

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE



Mémoire

En vue de l'obtention de Master en Informatique option

Ingénierie des systèmes d'information

Thème

Conception et réalisation d'une application android pour l'optimisation des chemins de transport cas NAFTAL

Encadré par :

Mr Habet Mouhend Said

Mr Chaba Kamel

Réaliser par :

Alik Amrane

Amrouni Ahcene

2017 / 2018

Sommaire

1	INTRODUCTION GENERALE	5
2	Présentation de l'organisme d'accueil	7
2.1	Introduction :	7
2.2	Historique :	7
2.3	Activité de NAFTAL :	8
2.4	Les infrastructures de NAFTAL :	8
2.5	Présentation du district de Tizi-Ouzou :	9
2.5.1	Aspect Organisationnel du district :	9
2.5.2	Mission du district :	9
2.6	Organigramme générale du district de Tizi-Ouzou :	10
2.7	Présentation du département commercial :	10
2.8	Problématique :	11
3	Conception	12
3.1	Introduction :	12
3.2	Présentation d'UML :	13
3.3	Identification des acteurs :	14
3.4	Spécification des tâches :	14
3.5	Spécification des cas d'utilisations :	15
3.5.1	Description textuelle du cas d'utilisation «Authentification» :	16
3.5.2	Description textuelle du cas d'utilisation « Ajouter commande » :	17
3.5.3	Description textuelle du cas d'utilisation «Rechercher une commande» :	18
3.5.4	Description textuelle du cas d'utilisation «Supprimer une commande» :	19
3.5.5	Description textuelle du cas d'utilisation «Affecter une commande a un client» :	20
3.5.6	Description textuelle du cas d'utilisation «Ajouter un client» :	21
3.5.7	Description textuelle du cas d'utilisation «Supprimer un client» :	22
3.5.8	Description textuelle du cas d'utilisation «Ajouter un Chauffeur» :	23
3.5.9	Description textuelle du cas d'utilisation « Supprimer un chauffeur » :	24
3.5.10	Description textuelle du cas d'utilisation «Consulter l'itinéraire du jour» :	25
3.5.11	Description textuelle du cas d'utilisation « Valider la livraison d'une commande » :	26
3.6	Diagramme de cas d'utilisation :	27
3.7	Diagramme de séquence :	28
3.7.1	Diagramme de séquence pour le cas d'utilisation ' Authentification d'un chauffeur ' :	29
3.7.2	Diagramme de séquence pour le cas d'utilisation ' Consulter l'itinéraire du jour ' :	30
3.7.3	Diagramme de séquence pour le cas d'utilisation ' valider la livraison d'une commande ' :	31
3.8	Diagramme de classe :	32
3.9	Le modèle logique de données :	33
3.9.1	Le schéma relationnelle correspondant au diagramme de class :	33
3.10	Les tables de base de données :	34
3.10.1	Les tables de base de données :	34
3.10.2	La table chef de parc :	34
3.10.3	La table client :	34
3.10.4	La table commande :	34
3.11	Optimisation de l'itinéraire :	35

3.12	Conclusion :	37
4	Réalisation	38
4.1	Outils et langages de programmation :	38
4.1.1	Introduction :	38
4.1.2	Netbeans :	38
4.1.3	MYSQL :	39
4.1.4	Glassfish :	40
4.1.5	Android Studio :	40
4.1.6	Plateform Java EE (Edition Entreprise) :	41
4.1.7	Langage de programmation JAVA :	42
4.1.8	JSON :	43
4.1.9	SQL :	44
4.1.10	Organisation de l'exécution des instructions sous android :	44
4.2	Architecture de notre système :	46
4.2.1	Architecture de système :	46
4.2.2	première partie la partie JEE WebApplication :	46
4.2.3	Structure des tables :	47
4.2.4	Service web disponible sur l'application :	49
4.2.5	Deuxième partie l'application android :	50
4.2.6	Le calcul de l'itinéraire :	51
4.2.7	Confirmation d'une livraison :	52
4.3	Interface de l'application JEE Web :	53
4.3.1	La page d'accueil :	53
4.3.2	la page d'authentification pour le chef de parc :	54
4.3.3	la page ajouter un chauffeur :	55
4.3.4	la page supprimer chauffeur :	56
4.3.5	la page ajouter une commande :	57
4.3.6	la page supprimer une commande :	58
4.3.7	affecter une commande a un chauffeur :	59
4.3.8	La page ajouter une nouvelle commande :	60
4.3.9	la page recherche d'une commande :	61
4.3.10	la page ajouter client :	62
4.3.11	la page supprimer un client :	63
4.3.12	l'interface d'authentification chauffeur :	64
4.3.13	l'interface feuille de route :	65
4.3.14	affichage de l'itinéraire :	66
4.3.15	affichage de la liste de commande a confirmer :	67
4.3.16	affichage de la liste de commande a confirmer :	68
4.3.17	Conclusion :	68
4.4	Conclusion général :	69
4.5	Annex :	70
4.6	Concepts de base sur Android :	70
4.6.1	Introduction :	70
4.6.2	Historique des versions :	70
4.6.3	Android :	71
4.6.4	Plateform android :	71
4.6.5	Dalvik et ART :	72
4.6.6	Le plugin de développement d'Eclipse ADT :	73
4.6.7	L'environnement Android Studio :	73
4.6.8	L'architecture d'un projet Android Studio :	74
4.6.9	Les vues dans Android Studio :	75
4.6.10	Les éléments d'une application :	76
4.6.11	Cycle de vie d'une activité :	77
4.7	Le geoweb :	79
4.7.1	Introduction :	79

4.7.2	L'infrastructure du GéoWeb :	80
4.7.3	Le GéoWeb volontaire et contributif :	81
4.7.4	Conclusion :	82
4.8	La géolocalisation :	83
4.8.1	Introduction :	83
4.8.2	Les systèmes existants :	83
4.8.3	Intérêt de la localisation par satellite :	84
4.8.4	Avantages et inconvénients de la géolocalisation GPS :	85
4.8.5	Géolocalisation par GSM :	85
4.8.6	Avantage et inconvénients de la géolocalisation par GSM :	85
4.8.7	Géolocalisation par Wifi :	85
4.8.8	Avantages et inconvénients de la géolocalisation par Wifi :	86
4.8.9	comparaison entre ces trois technologies :	86
4.8.10	Conclusion :	87
4.9	Les services Web :	88
4.9.1	Introduction :	88
4.9.2	Définitions des services Web :	88
4.9.3	Les principaux avantages d'un service Web :	89
4.9.4	L'intérêt d'un Service Web :	89
4.9.5	Les caractéristiques d'un service Web :	90
4.9.6	Architecture d'un service web :	90
4.9.7	Définition de service restful :	91
4.9.8	Les avantages et les inconvénients de Rest :	92
4.9.9	Conclusion :	92

Table des figures

2.1	Organigramme du district de naftal.	10
3.1	diagramme de cas d'utilisation pour un chauffeur.	27
3.2	Diagramme de cas d'utilisation pour le chef de parc.	27
3.3	diagramme de séquence pour le cas d'utilisation 'Authentification d'un chauffeur'.	29
3.4	diagramme de séquence pour le cas d'utilisation ' Consulter l'itinéraire du jour'.	30
3.5	diagramme de séquence pour le cas d'utilisation ' valider la livraison d'une commande'.	31
3.6	diagramme de classe générale.	32
4.1	Netbeans	39
4.2	MYSQL	39
4.3	MYSQL	40
4.4	Android Studio	41
4.5	Plateform Java EE	42
4.6	java.	43
4.7	JSON	43
4.8	SQL	44
4.9	l'exécution des applications sous une plateforme android.	45
4.10	la composition d'un processeur sous une plateforme android	46
4.11	Architecture de système.	47
4.12	illustration d'une tâche asynchrone sous android	52
4.13	La page d'accueil.	53
4.14	la page d'authentification pour le chef de parc	54
4.15	la page ajouter un chauffeur	55
4.16	la page supprimer chauffeur	56
4.17	la page ajouter une commande	57
4.18	la page supprimer une commande	58
4.19	affecter une commande a un chauffeur	59
4.20	La page ajouter une nouvelle commande.	60
4.21	la page recherche d'une commande	61
4.22	la page ajouter client	62
4.23	la page supprimer un client	63
4.24	l'interface d'authentification chauffeur	64
4.25	l'interface feuille de route	65
4.26	affichage de l'itinéraire.	66
4.27	affichage de la liste de commande a confirmer	67
4.28	affichage de la liste de commande a confirmer	68
4.29	Anatomie d'un déploiement.	72
4.30	Les vues dans Android Studio.	75
4.31	Cycle de vie d'une activité.	77
4.32	satellite GPS.	84
4.33	Tableau récapitulatif des systèmes localisation mobile.	86
4.34	architecture d'un service web.	91

Chapitre 1

INTRODUCTION GENERALE

Les smart phones sont aujourd'hui devenus les « couteaux suisses » de prédilection d'un individu hyperconnecté de plus en plus mobile. De ces applications naissent de nouveaux usages, de telle sorte qu'en plus de téléphoner, naviguer sur Internet ou partager une vaste quantité d'informations sur des réseaux sociaux, le smart phones permettent de se repérer dans l'espace... mais aussi de localiser des entités géographiques comme des lieux ou des personnes.

Un système de géolocalisation permet de localiser une personne, un véhicule ou un objet sur une carte, à l'aide de ses coordonnées géographiques.

Initialement conçue pour une utilisation exclusivement militaire, nombreuses sont les entreprises à se poser la question de l'utilité d'investir dans un système de géolocalisation qui peut être très pratique pour les entreprises de transport de marchandise. En effet, un système de géolocalisation propose diverses fonctionnalités : **aide à la navigation**, suivi de marchandises, **repérage d'un véhicule** suite à un vol, localisation de personnes (routiers, commerciaux, personnes dépendantes, travailleurs itinérants... .

Naftal est une entreprise Algérienne, filiale à 100% de SONATRACH, elle est spécialisée dans la conception, l'élaboration et la distribution de lubrifiants pour moteurs (deux roues, automobiles et autres véhicules) ainsi que pour l'industrie.

Pour la distribution et la commercialisation des produits lubrifiants et pneumatique, naftal ne dispose d'aucun système informatique lui permettant d'optimiser et améliorée son service clientèle, ainsi la société compte sur les compétences et l'expérience de ces conducteurs pour acheminer la marchandise.

En réduisant le temps passé sur la route, les solutions d'optimisation de tournées permettent d'or-

donnancer les déplacements et de définir des parcours optimaux et cohérents, en plus de réduire les kilomètres parcourus ça permet aussi de visiter plus de clients et de redimensionner la flotte de véhicules. Et ainsi améliore rapidement la rentabilité de l'entreprise.

C'est dans cette optique que s'inscrit notre projet intitulé « Conception et réalisation d'une application Android pour l'optimisation des chemins de transport cas NAFTAL » dans l'objectif est de concevoir une application dédiée aux appareil mobiles, doté de la plateforme Android, permettant a son propriétaire la visualisation de l'itinéraire a suivre a fin de mener ça livraison, grâce aux outils et techniques de géolocalisation et d'orientation.

Chapitre 2

Présentation de l'organisme d'accueil

2.1 Introduction :

Naftal est une société par actions (SPA) au capital social de 15 650 000 000 DA. Fondée en 1982 et filiale à 100% du Groupe Sonatrach, elle est rattachée à l'activité commercialisation. Elle a pour mission principale, la distribution et la commercialisation des produits pétroliers et dérivés sur le marché national.

Naftal est spécialisée dans la conception, l'élaboration et la distribution de lubrifiants pour moteurs (deux-roues, automobiles et autres véhicules) ainsi que pour l'industrie.

2.2 Historique :

Par transfert du monopole et de biens et personnels détenus ou gérés par Sonatrach, le 6 avril 1981 par décret N° 80/101, il est créé une entreprise nationale dénommée « Entreprise nationale de raffinage et de distribution de produits pétroliers » par abréviation (ERDP), entreprise socialiste à caractère économique.

Entrée en activité le 1^{er} janvier 1982, elle est chargée de l'industrie du raffinage des hydrocarbures liquides et de la distribution des produits raffinés sur le territoire national.

Le 5 février 1983, par décret n° 83-112, il est procédé à la modification de la dénomination de l'entreprise nationale de raffinage et de distribution de produits pétroliers de ERDP en Naftal.

Le 25 août 1987, par la promulgation du décret n° 87-190 portant création, par transfert de l'activité raffinage de Naftal, de l'entreprise nationale de raffinage des produits pétroliers sous le

sigle Naftec.

Naftal est désormais chargée de la commercialisation et de la distribution des produits pétroliers et dérivés.

Le 18 avril 1998, elle change de statut avec la transformation de Naftal en société par actions au capital social de 6 650 000 000 DA, filiale à 100 % du holding Sonatrach valorisation des hydrocarbures (SVH). Le 18 avril 1998, elle change de statut avec la transformation de Naftal en société par actions au capital social de 6 650 000 000 DA, filiale à 100 % du holding Sonatrach valorisation des hydrocarbures (SVH).

2.3 Activité de NAFTAL :

Naftal a pour mission principale, la distribution et la commercialisation des produits pétroliers sur le marché national.

Elle intervient dans les domaines de :

- L'enfûtage GPL .
- La formulation de bitumes .
- La distribution, stockage et commercialisation des carburants, GPL, lubrifiants, bitumes, pneumatiques, GPL/carburant, produits spéciaux .
- Transport des produits pétroliers.

Pour assurer la disponibilité des produits sur tout le territoire, Naftal met à contribution plusieurs modes de transport :

- Le rail pour le ravitaillement des dépôts à partir des entrepôts.
- Le cabotage et les pipes, pour l'approvisionnement des entrepôts à partir des raffineries.
- La route pour livraison des clients et le ravitaillement des dépôts non desservis par le rail.

2.4 Les infrastructures de NAFTAL :

Les infrastructures opérationnelles :

- 47 Dépôts carburants terre.

- 42 Centres et mini-centres GPL.
- 09 Centres vrac GPL.
- 47 Dépôts relais.
- 30 Centre et dépôts aviation.
- 06 Centres marine.
- 15 Centres bitumes.
- 24 Centres lubrifiants et pneumatiques.
- Un réseau de transport pipelines d'une longueur de (2 720 km).
- Un parc roulant de 3 300 unités.
- Un réseau de stations-service de 674, dont 338 stations-service en gestion directe.

2.5 Présentation du district de Tizi-Ouzou :

2.5.1 Aspect Organisationnel du district :

Issu des restructurations passées, la zone porte maintenant la nouvelle appellation de DISTRICT et ça suite à la décision n05699 du 26/05/2002 portant sur l'organisation de la macrostructure de la société NAFTAL donc dans la suite de ce travail nous utiliserons le terme DISTRICT.

2.5.2 Mission du district :

La structure est commune dans tous les DISTRICT avec des missions communes mais qui doivent s'adapter aux données du terrain, mais avant de passer aux missions des différents départements, commençons par les missions globales du DISTRICT :

- Gestion, organisation, promotion et développement de l'activité de distribution des carburants terre, lubrifiants spéciaux et pneumatiques.
- Commercialisation des produits au niveau du DISTRICT.
- Développement et modernisation du réseau.
- Analyse du marché du DISTRICT, position et mise en œuvre de toute action susceptible de renforcer les positions de la division et de l'entreprise par rapport aux concurrents.
- Promotion des produits et de l'image de marque de l'entreprise.

2.6 Organigramme générale du district de Tizi-Ouzou :

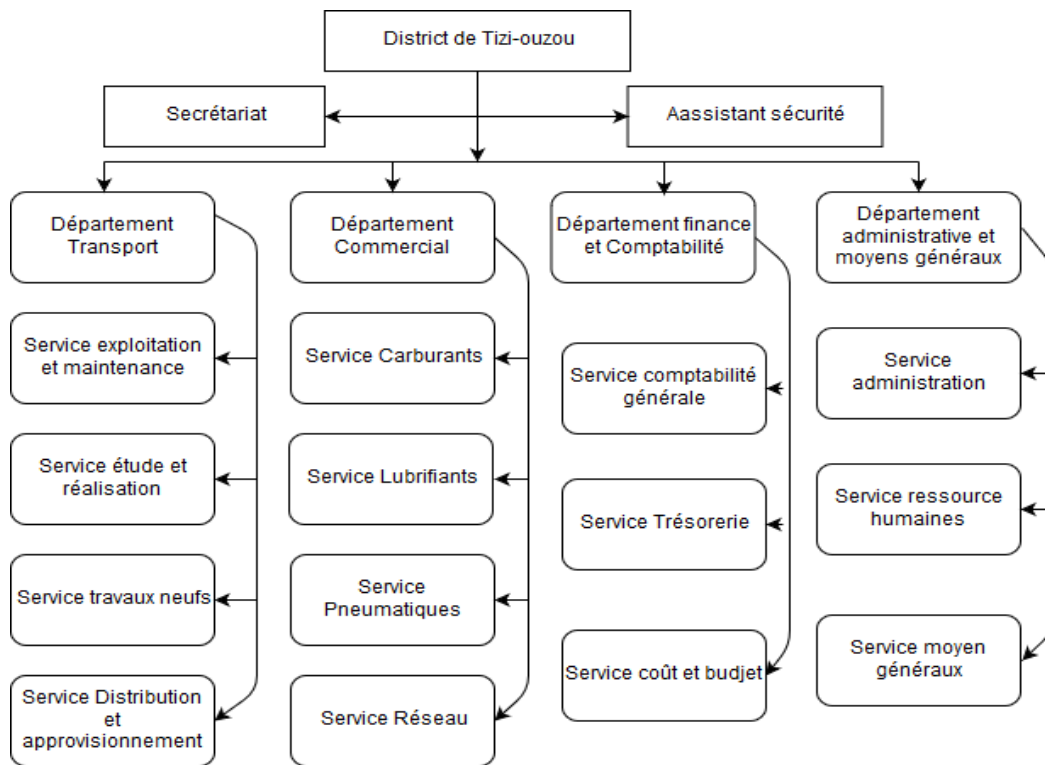


FIGURE 2.1 – Organigramme du district de naftal.

2.7 Présentation du département commercial :

Le département commercial est chargé de gérer toutes les transactions de ventes et d'approvisionnement des produits, et il est partitionné en 4 service :

1. Le service Carburants :

Ce service s'occupe de la distribution et la commercialisation du carburant.

2. Le service réseau :

Ce service s'occupe de la gestion du réseau de point de vente du district, il s'occupe de la gestion des :

- Stations en gestion directe i.e. les stations de naftal.
- Des points de vente tiers i.e. les points de vente agréés qui sont des stations privées agréées par naftal et les stations en gestion libre qui représentent des stations de service de naftal gérées par des privés.

3. Le service lubrifiant :

Ce service prend en charge toutes les transactions portant sur les lubrifiants de l'approvisionnement jusqu'au service après-vente en cas d'un produit non conforme.

4. Le service pneumatique :

Ce service s'occupe de l'approvisionnement et la commercialisation des pneumatiques.

2.8 Problématique :

L'objectif de toutes sociétés et de réalisé un meilleur profit en satisfaisant au mieux ça clients, spécifiquement les sociétés de distribution, la livraison rapide et optimal du produit est indispensable pour atteindre cet objectif.

Pour cela un système d'optimisation des chemins de transport est un moyen efficace pour Naftal afin d'améliorés ces services ainsi promouvoir son image sur le marché.

Chapitre 3

Conception

3.1 Introduction :

La mise en œuvre de notre système d'optimisation des chemain de transport passe par une analyse des besoins fonctionnels et par la modélisation du système. Cela suppose donc l'adoption d'une méthodologie, un langage pour une modélisation du système.

La modélisation permet de mieux comprendre le fonctionnement d'un système, c'est un moyen de maîtriser la complexité du système et d'assurer sa cohérence.

Il existe différents méthodes de modélisation , les plus connus sont :

- MERISE qui est basée sur la séparation des données et des traitements à effectuer en plusieurs modèle conceptuels et physiques.
- Le langage de modélisation UML (Unified Modeling Language) qui est un langage orienté objet.

Pour l'analyse et la conception de notre système , notre choix s'est porté sur le langage de modélisation unifié UML.

En outre UML est devenu le langage modélisation orienté objet de référence depuis l'apparition de sa première version en 1996 et de sa version UML 2 en 2006. Il ne se limite pas au domaine informatique, il s'applique à toutes sortes de système. C'est un standard défini par l'Object Management Group (OMG).

3.2 Présentation d'UML :

UML (Unified Modeling Language) est un langage de modélisation des spécifications d'un Système informatique. Il s'appuie sur la conception orientée objet pour la spécification des systèmes. UML 2 comporte treize (13) diagrammes, quatre (4) de plus que la première version. Chacun de ces 13 diagrammes permet d'avoir une vue spécifique du système à modéliser. Les diagrammes d'UML sont d'une utilité variable selon le projet et ne sont pas nécessairement tous produits dans le cadre d'une modélisation. UML est basé sur deux modes de représentation, le mode statique et le mode dynamique.

