme une boîte d'outils dans n'importe quel atelier ,dans ce chapitre on a traité les outils nécessaires utilisés dans le développement , les qualités de chaque outils sont pris en considération dans le choix , et en conséquence l'architecture peut être décidé d'une façon claire et solide .

# **Chapitre III**

## **Analyse et Conception du système**

### III.1 Introduction

Lors de ces dernières années, les systèmes de suivi sont devenus d'actualité. Dans notre projet de fin d'étude, nous nous sommes intéressés aux systèmes de suivi de véhicules de transport de marchandise. Pour la conception de notre projet, nous avons analysé les besoins, identifié et décrit les cas d'utilisation, pour ainsi réaliser le système WinRah.

Notre contribution se caractérise par des processus de niveau applicatif et de niveau de données.

Le niveau applicatif s'appuie essentiellement sur les diagrammes de modélisation :

- Après que le diagramme de cas d'utilisation est élaboré et les différents acteurs sont identifiés, nous traduisons chaque cas d'utilisation à un scénario qui devient l'objet d'une modélisation dans un diagramme de séquence puis dans un diagramme d'activités.
- Comme synthèse diagramme de classes sont établies.

Durant la partie analyse nous avons pu identifier les données nécessaires et indispensables au bon fonctionnement de l'application, Le niveau des données concerne l'organisation conceptuelle, logique et physique des données manipulées. En choisissant une architecture qui convient, et à travers la conception du niveau, nous allons sélectionner des classes significatives permettant d'élaborer la conception des bases de données.

Ce présent chapitre se divise en deux parties. La première partie concerne l'analyse du système et la deuxième partie la réalisation du système WinRah.

### III.2 Analyse du système :

#### III.2.1 Besoin fonctionnel

La réalisation d'un système dépend des besoins utilisateurs. Un système doit offrir un ensemble de fonctionnalités pour répondre aux besoins utilisateurs. Le système WINRAH doit satisfaire les besoins fonctionnels suivants :

##### **a. Le module application de bureau (ERP)**

Ce module assure le paramétrage de notre plate-forme. Cette partie intégrera : la gestion des administrateurs, la gestion des comptes clients professionnelles, la gestion des comptes clients particuliers, et la gestion des trois types de trackeurs. La gestion des comptes se traduit par l'ajout, la caractérisation, la recherche l'affectation et la suspension des comptes ; comme l'ajout, l'affectation et la caractérisation des trackeurs.

##### **b. Le module application :**

Ce module permet d'avoir une vue générale sur la géolocalisation des trackeurs en temps réel à un utilisateur connecté. Cette vue générale sur la géolocalisation concerne la position actuelle de tous

les trackers et leur déplacement dans le temps.

**c. Le module application Mobile :**

Vue la masse des utilisateurs des module PDA en générale et spécialement les téléphones mobiles, donc un module utilisateur assurant les mêmes fonctionnalités que l'application web est nécessaire. Ce module assure la géolocalisation, le contrôle du service d'un trackers, en plus de la fonctionnalité web une consultation direct de la géolocalisation d'un tracker via le code QR.

**III.2.2 Besoins non fonctionnels**

Tous les systèmes d'information à un certain point dans leur cycle de vie doivent considérer des besoins non-fonctionnels et leurs tests. Pour certains projets, ces besoins demanderont un travail très important et pour d'autres un contrôle rapide suffira. Les besoins non fonctionnels de notre système comme suit :

**a. Performance :**

Le système doit assurer un temps de réponse minime, tout en répondant aux exigences de l'utilisateur, pour tous ses modules.

**b. Sécurité :**

Les données recueillies par notre système ne doivent être accessibles que par le personnel autorisé, et l'accès doit établie en utilisant les protocoles de sécurité.

**c. Portabilité :**

Pour la plupart des modules logicielles la portabilité est un facteur prioritaire vue que notre système est destiné à des utilisateurs particuliers et professionnelles

**d. Ergonomie :**

Les interfaces doivent être ergonomiques et simples à utiliser. Les informations doivent être lisibles et facilement interprétables. Les actions et le système doit offrir une prise en main rapide et facile sans beaucoup d'efforts.

**e. Modularité du code :**

Notre système doit être extensible et lisible afin de simplifier la maintenance et pour cette raison le code source est bien structuré.

**III.2.3 Identification des cas d'utilisation :**

Chacune des fonctionnalités du système mène à une exigence dans le cas d'utilisation. Une fonctionnalité est une suite d'actions et d'événements qui se déroulent pour atteindre un état final et chaque suite (actions, événement) nous donne un scénario qu'on le modélise par les diagrammes

reconnus d'état et d'activité.

**a. Client particulier**

**Le Tableau III-1 décrit les cas d'utilisations Client particulier**

N°	Cas d'utilisation
1	S'identifier
2	Consultation de la localisation de l'ensemble des Trackers associés
3	Consultation détaillée du parcours d'un Tracker sélectionné
4	Paramétrage d'un tracker sélectionné

**Tableau III.1 - Description des cas d'utilisations du Client particulier**

**b. Client Professionnelle**

**Le Tableau III-2 décrit les cas d'utilisations Client Professionnelle**

N°	Cas d'utilisation
1	S'identifier
2	Consultation de la localisation d'un ensemble de Trackers non affectés
3	Consultation détaillée du parcours d'un Tracker sélectionné
4	Paramétrage d'un tracker sélectionné
5	Enregistrer un Client Fils
6	Affectation d'un tracker a un Client fils

**Tableau III.2 - Description des cas d'utilisations du Client Professionnelle**

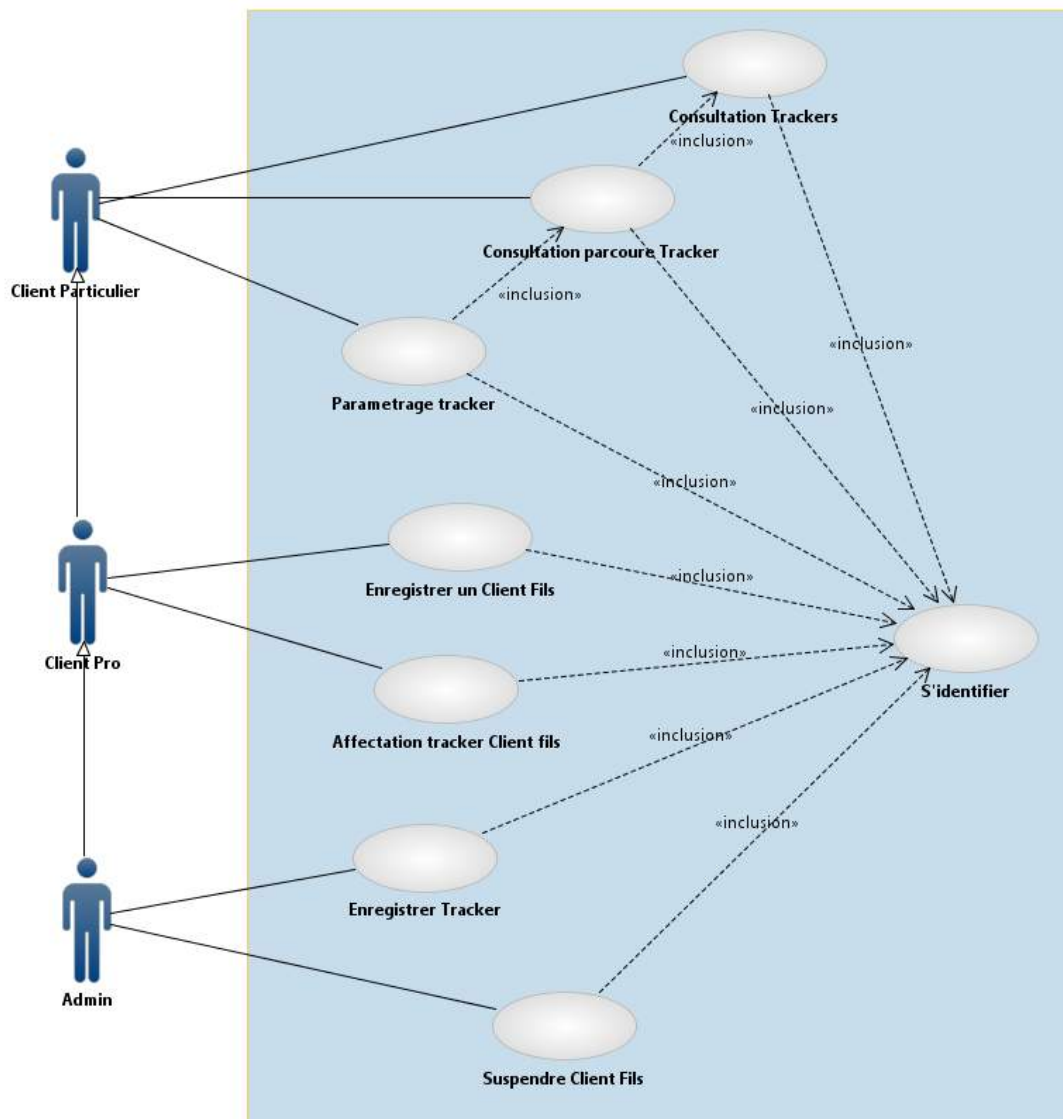
**c. Administrateur**

**Le Tableau III-3 décrit les cas d'utilisations Administrateur**

N°	Cas d'utilisation
1	S'identifier
2	Consultation localisation de l'ensemble des Trackers non affectés
3	Consultation détaillée du parcours d'un Tracker sélectionné
4	Paramétrage d'un tracker sélectionné
5	Enregistrer un Client Fils
6	Suspendre un Client Fils
7	Enregistrer un Tracker
8	Affectation d'un tracker à un Client fils

**Tableau III.3 - Description des cas d'utilisations Administrateur**

#### d. Diagramme des cas d'utilisation



**Figure III.1 – Diagramme Général de cas d'utilisation**

#### III.2.4 Descriptions de cas d'utilisations

Un cas d'utilisation est la description d'un ensemble de séquences d'actions qu'un système effectue pour produire un résultat observable à un acteur. Un cas d'utilisation représente une exigence fonctionnelle de votre système dans son ensemble. Les diagrammes de cas d'utilisation décrivent ce qu'un système fait du point de vue d'un observateur externe. L'accent est mis sur ce qu'un système fait, plutôt que sur la façon dont il le fait.

##### a. Description du cas d'utilisation « S'authentifier »

Cas d'utilisations « S'authentifier »

<b>Titre</b>	S'authentifier
<b>Acteur</b>	Client Particulier, Client Professionnel, Administrateur
<b>Résumé</b>	Un acteur s'identifie en saisissant le pseudo et mot de passe pour accéder à son espace.
<b>Descriptions des scénarios</b>	
<b>Préconditions</b>	L'utilisateur doit être créer par l'administrateur (Client Professionnel) ou créé et activé (Client Particulier)
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. Le système affiche le formulaire pour saisir les pseudos et le mot de passe</li> <li>2. L'acteur saisit ses coordonnées.</li> <li>3. le système contient une application web et application mobile. Dans les deux cas, la carte avec les marqueurs des trackeurs associées s'affiche</li> </ol> <p>- cas ERP pour Client Pro et Administrateur le tableau de bord s'affiche et exception pour Client Particulier</p>
<b>Enchaînement alternatif</b>	<p>Un ou les deux coordonnées sont incorrecte</p> <ol style="list-style-type: none"> <li>1- un message d'exception s'affiche indiquant que les coordonnées sont incorrecte</li> <li>2-Le cas d'utilisation reprend de l'étape 1 du scénario nominal</li> </ol>
<b>Postconditions</b>	L'utilisateur est authentifié et accède aux fonctionnalités dédiées.