La représentation statique ou structurelle décrit le système. Cette représentation est assurée par six (6) diagrammes :

- diagramme de classes (Class diagram) .
- diagramme d'objets (Object diagram) .
- diagramme de composants (Component diagram) .
- diagramme de déploiement (Deployment diagram) .
- diagramme de paquetages (Package diagram) .
- diagramme de structures composites (Composite structure diagram).

La représentation dynamique ou comportementale s'appuie quant à elle sur les sept (7) autres diagrammes :

- diagramme de cas d'utilisation (Use case diagram) .
- diagramme d'activités (Activity diagram) .
- diagramme d'états-transitions (State machine diagram) .
- diagramme de séquence (Sequence diagram) .
- diagramme de communication (Communication diagram) .
- diagramme global d'interaction (Interaction overview diagram).
- diagramme de temps (Timing diagram).

Ces diagrammes ne sont pas tous implémentés dans le cadre d'un même projet. En outre le choix des diagrammes à mettre en œuvre est généralement en fonction de la nature du projet, sa complexité et sa taille.

Dans le cadre de la conception d'un système d'optimisation des chemain de transport , nous utiliserons principalement le diagramme des cas d'utilisation , le diagraphme de séquence et le diagramme de classes .

3.3 Identification des acteurs :

Pour ce système, nous avons identifié deux acteurs essentiels qui peuvent interagir avec l'application afin de répondre au mieux aux besoins spécifier :

- Le chef de parc : qui est le responsable de la gestion des stocks en ce qui concerne l'entrer ou la sortie des produits du dépôt et l'affectation des feuilles de route pour les chauffeurs
- Les chauffeurs : qui sont responsables sur les produits durent la phase du transport du produit du dépôt aux clients.

3.4 Spécification des tâches :

Les taches principales qui seront accomplies par les deux acteurs sur le système sont :

Acteurs	Taches	Commentaires
Chef de parc	T 01 : Authentification	
	T02 : gérer les commandes	T02 : Gérer = Ajouter, supprimer, affecter les commandes pour les chauffeur et rechercher ou consulter
	T03 : gérer les clients	T03, T04 : Gérer = Ajouter ou supprimer
	T04 : gérer les chauffeurs	
Chauffeur	T05 : Authentification	
	T06 : Consulter les commandes du jour	
	T07 : Consulter l'itinéraire du jour	
	T08 : Valider la livraison d'une commande	

3.5 Spécification des cas d'utilisations :

Une spécification de cas d'utilisation fournit des détails textuels pour un cas d'utilisation. La description d'un cas d'utilisation permet de :

- Clarifier le déroulement de la fonctionnalité .
- D'identifier les parties redondantes pour en déduire des cas d'utilisation plus précises qui seront utilisées par inclusion, extension ou généralisation/spécialisation .
- D'indiquer d'éventuelles contraintes déjà connues et dont on doit tenir compte lors de la réalisation.

3.5.1 Description textuelle du cas d'utilisation «Authentification» :

Titre : Authentification.
Auteur : Chauffeur ou Chef de parc.
Objectif : ce cas d'utilisation permet au chauffeur ou au chef de parc de s'authentifier afin d'accéder à son compte utilisateur.
Date de création : 02/06/2018.
Date de mise à jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : L'utilisateur doit renseigner le formulaire d'authentification.
Scénario nominal : 1 - Le système affiche un formulaire d'authentification. 2 - L'utilisateur saisit son login et son mot de passe et valide le formulaire. 3 - Le système vérifie si les informations reçues sont correctes. 4 - L'utilisateur accède à son compte utilisateur.
Enchaînement alternatif : si le login et le mot de passe ne sont pas corrects 3 - Le système affiche un message d'erreur à l'utilisateur indiquant que le login ou le mot de passe est incorrect. Le scénario nominal reprend au point 1.
Post condition : Authentification réussie, l'utilisateur pourra se déconnecter de son compte.

3.5.2 Description textuelle du cas d'utilisation « Ajouter commande » :

Titre : Ajouter commande.
Auteur : Chef de parc.
Objectif : ce cas d'utilisation permet au chef de parc d' enregistrer une nouvelle commande.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier. - Le client concerné par la commande doit exister dans la base de données .
Scénario nominal : 1 - Le système affiche un formulaire d'ajouter d'une nouvelle commande. 2 - L'utilisateur remplit et valide le formulaire. 3 - Le système vérifie si les informations reçues et les enregistre dans la base de données. 4 - Le système affiche le succès de l'opération .
Enchaînement alternatif : si les informations envoyées ne sont pas correctes (non respect des types de données) ou manquantes : 3 - Le système affiche un message d'erreur à l'utilisateur indiquant l'échec de l'opération et demande à l'utilisateur de vérifier ces informations . Le scénario nominal reprend au point 1 Si le client concerné par la commande n'existe pas : 3 - Le système affiche un message d'erreur et propose à l'utilisateur d'ajouter ce client à la base de données. Si la commande existe déjà dans la base de données : 3 - Le système affiche un message indiquant que la commande existe déjà. Le scénario nominal reprend au point 1.
Post condition : Authentification réussie , l'utilisateur pourra se déconnecter de son compte.

3.5.3 Description textuelle du cas d'utilisation «Rechercher une commande» :

Titre : Rechercher une commande.
Auteur : Chef de parc.
Objectif : ce cas d'utilisation permet au chef de parc de vérifier si une commande existe dans la base de données.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier. - L'utilisateur doit renseigné le formulaire de recherche.
Scénario nominal : 1 - Le système affiche un formulaire de recherche. 2 - L'utilisateur remplit et valide le formulaire. 3 - Le système cherche la commande correspondante a la recherche. 4 - Le système affiche la commande .
Enchaînement alternatif : si la commande n'existe pas dans la base de données : 3 - Le système affiche un message indiquant la commande n'existe pas et propose a l'utilisateur de la créer en ajoutant un lien vers l'ajout d'une nouvelle commande
Post condition : / .

3.5.4 Description textuelle du cas d'utilisation «Supprimer une commande» :

Titre : Supprimer une commande.
Auteur : Chef de parc.
Objectif : ce cas d'utilisation permet au chef de parc de supprimer une commande de la base de données.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier . - L'utilisateur doit renseigné le formulaire de suppression d'une commande.
Scénario nominal : 1 - Le système affiche un formulaire de suppression . 2 - L'utilisateur remplit et valide le formulaire. 3 - Le système cherche et supprime la commande concerné dans la base de donnée. 4 - Le système affiche le succès de l'opération.
Enchaînement alternatif : si la commande n'existe pas dans la base de donnée : 3 - Le système affiche un message indiquant que la commande n'existe pas .
Post condition : / .

3.5.5 Description textuelle du cas d'utilisation «Affecter une commande a un client» :

Titre : Affecter une commande a un client.
Auteur : Chef de parc.
Objectif : ce cas d'utilisation permet au chef de parc d'affecter une commande a un client.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier . - la commande et le chauffeur doivent existé dans la base de données.
Scénario nominal : 1 - Le système affiche un formulaire affectation. 2 - L'utilisateur remplit et valide le formulaire. 3 - Le système vérifie l'existence du chauffeur et de la commande dans la base de donnée et ajoute l'identifiant du chauffeur a la commande. 4 - Le système affiche le succès de l'opération.
Enchaînement alternatif : si la commande ou le chauffeur n'existe pas dans la base de données : 3 - Le système affiche un message indiquant que la commande ou le chauffeur n'existe .
Post condition : le client concerné par la commande s'affichera sur l'itinéraire du chauffeur concerné.

3.5.6 Description textuelle du cas d'utilisation «Ajouter un client» :

Titre : Ajouter un client.
Auteur : Chef de parc.
Objectif : ce cas d'utilisation permet au chef de parc d'ajouter un nouveau client.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier .
Scénario nominal : 1 - Le système affiche un formulaire d'ajout d'un nouveau client. 2 - L'utilisateur remplit le formulaire. 3 - L'utilisateur cherche l'emplacement du client sur la carte et fait un clic droit afin de récupérer ces coordonnées GPS . 4 - l'utilisateur valide le formulaire. 5 - Le système vérifie si les informations reçues et les enregistre dans la base de données. 6 - Le système affiche le succès de l'opération.
Enchaînement alternatif : si les informations concernant le client sont incorrectes (non respect des types de données) ou manquantes : 3 - Le système affiche un message indiquant l'échec de l'opération . Si le client existe déjà dans la base de données : 3 - Le système affiche un message indiquant que le client existe déjà.
Post condition : l'utilisateur peut ajouter des commandes concernant ce client.

3.5.7 Description textuelle du cas d'utilisation «Supprimer un client» :

Titre : Supprimer un client.
Auteur : Chef de parc.
Objectif : ce cas d'utilisation permet au chef de parc de supprimer un client.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier . - L'utilisateur doit renseigné le formulaire de suppression d'une commande.
Scénario nominal : 1 - Le système affiche un formulaire de suppression d'un client. 2 - L'utilisateur remplit et valide le formulaire. 3 - Le système cherche et supprime le client concerné dans la base de donnée. 4 - Le système affiche le succès de l'opération..
Enchaînement alternatif : si le client n'existe pas dans la base de donnée : 3 - Le système affiche un message indiquant que le client n'existe pas .
Post condition : / .

3.5.8 Description textuelle du cas d'utilisation «Ajouter un Chauffeur» :

Titre : Ajouter un chauffeur.
Auteur : Chef de parc.
Objectif : ce cas d'utilisation permet au chef de parc d'ajouter un nouveau chauffeur.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier.
Scénario nominal : 1 - Le système affiche un formulaire d'ajout d'un nouveau chauffeur. 2 - L'utilisateur remplit et valide le formulaire. 3 - Le système vérifie si les informations reçues et les enregistre dans la base de données. 4 - Le système affiche le succès de l'opération.
Enchaînement alternatif : si les informations concernant le client sont incorrectes (non respect des types de données) ou manquantes : 3 - Le système affiche un message indiquant l'échec de l'opération . Si le Chauffeur existe déjà dans la base de données : 3 - Le système affiche un message indiquant que le chauffeur existe déjà.
Post condition : l'utilisateur peut ajouter affecter une commande à ce chauffeur .

3.5.9 Description textuelle du cas d'utilisation « Supprimer un chauffeur » :

Titre : Supprimer un chauffeur.
Auteur : Chef de parc.
Objectif : ce cas d'utilisation permet au chef de parc de supprimer un chauffeur.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier . - L'utilisateur doit renseigné le formulaire de suppression d'un chauffeur.
Scénario nominal : 1 - Le système affiche un formulaire de suppression d'un chauffeur. 2 - L'utilisateur remplit et valide le formulaire. 3 - Le système cherche et supprime le chauffeur concerné dans la base de donnée. 4 - Le système affiche le succès de l'opération.
Enchaînement alternatif : si le chauffeur n'existe pas dans la base de donnée : 3 - Le système affiche un message indiquant que le chauffeur n'existe pas.
Post condition : / .

3.5.10 Description textuelle du cas d'utilisation «Consulter l'itinéraire du jour» :

Titre : Consulter l'itinéraire du jour.
Auteur : Chauffeur.
Objectif : ce cas d'utilisation permet au chauffeur de visualiser l'itinéraire a suivre pour effectuer ces livraison.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier .
Scénario nominal : 1 - Le système affiche une interface d'accueil. 2 - L'utilisateur clique sur afficher l'itinéraire du jour. 3 - Le système affiche l'itinéraire sur la carte.
Enchaînement alternatif : si aucune commande ne lui a été affecter : 3 - Le système affiche un message indiquant qu'il ne y a pas de commande a livré .
Post condition : / .

3.5.11 Description textuelle du cas d'utilisation « Valider la livraison d'une commande » :

Titre : Valider la livraison d'une commande.
Auteur : Chauffeur.
Objectif : ce cas d'utilisation permet au chauffeur de valider la livraison d'une commande.
Date de création : 02/06/2018.
Date de mise a jour : 02/06/2018.
Version : 1.0.
Responsable : Alik Amrane.
Précondition : - L'utilisateur doit s'authentifier .
Scénario nominal : 1 - Le système affiche une interface d'accueil. 2 - L'utilisateur clique sur valider la livraison d'une commande. 3 - Le système affiche une liste de commande affecter a ce chauffeur. 4 - L'utilisateur fais un long click sur la commande livrer . 5 - Le système affiche le succès de l'opération .
Enchaînement alternatif : si aucune commande ne lui a été affecter : 3 - Le système affiche un message indiquant qu'il ne y as pas de commande a livré.
Post condition : / .

3.6 Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs.

Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne.

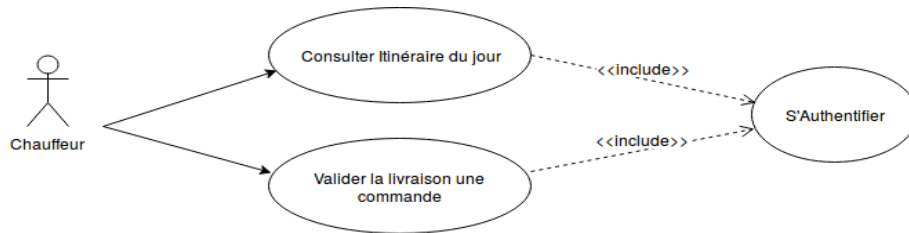


FIGURE 3.1 – diagramme de cas d'utilisation pour un chauffeur.

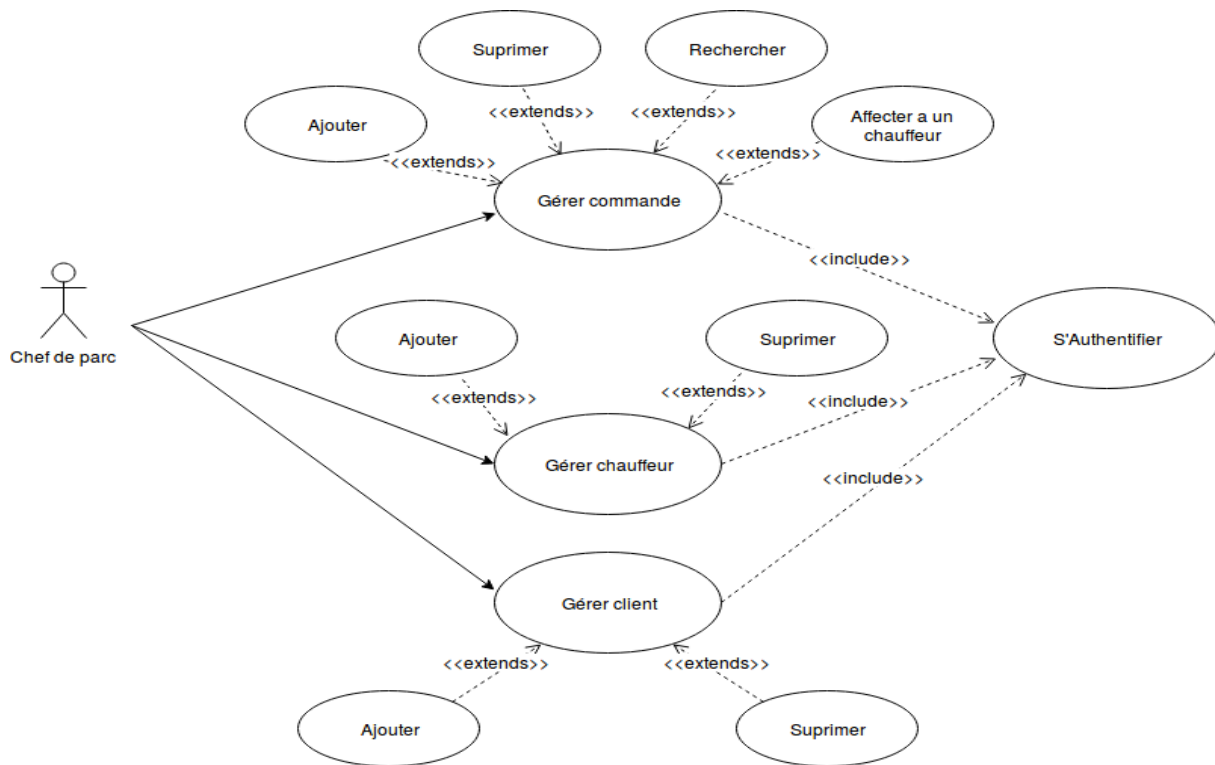


FIGURE 3.2 – Diagramme de cas d'utilisation pour le chef de parc.

3.7 Diagramme de séquence :

Un diagramme de séquence est un diagramme UML (Unified Modeling Language) qui représente la séquence de messages entre les objets au cours d'une interaction, et comprend un groupe d'objets représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction.

Le formalisme utilisé pour modéliser les diagrammes de séquence inclut les objets suivants :

- **objet d'interface** : ils représentent l'interface entre l'acteur et le système.

L'icone :



- **Objets contrôles** : il représente un processus (une activité du système)

L'icone :



- **Objets entité** : c'est un objet décrits dans le cas d'utilisation.

L'icone :



3.7.1 Diagramme de séquence pour le cas d'utilisation ' Authentification d'un chauffeur ' :

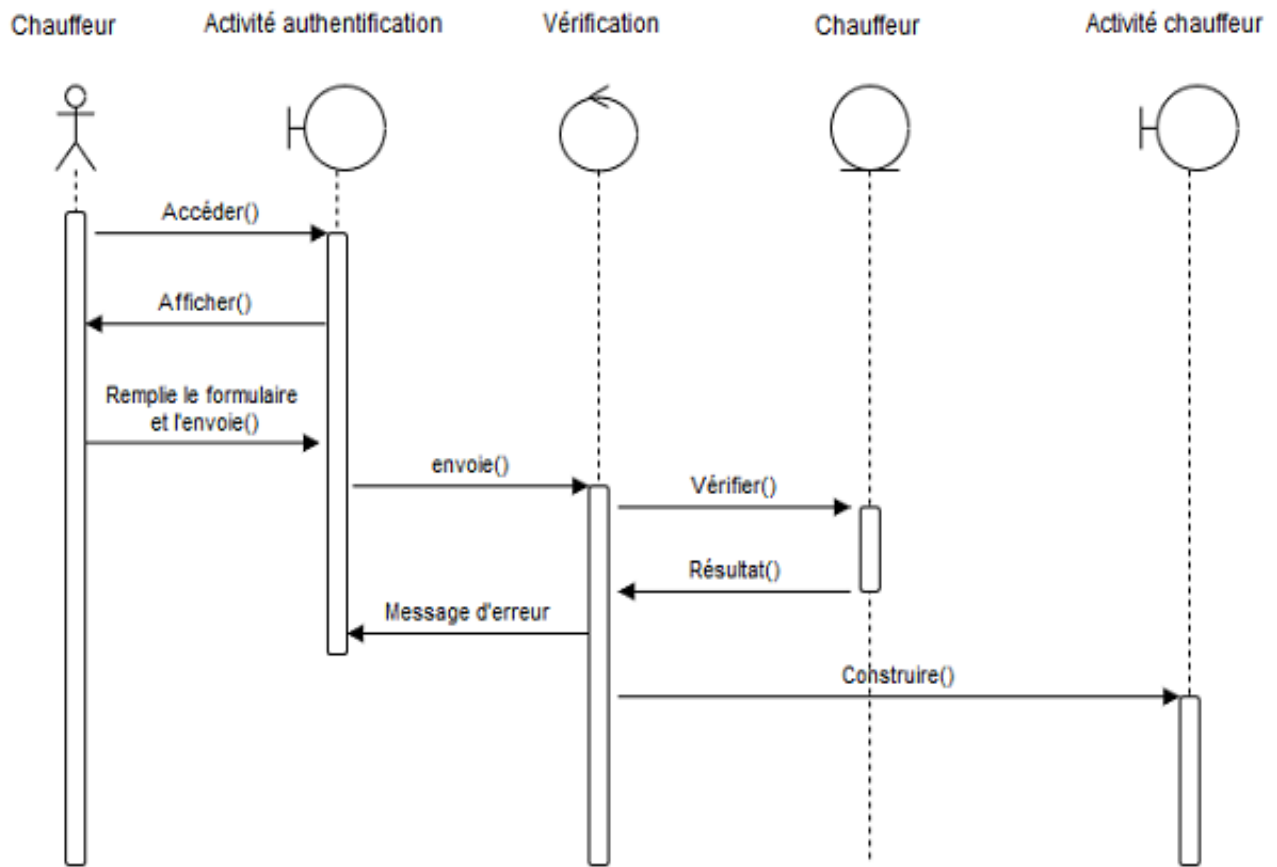


FIGURE 3.3 – diagramme de séquence pour le cas d'utilisation 'Authentification d'un chauffeur'.

3.7.2 Diagramme de séquence pour le cas d'utilisation 'Consulter l'itinéraire du jour' :

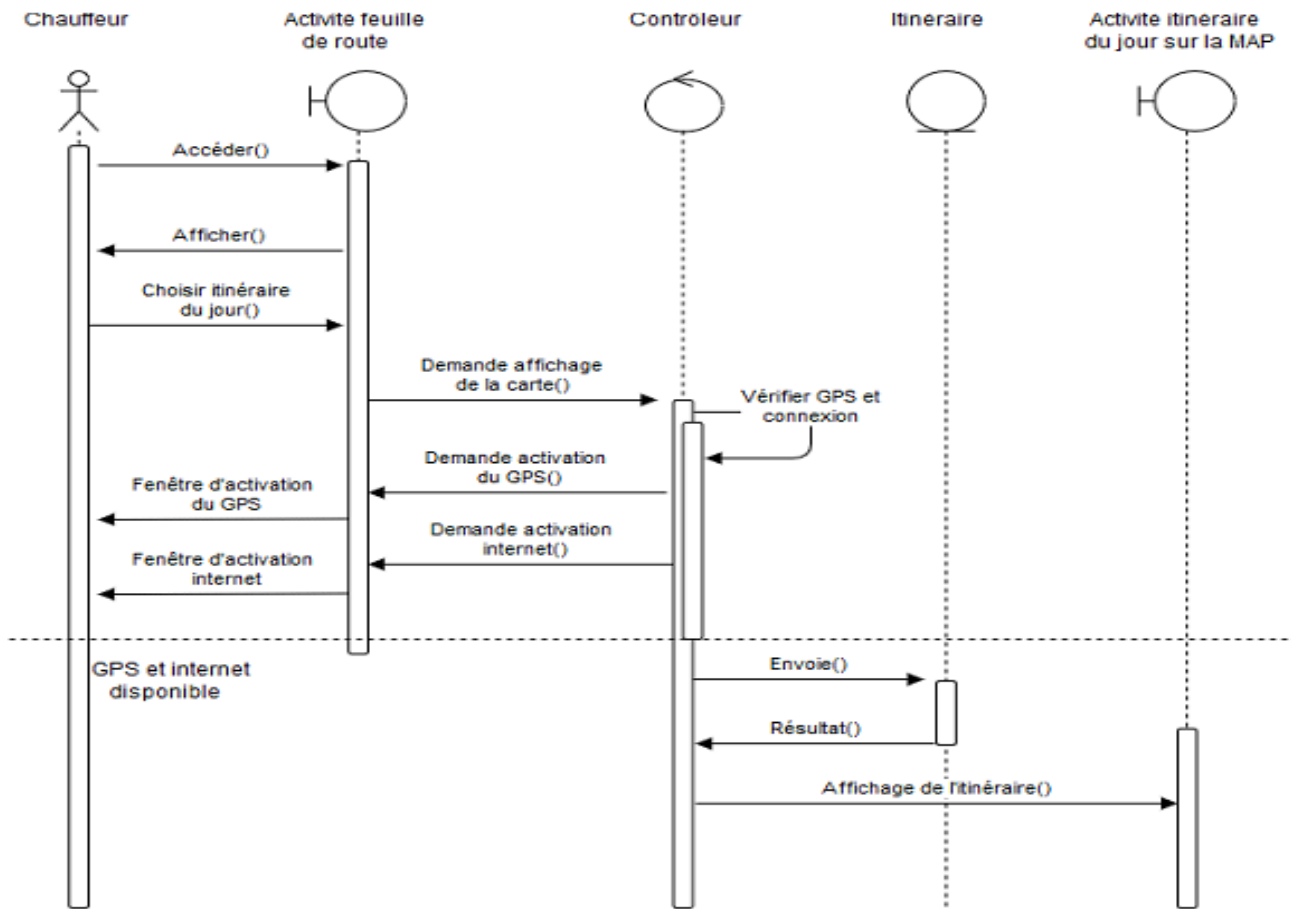


FIGURE 3.4 – diagramme de séquence pour le cas d'utilisation ' Consulter l'itinéraire du jour'.

3.7.3 Diagramme de séquence pour le cas d'utilisation 'valider la livraison d'une commande' :

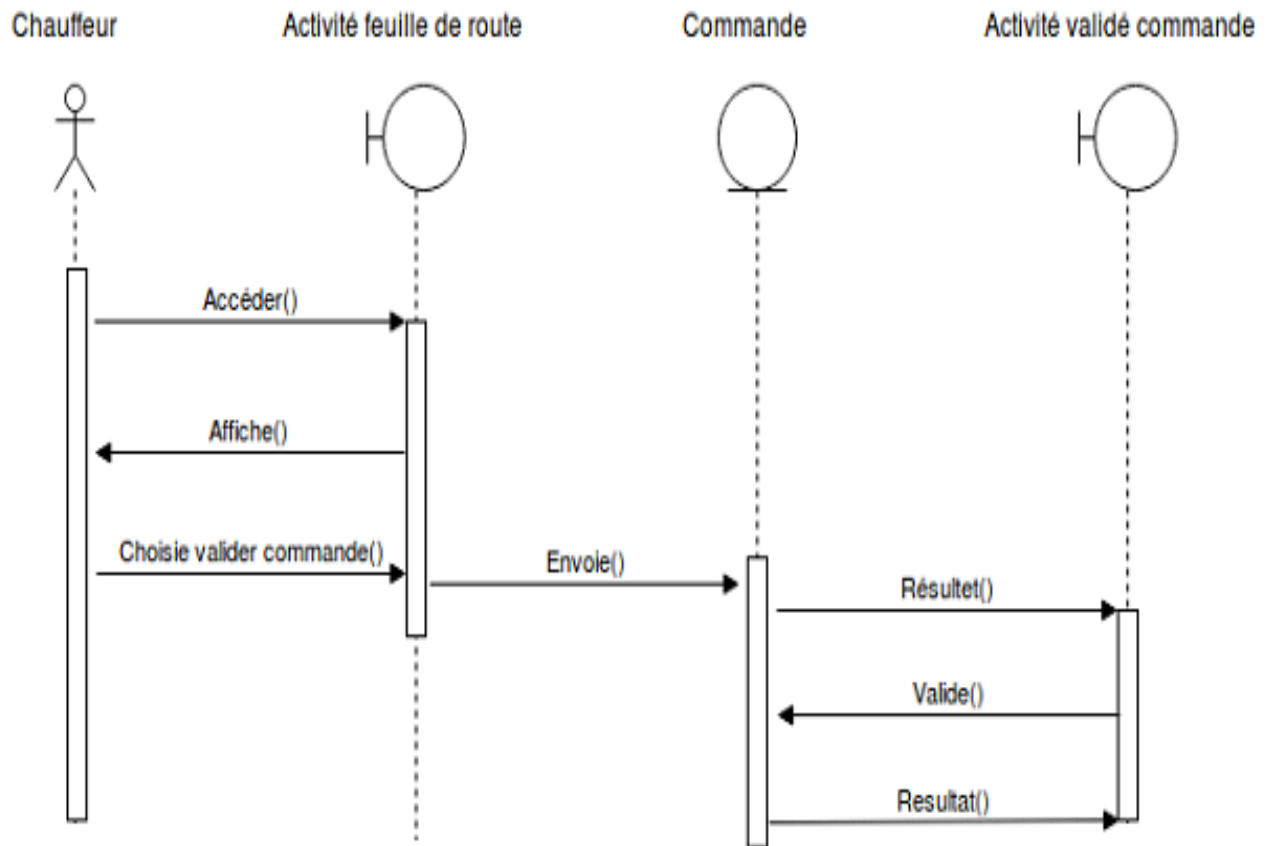


FIGURE 3.5 – diagramme de séquence pour le cas d'utilisation ' valider la livraison d'une commande'.

3.8 Diagramme de classe :

Les diagrammes de classes sont fondamentaux pour le processus de modélisation des objets et modélisent la structure statique d'un système, et sont principalement utilisés pour afficher les relations entre les objets et pour décrire ce que ces objets font et les services qu'ils fournissent.

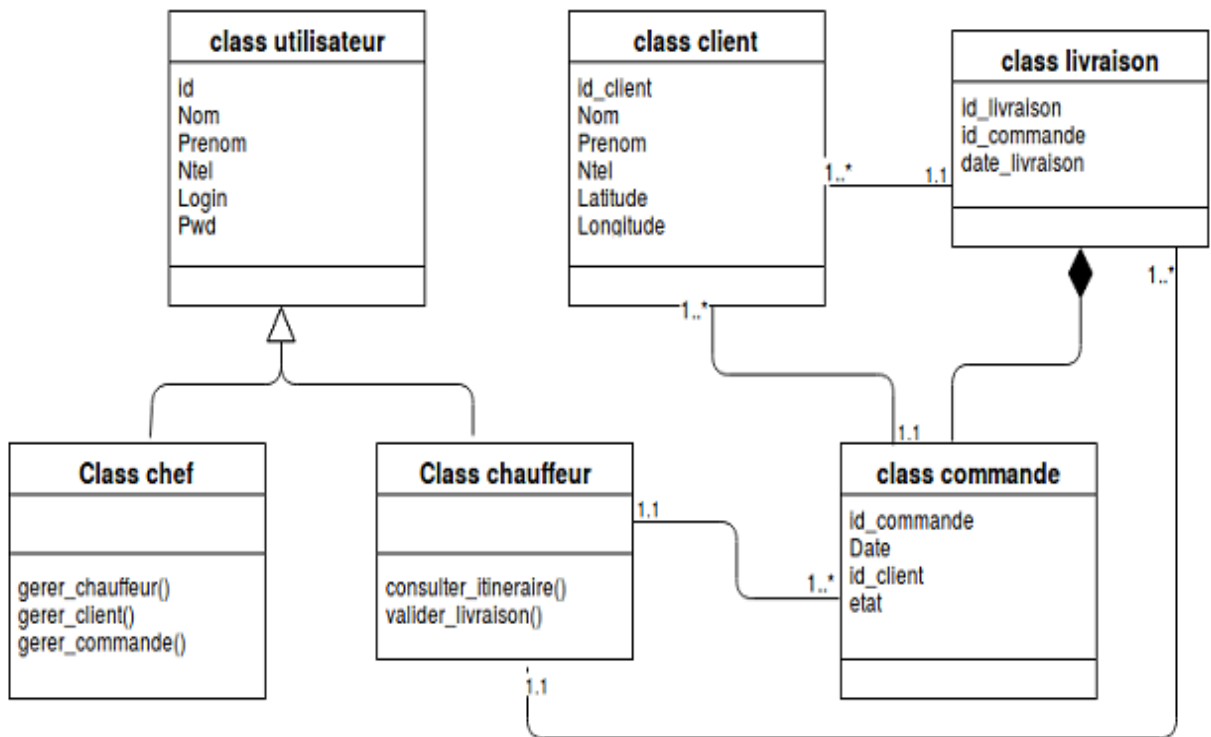


FIGURE 3.6 – diagramme de classe générale.

3.9 Le modèle logique de données :

Une fois le diagramme de classe final établi (modèle conceptuel de données), il est nécessaire de faire le passage au modèle logique des données, tout en respectant des règles de passage.

Pour représenter le niveau logique des données, nous avons opté pour le modèle relationnel, qui offre une modélisation rigoureuse et simple. Aussi, il a permis aux SGBD relationnels de s'imposer dans les projets de développement des systèmes d'information.

3.9.1 Le schéma relationnelle correspondant au diagramme de class :

chef(Id , login , MotDePasse)

chauffeur(matricule , nom , prenom , NTel , login , MotDePasse)

client (IdClient , nom , prenom , NTel , latitude , langitude)

commande (NumCom , date , IdClient* , etat)

livraison (IdLivraison , date , NumCom* , matricule*)

3.10 Les tables de base de données :

Les tables de base de données qui seront implémenté dans notre système sont : Les abréviations suivantes signifient :

A : Alphabétique.

N : Numérique.

3.10.1 Les tables de base de données :

id	Login	Mot de pass
N	AN	AN

3.10.2 La table chef de parc :

Matricule	Nom	Prénom	NTel	Latitude	Longitude	Login	Mot de passe
AN	A	A	N	N	N	AN	AN

3.10.3 La table client :

IdClient	Nom	Prénom	NTel	Latitude	Longitude
AN	A	A	N	N	N

3.10.4 La table commande :

NumCommande	Date	IdClient(foreign key)	Etat
AN	DATE	AN	A

3.11 Optimisation de l'itinéraire :

Dans le contexte de notre application l'optimisation de l'acheminement des produits aux clients revient à trouver le plus court chemin adapté au type de transport utilisé (principalement des camions).

En théorie on suppose un graphe qui a comme sommets les points de ventes (clients) et comme arête le chemin reliant deux destinations, donc il s'agit de réduire le poids des arêtes tout en conservant un graphe composé d'une seule composante connexe.

Une composante connexe C d'un graphe est un ensemble de nœuds tel que pour tout couple de nœuds $(s, t) \in C^2$, il existe une chaîne reliant s et t et dont tous les nœuds intermédiaires sont inclus dans C .

de manière globale notre problématique ressemble à un problème connu en mathématique, le problème du voyageur de commerce .

Le problème du voyageur de commerce est un problème d'optimisation qui, étant donné une liste de villes, et des distances entre toutes les paires de villes, détermine un plus court chemin qui visite chaque ville une et une seule fois et qui termine dans la ville de départ.

Malgré sa simplicité, il s'agit d'un problème d'optimisation pour lequel on ne connaît pas d'algorithme permettant de trouver une solution exacte rapidement dans tous les cas. Plus précisément, on ne connaît pas d'algorithme en temps polynomial, et sa version décisionnelle (pour une distance D , existe-t-il un chemin plus court que D passant par toutes les villes et qui termine dans la ville de départ ?) est un problème NP-complet, ce qui est un indice de sa difficulté.

Un problème NP-complet (un problème complet pour la classe de complexité NP) est un problème de décision qui fait partie des problèmes les plus difficiles à résoudre de cette classe. En ce sens, il est un représentant de la classe. C'est une notion centrale en complexité. Elle permet notamment d'établir des inclusions entre les classes en ne considérant qu'un seul problème.

En pratique, lorsqu'on travaille sur un problème NP-complet, il vaut mieux chercher des solutions approchées en utilisant des algorithmes d'approximation ou utiliser des heuristiques pour trouver des solutions acceptables. Comme algorithme d'approximation on trouve l'algorithme de Christofides qui ne résout pas l'intégralité du problème mais des sous problèmes comme l'arbre couvrant de poids minimal et celui du couplage de poids minimum.

Dans notre cas on a opté pour une solution simple mais qui donne un résultat rapide est satisfaisant, l'algorithme voisin le plus proche qui consiste à construire une tournée en partant de rien, de façon exacte l'algorithme a comme entrée un tableau de sommets (les sommets constituent le circuit) puis les réordonne en minimisant la distance entre 2 sommets voisins dans le tableau ie minimiser $C(v_{k-1}, v_k)$ et offre en sortie un tableau ordonné en voisin plus proche.

Algorithme voisin le plus proche :

Début

Choisir un sommet $v_1 \in V$

Poser $k \leftarrow 1$

Tant que $k < n$ faire

$k \leftarrow k + 1$

Choisir v_k dans $V - \{v_1, v_2, \dots, v_{k-1}\}$

// a chaque fois qu'on choisi un v_k on l'enlève de l'ensemble v

qui minimise C_{v_{k-1}, v_k}

Fin Tant que

Retourne le tour (v_1, v_2, \dots, v_n)

Fin

A la fin d'exécution de cette algorithme on aura comme résultat un circuit hamiltonien de précision de 1.26 fois plus élevé que la valeur optimale .

3.12 Conclusion :

Dans ce chapitre nous avons conçu le système d'optimisation des chemins de transport a l'aide du langage de modélisation graphique UML qui fourné une méthode normalisée pour visualiser la conception d'un système.

Dans le chapitre suivant nous allons présenter les déffirente étape de réalisation de notre système ainsi que les langages et outils de développement utilisé.

Chapitre 4

Réalisation

4.1 Outils et langages de programmation :

4.1.1 Introduction :

Dans ce chapitre nous allons présenter la création et la mise en oeuvre des différents programmes, interface et base de données qui constituent notre système et ces fonctionnalités.

Nous allons commencer par présenter l'environnement de création du système et de la base de données, ensuite nous présenterons quelques interfaces résultantes.

4.1.2 Netbeans :

L'EDI NetBeans est un environnement de développement - un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java - mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'EDI NetBeans. L'EDI NetBeans est un produit gratuit, sans aucune restriction quant à son usage.

Également disponible, La Plateforme NetBeans ; une fondation modulaire et extensible utilisée comme brique logicielle pour la création d'applications bureautiques. Les partenaires privilégiés fournissent des modules à valeurs rajoutées qui s'intègrent facilement à la Plateforme et peuvent être utilisés pour développer ses propres outils et solutions.

Les deux produits sont open source et gratuits pour un usage commercial et non-commercial. Le code source est disponible pour réutilisation sous la Common Development and Distribution License (CDDL). [1]



FIGURE 4.1 – Netbeans .

4.1.3 MYSQL :

MySQL – un système populaires pour créer et contrôler des bases de données. Le logiciel possède une haute vitesse du traitement des bases de données et réduit des dépenses d'extension des possibilités du système ou de gestion des bases de données. Avec le serveur interne, l'utilisateur a la possibilité d'ajouter MySQL aux logiciels autonomes. Le logiciel soutient aussi de nombreux types de tableaux qui fournissent la recherche de texte complet et les transactions au niveau des notes individuelles.



FIGURE 4.2 – MYSQL .

4.1.4 Glassfish :

Glassfish est un serveur d'applications Java EE dont le développement a été initié et est aujourd'hui encore dirigé par SUN. Le serveur d'application de SUN est disponible sous une double licence Common Development and Distribution License (CDDL) et GNU General Public License (GPL) avec exception du classpath. Le projet glassFish a été initié en 2005 et est aujourd'hui considéré comme un produit stable déjà utilisé en production .



FIGURE 4.3 – MYSQL .

4.1.5 Android Studio :

Android Studio est un nouvel environnement pour développement et programmation entièrement intégré qui a été récemment lancé par Google pour les systèmes Android. Il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l'IDE le plus utilisé.

Lorsque vous commencez un nouveau projet avec Android studio, la structure des projets apparaîtra avec pratiquement tous les fichiers du répertoire SDK, ceci vous redirigera vers un système de gestion conçu en Gradle offrant encore plus de flexibilité au processus de construction.

Android Studio permet de voir chacun des changements visuels que vous effectuez sur votre application et en temps réel, vous pourrez voir aussi son effet sur différents appareils Android, chacune avec différentes configurations et configurations simultanément.



FIGURE 4.4 – Android Studio .

4.1.6 Plateform Java EE (Edition Entreprise) :

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

- un ensemble de spécifications pour une infrastructure dans laquelle s'exécutent les composants écrits en Java : un tel environnement se nomme serveur d'applications.
- un ensemble d'API qui peuvent être obtenues et utilisées séparément. Pour être utilisées, certaines nécessitent une implémentation de la part d'un fournisseur tiers.

L'utilisation de Java EE pour développer et exécuter une application offre plusieurs avantages :

- une architecture d'applications basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement .
- la possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA ...
- la possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres .

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

- **la partie cliente** : c'est la partie qui permet le dialogue avec l'utilisateur. Elle peut être composée d'une application standalone, d'une application web ou d'applets .
- **la partie métier** : c'est la partie qui encapsule les traitements (dans des EJB ou des JavaBeans).
- **la partie données** : c'est la partie qui stocke les données .[2]



FIGURE 4.5 – Plateform Java EE .

4.1.7 Langage de programmation JAVA :

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java. La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.



FIGURE 4.6 – java.

4.1.8 JSON :

JSON (JavaScript Object Notation) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée comme le permet XML par exemple. Créé par Douglas Crockford entre 2002 et 2005, il est décrit par la RFC 7159 de l'IETF. Un document JSON a pour fonction de représenter de l'information accompagnée d'étiquettes permettant d'en interpréter les divers éléments, sans aucune restriction sur le nombre de celles-ci.

Un document JSON ne comprend que deux types d'éléments structurels :

- des ensembles de paires nom / valeur .
- des listes ordonnées de valeurs.

Ces mêmes éléments représentent trois types de données :

- des objets .
- des tableaux .
- des valeurs génériques de type tableau, objet, booléen, nombre, chaîne ou null.



FIGURE 4.7 – JSON .

4.1.9 SQL :

Le SQL (Structured Query Language) est un langage permettant de communiquer avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d'un site web. SQL.sh recense des cours de SQL et des explications sur les principales commandes pour lire, insérer, modifier et supprimer des données dans une base.



FIGURE 4.8 – SQL .

4.1.10 Organisation de l'exécution des instructions sous android :

Lorsqu'une application s'exécute sur un environnement android , elle est exécuté sur un processus unique contenant une plage mémoire qui lui est propre .

Un processus exécute des instructions dans une ou plusieurs files d'exécution, appelé également Thread. Ces instructions seront exécutées sont exécuter séquentiellement .