**Tableau III.4 - Description du cas d'utilisation « S'authentifier »**

**b. Description du cas d'utilisation « Consultation Parcours Tracker »**

<b>Cas d'utilisations « Consultation du Parcours d'un Tracker »</b>	
<b>Titre</b>	Consultation du Parcours d'un Tracker
<b>Acteur</b>	Client Particulier, Client Professionnel, Administrateur

<b>Résumé</b>	Un acteur sélectionne un marqueur sur la carte
<b>Descriptions des scénarios</b>	
<b>Préconditions</b>	L'utilisateur peut s'authentifier sur le service web ou sur l'application mobile et ainsi avoir accès à la carte de trackeurs
<b>Scénario nominal</b>	L'application web ou l'application mobile affiche les marqueurs des trackeurs associer à un utilisateur connecté. 1- L'acteur click sur le marqueur représentant un trackeur. 2- la carte marque et affiche les points du parcours du trackeurs.
<b>Enchaînement alternatif</b>	Aucun
<b>Postconditions</b>	La carte avec les marqueurs de trackeurs

**Tableau III.5 - Description du cas d'utilisation «Consultation Parcoure Tracker»**

**c. Description du cas d'utilisation « Enregistrer un Trackeur »**

<b>Cas d'utilisations « <u>Enregistrer un Trackeur</u> »</b>	
<b>Titre</b>	Enregistrer un Trackeur
<b>Acteur</b>	Administrateur
<b>Résumé</b>	Un Administrateur saisit les informations d'un nouveau trackeur
<b>Descriptions des scénarios</b>	
<b>Préconditions</b>	L'utilisateur doit être authentifier dans l'application de bureau (ERP) pour avoir accès au tableau de bord administrateur
<b>Scénario nominal</b>	L'application de bureau (ERP) affiche le tableau de bord de l'administrateur 1- L'acteur click sur le bouton Add Tracker

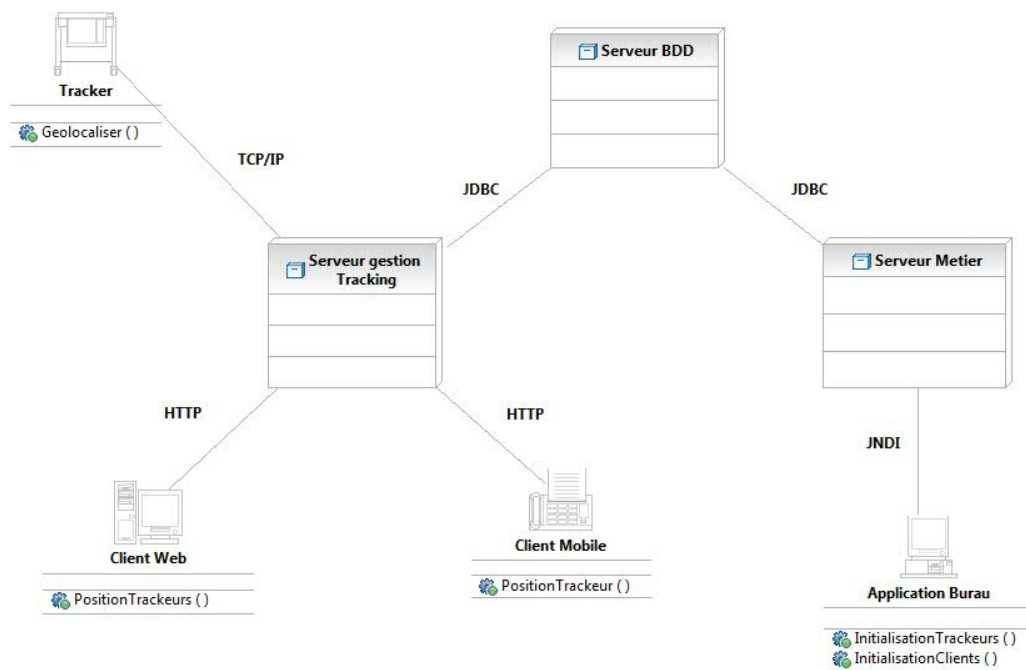
	<p>2- La fenêtre formulaire d'inscription de trackeur s'affiche</p> <p>3- L'acteur sélectionne le type de trackeur à enregistrer et click sur le bouton New</p> <p>4- L'acteur saisit les cordones du trackeur</p> <p>5- L'acteur valide les cordonnées en cliquant sur le bouton Add.</p>
<b>Enchaînement alternatif</b>	<p>-Le code du trackeur existe déjà : un message d'exception s'affiche</p> <p>-Un des cordonnées n'est pas valide : un message d'exception s'affiche</p>
<b>Postconditions</b>	<p>- L'acteur sélectionne le type de trackeur</p> <p>- L'acteur ferme le formulaire</p>

**Tableau III.6 - Description du cas d'utilisation « Enregistrer un Trackeur »**

### III.3 Conception du système « WinRah »

#### III.3.1 Diagramme de déploiement :

Les diagrammes de déploiement sont utilisés pour visualiser les processeurs matériels, les nœuds et les dispositifs d'un système, les liens de communication entre eux et l'emplacement des fichiers logiciels sur ce matériel.



**Figure III.2 – Diagramme de Déploiement de la Plate-forme WinRah**

Le diagramme de déploiement de notre application est représenté par le diagramme ci-dessus.

### III.3.2 Diagrammes de Séquences :

La plate-forme de notre projet de fin d'étude peut être modélisé en trois scenarios.

#### a. Diagrammes de séquences pour une application de bureau :

Dans les Diagrammes ci-dessous nous modélisons, les diagrammes de séquences de l'application de bureau pour un administrateur qui initialise un compte client particulier