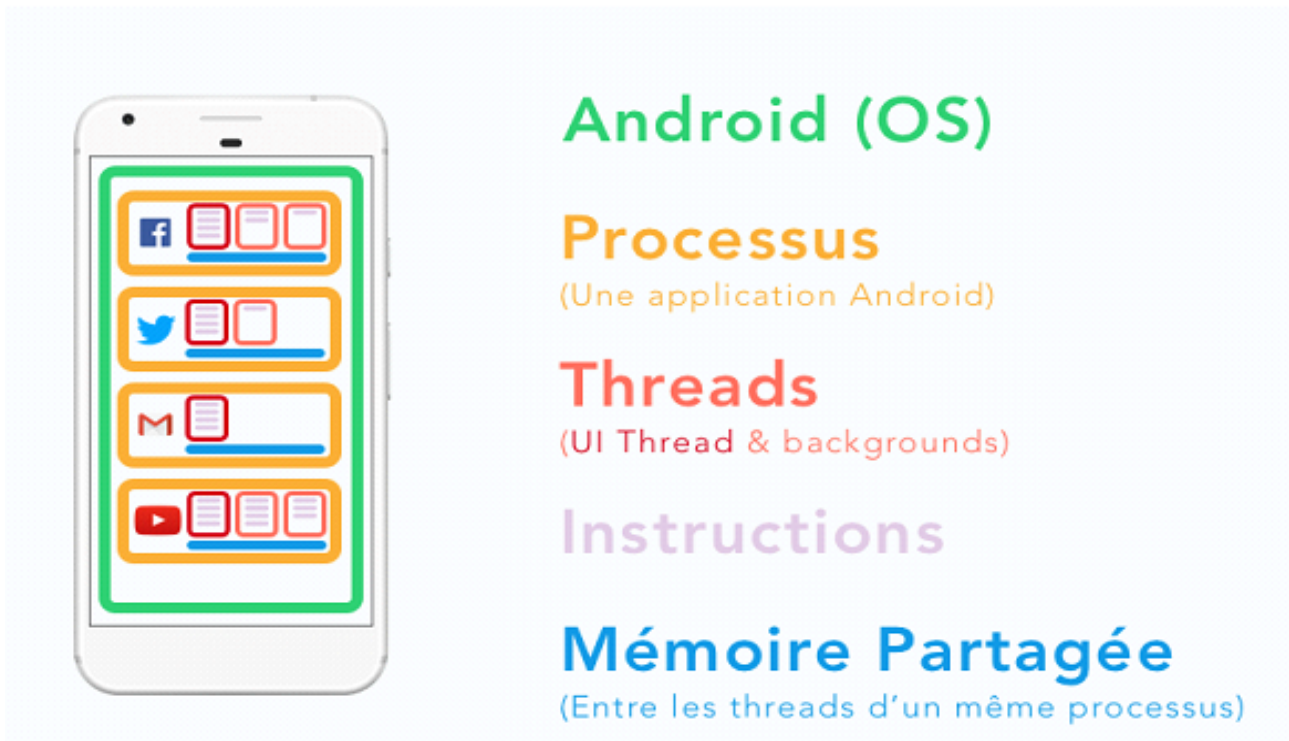


FIGURE 4.9 – l'exécution des applications sous une plateforme android.

Le fil d'exécution principal sur Android est appelé Main Thread (parfois aussi UI Thread, car essentiellement réservé à l'interface graphique de l'application) et c'est sur cette file que toutes les opérations sont exécutées par défaut. A fin de mieux exploiter les ressources et le temps d'exécution dans un processeur on peut créer une deuxième file d'instruction qui s'exécutera indépendamment de la file par défaut .

- Le Thread Principal (Main Thread) est généralement dédié aux opérations courtes qui consiste à mettre à jour un élément de l'interface graphique (UI).
- Les nouveaux Threads (Backgrounds) seront créés dès que l'on veut effectuer des tâches longues, afin d'éviter de bloquer l'interface graphique pour l'utilisateur.

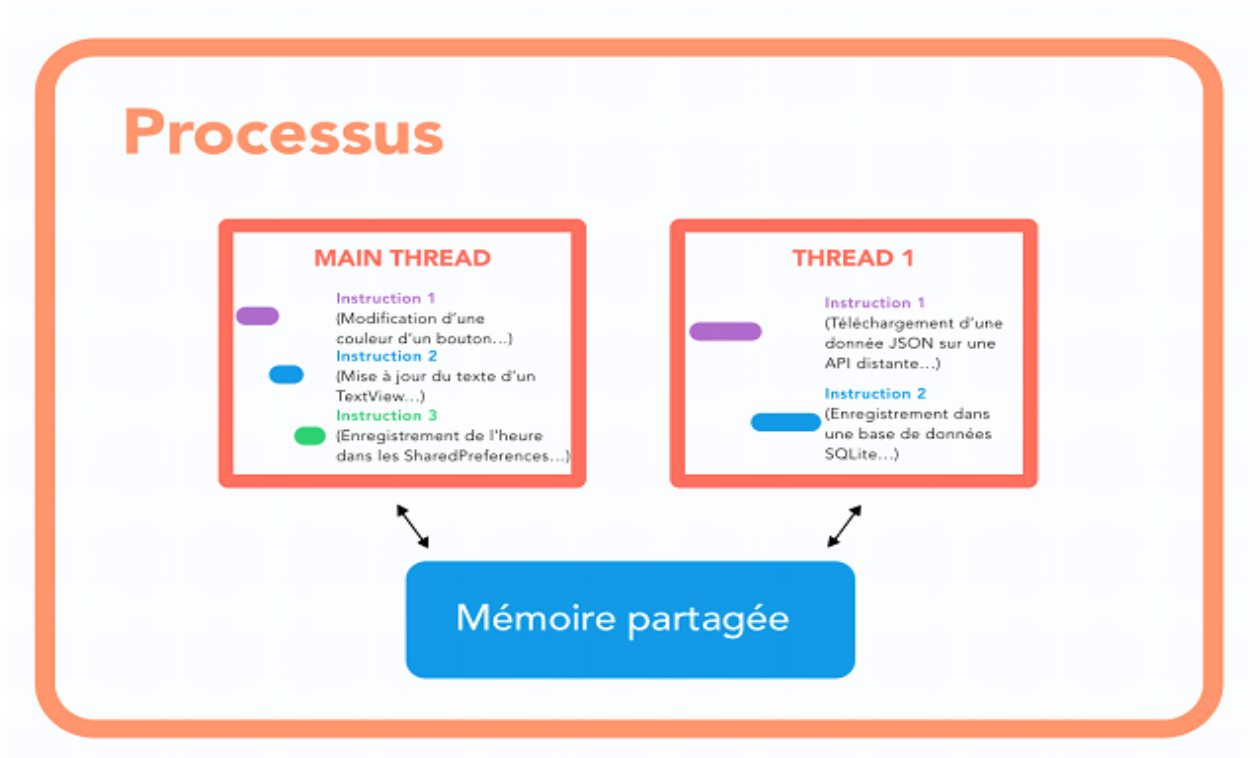


FIGURE 4.10 – la composition d'un processus sous une plateforme android .

4.2 Architecture de notre système :

4.2.1 Architecture de système :

On peut décomposer notre système en deux parties essentielles :

- JEE WebApp.
- Android Application.

4.2.2 première partie la partie JEE WebApplication :

Dans cette partie on a conçu une application web basée sur Java Edition Enterprise qui prend en main l'approvisionnement du système en matière de données, c'est dans cette application que l'administrateur du système entre toutes les données essentielles pour le fonctionnement de notre système.

Dans cette partie l'administrateur peut :

- Gérer les commandes .
- Gérer les chauffeurs .
- Gérer les Clients .

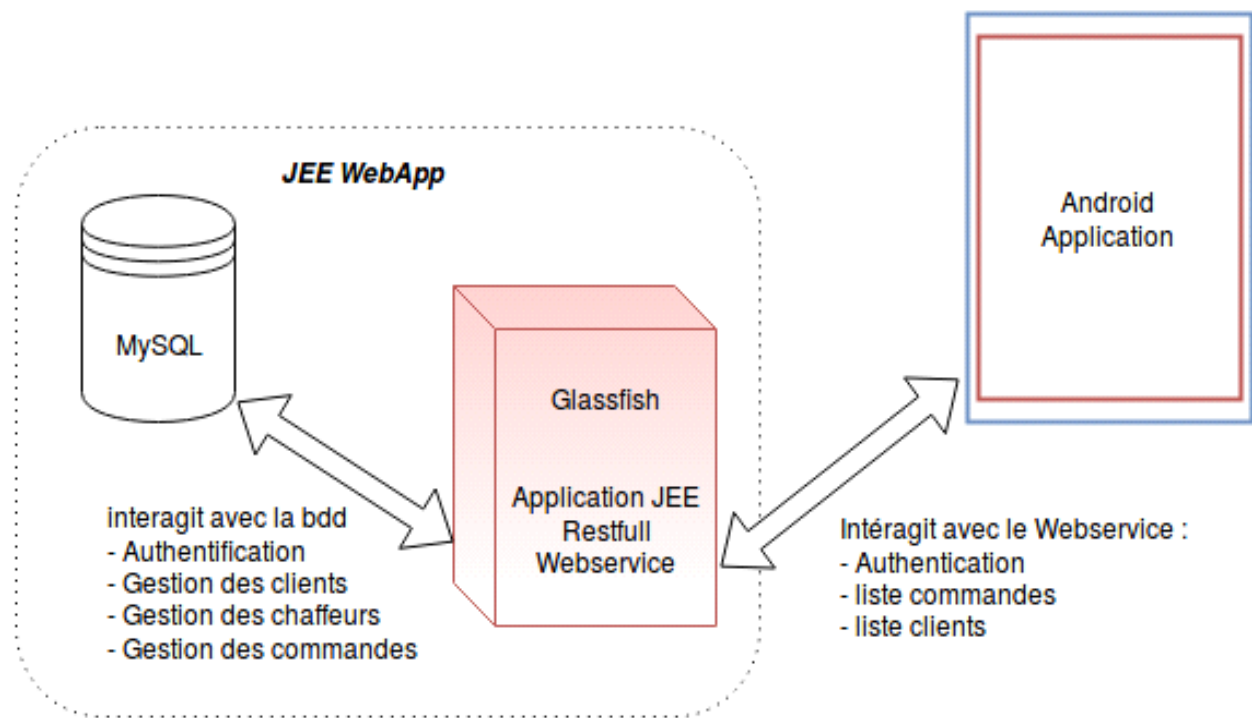


FIGURE 4.11 – Architecture de système.

L'application interagit avec une base de donnée MySQL sur la quelle est stocké toutes les informations, et fournit des service web de type restfull a fin d'assuré l'interopérabilité entre l'application mobile et l'application web.

4.2.3 Structure des tables :

— La table chef de parc (administrateur du système) :

Nom du champs	Type de données	Taille	Description
Id	Numérique	2	Identificateur admin
login	Alphanumérique	50	Nom d'utilisateur
pwd	Alphanumérique	50	Mot de passe utilisateur

— La table chauffeur :

Nom du champs	Type de données	Taille	Description
Matricule	Alphanumérique	10	Identificateur chauffeur
Nom	Alphabétique	30	Nom du chauffeur
Prénom	Alphabétique	30	Prénom du chauffeur
Login	Alphanumérique	50	Nom d'utilisateur
Pwd	Alphanumérique	50	Mot de passe utilisateur
idLivraison	Numérique	10	Clé étrangère de la table livraison

— La table livraison :

Nom du champs	Type de données	Taille	Description
idLivraison	Numérique	10	Identificateur livraison
DateLivraison	date		Date de livraison
Matricule	Alphanumérique	10	Identificateur chauffeur
idCom	Alphanumérique	10	Identificateur commande

— La table client :

Nom du champs	Type de données	Taille	Description
idClient	Alphanumérique	10	Identificateur client
Nom	Alphabétique	30	Nom du client
Prénom	Alphabétique	30	Prénom du client
Ntel	Numérique	10	Numéro de téléphone du client
Lat	Réel		Latitude de la position du client
Lng	Réel		Longitude de la position du client

— La table commande :

Nom du champs	Type de données	Taille	Description
idCom	Alphanumérique	10	Identificateur commande
Date	Date		Date de commande
Etat	Booleén		Si la commande est livré ou pas
idClient	Alphanumérique	10	Identificateur client

4.2.4 Service web disponible sur l'application :

- **Service Authentification** :ce service permet de récupérer la liste des commandes affecté un chauffeur pour livraison en fournissant le matricule du chauffeur en paramètre , ce service est disponible sur :

<http://adresse ip de la machine server :8080/NAFTAL/webresources/commandes?mat=XXX>.

- **Service client concerné par les commande du jour** :ce service permet de récupérer la liste des commandes affecté un chauffeur pour livraison en fournissant le matricule du chauffeur en paramètre , ce service est disponible sur :

<http://adresse ip de la machine server :8080/NAFTAL/webresources/commandes?mat=XXX>

- **Service client concerné par les commande du jour** :ce service permet de récupérer la liste des clients dans le but de les afficher sur la carte de l'application cliente en fournissant le matricule du chauffeur en paramètre , ce service est disponible sur :
[http ://adresse ip de la machine server :8080/NAFTAL/webresources/clients ?mat=XXX](http://adresse ip de la machine server :8080/NAFTAL/webresources/clients ?mat=XXX) .
- **Service de validation d'une livraison** :ce service permet de modifier l'état d'une commande en une commande livré une fois livraison effectuer , ce service est disponible sur :
[http ://adresse ip de la machine server :8080/NAFTAL/webresources/validate ?idcom=XXX](http://adresse ip de la machine server :8080/NAFTAL/webresources/validate ?idcom=XXX).

4.2.5 Deuxième partie l'application android :

Dans cette partie on a conçu une application mobile basé sure java android qui fournit l'itinéraire a suivre pour le chauffeur et une interface pour confirmé la livraison d'une commande une fois livré. L'affichage de l'itinéraire pour le chauffeur ce fait sur une carte basé sur l'API GOOGLE MAP qui offre des services adéquat a notre problématique a savoir :

- un itinéraire adapté ou camion .
- le plus court chemin entre deux point .
- la distance entre deux points .

et qu'on associe avec l'algorithme du plus proche voisin qui complete cette tache en offrant un chemin non pas optimal mais acceptable vu la complexité de la problématique avec prise en compte de toutes les destination prévue lors de la livraison.

4.2.6 Le calcul de l'itinéraire :

Pour implémenter le calcul de l'itinéraire on a créé une tâche asynchrone qui a pour rôle d'appeler l'API Google Direction, afin de récupérer tous les chemins possible entre les nœuds entrés en paramètre et de mettre à jour notre carte.

Une tâche asynchrone sur Android a 3 paramètres exemple :

`AsyncTask < Void (1), Integer (2), Boolean (3) >`.

- Ce paramètre est le type des paramètres fournis à la tâche c'est à dire les données en entrée.
- Ce paramètre est le type de données transmises durant la progression du traitement.
- Ce paramètre est le type du résultat de la tâche, c'est à dire les données renvoyées par la tâche.

Et elle est composée de 3 fonctions essentielles :

- **onPreExecute()** : Cette méthode s'exécute sur `MainThread`, elle permet ainsi par exemple, de mettre à jour notre UI en lançant une animation de chargement (`ProgressBar`).
- **DoInBackground ()** : Cette méthode s'exécute dans un `Thread` à part, c'est une tâche longue destinée à être lancée en arrière-plan.
- **OnPostExecute()** : Cette méthode s'exécute sur `MainThread`, une fois que la tâche longue (`DoInBackground()`) soit terminée. Ça permet d'arrêter l'animation de chargement par exemple (`ProgressBar`).

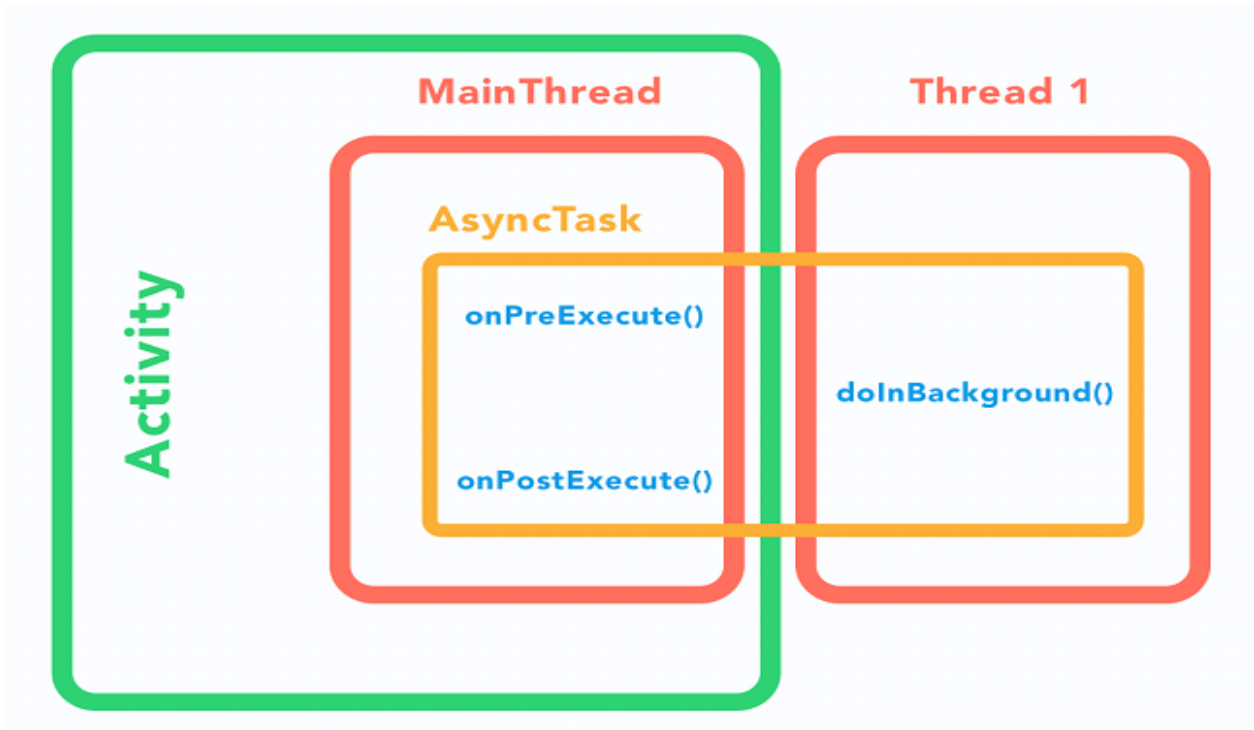


FIGURE 4.12 – illustration d’une tâche asynchrone sous androïde .

Dans notre cas nous avons implémenter dans la méthode `doInBackground` notre algorithme d’optimisation en utilisant l’API Distance Matrix de google map .

l’API Distance Matrix est utiliser dans une boucle a fin de récupérer le sommet qui satisfais la condition minimiser $C(v_{k-1}, v_k)$ ie un voisin plus proche d’un sommet choisie au début de la boucle. En fin dans la méthode `onPostExecute ()` on met a jour notre carte et la tâche est fini .

4.2.7 Confirmation d’une livraison :

En fin pour la confirmation de la livraison d’une commande on a prévu une activité sur la quelle on affiche la liste `ListView` cliquable , une fois livraison effectuée le chauffeur fais un long click sur la commande livrer pour la valider .

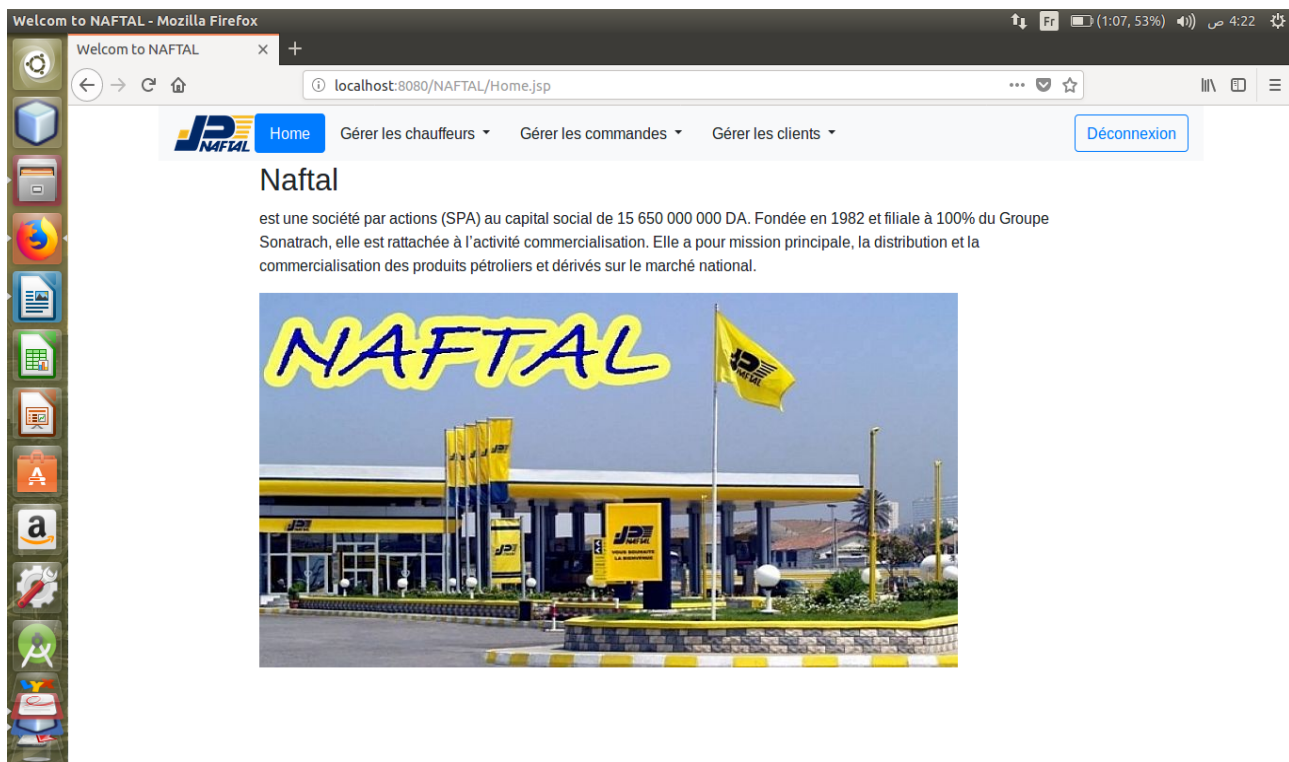


FIGURE 4.13 – La page d'accueil.