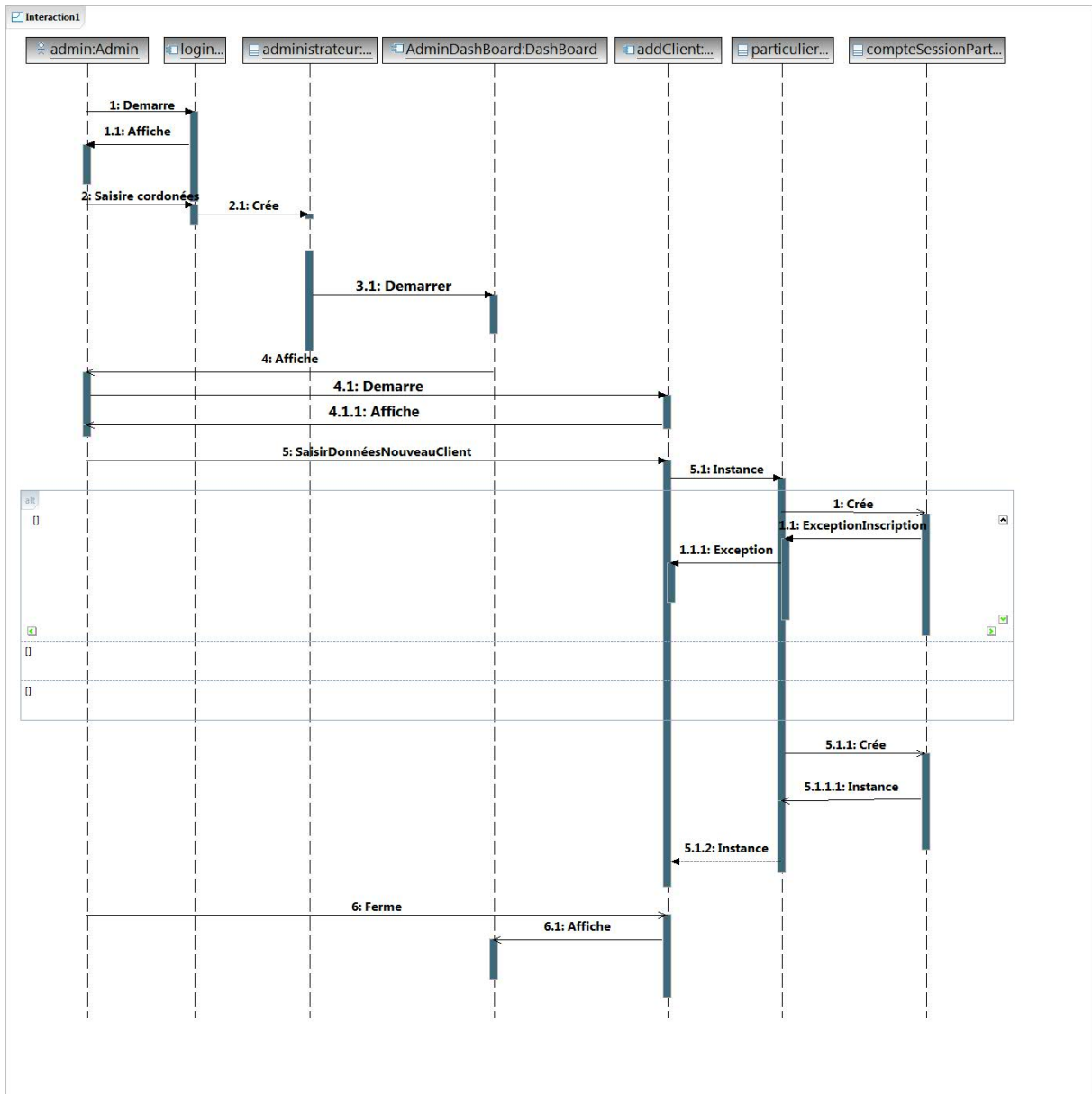


Figure III.3 – Diagramme de séquences pour la Création d'un Compte Particulier

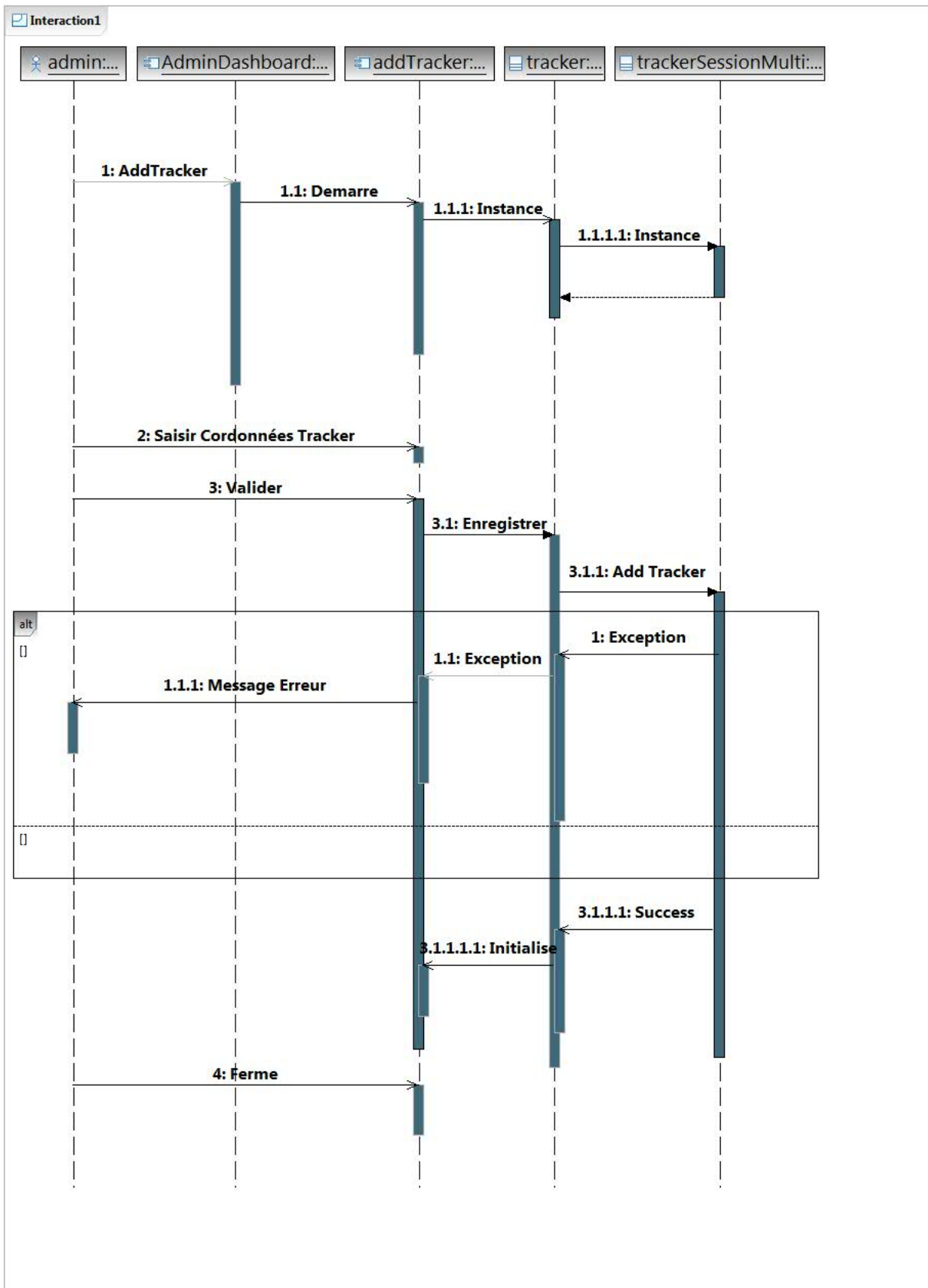


Figure III.4 - Diagramme de séquences pour la Création de TrackerMULTI

Le Diagramme ci-dessous modélise le scénario de l'association d'un Tracker Pro a un Compte Pro

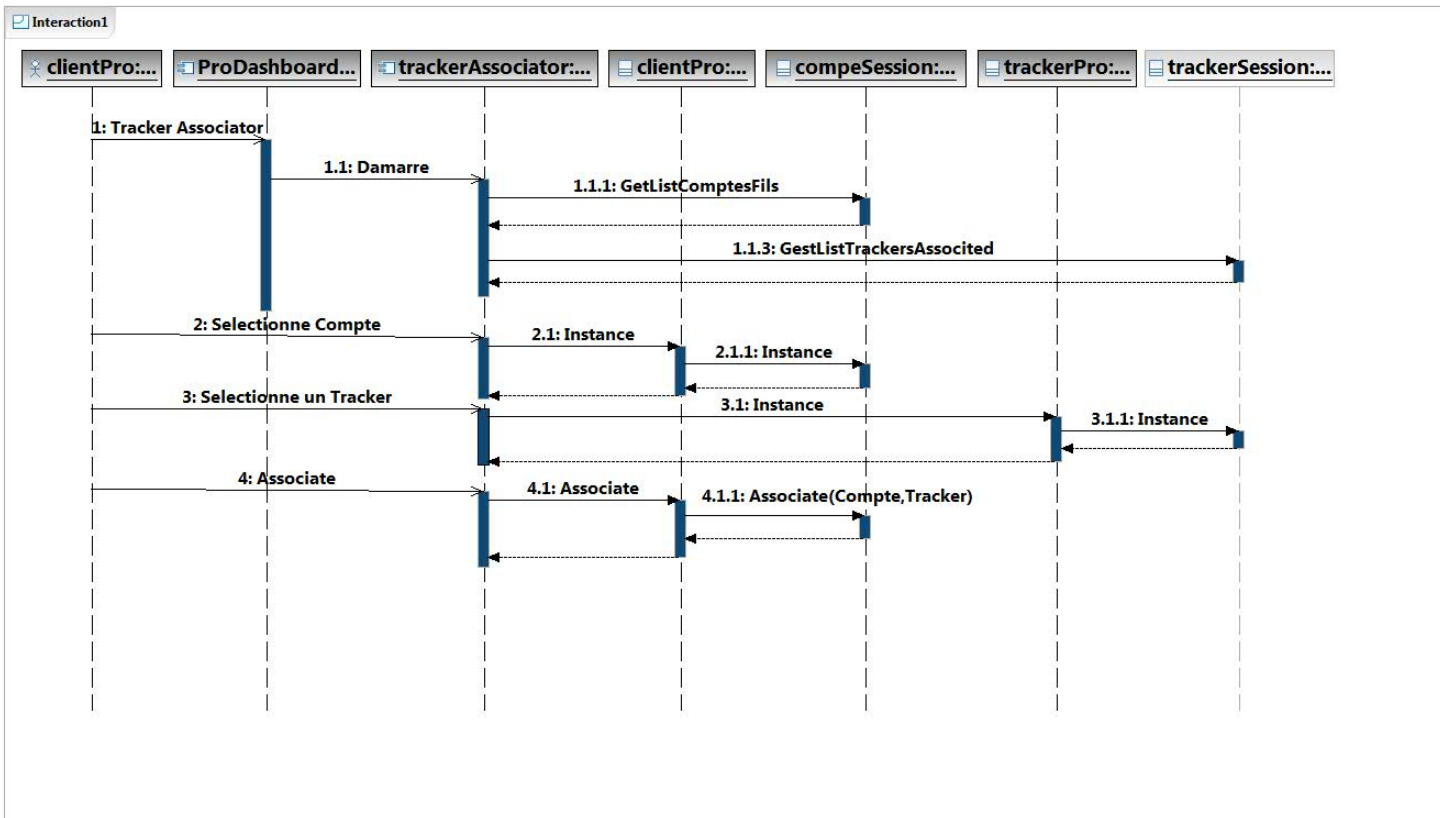


Figure III.5 – Diagramme de séquences pour l'association d'un trackeur à un ComptePro

Ainsi l'application bureau est l'outil pour la gestion initiale des acteurs principaux de la plateforme et elle est considéré comme la mise en marche du système.

**b. Diagramme de séquence Cas client Web :**

Après l'inscription d'un trackeur et son association à un Compte Client, la géolocalisation est acceptée par le serveur de gestion de Tracking et les données peuvent être stocké pour consultation. Nous pouvons modéliser les scénarios d'accès et de consultation de la géolocalisation par le diagramme de séquence ci-dessous.

Le Trackeur alimente la base de données par les données de géolocalisation indépendamment de toute interaction de la part de l'utilisateur, et le chargement de la nouvelle position se fait en boucle dans le temps.

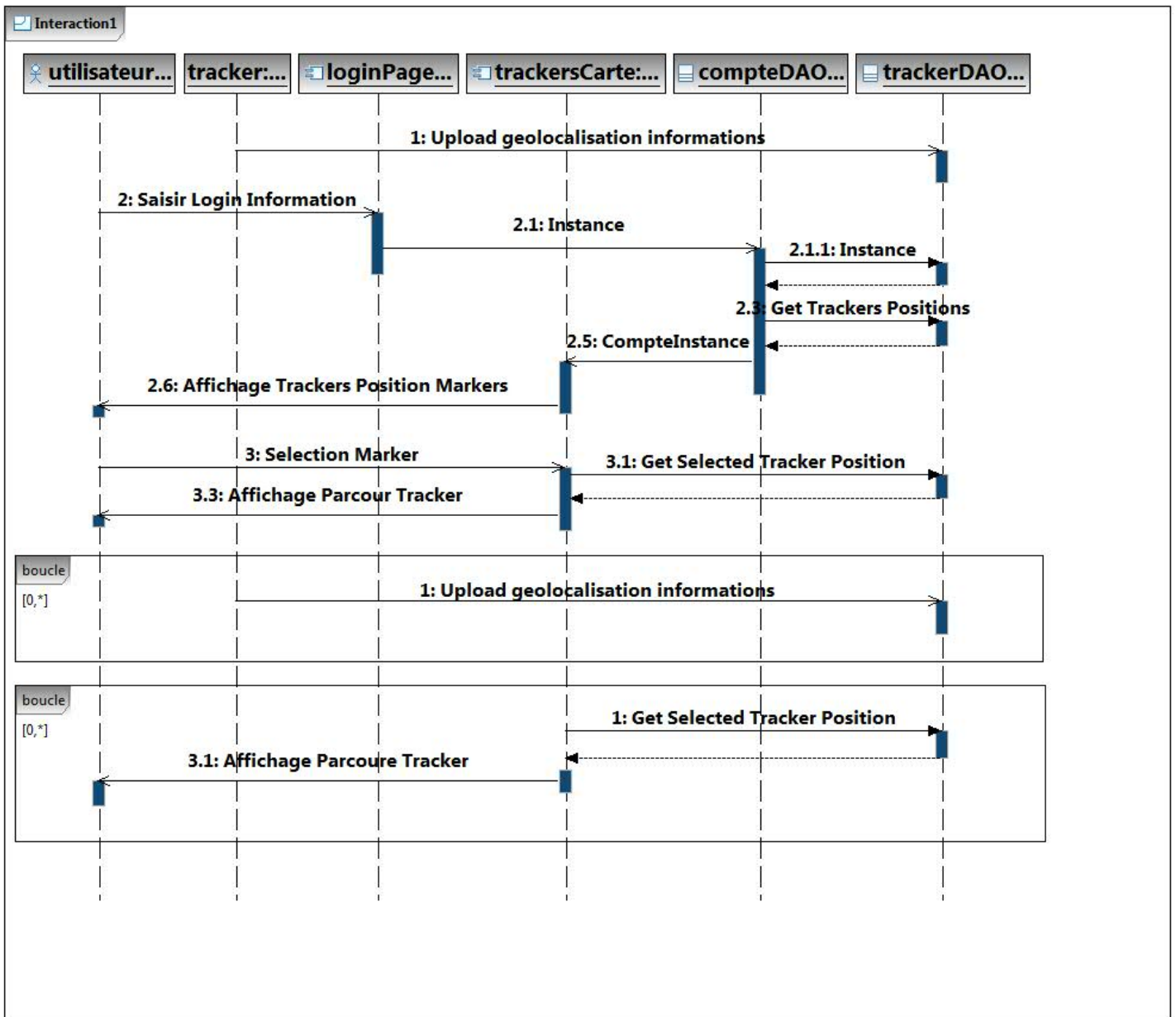
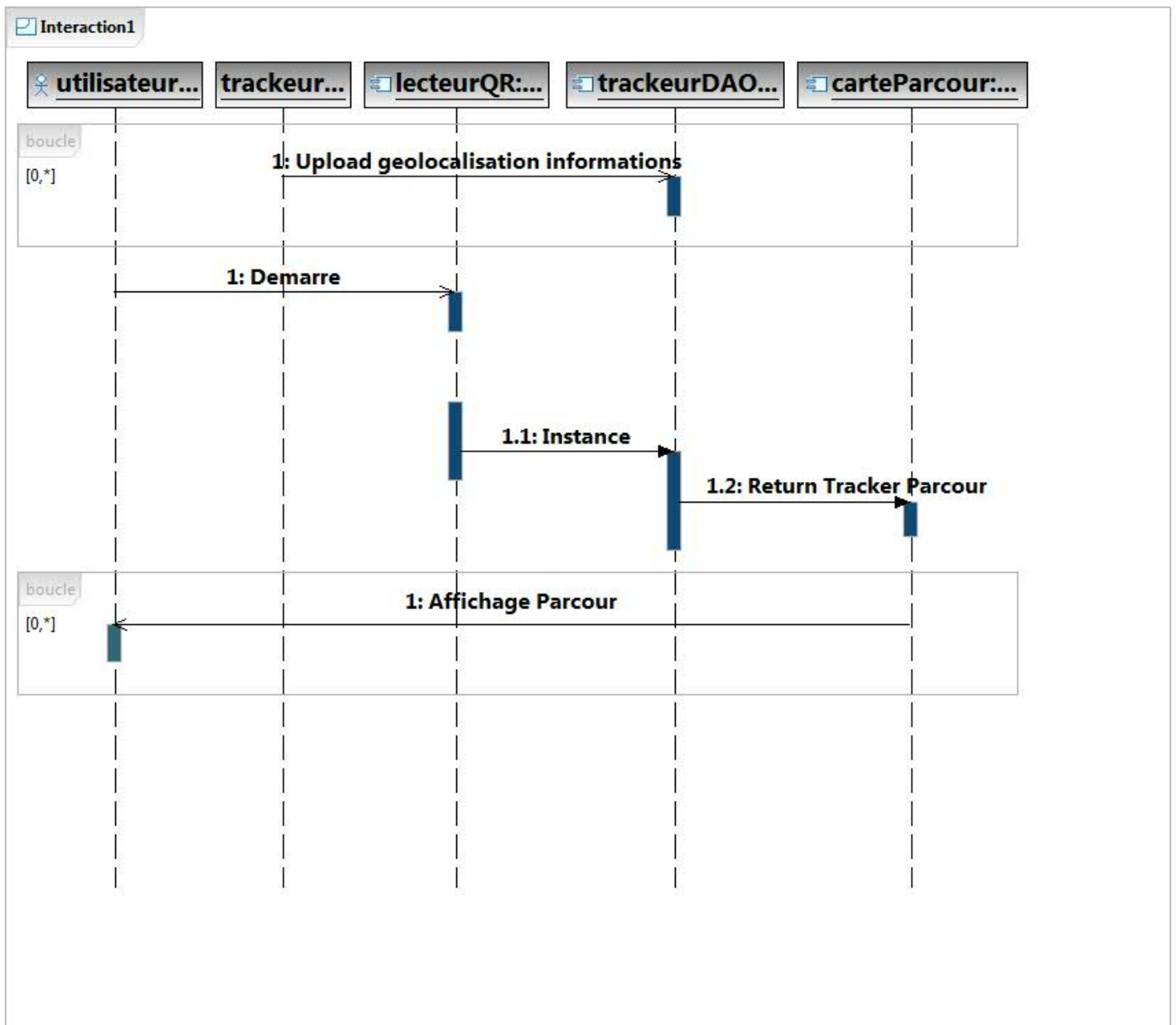