4.3 Interface de l'application JEE Web :

4.3.1 La page d'accueil :

4.3.2 la page d'authentification pour le chef de parc :

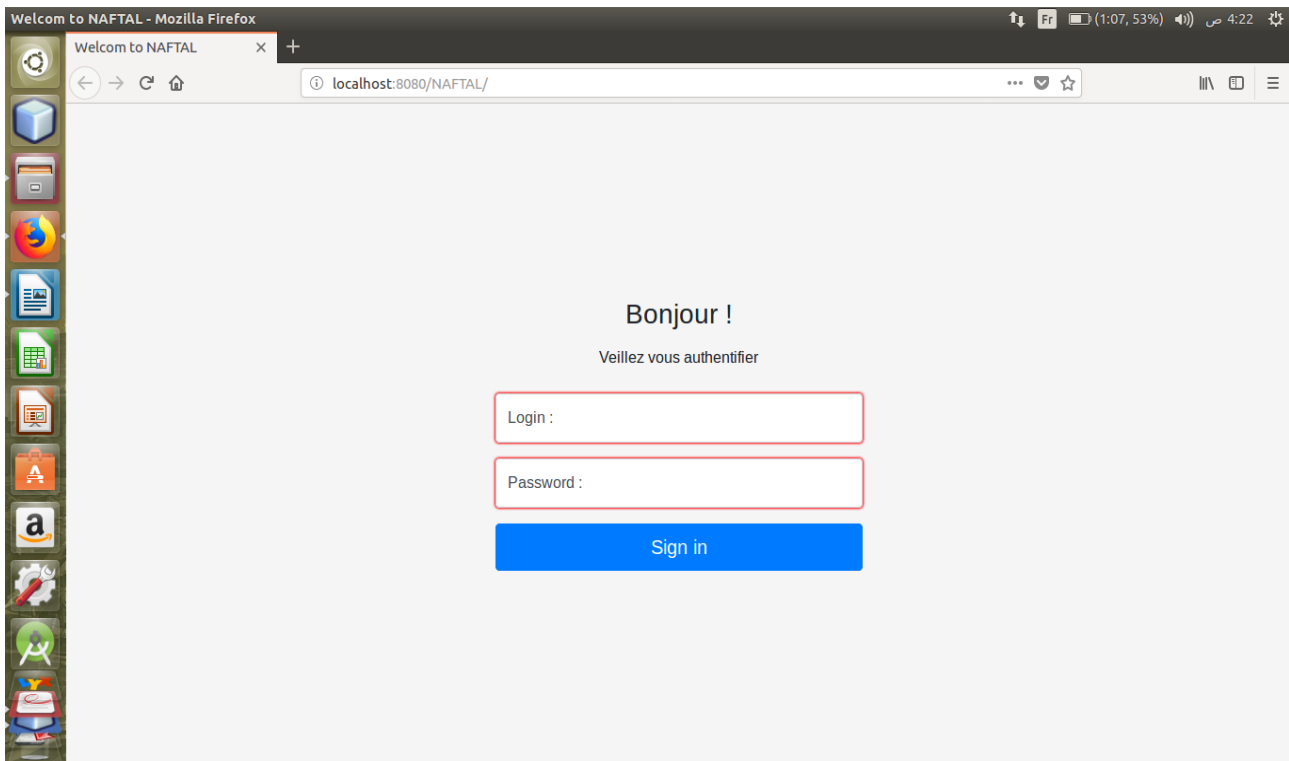


FIGURE 4.14 – la page d'authentification pour le chef de parc .

4.3.3 la page ajouter un chauffeur :

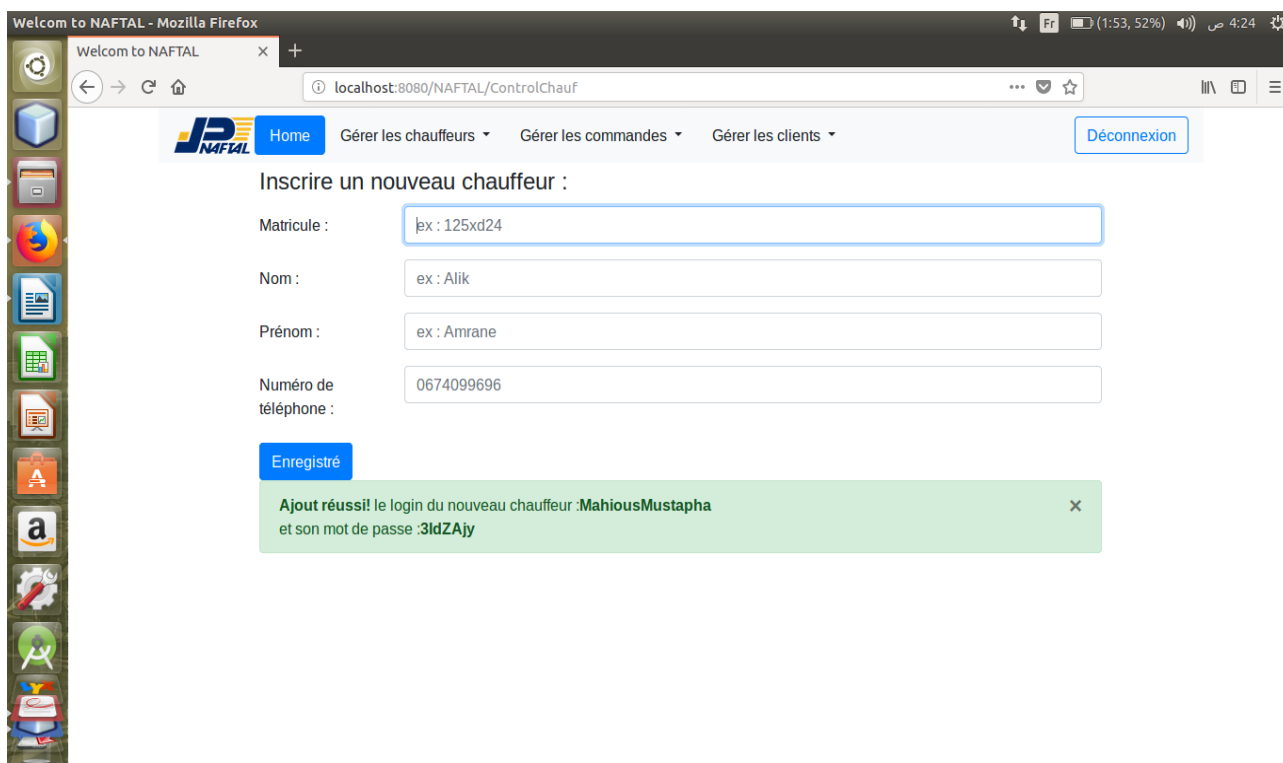


FIGURE 4.15 – la page ajouter un chauffeur .

pour l'ajout du chauffeur le chef de parc doit saisir les information concernant le chauffeur et le système générera son login et mot de passe .

4.3.4 la page supprimer chauffeur :

pour supprimer un chauffeur l administrateur doit saisir le matricule du chauffeur et valider .

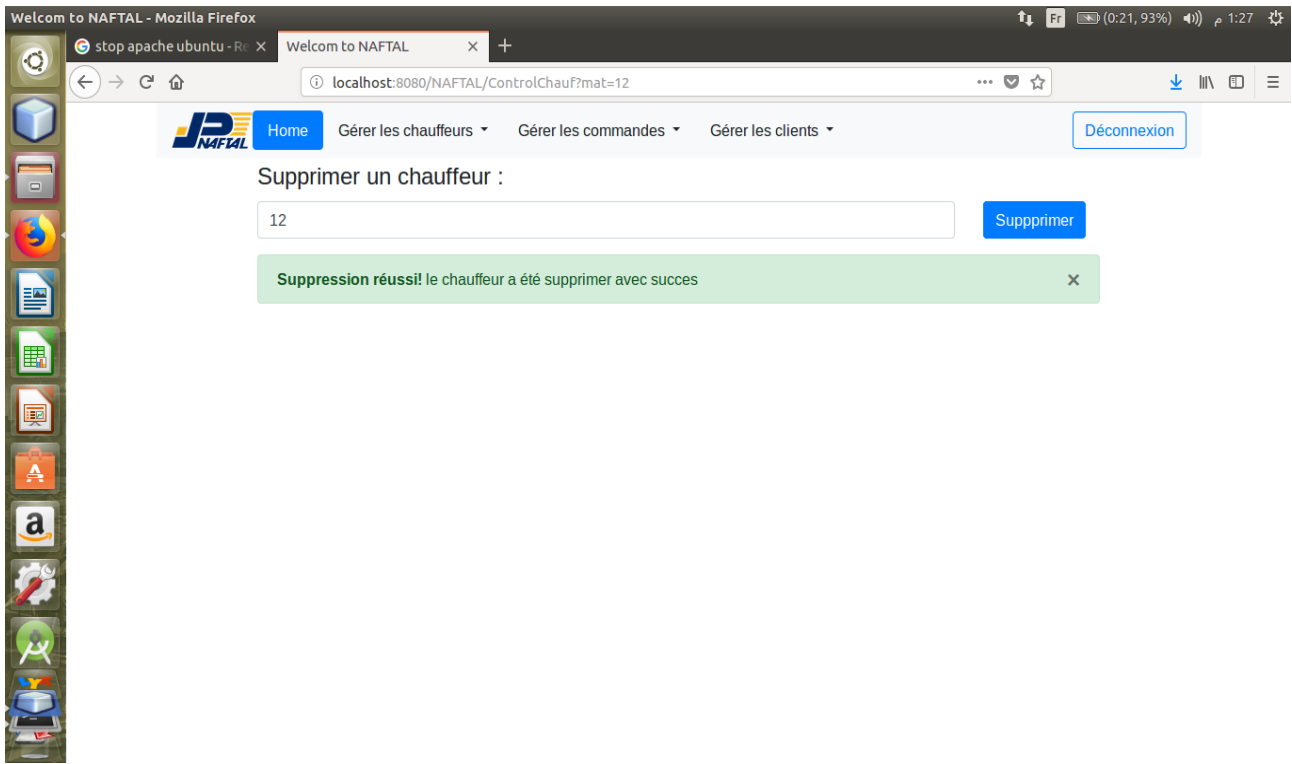


FIGURE 4.16 – la page supprimer chauffeur .

4.3.5 la page ajouter une commande :

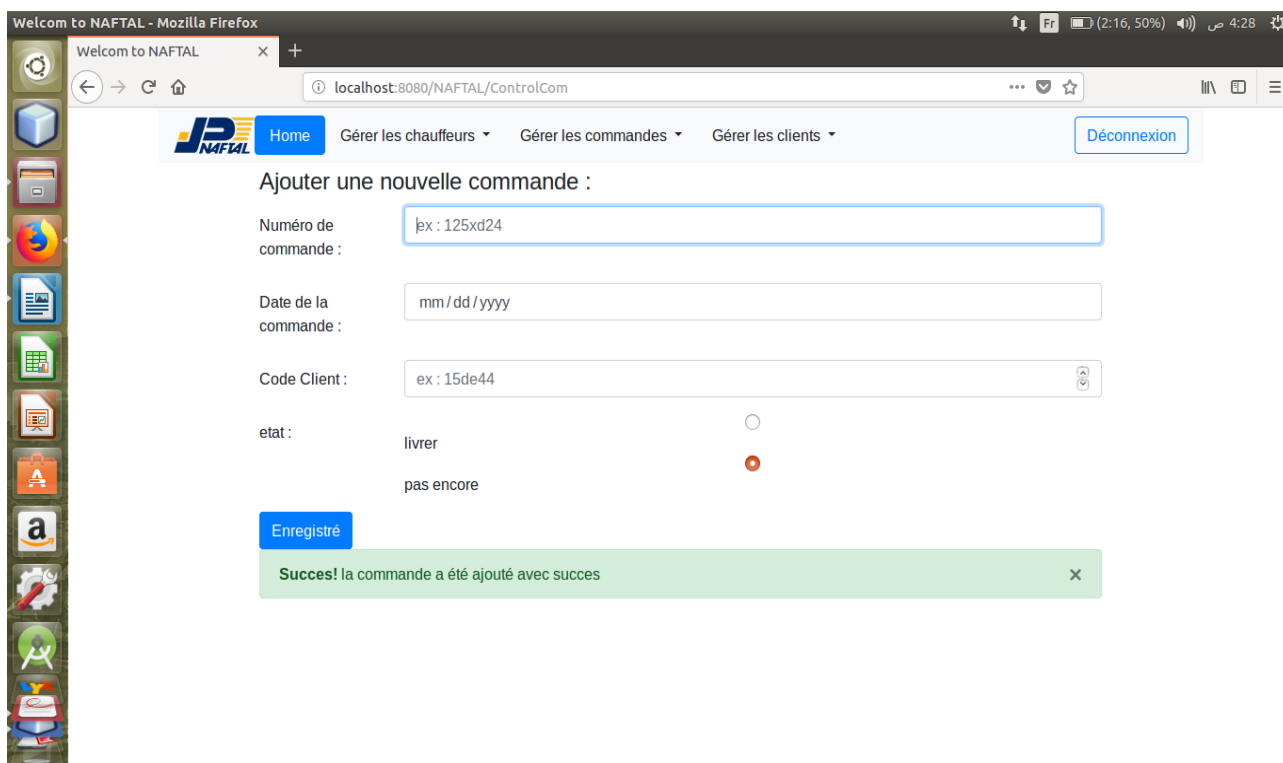


FIGURE 4.17 – la page ajouter une commande .

si le code client ne correspond a aucun client enregistré le système propose a l'administrateur de le crée en lui affichant un lien vers la page créé client.

4.3.6 la page supprimer une commande :

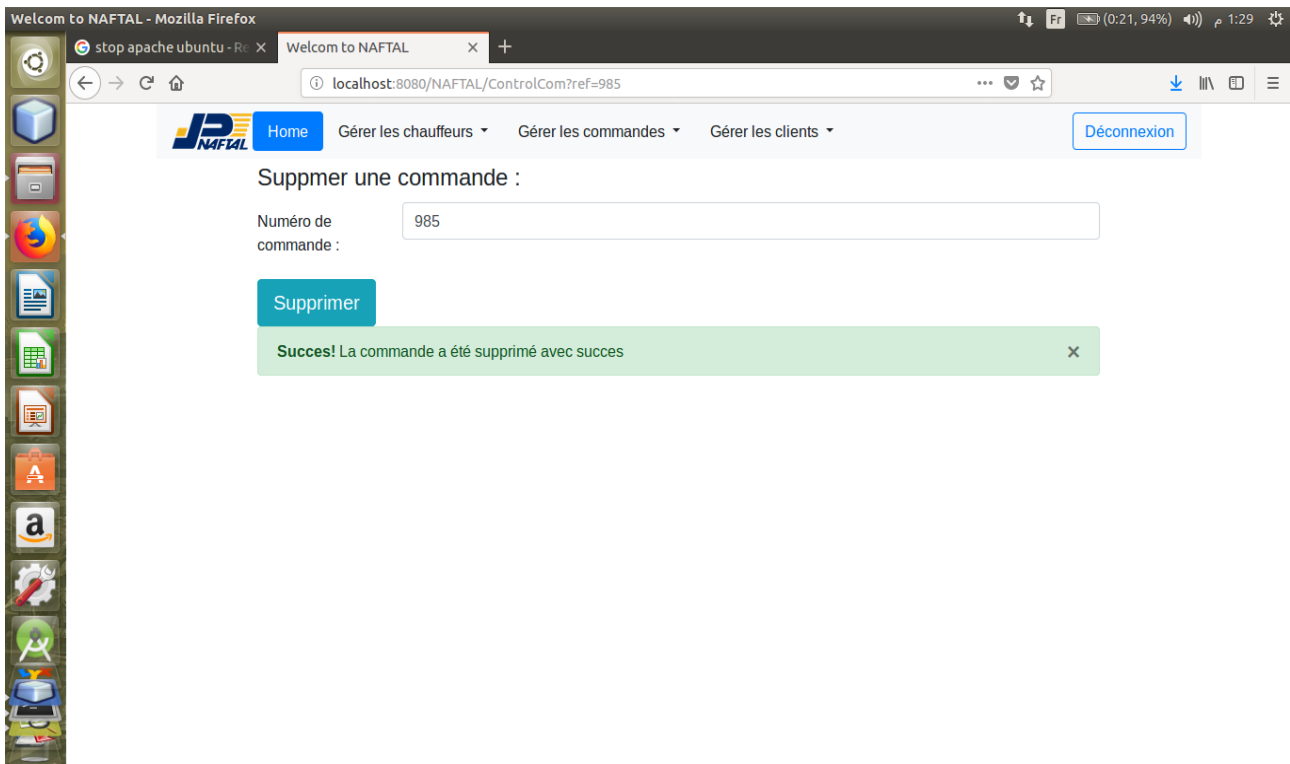


FIGURE 4.18 – la page supprimer une commande .

4.3.7 affecter une commande a un chauffeur :

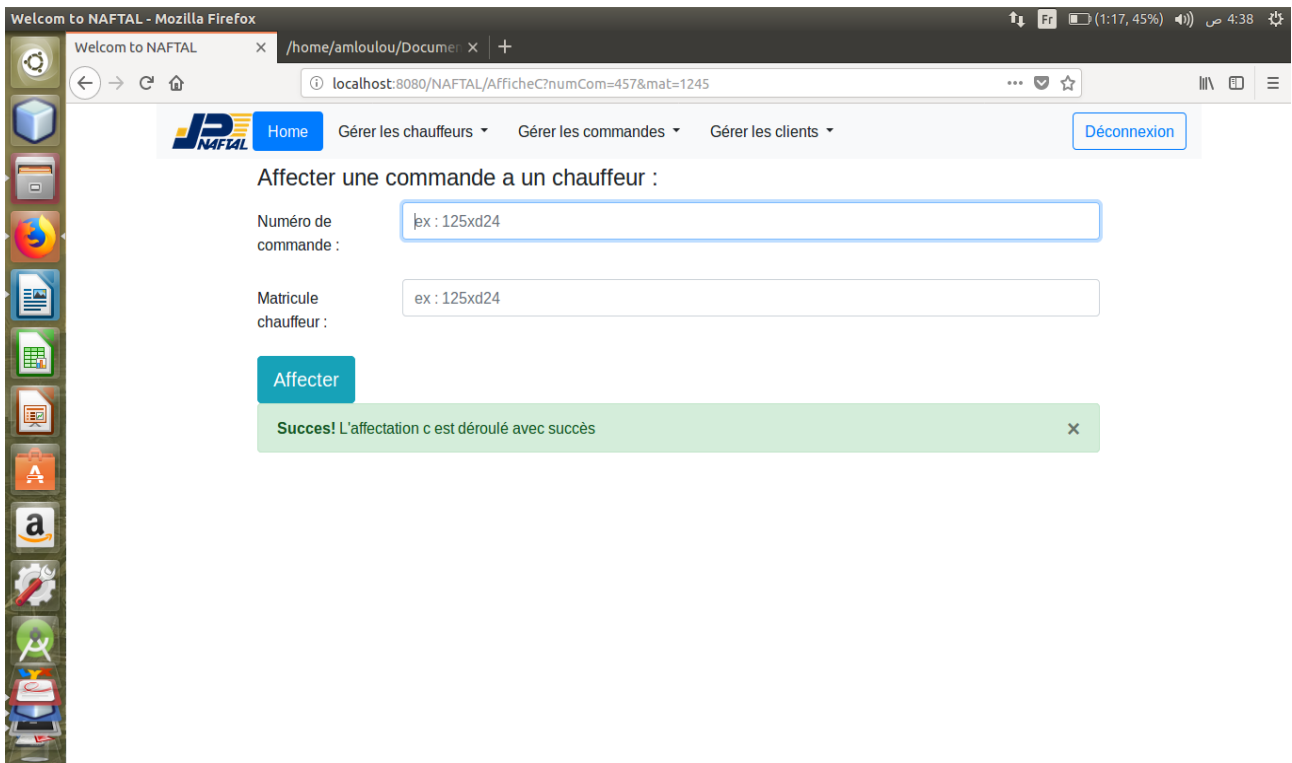


FIGURE 4.19 – affecter une commande a un chauffeur .