Figure III.6 – Diagramme de séquences de l'Interface de consultation Web

### c. Diagramme de séquence Cas client Application Mobile :

Le Diagramme de séquences pour l'application mobile est identique à celui de l'interface web pour un client qui utilise son compte utilisateur. Une autre fonctionnalité peut être ajoutée, est la consultation de la localisation en utilisant la lecture des coordonnées d'un trackeur sous forme de code QR généré et affiché dans la page de consultation de chemin du service web, cela indépendamment du compte utilisateur.

Ainsi ci-dessus le diagramme de séquences qui modélise le scénario de la consultation du parcours d'un trackeur à base de la lecture de son code QR



**Figure III.7 – Diagrammes de séquence de Consultation de localisation par QR (Application Mobile)**

### III.3.3 Diagrammes de classes :

Un diagramme de classe décrit clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.

Le concept Classe est abstrait, où toute entité peut être modéliser ; un immeuble, un véhicule une raquette, un bouton, une image ou tout autre élément peut être une partie d'un modèle pour avoir un code métier généré.

## a. Diagrammes de classes Au Module Web

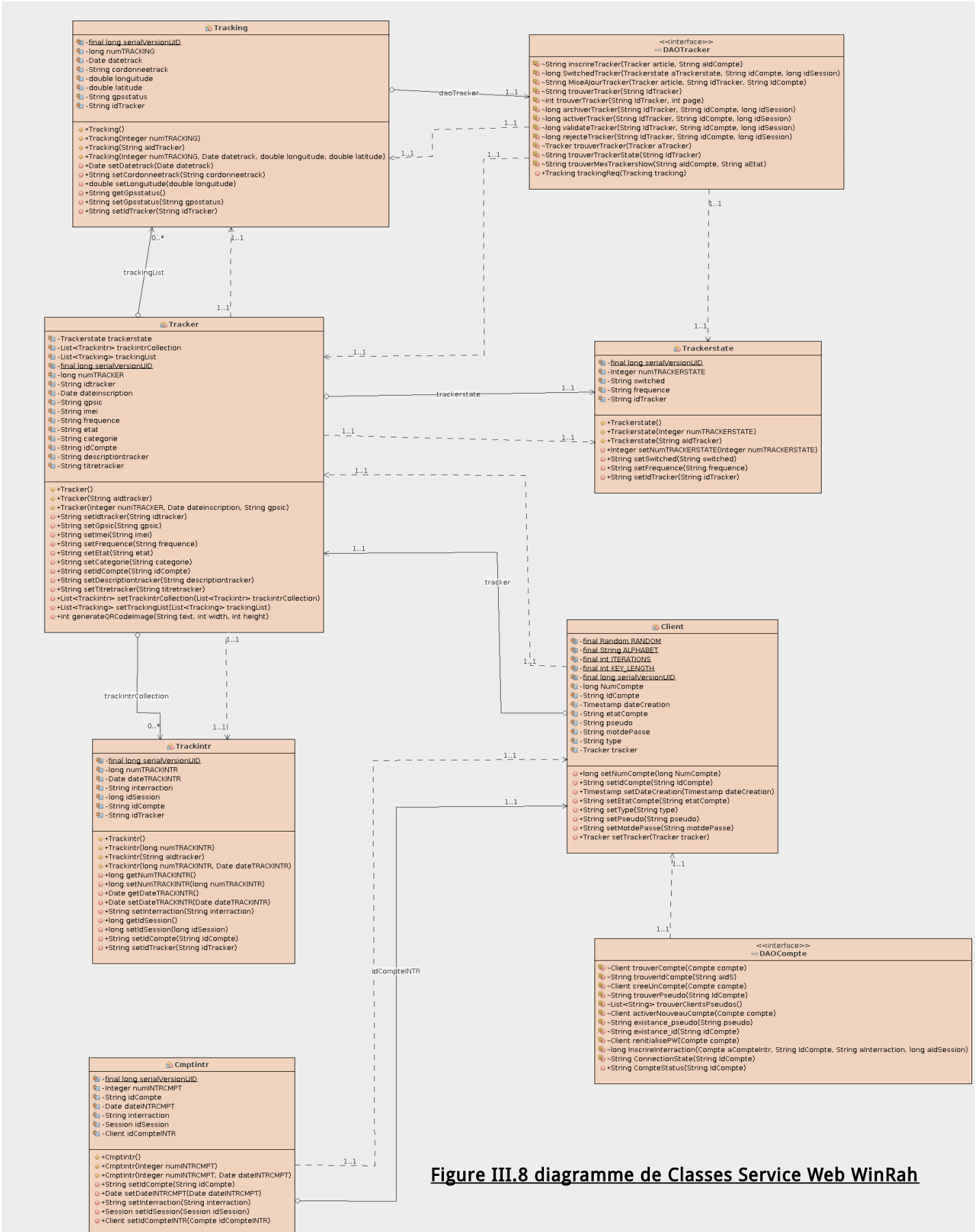
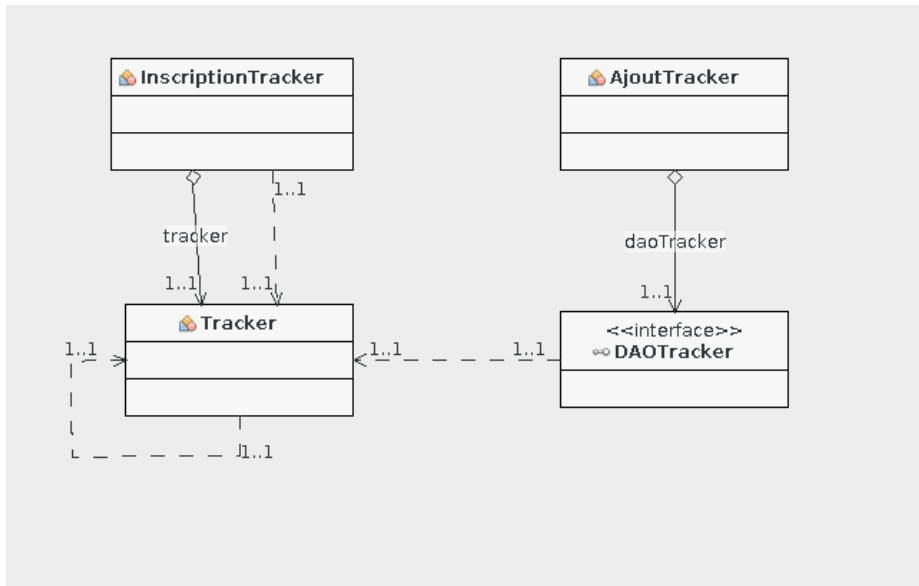


Figure III.8 diagramme de Classes Service Web WinRah

Les Classes du Service Web, par le biais des servlets, sont modélisées par le diagramme représenté dans la figure ci-dessus. Sachant que le choix été pour le pattern DAO à cause de la nécessité de l'existence d'une interface de bas niveau pour connecter le trackeur et une interface de niveau haut pour les clients, dans un environnement passage par objet.



**Figure III.9 – Diagramme De Classe Service WEB -Partie Ajout D'un Tracker-**

La Servlet **AjoutTracker** sert à l'instanciation d'un objet **DAOTrackerImp**, qui hérite de l'interface **DAOTracker**, responsable des requêtes sur la table **tracker** dans la base de données. L'objet **DAOTrackerImp** utilise des objets **Tracker** dans le processus de passage de données par objet. Un Objet **InscriptionTracker** est responsable de la validation des attributs venant des requêtes web, utilisé pour l'initialisation d'un objet **Tracker**, et génère une exception dans le cas où une attribut n'est pas valide.

**b. Diagrammes de classes Au Module Bureau :**

Pour avoir une bonne gestion des fonctionnalités fondamentales des acteurs de notre système, un administrateur a été chargé de la gestion de l'initiation et du suivie de ces acteurs par un module de bureau. On peut le définir comme une application bureau avec des fonctionnalités qu'on trouve dans un **ERP** , autrement dit un **pseudo ERP**.

Le choix architecturale été pour le pattern **Abstract Factory** déployé en utilisant L'EJB sous un **Wildfly** serveur de sessions **@Remote** avec **JNDI** .

Comme c'est montré dans le diagramme de classe ci-dessous deux **Abstract Factories** sont élaborées, une chargée des comptes et du niveau de privilège, et un deuxième chargée des Trackers. **L'instanciation ce fait au niveau de la Classe Abstraite ComptesWinRah .**

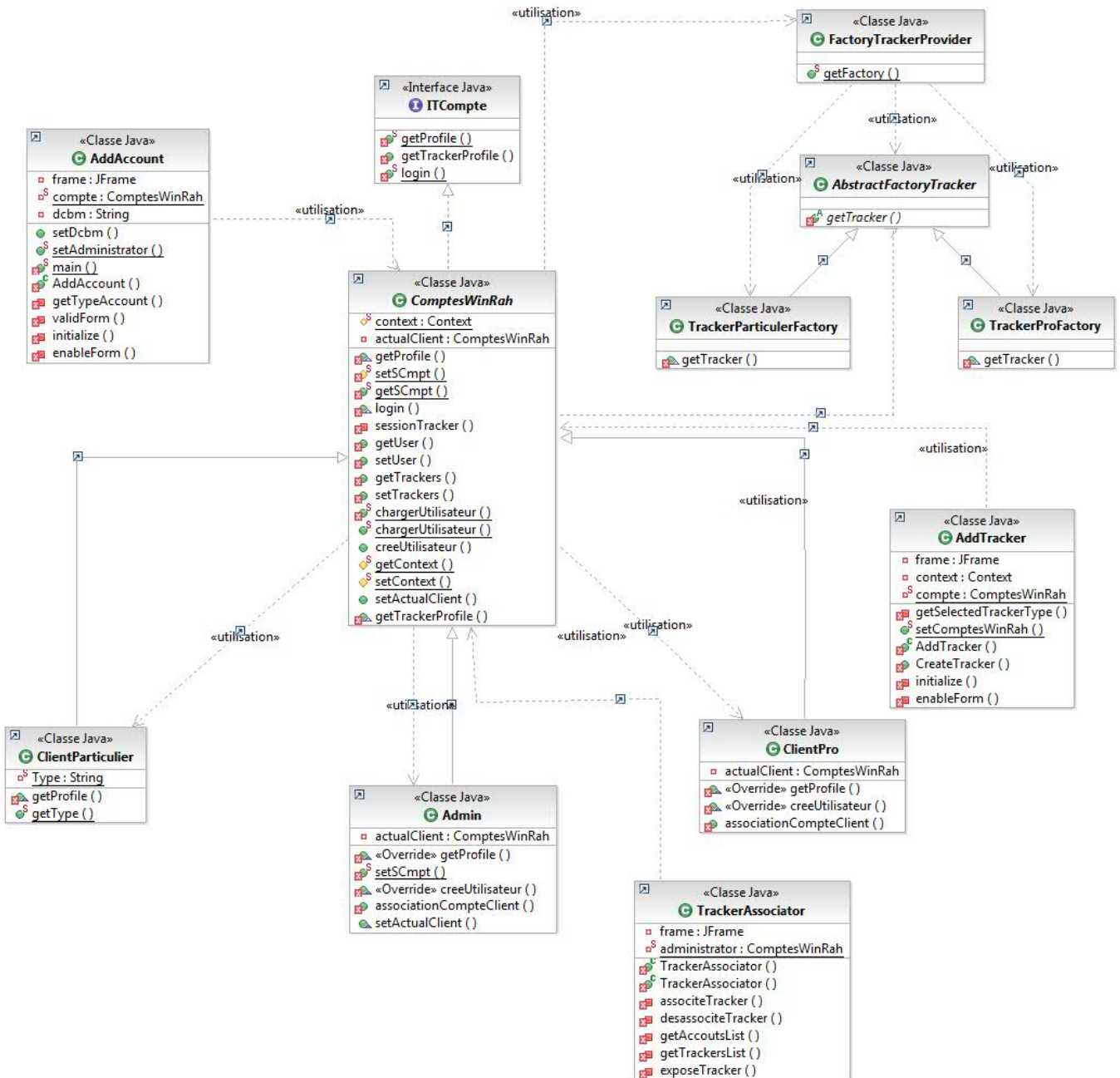
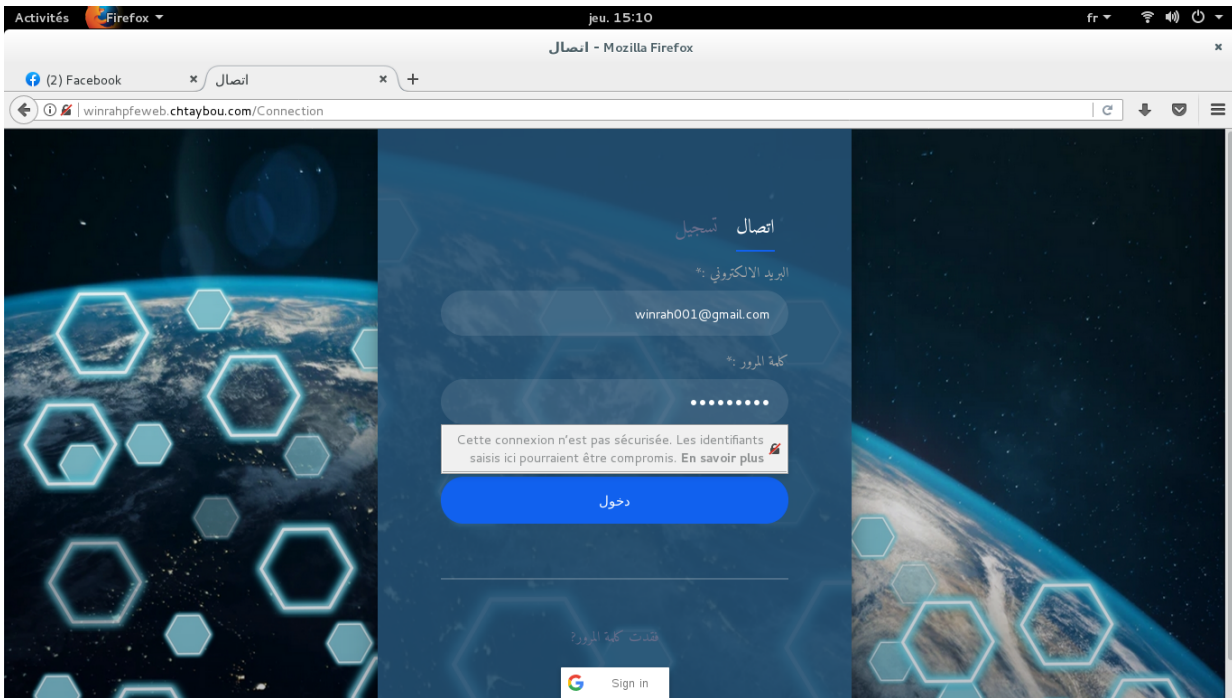


Figure III.10 – Diagramme De Classe Module Bureau

### III.4 Exploitation des modules de la plate-forme :

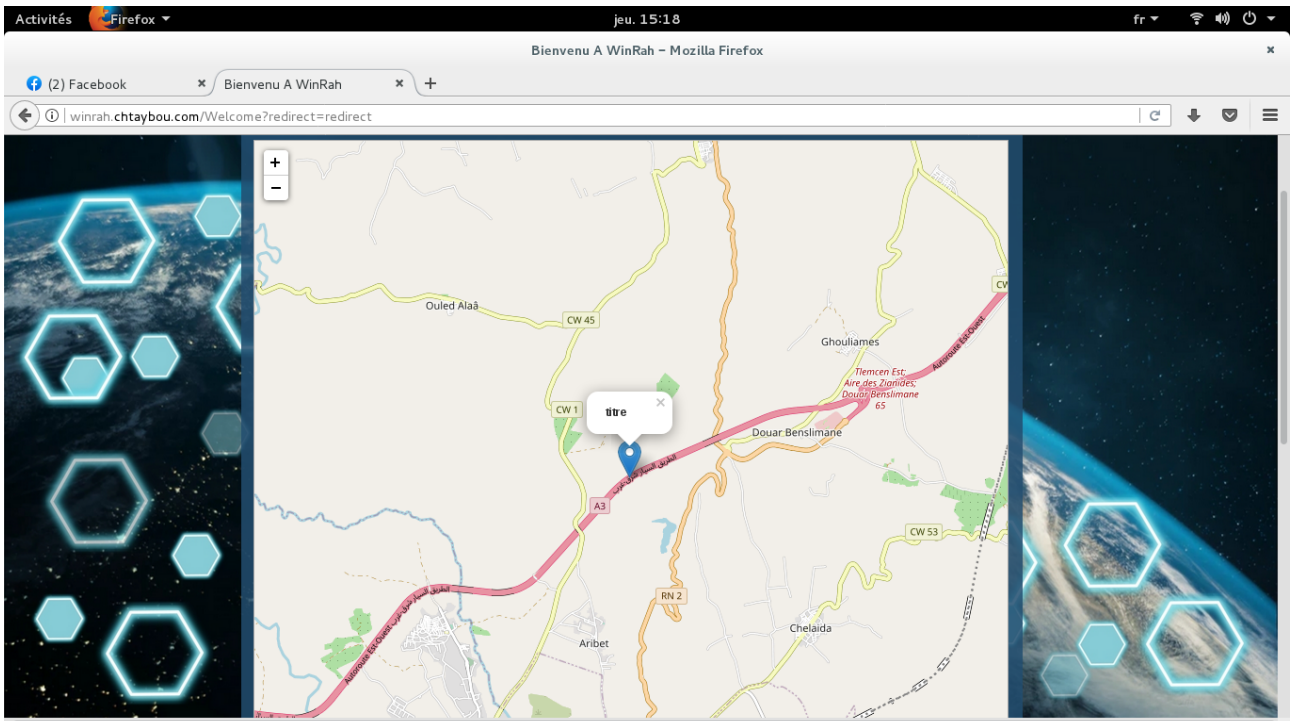
#### III.4.1 Exploitation de l'Application WEB :

On a réalisé trois interfaces que un client particulier exploitera toujours pour la consultation de la position de son/ses trackers ci dessous des captures d'écran pour chaque interface