4.3.8 La page ajouter une nouvelle commande :

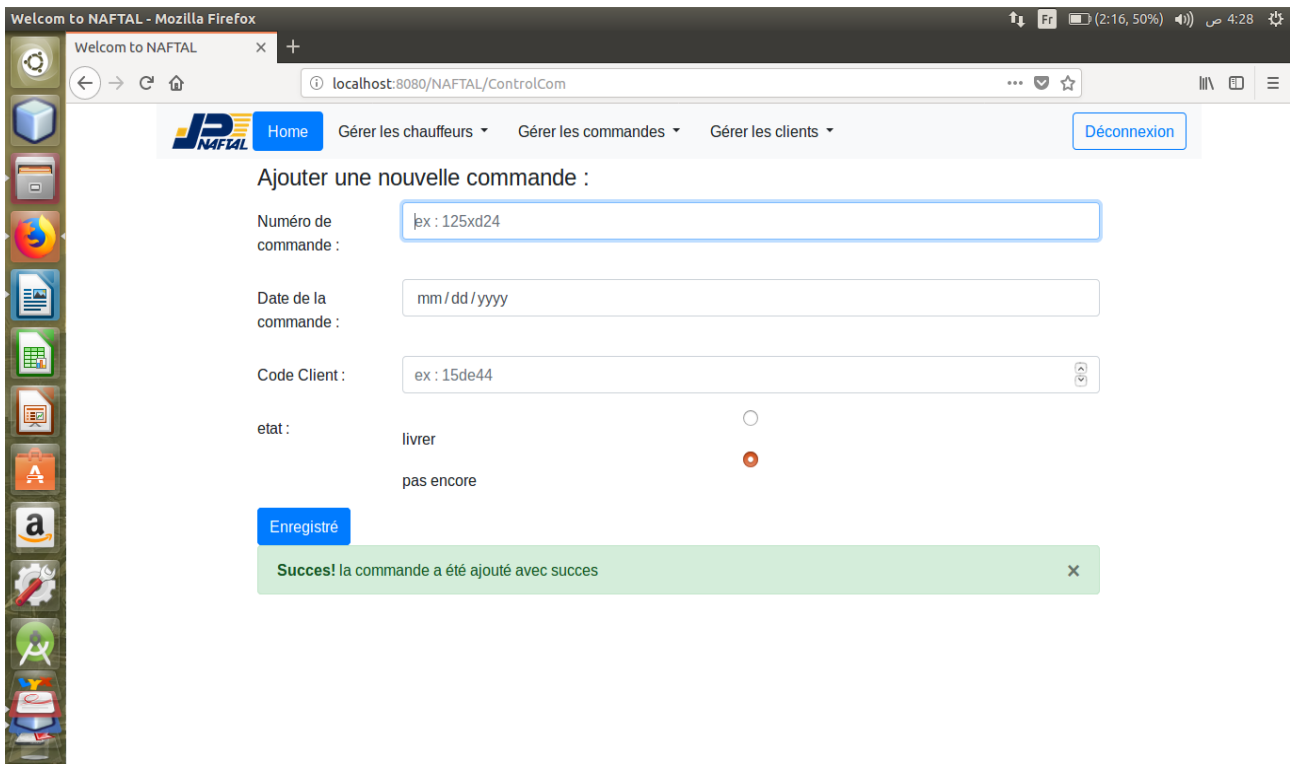


FIGURE 4.20 – La page ajouter une nouvelle commande.

4.3.9 la page recherche d'une commande :

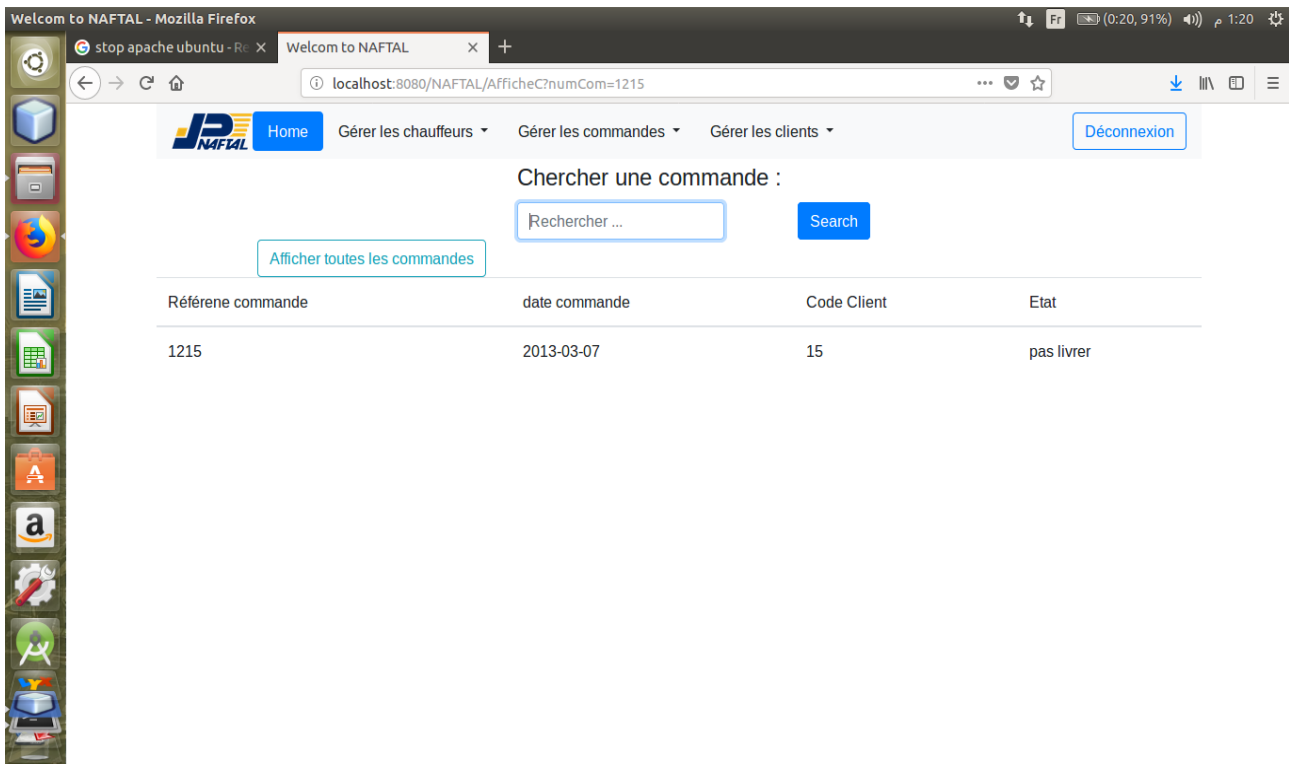
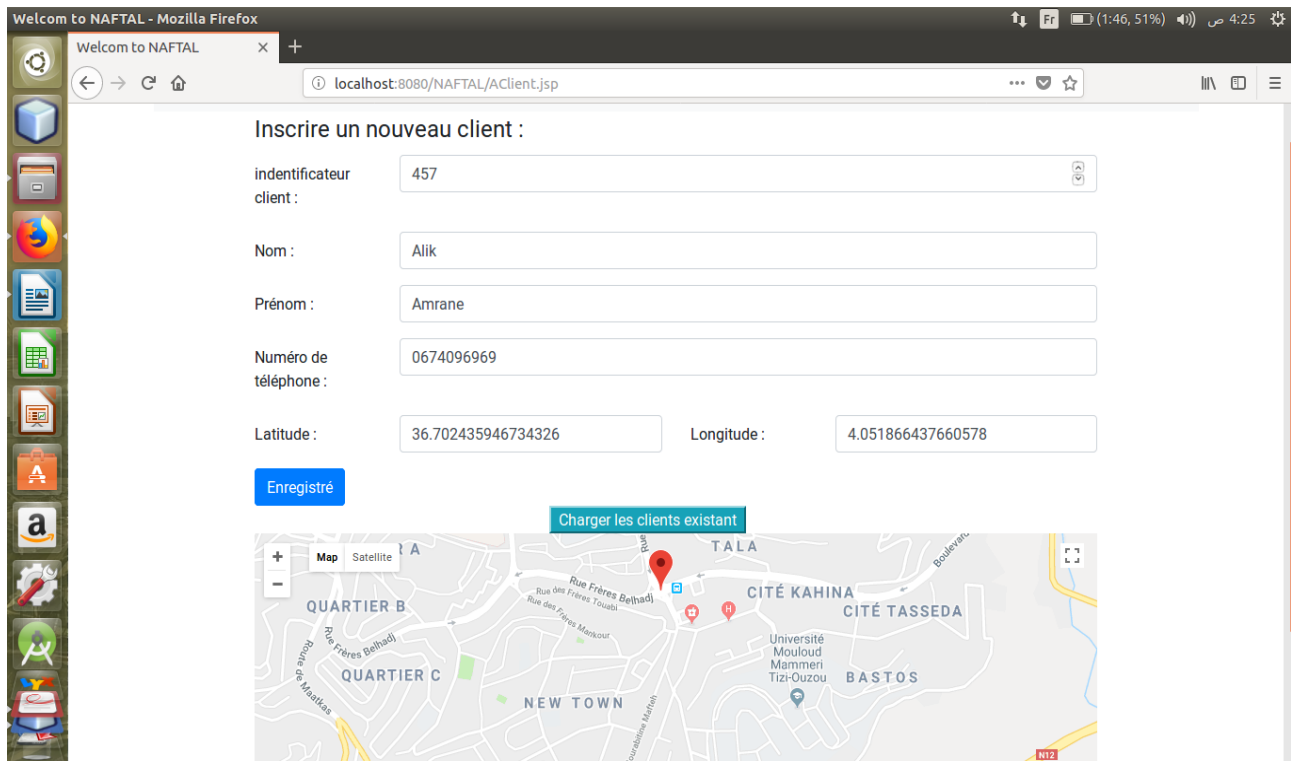


FIGURE 4.21 – la page recherche d'une commande .

4.3.10 la page ajouter client :



The screenshot shows a web browser window with the title "Welcom to NAFTAL - Mozilla Firefox". The address bar displays "localhost:8080/NAFTAL/AClient.jsp". The main content area is titled "Inscrire un nouveau client :" and contains a registration form with the following fields:

- identificateur client : 457
- Nom : Alik
- Prénom : Amrane
- Numéro de téléphone : 0674096969
- Latitude : 36.702435946734326
- Longitude : 4.051866437660578

Below the form is a blue button labeled "Enregistré". To the right of the form is a button labeled "Charger les clients existant". Below these buttons is a map showing a street grid with labels for "QUARTIER B", "QUARTIER C", "NEW TOWN", "CITÉ KAHINA", "CITÉ TASSEDA", "BASTOS", and "TALA". A red location pin is placed on the map, and a blue location pin is also visible. The map includes a search bar with the letter "A" and a "Map Satellite" toggle.

FIGURE 4.22 – la page ajouter client .

4.3.11 la page supprimer un client :

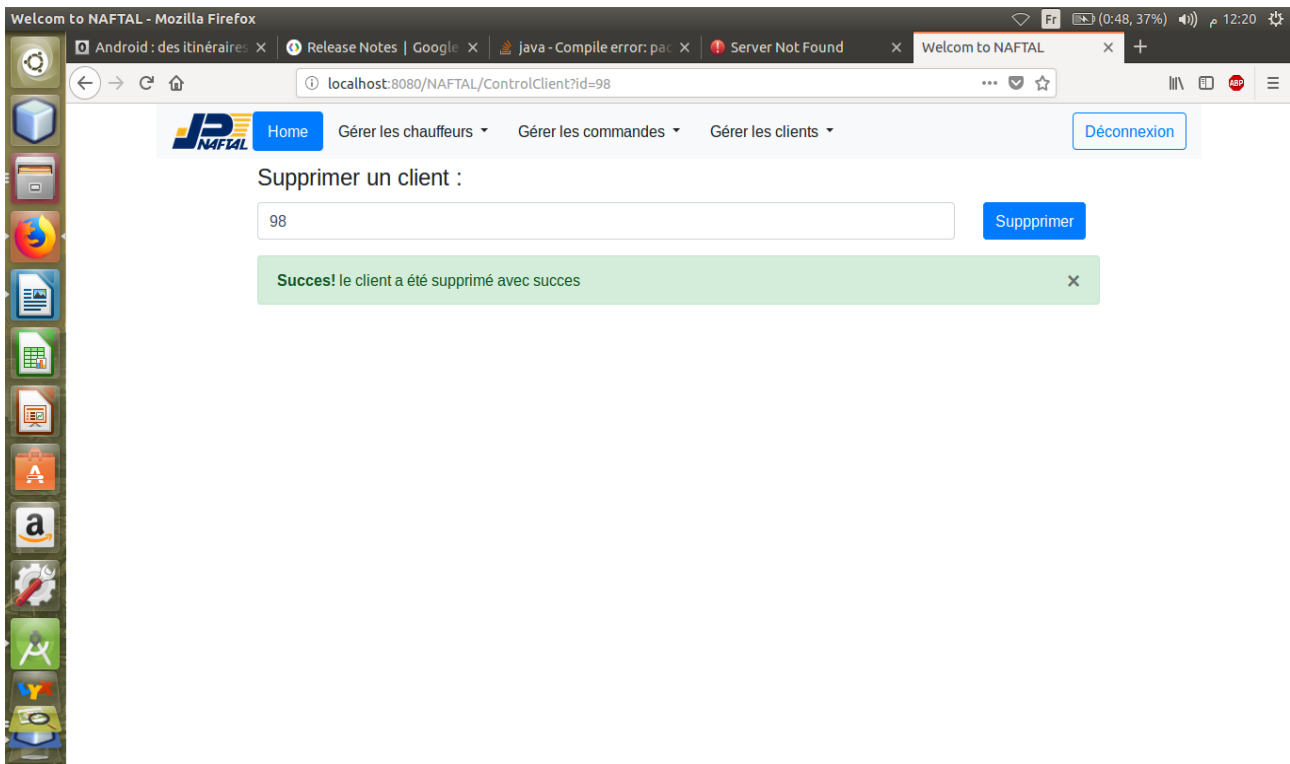


FIGURE 4.23 – la page supprimer un client .

4.3.12 l'interface d'authentification chauffeur :

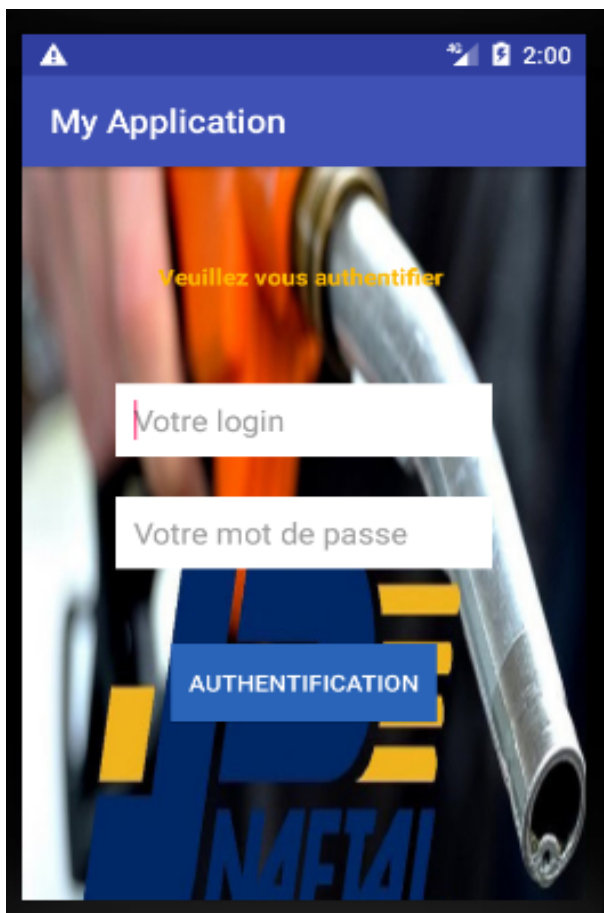


FIGURE 4.24 – l'interface d'authentification chauffeur .

4.3.13 l'interface feuille de route :

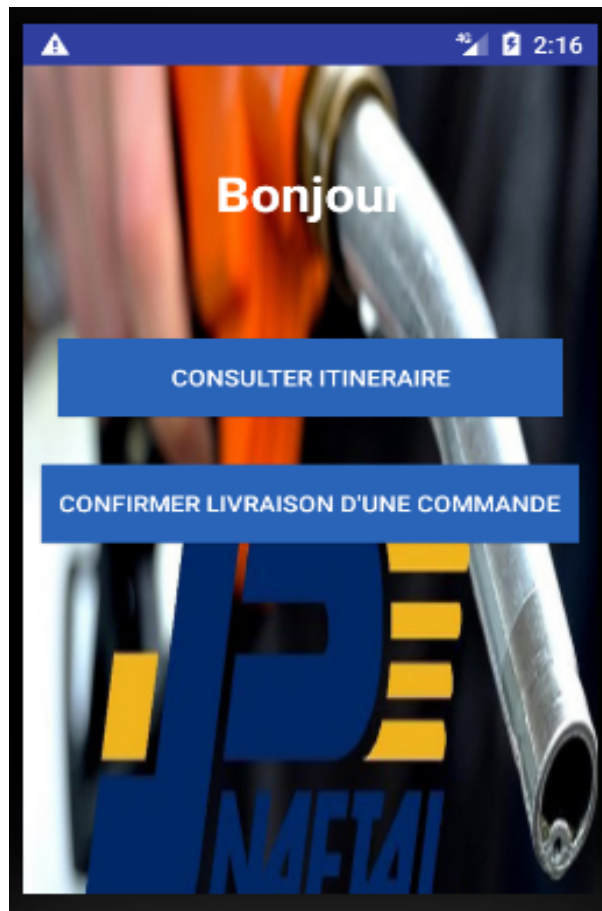


FIGURE 4.25 – l'interface feuille de route .

4.3.14 affichage de l'itinéraire :

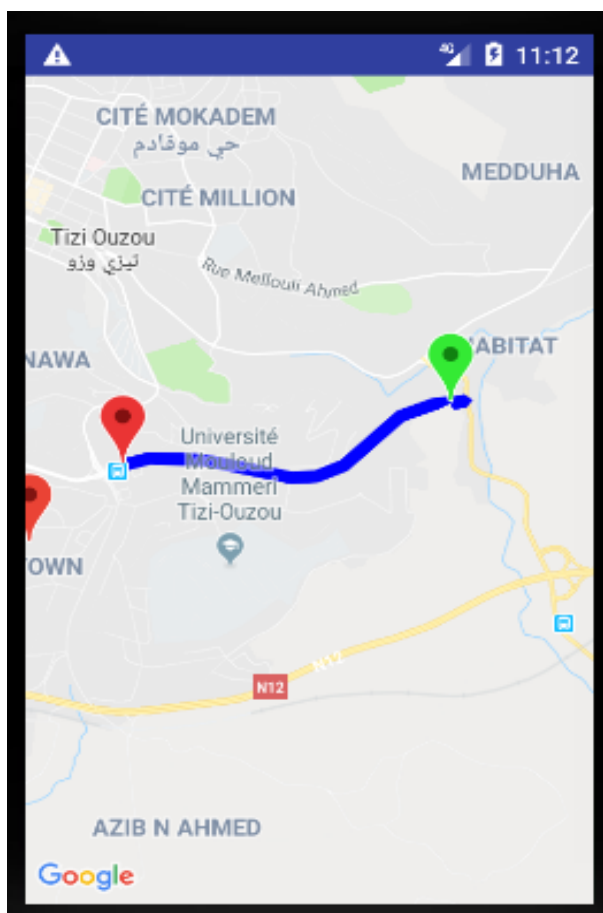


FIGURE 4.26 – affichage de l'itinéraire.