**Figure III.11 – Interface de connexion au module WEB**

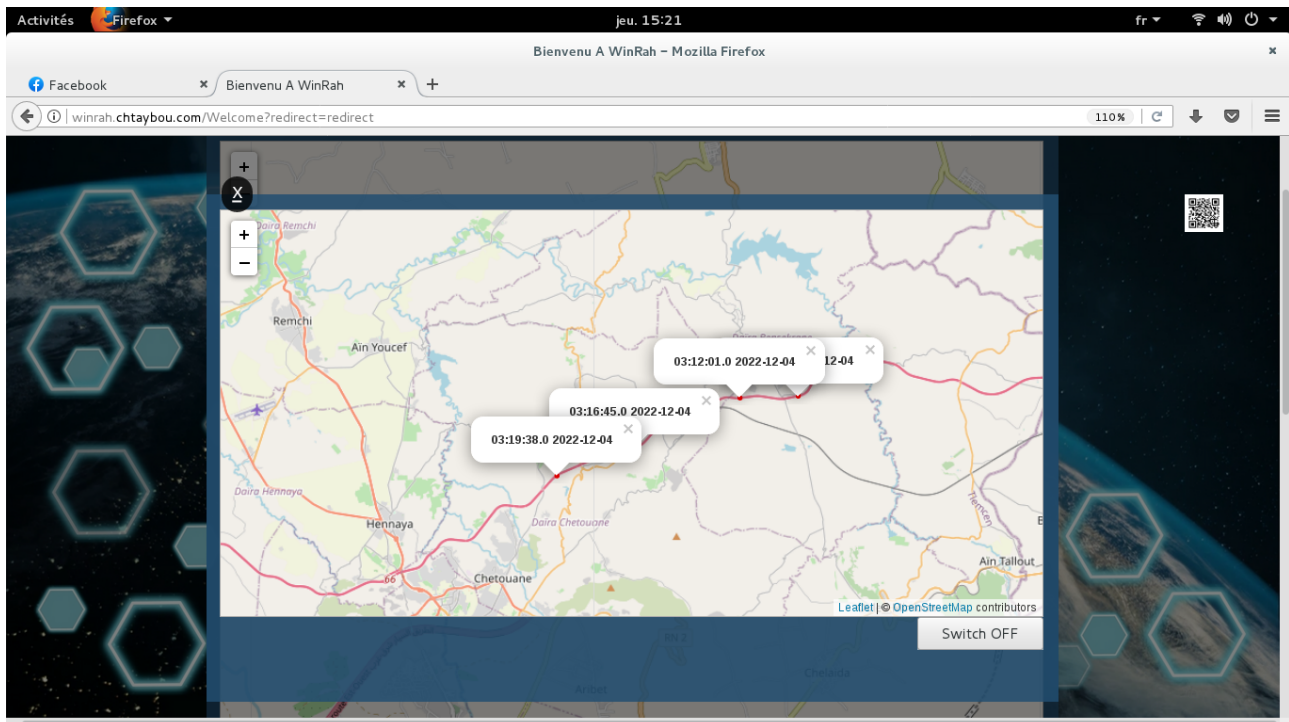
L'interface ci-dessus se compose d'un formulaire pour les coordonnées de connexion  
 Dans le cas de coordonnées erroné un message d'erreur s'afficher selon le type d'erreur  
 Dans le cas de coordonnées correcte une redirection se déclenche pour l'interface de visualisation  
 général représenté dans la figure ci dessous.



**Figure III.12 – Interface de Consultation générale des trackeurs au module WEB**

On peut voir la carte chargé en ligne du SIG selon la position de la dernière position inscrite par un  
 des trackeurs affecté a compte utilisateur en connexion

Ainsi pour chaque trackeur un marqueur pour la dernière position inscrite est affiché sur la carte avec le titre associé au trackeur. On passe au détail a propos du trackeur sélectionné en cliquant sur son marqueur sur une seconde carte comme ce voie dans la figure ci-dessou



**Figure III.13 – Interface de Consultation détaillée d'un trackeurs en WEB**

Chaque point représente une position de localisation avec son horaire,et un chargement continue de la dernière position ce produit après une courts laps de temps.

### III.4.2 Exploitation de l'Application Mobile :



**Figure III.14 – Interface de connexion au module Mobile**

Les interfaces dans l'application mobile sans de même approche que dans le module web et le but de son développement été a cause du faite que a nos jours les mobiles sont beaucoup plus populaire que les ordinateurs,et il sont plus facile a utilisé.

Un Fonctionnalité additionnelle est la connexion via le code QR que après son scanne l'interface de représentation détaille s'affiche comme on peut voir dans la figure ci-dessous .

On a ajouté aussi la possibilité de contrôle du chargement continue ,on peut l'activé ou le désactivé

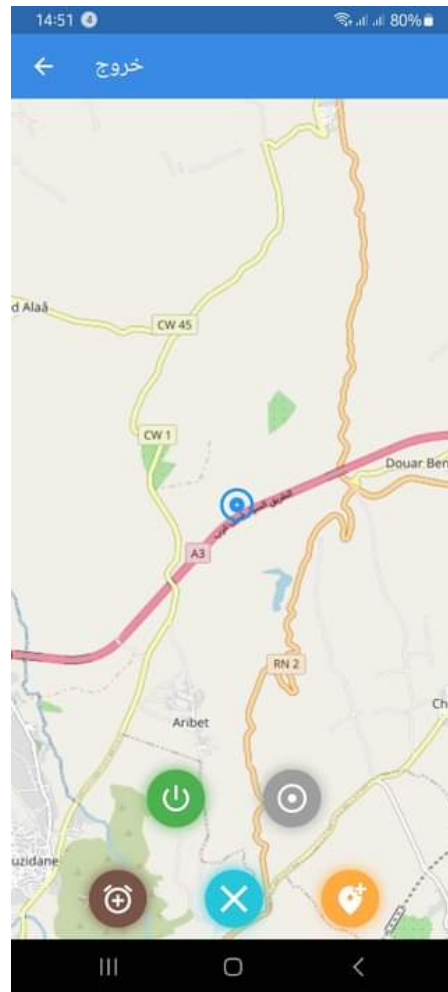


Figure III.15 – Interface de Consultation détaillée d'un trackeurs en Mobile

### III.4.3 Exploitation de l'Application de bureau :

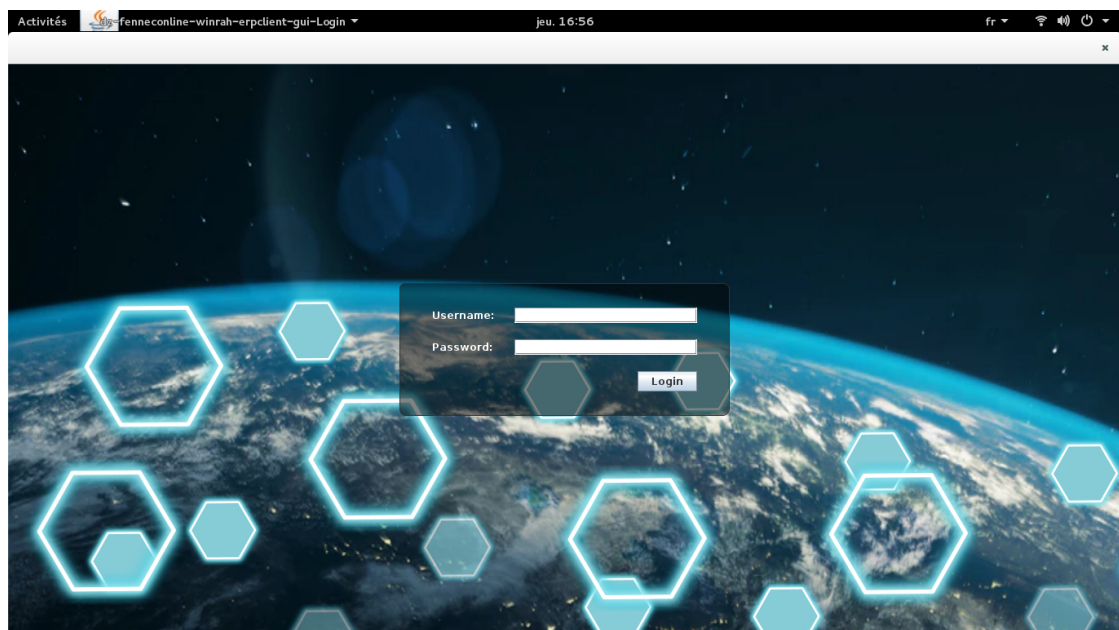
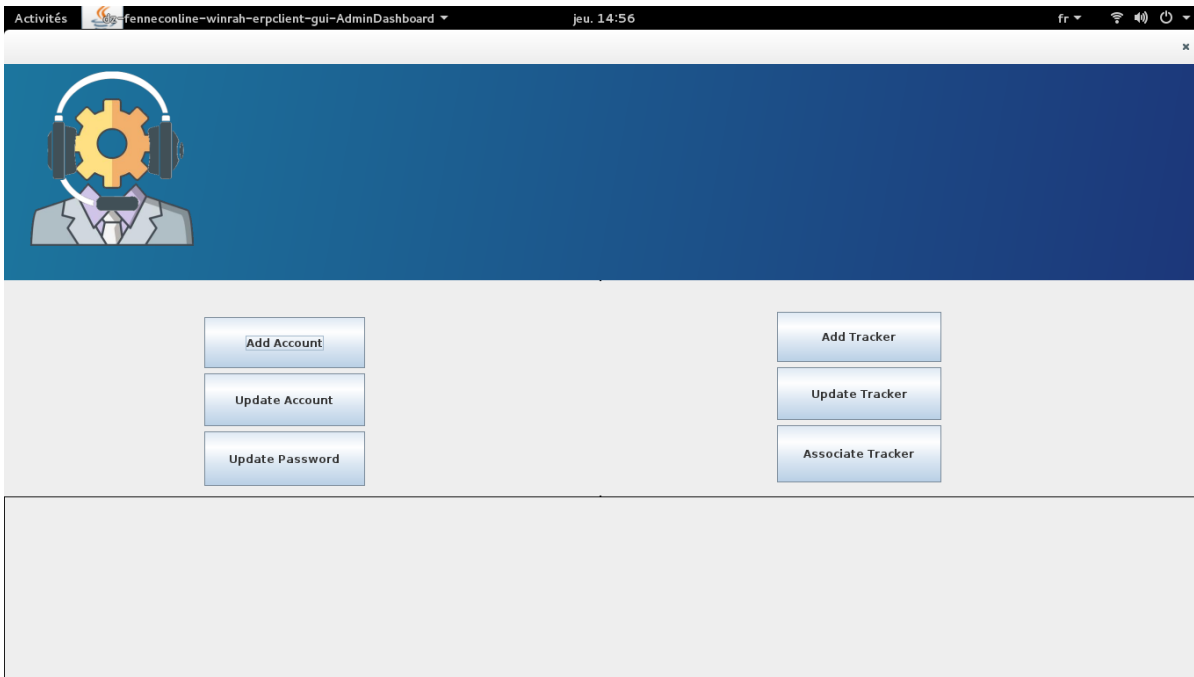


Figure III.16 – fenêtre de connexion au module Bureau

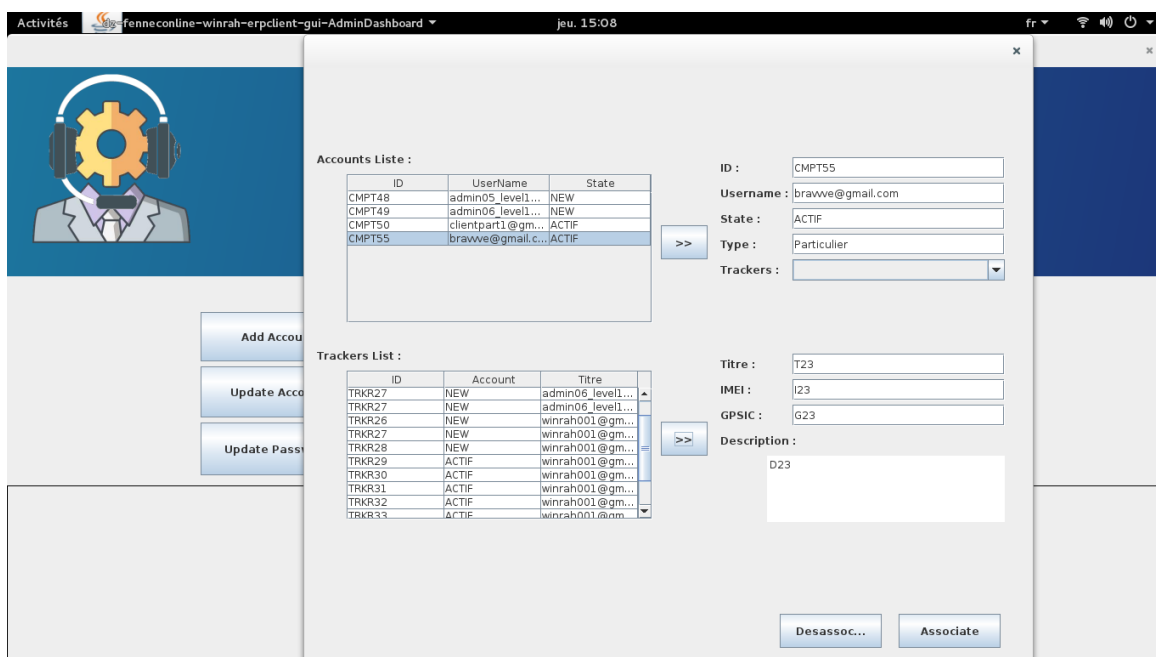
Ce module rentre dans le cadre d'un ERP pour le service de la gestion des réseau de tracking et multiple interface qu'on a développé . Et on a utilisé le bibliothèque Swing pour cela sous l'ide eclipse

Ci dessus la fenêtre de connexion .

Après la connexion un tableau de bord doit être affiché,une remarque essentiel que ce module est que pour les compte de gestion et le client ne peut pas l'utilisé .



**Figure III.17 – Tableau de bord du module Bureau**



**Figure III.18 – fenêtre d'association des trackeurs au module Bureau**

Chaque bouton on l'utilise pour une opération sur les comptes,ou sur les traceurs, en exception le bouton associate Tracker fait la gestion des deux .

## **Conclusion**

Dans ce chapitre, nous avons présenté les besoins permettant de décrire les fonctionnalités du système de manière générale. Nous avons alors identifié les acteurs, les besoins fonctionnels et non fonctionnels du système, et nous avons réalisé un diagramme des cas d'utilisation. Cela nous a permis de faire des descriptions pour les cas d'utilisation. Après nous avons procédé à la mise en places des facteurs essentiels dans les modules du système d'un façon généralisé en premier lieu, et ensuite nous avons détaillé les sous-facteurs dans le but de faire l'abstraction traduite en code à la fin .

## **Conclusion générale :**

L'objectif principal de ce projet de fin d'étude était de concevoir et de construire un système efficace pour suivre la position ou les mouvements des véhicules à l'aide d'un système de positionnement basé sur le réseau GSM. Nous pouvons dire que l'objectif principal a été atteint.

En conclusion de ce manuscrit, nous avons pu atteindre l'objectif principal de ce projet de maîtrise en Génie Logiciel , étant l'acquisition de connaissances dans le domaine du suivi des mouvements de véhicules, Ces connaissances théoriques et pratiques acquises , aboutissant à la conception et au développement d'un système de travail pour l'organisation du client prévu.

Le rôle principal de notre système est de suivre et tracer l'emplacement du parcours des véhicule sur une carte de base. Notre système est extensible à de nouvelles applications et adaptable à des plates-formes différentes avec un minimum de modifications logicielles supplémentaires, car il a été développé à l'aide de modèles d'architectures connue.

Des améliorations avancées pourraient être apportées à notre système, telles que des moyens de communication considérablement améliorés tels que la précision GPS, des capteurs embarqués améliorés ou la miniaturisation des systèmes embarqués.

À des fins de démonstration avec ce projet pilote, le système effectue le suivi des mouvements pour seulement deux véhicules de suivi simultanément dans la région de Tlemcen. Or le système peut passer à l'échelle. C'est-à-dire, il a été conçu et développé de telle manière qu'il peut suivre n'importe quel nombre de véhicules simultanément n'importe où lorsqu'il existe une couverture GSM. Le système de suivi fonctionne avec succès et des travaux supplémentaires seraient nécessaires pour le mettre en œuvre en tant que produit de qualité commerciale et de base. La miniaturisation de l'unité de voiture serait nécessaire pour une mise en œuvre au niveau commercial.

## References Bibliographiques

- [1] Vehicle Tracking System Market to Grow Three-Fold Amid Growing Focus Towards Improving Vehicle and Passenger Safety, <https://www.bloomberg.com/>
- [2] Neil Ackroyd, Robert Lorimer ,Global Navigation: A GPS User's Guide, Livre,1994
- [3] Stephen Stelting, Olav Maassen,Applied Java Patterns,Livre,2002
- [4] Marty Hall, EJB3: Session Beans,Polycopier ,janvier 2023
- [5] Christopher Ritchie,WildFly Configuration, Deployment, and Administration ,Livre,2014
- [6] Luciano Manelli,Developing a Java Web Application in a Day: Step by Step ,Livre,2016
- [7] R. Dharani,Learn Java Design Patterns,Livre,2018
- [8] Partha Kuchana,Software Architecture Design Patterns in Java,Livre,2004
- [9] Martin W. Murhammer, Orcun Atakan, Stefan Bretz,Larry R. Pugh, Kazunari Suzuki, David H. Wood,TCP/IP Tutorial and Technical Overview,Livre,1998
- [10] Fu Cheng,Flutter Recipes Mobile Development Solutions for iOS and Android,Livre, 2019
- [11] Dale Wheat,Arduino Internals,Livre,2011
- [12] Flutter: Beautiful native apps in record time. <https://flutter.dev/>
- [13] GLENFORD J. MYERS,TOM BADGETT,COREY SANDLER,THE ART OF SOFTWARE TESTING ,Third Edition,2012
- [14] Hardware/software – Arduino. <https://www.arduino.cc/>

# **Annexes**

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة أوبكر بلقايد - تلمسان -

## Business Model Canevas

### نموذج العمل التجاري

تاريخ الإيداع: 2023/07/02

الطالب:

الاسم : عبد الرحمن

اللقب : الصغير

رمز المشروع : S\_22

اسم المشروع : Système de Gestion de Tracking GPS Orienté Transport Routier WinRah

المؤطر:

الإسم : مريم بشرى

اللقب : بن مهدي

الدورة الجامعية : 2023/2022

## 1- Value proposition



## 1- القيمة المقترحة

La Service de Tracking GPS WinRah est un système qui fait la collection et l'exposition des positions parcourue par un ou plusieurs trackers ,chaqu'un construit spécialement pour la plateforme, en temp réel

Grace a la facilité d'utilisation du module de gestion réseau administrateur/Trackers le client peut géré sont personnel vis-a-vis les trackers avec l'association des trackeurs au compte chargé de supervision

Le module web permet au client une souplesse d'accès a sont compte pour visualisé ses trackers associés

Une application mobile Téléchargeable a Google Play Store permet au client l'accès a son compte a l'aide des équipements smarts :Telephone,Tablette,Télévision

L'équipement trackeur est adapter a tous les operateurs de téléphonie mobile existant au marché algerien

D'autre part le trackeur est Independent de l'objet sujet du track,et pour la gamme MONO/MULTI une ergonomie qui peut atteindre les 8 heures ,aussi le module de track est équipé d'un panneaux photovoltaïque et peut ce chargé

Un service client avec un contrôle total sur tous les composants du système logiciel et matériel

Un Prix très raisonnable en le comparant aux solutions existante dans le marché qui propose un service similaire



La plateforme WinRah est destinée aux organismes et aux particuliers chercheurs de service de tracking

GPS pour seul ou multiple trackers, et pour cela trois types de services sont disponibles

Les particuliers seront dotés d'un compte particulier pour visualiser un ou plusieurs trackers

Pour les organismes Deux offres sont disponibles au choix

Les organismes qui gèrent un réseau de compte particulier, ces organismes ont la possibilité de gérer juste un niveau compte effectif de consultation des trackers, pour un réseau de transport, on fournit un compte professionnel pour cette catégorie de clients.

Les organismes qui gèrent un réseau de compte de client professionnel, et font le basculement des trackers entre ces comptes vont avoir un niveau de privilège plus haut qui permet de construire un réseau de réseaux

Un autre type de client ce sont ceux qui accèdent avec le code QR sans aucun privilège sur le tracker

On peut avoir d'autres types de clients fournisseurs de service de tracking pour d'autres clients indépendamment

Un Service Client est assuré au niveau du siège social pour achat, abonnement et accompagnement de proche

Un agent assure le suivi des messages et des appels, de façon régulière et donne des réponses, et des solutions

Un bureau de service après-vente est assuré pour recevoir les réclamations des clients et les modules tracking défectueux

Le site Web assure la demande d'un module de tracking

Le Site Web permet la visualisation des trackers associés au compte du client

L'application mobile permet la visualisation des trackers associés au compte du client

Une assistance via le téléphone est assurée pour la vérification des trackers d'un client professionnel

Les campagnes publicitaires sur les réseaux sociaux sont programmées régulièrement

Le porte a porte pour les entreprises ayant un réseau de transport ou il font des contrat avec des transporteurs

Les réseaux sociaux pour la diffusion des annonces ,la recherche des client professionnel potentiels et la gestion des messages des clients

Le site Web pour l'achat des trackeur et la visualisation du tracking et aussi le contrôle du trackeur

Les Services de livraison pour livré les trackeurs au Clients Particuliers .

L'Application Mobile pour l'achat des trackeur et la visualisation du tracking et aussi le contrôle du trackeur .

Une assistance après vente via un bureau chargé de la gestion des abonnements et de recevoir les modules en panne

**5- Key partners:**



**5- الشراكات الرئيسية :**

Les Fournisseur d'énergie électrique

Le Fournisseur d'ACCES Internet Filaire

Les Operateurs de Service de Téléphonie Mobile Avec accès Internet.

Fournisseurs de composante électronique

Fournisseurs des articles d'emballage .

Entreprise Spécialiste en livraison

Les fournisseurs d'hébergement VPS.

Les Autorité locale responsable de fournir les agréments nécessaires

Les autorités responsable du payement électronique.

Les autorités responsable du commerce électronique.

**6- Key activities :**



**6- الأنشطة الرئيسية:**

Les Activités clés pour sont :

L'étude et la réalisation électronique l'emballage, la mise en marche et L'inscription, et en fin le test.

La gestion Des Comptes Client Professionnel et l'association des Trackeur au Comptes.

La Mise A Jour Des Applications De La Plateforme: Module Web, Application Mobile et ERP.