4.3.15 affichage de la liste de commande a confirmer :

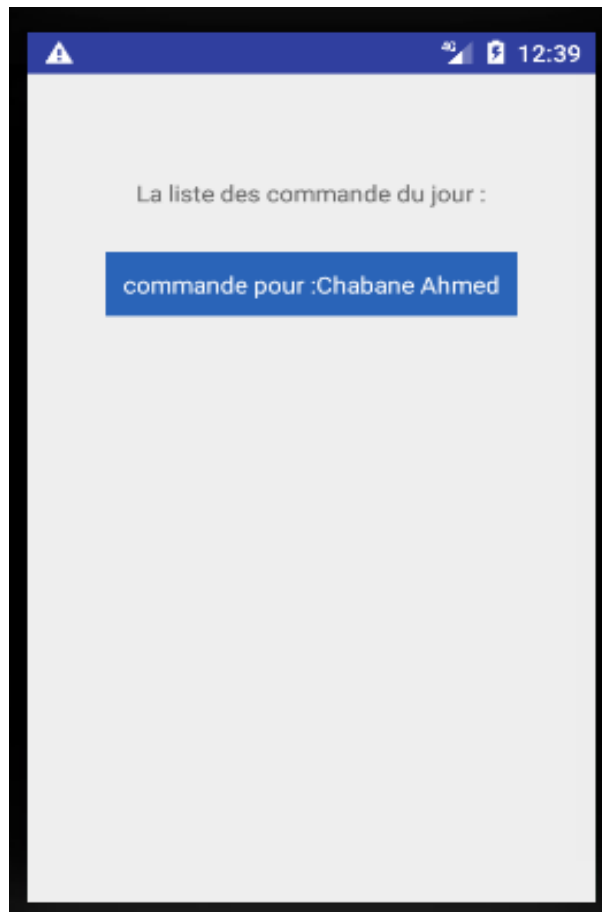


FIGURE 4.27 – affichage de la liste de commande a confirmer .

4.3.16 affichage de la liste de commande a confirmer :



FIGURE 4.28 – affichage de la liste de commande a confirmer .

4.3.17 Conclusion :

Dans ce chapitre nous avons présenté les différentes technologie qui nous ont permet d'implémenter notre système ainsi que sont architecture détaillé , nous avons aussi présenter les table qui constitue notre base de donnée et en fin quelque capture d'écran correspondante a toute les vus de notre système .

4.4 Conclusion général :

Dans ce projet , nous avons réalisé un système d'optimisation des chemins de transport pour le service de commercialisation des lubrifiants et pneumatique du district NAFTAL de tizi ouzou leurs permettant d'améliorer leur service de livraison .

Toutefois le problème d'optimisation auquel on a été confronté dans ce projet peut être amélioré avec des heuristiques qui permettent de répondre aux exigences de performance à savoir trouver un circuit hamiltonien optimal avec un temps de réponse raisonnable .

Enfin notre solution apportée à la réalisation de ce système n'est qu'une tentative parmi d'autres à la résolution de l'un des problèmes les plus célèbres dans le domaine mathématique auquel jusqu'à ce jour ne connais pas d'algorithme exact.

Et pour conclure nous souhaitons que ce travail satisfera les attentes de l'organisme d'accueil auquel il est destiné .

4.5 **Annex :**

4.6 **Concepts de base sur Android :**

4.6.1 **Introduction :**

Android est un système d'exploitation basé sur le noyau Linux. Le projet pour le développement du système Android est appelé Android Open Source Project (AOSP) et est géré principalement par Google.

Le système Android supporte les tâches de fond, fournit une riche bibliothèque pour l'interface utilisateur, supporte la 2D et la 3D en utilisant le standard OpenGL-ES (OpenGL pour faire court) et fournit un accès au système de fichiers ainsi qu'à une base de données SQLite embarquée.

4.6.2 **Historique des versions :**

Nom	Version	Date
Android	1.0	09/2008
Petitfour	1.1	02/2009
Cupcake	1.5	04/2009
Donut	1.6	09/2009
Gingerbread	2.3	12/2010
Honeycomb	3.0	02/2011
Ice Cream Sandwich	4.0.1	10/2011

Nom	Version	Date
jelly Bean	4.1	07/2012
KitKat	4.4	10/2013
Lollipop	5.0	10/2014
Marshmallow	6.0	09/2009
Gingerbread	2.3	05/2015
Nougat	7.0	09/2016
Oreo	8.0	08/2017
Popcorn	9.0	03/2018

4.6.3 Android :

L'écosystème d'Android s'appuie sur deux piliers :

- Le langage Java.
- Le SDK qui permet d'avoir un environnement de développement facilitant la tâche du développeur.

Le kit de développement donne accès à des exemples, de la documentation mais surtout à l'API de programmation du système et à un émulateur pour tester ses applications. Stratégiquement, Google utilise la licence Apache pour Android ce qui permet la redistribution du code sous forme libre ou non et d'en faire un usage commercial.

Le SDK était :

- Anciennement manipulé par un plugin d'Eclipse (obsolète).
- Maintenant intégré à Android Studio (IntelliJ).

4.6.4 Plateform android :

Android est un système de la famille des Linux, pour une fois sans les outils GNU. L'OS s'appuie sur :

- Un noyau Linux (et ses drivers).

- Une couche d'abstraction pour l'accès aux capteurs (HAL).
- Une machine virtuelle : Dalvik Virtual Machine (avant Lollipop).
- Des applications (navigateur, gestion des contacts, application de téléphonie...).
- Des bibliothèques (SSL, SQLite, OpenGL ES, etc...).
- Des API d'accès aux services Google .

Anatomie d'un déploiement :

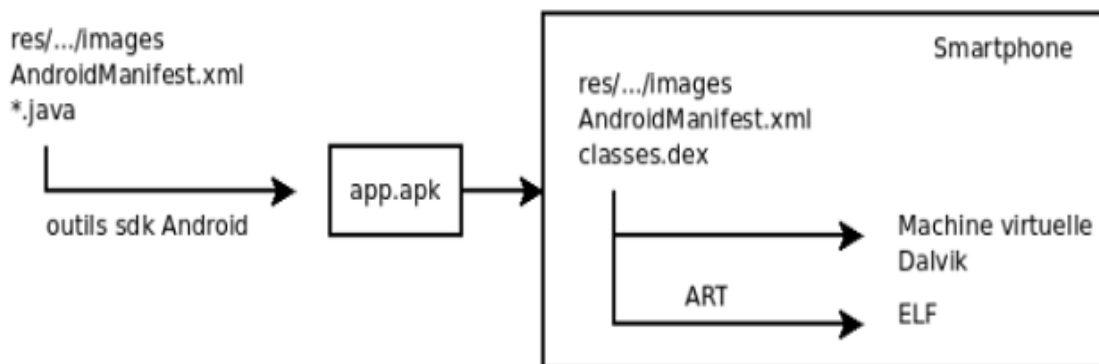


FIGURE 4.29 – Anatomie d'un déploiement.

4.6.5 Dalvik et ART :

Dalvik est le nom de la machine virtuelle open-source utilisée sur les systèmes Android. Cette machine virtuelle exécute des fichiers `.dex`, plus ramassés que les `.class` classiques. Ce format évite par exemple la duplication des String constantes. La machine virtuelle utilise elle-même moins d'espace mémoire et l'adressage des constantes se fait par un pointeur de 32 bits.

Dalvik n'est pas compatible avec une JVM du type Java SE ou même Java ME. La librairie d'accès est donc redéfinie entièrement par Google.

A partir de Lollipop, Android dispose d'ART qui compile l'application au moment du déploiement (Ahead-of-time compilation).

4.6.6 Le plugin de développement d'Eclipse ADT :

[Obsolète] Un projet basé sur le plugin ADT était décomposé de la manière suivante :

- **src/** : les sources Java du projet.
- **libs/** : bibliothèques tierces.
- **res/** :
 - **res/drawable** : ressources images.
 - **res/layout** : description des IHMs en XML.
 - **res/values** : chaînes de caractères et dimensions.
- **gen/** : les ressources auto générées par ADT.
- **assets/** : ressources brutes (raw bytes).
- **bin/** :
 - **bin/classes** : les classes compilées en class.
 - **bin/classes.dex** : exécutable pour la JVM Dalvik.
 - **bin/myapp.apk** : application empaquetée avec ses ressource et prête pour le déploiement .

4.6.7 L'environnement Android Studio :

Depuis mi-2015, il faut utiliser Android Studio (IDE IntelliJ ayant subi l'intégration de fonctionnalités de développement Android).

Avantages :

- Meilleure intégration du SDK dans Android Studio.
- Puissance de l'IDE IntelliJ .
- Nouvelle architecture des répertoires.
- Meilleure gestion des dépendances avec gradle.

Inconvénients :

- Lourdeur de l'IDE IntelliJ.
- Moins d'outils standalone (gestion des émulateurs, du SDK) .

- Obligation de s'habituer à un autre IDE.

4.6.8 L'architecture d'un projet Android Studio :

- **app** : le code de votre application.
 - **build** : le code compilé.
 - **lib** : les librairies natives.
 - **src** : les sources de votre applications.
 - **main/java** : vos classes.
 - **main/res** : vos ressources (XML, images, ...).
 - **test** : les tests unitaires.
- Des fichiers de configuration :
 - **build.gradle (2 instances)** : règles de dépendance et de compilation.
 - **settings.gradle** : liste de toutes les applications à compiler (si plusieurs).

4.6.9 Les vues dans Android Studio :

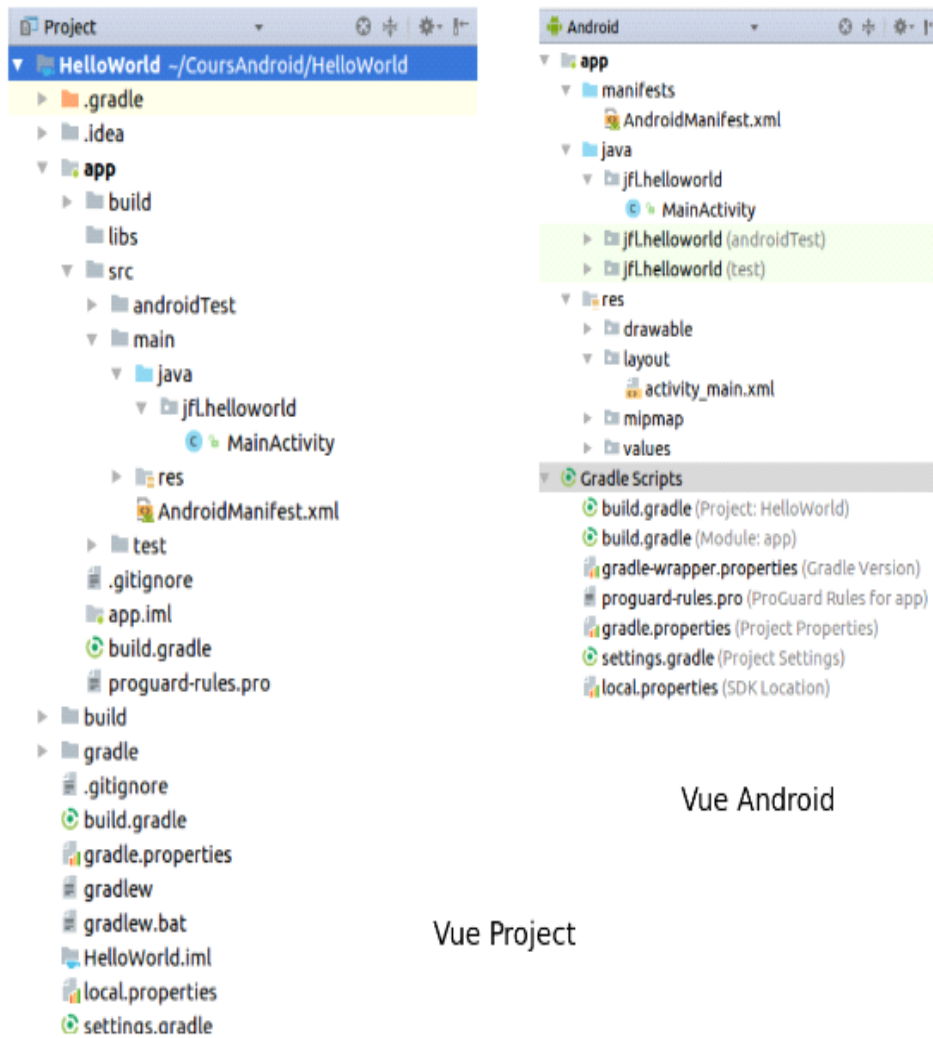


FIGURE 4.30 – Les vues dans Android Studio.

4.6.10 Les éléments d'une application :

Une application Android peut être composée des éléments suivants :

- **des activités** (`android.app.Activity`) : il s'agit d'une partie de l'application présentant une vue à l'utilisateur.
- **des services** (`android.app.Service`) : il s'agit d'une activité tâche de fond sans vue associée.
- **des fournisseurs de contenus** (`android.content.ContentProvider`) : permet le partage d'informations au sein ou entre applications.
- **des widgets** (`android.appwidget.`) : une vue accrochée au Bureau d'Android.
- **des Intents** (`android.content.Intent`) : permet d'envoyer un message pour un composant externe sans le nommer explicitement .
- **des récepteurs d'Intents** (`android.content.BroadcastReceiver`) : permet de déclarer être capable de répondre à des Intents.
- **des notifications** (`android.app.Notifications`) : permet de notifier l'utilisateur de la survenue d'événements.

4.6.11 Cycle de vie d'une activité :

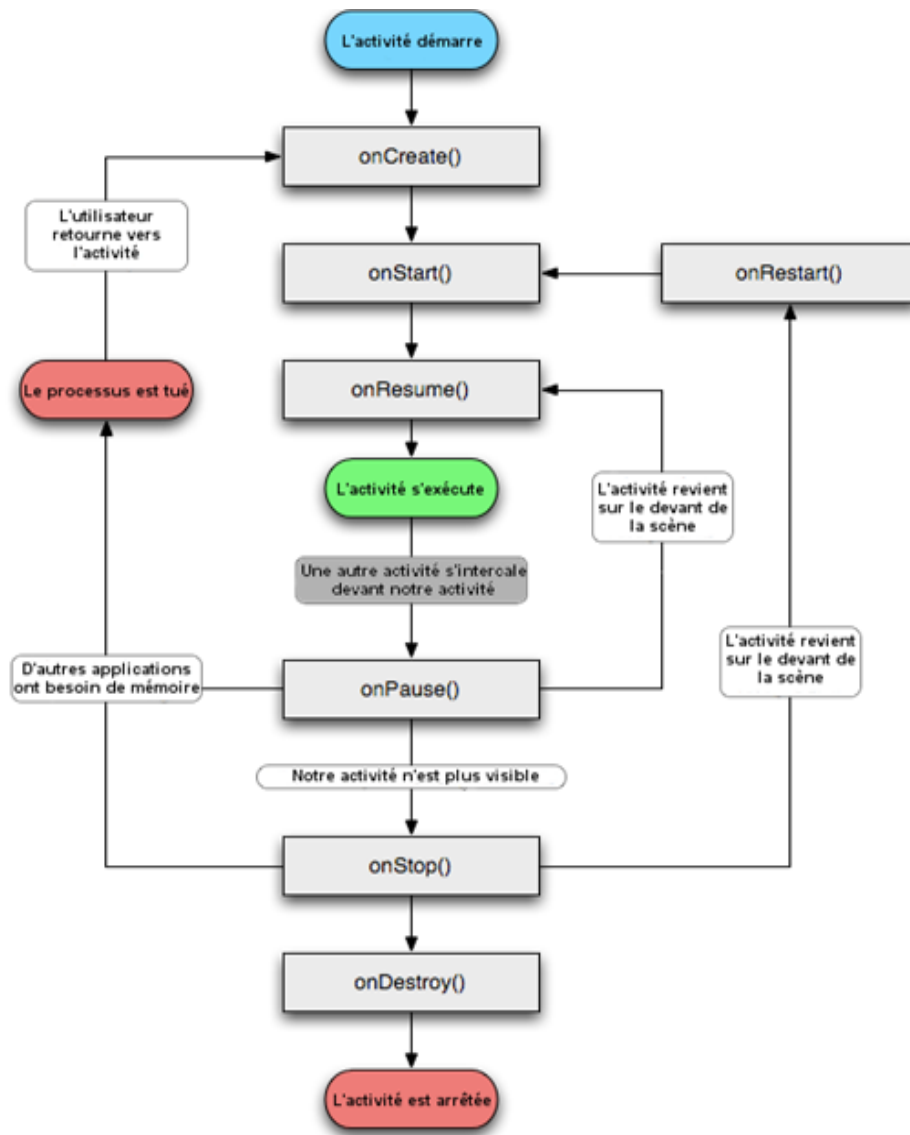


FIGURE 4.31 – Cycle de vie d'une activité.

- **onCreate()** / **onDestroy()** : permet de gérer les opérations à faire avant l'affichage de l'activité, et lorsqu'on détruit complètement l'activité de la mémoire. On met en général peu de code dans **onCreate()** afin d'afficher l'activité le plus rapidement possible.
- **onStart()** / **onStop()** : ces méthodes sont appelées quand l'activité devient visible/invisible pour l'utilisateur.
- **onPause()** / **onResume()** : une activité peut rester visible mais être mise en pause par le fait qu'une autre activité est en train de démarrer, par exemple B. **onPause()** ne doit pas être trop long, car B ne sera pas créé tant que **onPause()** n'a pas fini son exécution.

- **OnRestart()** : cette méthode supplémentaire est appelée quand on relance une activité qui est passée par **onStrop()**. Puis **onStrop()** est aussi appelée. Cela permet de différencier le premier lancement d'un relancement. [3]

4.7 Le geoweb :

4.7.1 Introduction :

Depuis 2005 et la fourniture du service de cartographie Web proposé par Google, Google Maps , on peut parler de l'émergence d'un GéoWeb, une organisation de l'information sur Internet par l'espace et le positionnement géographique direct ou indirect. Après Google Maps, l'offre d'outils cartographiques en ligne a explosé, avec de multiples outils et services et l'apparition d'usages différenciés. En fait Google Maps n'est pas un logiciel, ni même un simple service mais un service de services . Des développeurs informatiques peuvent assez facilement construire de nouveaux services de cartographie ou de géolocalisation qui s'appuient sur les fonctions et les données fournies par Google Maps. Les concurrents de Google (Microsoft, Yahoo, ...) ont réagi et proposé eux aussi des services cartographiques sur lesquels on peut construire de nouvelles briques. En réaction à ces initiatives des grands groupes qui dominent l'information sur le Web, d'autres groupes se sont lancés dans des opérations du même type. L'IGN (Institut géographique national) a aussi proposé un service d'accès en ligne à ses données numériques, le géoportail . Enfin une initiative a été lancée pour construire une base de données libres , ni publique ni privée, par contribution des Internautees , Openstreet Map.

Les services cartographiques de base qui forment l'infrastructure du GéoWeb :

- Google Maps.
- Bing Maps.
- Géoportail.
- Openstreetmap.

Leurs extensions 3D sous forme de globes virtuels :

- Google Earth.
- Bing Earth.
- Geopotail 3D.

4.7.2 L'infrastructure du GéoWeb :

Il existe une infrastructure matérielle du GéoWeb : des satellites pour capter des données, relayer des messages ou envoyer un signal de localisation, des serveurs informatiques et des réseaux pour stocker et diffuser l'information et des dispositifs variés pour consulter et interagir avec elle : ordinateurs, téléphones et appareils photographiques équipés de GPS ... Il est assez difficile de dresser un état de cette infrastructure matérielle et surtout de la distinguer de celle qui fait fonctionner Internet en général. En revanche, l'infrastructure logicielle du GéoWeb est un peu plus spécifique et elle s'appuie sur des services cartographiques dont certains sont très connus :

- **Google Maps** : premier service fournissant à l'échelle de la planète des données cartographiques systématiques de type carte routière et images aériennes. Il existait déjà des services cartographiques (Mapquest, Mappy, ViaMichelin ...) qui offraient et continuent à offrir des fonctions de navigation et de calcul d'itinéraires du type de ceux proposés par Google Maps. Mais d'une part, ils étaient réservés à une zone géographique, la France, les Etats-Unis ou l'Europe, tandis que Google Maps s'est d'emblée positionné au niveau mondial. D'autre part, ils étaient limités à ces fonctions de navigation alors que Google Maps est conçu non comme un service mais comme un service de services. Google fournit des données sous formes de couches d'information géographiques (images aériennes, plans routiers, ...), des fonctions logicielles de visualisation et d'interrogation et ce qu'on appelle une API qui permet à des développeurs informatiques de construire facilement de nouveaux services de cartographie ou de géolocalisation à partir des fonctions et les données fournies par Google Maps.

Les concurrents ont proposé rapidement des services cartographiques du même type que ceux de Google et sur lesquels on peut aussi construire de nouvelles briques. Microsoft propose ainsi Bing Maps, dont les fonctionnalités sont proches de Google Maps. Yahoo a aussi son Yahoo Maps, moins utilisé sauf par le site de partage de photographies Flickr dont je parle plus loin.

Les agences cartographiques nationales ou régionales ont aussi réagi avec l'idée de fournir un service cartographique public. Une des plus belles réalisations est le Géoportail de l'Institut Géographique National qui est un service d'accès en ligne aux données numériques françaises. Les données sont beaucoup plus variées et riches que celles de Google mais limitées au territoire français.

Enfin une initiative a été lancée au niveau mondial pour construire une base de données géogra-

phiques, et donc une infrastructure du GéoWeb, libre de droit entièrement construite par contribution des Internautes : Openstreetmap .