La gestion des commandes la livraison des trackeurs réservé par les clients particuliers

La maintenance de Trackeurs défectueux pour es Clients professionnel

L'assistance aux Client par téléphonie ou par réseaux spéciaux

L'étude de marché des professionnel exerçant dans le domaine de transport

Mettre en œuvre des campagne publicitaire ciblant les client particuliers

**7- Key  
Activities:**



**7- الموارد الرئيسية:**

Le ressources clés sont les suivant :

Siege pour les bureau Service client, après vente, atelier de montage et maintenance équipé

de bureautique et d'équipement de maintenance électronique

Serveurs physique, ou VPS dédié ou Cloud pour l'hébergement des différents module de la

plateforme

Nom de domaine

Abonnement internet de vitesse d'au moins 20 Mbps

Composants électroniques utilisés dans la fabrication des trackeurs

Equipements et consommable pour fabrication d'emballage des trackeurs : imprimante 3D et filament

Véhicule de service

Pc de caractéristiques adapté au développent et a la création digital pour le développent des différent

module de la plateforme

**8- Cost structure:**



**8- هيكل التكاليف:**

Le service est catégorisé en deux gamme vis a vis le module de tracking : Mono/Multi tracking et

Mega Tracking

Les Charge de la mise en marche d'un trackeur les suivant:

1. Pour un trackeur Mini/Multi le prix estimatif pour le montage de chaque trackeur est de 5000 DA
2. Pour les trackeur Mega le prix du produit fini est de 6500 DA

Pour ce qui concerne les charges de la mise en service de la plateforme on peut les détaillé comme suit:

L'abonnement Internet en fibre optique 3500 DA/Mois

Frais de location d'un nom de domaine 2000DA/Mois

Frais de Location de serveur VPS 1500DA/Mois

Bureautique 300000 DA

Location de Local 20000DA/Mois

Salaires des Employés 120000DA/Mois

Frais de roulement :Energie ,Carburant, Eau et Transport 100000DA/Mois

## 9- Revenue streams:



## 9- مصادر الإيرادات:



Le pris de vente d'un trackeur Mini/Multi est de 7500DA










Le prix de vente d'un trackeur Mega est de 9000DA

Les charges de l'utilisation du system sont incluses au pris du trackeur dans les 3 premiers mois,par la suite l'abonnement est de 2000DA/Mois pour un compte Particulier

Les frais pour avoire un compte profesionel sont fixé a 1000DA par mois et par trackeur

Un type d'abonnement FORFAIT pour les grandes entreprises qu'on decidera son prix selon le type de compte superviseur,les comptes herités,et les type des trackeurs associées .

## Business Model Canvas

<p><b>Partenaires clés</b> </p> <ul style="list-style-type: none"> <li>•Fournisseurs : Energie , internet filaire, téléphonie mobile, module emballage, serveurs VPS, composants électroniques service livraison</li> <li>•Autorités Locales</li> <li>•Autorités E-payment</li> <li>•CNRC</li> </ul>	<p><b>Activités</b> </p> <p>Administration service Service Client Confection, vente et maintenance trackeurs Mise a jour logiciel Développement marché</p> <p><b>Ressources clés</b> </p> <p>Serveur VPS,et domaine Pièces pour trackeurs Abonnement Internet Local bureau équipé Materiel developement Materiel et pièces fabrication et montage tracker Moyen de transport</p>	<p><b>Propositions valeur</b> </p> <p>Localisation GPS Localisation Temps Réel Gestion Son Réseau Trackeurs Exploitation Facile de parcoure Multi Plateforme Adapté au marché Local Possibilité du développement du service Prix service Etudié</p>	<p><b>Relation</b> </p> <p>Service Client Gestion de vente des trackeurs Service Apres vente Consultation réseau tracking</p> <p><b>Canaux</b> </p> <p>Porte a Porte Site Web Réseaux sociaux Siège Bureau Services de livraison</p>	<p><b>Clients</b> </p> <p>Les particuliers chercheur de mono ou multi tracking</p> <p>Les Organismes cherchant une administration de réseau de tracking</p>
<p><b>Coûts</b> </p> <p>Abonnement VPS, DNS Montage Module trackeurs Frais : Internet,Energie,Location Bureau,Moyen Transport/Vehicules Services Equipement Bureautique et informatique Salaires Employées</p>		<p><b>Revenus</b> </p> <p>Vente des trackeurs Abonement Service</p>		

## Liste de figures

<b><u>Figure II.1</u></b> Architecture des Servlets.....	18
<b><u>Figure II.2</u></b> Les portées des variables en JEE .....	23
<b><u>Figure II.3</u></b> Schéma pin de Arduino UNO R3 Board .....	30
<b><u>Figure II.4</u></b> Sim808 Board.....	31
<b><u>Figure II.5</u></b> Outils Logiciel de développement.....	32
<b><u>Figure II.6</u></b> Le Déploiement du pattern DAO.....	33
<b><u>Figure II.7</u></b> Le Packaging EAR.....	37
<b><u>Figure III.1</u></b> Diagramme Général de cas d'utilisation .....	43
<b><u>Figure III.2</u></b> Diagramme Déploiement le la Plate-forme WinRah .....	46
<b><u>Figure III.3</u></b> Diagramme Séquence Pour Création de Compte Particulier .....	47
<b><u>Figure III.4</u></b> Diagramme Séquence Pour Création de TrackerMULTI .....	48
<b><u>Figure III.5</u></b> Diagramme Séquence Pour Association Tracker au ComptePro .....	49
<b><u>Figure III.6</u></b> Diagramme Séquence de l'Interface de consultation Web .....	50
<b><u>Figure III.7</u></b> Diagramme Séquence de Consultation de localisation par QR (Application Mobile) .....	51
<b><u>Figure III.8</u></b> Diagramme de Class Service Web WinRah .....	52
<b><u>Figure III.9</u></b> Diagramme De Classe Service WEB -Partie Ajout D'un Tracker.....	53
<b><u>Figure III.10</u></b> Diagramme De Classe Module Bureau .....	54
<b><u>Figure III.11</u></b> Interface de connexion au module WEB.....	55
<b><u>Figure III.12</u></b> Interface de Consultation générale des trackeurs au module WEB ..55	
<b><u>Figure III.13</u></b> Interface de Consultation détaillée d'un trackeurs en WEB .....	56
<b><u>Figure III.14</u></b> Interface de connexion au module Mobile .....	57
<b><u>Figure III.15</u></b> Interface de Consultation détaillée d'un trackeurs en Mobile.....	58
<b><u>Figure III.16</u></b> fenêtre de connexion au module Bureau .....	58
<b><u>Figure III.17</u></b> Tableau de board du module Bureau .....	59
<b><u>Figure III.18</u></b> fenêtre d'association des trackeurs au module Bureau.....	59

## Liste des tableaux

<u>Tableau I.1</u> : méthodes de positionnement mobile dans le reseau GSM .....	9
<u>Tableau I.2</u> : Attribut du marcher Tracking-as-service .....	14
<u>Tableau III.1</u> décrit les cas d'utilisations Client particulier .....	42
<u>Tableau III.2</u> décrit les cas d'utilisations Client Professionnelle .....	42
<u>Tableau III.3</u> décrit les cas d'utilisations Administrateur .....	42
<u>Tableau III.4</u> Description du cas d'utilisation « S'authentifier » .....	44
<u>Tableau III.5</u> Description du cas d'utilisation «Consultation Parcoure Tracker»...	45
<u>Tableau III.6</u> Description du cas d'utilisation «Enregistrer Trackeur».....	46

## **Liste des abréviations :**

**MSC** : Centre de commutation Mobile

**BSC** : Base Station Controller

**BTS** : Base Transceiver Stations

**FPGA** :Field-programmable gate array

**ARM** :Advanced RISC (Reduced Instruction Set Computing) Machines –microcontrollers

**ESRI** :Environmental Systems Research Institute

**HTTP** :Hypertext Transfer Protocol

**DNS** :Domain Name System

**LDAP** :Lightweight Directory Access Protocol

**NIS** :Network and Information Systems

**CoS** :Class of Service

**IP** :Internet Protocol

**ERP** :Enterprise resource planning

**QR** : Quick response

**TCAC** :Tax Credit Allocation Committee

**IoT** :Internet of Things

**URI** :Uniform Resource Identifier

**CMP** : Container Managed Persistence

**BMP** : Bean Managed Persistence

## Résumé

Ces dernière années, les systèmes de suivi sont de plus en plus utilisés dans le quotidien comme ou dans les entreprises de taxi ou visant par exemple la surveillance des personnes âgées ou des enfants. Dans notre contribution, nous avons mis en place un système de suivi de véhicules appelé WinRah. Ce système permet de localiser l'emplacement et de donner le parcours des véhicules avec le GPS. Pour ce faire, nous avons développé trois applications ; une application web, une application mobile et une application ERP. L'application web et l'application mobile permet l'accès des clients au système. L'application ERP permet la gestion du système.

Pour réaliser ce système, nous avons d'abord modéliser notre système avec le langage UML. Ensuite pour la mise en place du système, nous l'avons développé avec plusieurs outils logiciels comme Java, Flutter Dart, Mysql et C Arduino et nous avons confectionné le traceur.

### Mots clés

**plat-forme, système, information, localisation, position, carte, géographique, transport, parcoure, temps-réel, application, mobile**

## ملخص

في السنوات الأخيرة ، استخدام أنظمة التتبع يتم بشكل متزايد في الحياة اليومية ، كما هو الحال في شركات سيارات الأجرة أو تلك التي تهدف ، على سبيل المثال ، إلى مراقبة كبار السن أو الأطفال.

في هذا العمل ، قنا بإنشاء ما يتيح ذلك عبر نظام تتبع للمركبات والذي اطلقنا عليه اسم **WinRah** يتيح هذا النظام تحديد الموقع وتحديد مسار المركبات باستخدام نظام تحديد المواقع العالمي (**GPS**) ، للقيام بذلك ، قنا بتطوير ثلاثة تطبيقات ، تطبيق ويب وتطبيق جوال وتطبيق **ERP** .

يسمح تطبيق الويب وتطبيق الهاتف المحمول للعملاء بالوصول إلى النظام. في حين يستخدم تطبيق **ERP** في إدارة النظام.

لإنشاء هذا النظام ، قنا أولاً بالتصميم النموذجي لمنظومتنا بلغة **UML**. ثم من أجل وضعها حيز العمل ، قنا بتطويرها باستخدام العديد من أدوات ولغات البرمجة مثل **Java** و **Dark** و **Mysql** و **C Arduino** كما قنا بتركيب نموذج للمتبع.

## الكلمات الدلالية

منظومة ، نظام ، معلومات ، تتبع ، موقع ، متبع ، خريطة ، جغرافي ، نقل ، مسار ، تطبيق ، آني ، هاتف ذكي.

## **Summary**

In recent years, tracking systems have been increasingly used in everyday life, such as in taxi companies or aimed, for example, at tracking the elderly or children. In our contribution, we implemented a vehicle tracking system called WinRah. This system give possibility to get the location and give us the path of the vehicles with the GPS. To do this, we have developed three applications; a web application, a mobile application and an ERP application. The web application and the mobile application allow customers to access the system. The ERP application allows the management of the system.

To realize this system, we first modeled our system with the UML language. Then for the implementation of the system, we developed it with several software tools like Java, Dark, Mysql and Arduino C and we made the tracker.

## **Key words**

**platform, system, information, location, position, map, geographic, transport, path, real-time, application, mobile**