4.7.3 Le GéoWeb volontaire et contributif :

Sur ces services de base de localisation et de visualisation cartographiques, des services de « deuxième niveau » ont été développés qui permettent à des internautes sans compétence informatique particulière de contribuer à l'enrichissement en contenu du GéoWeb. Ces applications sont très nombreuses. Voici quelques exemples : [4]

- **My Maps** : c'est peut-être la première application, celle que Google a développé pour permettre à un internaute de saisir ses propres données sous forme de carte sur Google Maps.
- **Wikimapia** : se veut un peu le pendant cartographique de l'encyclopédie Wikipedia. Chacun peut délimiter des entités sur la surface terrestre et les caractériser ou les annoter, en renvoyant par exemple à un article de Wikipedia. [4]
- **Dismoiou** : on peut marquer sur ce type de site les lieux que l'on aime ou les restaurants que l'on conseille et les partager avec des internautes. [4]
- **Flickr ou Panoramio** : des sites où l'on peut partager des photographies en les localisant. Elles s'ajoutent sur les services cartographiques de Yahoo (Flickr) ou Google (Panoramio). On peut donc visualiser les lieux en naviguant entre les photographies depuis la carte. [4]
- **StreetAdvisor** : un service Web américain : l'objectif est de noter une rue et un voisinage selon différents critères. [4]
- **FixMyStreet** : un outil anglais de gestion locale : l'objectif est de décrire les problèmes de voirie ou de voisinage, de manière à les régler et à améliorer la vie quotidienne d'une communauté locale. [4]
- **Geocommons** : un outil pour fabriquer ses propres cartes à partir de ses données et les partager avec les utilisateurs.
- **Foursquare** : un outil récent au développement rapide qui croise une application de réseau social et de microblogging dans une activité de jeu et de « Cityguide ». Il s'agit de signaler au site grâce à un téléphone portable géolocalisé les lieux où l'on se trouve, ce qui permet d'y

donner rendez-vous à ses connaissances si elles adhèrent à Foursquare. On peut aussi décrire et documenter les lieux que l'on fréquente. On accumule alors des badges relatifs à ces lieux, qui permettent éventuellement d'obtenir des réductions ou des cadeaux. Enfin, on devient le « Maire » d'un lieu quand on est celui qui s'y est le plus souvent localisé. [4]

Il existe des centaines de services et il s'en invente tous les jours. Ces services peuvent être spécifiques ou intégrés sous forme d'interface cartographique à des sites Web généraux : publier des photographies anciennes des lieux, localiser les écoles équipées de caméras de vidéosurveillance, publier son journal de voyage ... Avec ces outils, des individus, amateurs ou professionnels, des associations, des groupes, des firmes ou des collectivités associent une localisation géographique à des objets variés : zones de projet, opérations d'intervention, lieux de vacances, lieux de tournage de films, endroits décrits par des livres... [4]

4.7.4 Conclusion :

On voit qu'à travers le GéoWeb, Internet stocke un volume très important et croissant d'information décrivant les lieux, les territoires et les relations que les individus entretiennent entre eux et avec eux. Bien sûr, cette information est biaisée. Tout le monde n'est pas connecté au Net, tout le monde n'y est pas actif de la même manière, sur le Net et nous ne sommes pas tous prêts à accepter de rendre publique notre localisation. Il n'empêche que devient disponible une information localisée qui présente pour un nombre croissant d'individus les lieux qu'ils fréquentent et apprécient, les photos qu'ils prennent, les activités qu'ils y font, les gens qu'ils y rencontrent ... Ce nuage d'information n'est pas unifié. Les données localisées sont partagées entre différents sites et Big Mother, la maman géographique et à l'échelle terrestre de Big Brother n'est pas (encore) là. Mais cette masse d'information c'est une manne pour les sociétés (suivi des salariés et des consommateurs et marketing, analyse des tendances, stratégies de localisation, surveillance des concurrents ...), pour l'Etat et les collectivités qui veulent surveiller, contrôler et aménager l'espace. Elle l'est aussi pour les scientifiques qui veulent comprendre les fréquentations des lieux et leurs images, les interactions qu'ils permettent et les dynamiques des territoires qu'ils reflètent ou contribuent à générer. Cela implique la maîtrise de nouveaux outils techniques, ceux du GéoWeb et une culture géographique solide. [4]

4.8 La géolocalisation :

4.8.1 Introduction :

Le développement de l'utilisation dans les transports terrestres de l'information de position et de temps délivrée par les systèmes satellitaires est récent. Néanmoins, l'usage de ces informations devient prédominant dans ce domaine par opposition aux domaines maritimes et aériens, qui se sont positionnés bien avant comme utilisateurs potentiels.

De nombreuses fonctionnalités se développent dans le cadre des systèmes existants, mais elles sont dépendantes avant tout du marché de l'offre et de la demande et des perspectives de rentabilité. Aujourd'hui, seul le système américain GPS est réellement opérationnel et constitue l'élément sur lequel s'est développé tout ce secteur.

4.8.2 Les systèmes existants :

Deux systèmes autonomes de positionnement par satellite existent actuellement déclarés opérationnels depuis fin 1993 : l'un américain, le GPS (Global Positioning System) et l'autre russe, Glonass (Global Orbiting Navigation Satellite System).

Le GPS :Le système GPS a été développé par le département de la défense des Etats-Unis et reste sa propriété. Initié en 1973, ce projet est complètement opérationnel depuis 1993. Le système se compose de trois segments majeurs : le segment spatial, le segment de contrôle et le segment utilisateur.

Ce système offrait jusqu'à récemment deux niveaux de services : Service standard et service de détermination de la position avec précision. Ce système permet ainsi, de façon mondiale et permanente, à un nombre illimité d'utilisateurs équipés du récepteur adéquat de se positionner et de synchroniser avec une très grande précision par rapport aux référentiels géographiques et temporels mondiaux. La gratuité de la fourniture de ces services a assuré une implantation et une utilisation rapide de GPS, et constitué une véritable industrie.

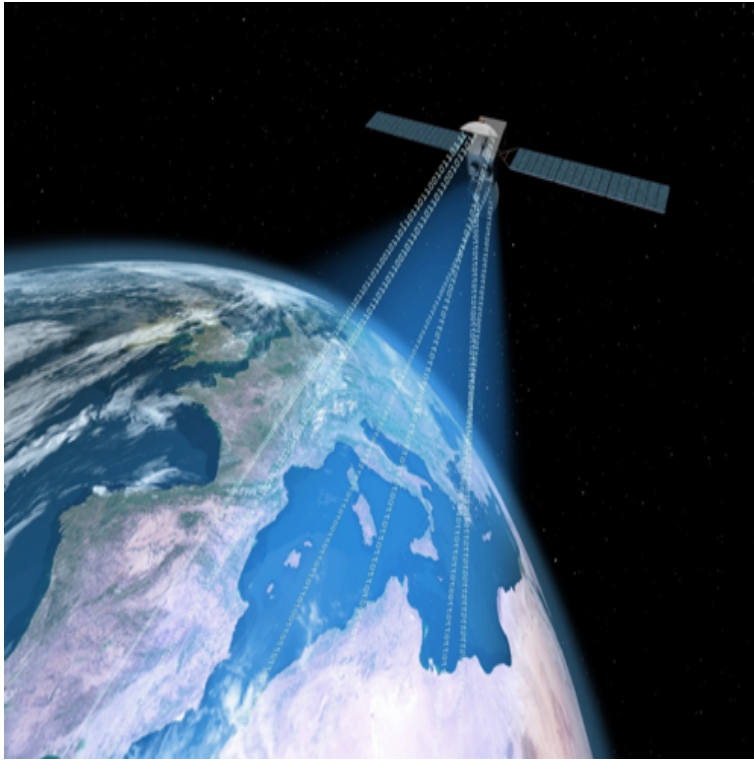


FIGURE 4.32 – satellite GPS.

4.8.3 Intérêt de la localisation par satellite :

On ne peut expliquer la croissance exponentielle de l'utilisation du GPS par sa seule gratuité. Mais c'est plutôt par les caractéristiques des services délivrés et leur compétitivité économique que l'on peut comprendre un tel développement, et la naissance de toute une industrie autour du GPS (segment spatial mais plus encore segment usager).

Mais c'est plus encore couplés à d'autres technologies que ces systèmes vont donner tout leur potentiel en permettant la fourniture d'un service à valeur ajoutée.

En effet, savoir où l'on se trouve n'est pas une fin en soi. Et c'est associées le plus souvent à des télécommunications et des traitements logiciels, ou reposant sur des SIG (systèmes d'informations géographiques) que ces données vont montrer toute leur pertinence et ouvrir des potentialités multiples.

4.8.4 Avantages et inconvénients de la géolocalisation GPS :

- **Avantages** : bonne précision.
- **Inconvénients** : le système ne fonctionne pas toujours en intérieur, il met du temps à déterminer la position lors du démarrage et il consomme beaucoup d'énergie.

4.8.5 Géolocalisation par GSM :

La géolocalisation par GSM détermine une localisation géographique en se basant sur les antennes GSM, c'est-à-dire les antennes relais qui servent habituellement à transférer les données aux téléphones mobiles. La précision de ce type de positionnement peut aller de 200 mètres à plusieurs kilomètres, selon la densité des antennes. En milieu urbain, la densité est plus élevée et permet donc un meilleur positionnement qu'en milieu rural.

4.8.6 Avantage et inconvénients de la géolocalisation par GSM :

- **Avantages** : consomme peu et est rapide à démarrer.
- **Inconvénients** : la couverture géographique dépend du réseau des antennes relais. Lorsque le nombre d'antennes est faible la géolocalisation est peu précise, et lorsqu'il n'y en a pas la géolocalisation est impossible.

4.8.7 Géolocalisation par Wifi :

La géolocalisation Wifi exploite la position connue de certains réseaux wifi pour déterminer la position d'un appareil. Un smartphone équipé d'une puce Wifi pourra alors se baser sur les connexions wifi qu'il détecte à proximité pour estimer son emplacement géographique. La précision dépend de la puissance des points d'accès Wifi, c'est-à-dire quelques dizaines de mètres.

A noter : Google utilise des voitures qui parcourent les villes pour prendre des photos utilisables dans Google Street View et pour obtenir une liste de toutes les connexions wifi qu'ils détectent. Ces données sont associées à la géolocalisation de la voiture et servent justement à renforcer la puissance de la géolocalisation Wifi.

4.8.8 Avantages et inconvénients de la géolocalisation par Wifi :

- **Avantages** : consomme peu et a une bonne précision.
- **Inconvénient** : presque inutilisable en zone rurale.

4.8.9 comparaison entre ces trois technologies :

	GPS	GSM	Wifi
Précision	👍	👎	👍
Consommation	👎	👍	👍
Démarrage	👎	👍	👍
Fonctionnement à l'intérieur	👎	👍	👍
Fonctionnement en zone urbaine	👍	👍	👍
Fonctionnement en zone rurale	👍	👎	👎

FIGURE 4.33 – Tableau récapitulatif des systèmes localisation mobile.

Sachant que chaque méthode possède des avantages et inconvénients, les smartphones combinent aisément les technologies pour bénéficier de la meilleure précision à tout moment et pour un minimum de consommation de la batterie du téléphone. Ainsi il est possible d'avoir un positionnement GSM pour obtenir rapidement une position, pendant que la puce GPS est activée et qu'elle détermine une localisation plus précise.

Il est par ailleurs intéressant de savoir que les smartphones enregistrent la dernière position connue pour économiser de la batterie et gagner en réactivité.

4.8.10 Conclusion :

Les systèmes de localisation par satellites sont maintenant partie prenante des applications civiles et constituent un progrès sans précédent dans l'histoire du développement des moyens de distribution de cette information.

4.9 Les services Web :

4.9.1 Introduction :

La technologie des services Web est un moyen rapide de distribution de l'information entre clients, fournisseurs, partenaires commerciaux et leurs différentes plates-formes. Les services Web sont basés sur le modèle SOA.

D'autres technologies telles que RMI, DCOM et CORBA ont précédemment adopté ce style architectural mais ont généralement échoué en raison de la diversité des plates-formes utilisées dans les organisations et aussi parce que leur usage n'était pas adapté à Internet (problème de passage à travers des FireWalls, etc.) d'où la lenteur, voire l'absence de réponses sur ce réseau. Les applications réparties fondées sur ces technologies offrent des solutions caractérisées par un couplage fort entre les objets. Les solutions proposées par les services Web, permettent néanmoins un couplage moins fort. De plus, l'utilisation des technologies standards du Web telles HTTP et XML par les services Web facilite le développement d'applications réparties sur Internet, et permet d'avoir des applications très faiblement couplées. L'intégration est sans doute le facteur essentiel qui favorise l'utilisation des services Web.

4.9.2 Définitions des services Web :

Un service Web est un composant logiciel identifié par une URI, dont les interfaces publiques sont définies et appelées en XML. Sa définition peut être découverte par d'autres systèmes logiciels. Les services Web peuvent interagir entre eux d'une manière prescrite par leurs définitions, en utilisant des messages XML portés par les protocoles Internet.

En d'autres termes, un service Web est tout simplement un programme accessible au moyen d'Internet, qui utilise un système de messagerie standard XML, et n'est lié à aucun système d'exploitation ou langage de programmation.

4.9.3 Les principaux avantages d'un service Web :

- son interface décrite d'une manière interprétable par les machines, qui permet aux applications clientes d'accéder aux services de manière automatique.
- son utilisation de langages et protocoles indépendants des plates-formes d'implantation, qui renforcent l'interopérabilité entre services.
- son utilisation des normes actuelles du Web, qui permettent la réalisation des interactions faiblement couplées et favorisent aussi l'interopérabilité.

4.9.4 L'intérêt d'un Service Web :

Les services Web fournissent un lien entre applications. Ainsi, des applications utilisant des technologies différentes peuvent envoyer et recevoir des données au travers de protocoles compréhensibles par tout le monde.

Les services Web sont normalisés car ils utilisent les standards XML et HTTP pour transférer des données et ils sont compatibles avec de nombreux autres environnements de développement. Ils sont donc indépendants des plates-formes. C'est dans ce contexte qu'un intérêt très particulier a été attribué à la conception des services Web puisqu'ils permettent aux entreprises d'offrir des applications accessibles à distance par d'autres entreprises. Cela s'explique par le fait que les services Web n'imposent pas de modèles de programmation spécifiques. En d'autres termes, les services Web ne sont pas concernés par la façon dont les messages sont produits ou consommés par des programmes. Cela permet aux vendeurs d'outils de développement d'offrir différentes méthodes et interfaces de programmation au-dessus de n'importe quel langage de programmation, sans être contraints par des standards comme c'est le cas de la plate-forme CORBA qui définit des ponts spécifiques entre le langage de définition IDL et différents langages de programmation. Ainsi, les fournisseurs d'outils de développement peuvent facilement différencier leurs produits avec ceux de leurs concurrents en offrant différents niveaux de sophistication.

Les services Web représentent donc la façon la plus efficace de partager des méthodes et des fonctionnalités. De plus, ils réduisent le temps de réalisation en permettant de tirer directement parti de services existants.

4.9.5 Les caractéristiques d'un service Web :

La technologie des services Web repose essentiellement sur une représentation standard des données (interfaces, messageries) au moyen du langage XML. Cette technologie est devenue la base de l'informatique distribuée sur Internet et offre beaucoup d'opportunités au développeur Web. Un service Web possède les caractéristiques suivantes :

- il est accessible via le réseau.
- il dispose d'une interface publique (ensemble d'opérations) décrite en XML.
- ses descriptions (fonctionnalités, comment l'invoquer et où le trouver ?) sont stockées dans un annuaire.
- il communique en utilisant des messages XML, ces messages sont transportés par des protocoles Internet (généralement HTTP, mais rien n'empêche d'utiliser d'autres protocoles de transfert tels : SMTP, FTP, BEEP...).
- l'intégration d'application en implémentant des services Web produit des systèmes faiblement couplés, le demandeur du service ne connaît pas forcément le fournisseur.

4.9.6 Architecture d'un service web :

Les services Web reprennent la plupart des idées et des principes du Web (HTTP, XML), et les appliquent à des interactions entre machines. Comme pour le World Wide Web, les services Web communiquent via un ensemble de technologies fondamentales qui partagent une architecture commune. Ils ont été conçus pour être réalisés sur de nombreux systèmes développés et déployés de façon indépendante. Les technologies utilisées par les services Web sont HTTP, WSDL, REST, XML-RPC, SOAP et UDDI.

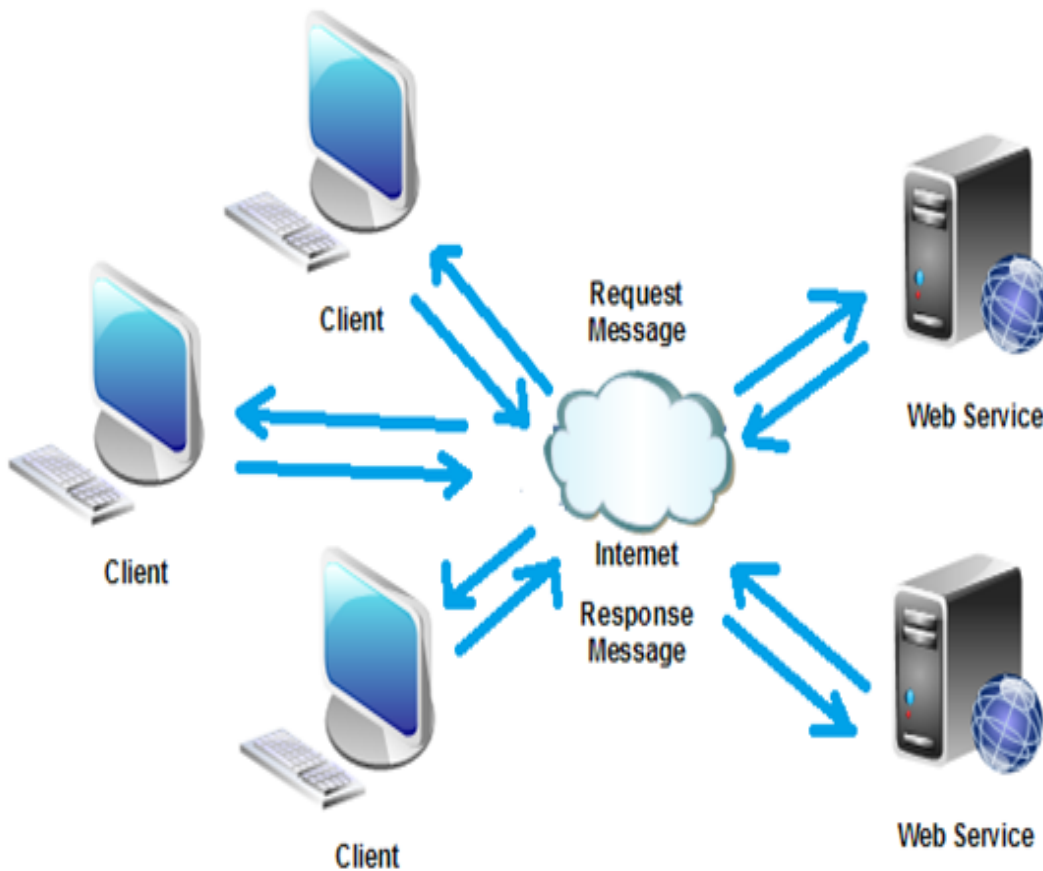


FIGURE 4.34 – architecture d'un service web.

4.9.7 Définition de service restful :

Une API compatible REST, ou « RESTful », est une interface de programmation d'application qui fait appel à des requêtes HTTP pour obtenir (GET), placer (PUT), publier (POST) et supprimer (DELETE) des données.

Aujourd'hui, l'utilisation croissante du Cloud entraîne l'apparition de différentes API visant à présenter les services Web. REST constitue un choix logique pour le développement d'API permettant à l'utilisateur final de se connecter à des services Cloud et d'interagir avec eux. De nombreux sites font appel à des API RESTful, notamment Google, Amazon, Twitter et LinkedIn.

4.9.8 Les avantages et les inconvénients de Rest :

- les avantages :

- Simple à mettre en place.
- Client moins dépendant du serveur.
- Pas besoin de couche supplémentaire.

- les inconvénients

- Sécurité moins évoluée.
- Web service lié au mode de transport http.

4.9.9 Conclusion :

Nous avons vu que les web services permettent l'échange d'informations à travers le web. C'est une notion à maîtriser, car tous les grands groupes ou presque ont leur web service, et que sans une connaissance théorique de ce qu'est un web service, leur utilisation est compliquée.

Dans le cas où vous devez créer un service web, il va falloir dans ce cas choisir entre REST et SOAP, celui qui correspond au mieux au besoin exprimé, car on ne peut pas dire que l'un est mieux que l'autre. Tout dépend de l'utilisation qu'on en a.

Bibliographie

- [1] [<https://netbeans.org/index-fr.html>.]
- [2] [<https://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm>.]
- [3] [<http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/enseignement/android/cours-android.pdf>.]
- [4] [<https://mondegeonumerique.wordpress.com/2010/06/24/le-geoweb-pour-les-nuls/>.]