

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

**Centre Universitaire
Abd Elhafid Boussouf Mila**

Institut des sciences et de la technologie

Département de Mathématiques et Informatique

Mémoire préparé En vue de l'obtention du diplôme de licence

En :-Filière : Informatique

**Conception et Réalisation d'une
Application Pour la Gestion d'une
Agence de Transport à la Demande**

Préparé par :

**-Boudjenana Khalil Errahmane.
-Fedali Imad.
-Hocine Youcef.**

- Encadré par :

Mme.Zekiouk Mounira

Année universitaire : 2014/2015



*AU TERME DE CE TRAVAIL, NOUS TENONS VIVEMENT À
REMERCIER TOUTES LES
PERSONNES QUI ONT CONTRIBUÉ DE PRÈS OU DE LOIN À
LA RÉALISATION DE CE
PRÉSENT MÉMOIRE.*

*TOUS D'ABORD NOUS TENONS À REMERCIER MME
ZEKIOUK MOUNIRA*

*NOTRE ENCADREUR POUR SA DISPONIBILITÉ, SON AIDE
PRÉCIEUSE ET SES CONSEILS*

QUI NOUS ONT ÉTÉ D'UNE UTILITÉ INDÉNIABLE

*ELLE NOUS AIDÉ EN NOUS ACCORDANT UNE GRANDE
CONFIANCE DANS LA DÉMARCHE*

*DE NOTRE TRAVAIL, SANS OUBLIER LEUR REGARDS
PERTINENTS ET LEUR CRITIQUES*

*CONSTRUCTIVES. CE TRAVAIL N'EXISTERAIT PAS LEURS
AIDES ET LEURS*

BIENVÉILLANCES.

*UN GRAND MERCI POUR TOUS NOS ENSEIGNANTS POUR
TOUTES LES CONNAISSANCES*

QU'ILS NOUS ONT INCULQUÉES.

KHALIL ERRAHMANE BOUDJENANA

AU DÉBUT ET AVANT TOUS, JE VEUX REMERCIER LE
DIEU QUI À PERMET LE
COURAGE À FAIRE ET FINIR CE MODESTE TRAVAIL.
C'EST AVEC UN GRAND PLAISIR ET UNE RÉELLE JOIE DE
FIERTÉ QUE JE DÉDIE
CE TRAVAIL.

À CELLE QUI M'A SOUTENU DURANT TOUT MES ANNÉES
D'ÉTUDE, QUI MÉRITÉ
MON AMOUR ÉTERNEL POUR SES CONSEILS PRÉCIEUSE,
SA TÂCHASSE, SA PATIENCE À
L'ESPRIT DE MA MÈRE *NACERRA* .
ET MON PÈRE *ABD EL HAMID*
À TOUS MES SOEURS , *IMAN,HADJER,BATOL,BARAA, ET*
MON .

À MES ONCLES ET TANTES, AINSI QUE MES COUSINS ET
COUSINS ;
À MA MEILLEUR AMIE POUR LA QUELLE JE GARDE DES
BONS SOUVENIRS ET
CELLE QUI M'A FAIT COMPRENDRE LE VRAI SENS DE
L'AMITIÉ, ET JE PARTAGE
MAINTENANT CE MODESTE TRAVAIL À : *YOUCEF,IMAD* ;
À MES CHÈRE AMIES: *DADI,AHMAD YACINE,ZAKI LBOBÉ*
AHMED RAHIM ;

À TOUS LA FAMILLE *BOUDJENANA* ET LA FAMILLE
OUKRID ;

À LA FIN, NOUS REMERCIONS TOUS CEUX QUI ONT AIDÉ
DE PRÈS OU DE LOIN À
RÉALISER NOTRE TRAVAIL.

KHALIL IMAD YOUCEF

IMAD FEDALI

AU DÉBUT ET AVANT TOUS, JE VEUX REMERCIER LE
DIEU QUI A PERMET LE
COURAGE À FAIRE ET FINIR CE MODESTE TRAVAIL.
C'EST AVEC UN GRAND PLAISIR ET UNE RÉELLE JOIE DE
FIERTÉ QUE JE DÉDIE
CE TRAVAIL.

À CELLE QUI M'A SOUTENU DURANT TOUT MES ANNÉES
D'ÉTUDE, QUI MÉRITÉ
MON AMOUR ÉTERNEL POUR SES CONSEILS PRÉCIEUSE,
SA TÂCHASSE, SA PATIENCE À
L'ESPRIT DE MA MÈRE *MALIKA*.
ET MON PÈRE *BACHIR*

À TOUS MES FRÈRE ET SOEUR,
RIMA, HICHAM, ADEL, AMANI, BOUTAINA, ET MON.
À MES ONCLES ET TANTES, AINSI QUE MES COUSINS ET
COUSINS ;

À MA MEILLEUR AMIE POUR LA QUELLE JE GARDE DES
BONS SOUVENIRS ET
CELLE QUI M'A FAIT COMPRENDRE LE VRAI SENS DE
L'AMITIÉ, ET JE PARTAGE

MAINTENANT CE MODESTE TRAVAIL À : *YOUCEF*
, KHALIL ;

À MES CHÈRE AMIES: *SEIF EDDINE, ALI*
CHIHA, BOUALAM, KHALIL, YOUCEF, LOTFI, MOH;

À TOUS LA FAMILLE *FEDALI* LA FAMILLE *FENNI*;

À LA FIN, NOUS REMERCIONS TOUS CEUX QUI ONT AIDÉ
DE PRÈS OU DE LOIN À
RÉALISER NOTRE TRAVAIL.

KHALIL IMAD YOUCEF



HOCINE YUCEF

AU DÉBUT ET AVANT TOUS, JE VEUX REMERCIER LE
DIEU QUI A PERMET LE
COURAGE À FAIRE ET FINIR CE MODESTE TRAVAIL.
C'EST AVEC UN GRAND PLAISIR ET UNE RÉELLE JOIE DE
FIERTÉ QUE JE DÉDIE
CE TRAVAIL.

À CELLE QUI M'A SOUTENU DURANT TOUT MES ANNÉES
D'ÉTUDE, QUI MÉRITÉ
MON AMOUR ÉTERNEL POUR SES CONSEILS PRÉCIEUSE,
SA TÂCHASSE, SA PATIENCE À
L'ESPRIT DE MA MÈRE *SAMIA*.

ET MON PÈRE *BELGHASEM*

À TOUS MES SOEUR, *AMINA, SAMIHA, ET MON*.

À MES ONCLES ET TANTES, AINSI QUE MES COUSINS ET
COUSINS ;

À MA MEILLEUR AMIE POUR LA QUELLE JE GARDE DES
BONS SOUVENIRS ET

CELLE QUI M'A FAIT COMPRENDRE LE VRAI SENS DE
L'AMITIÉ, ET JE PARTAGE

MAINTENANT CE MODESTE TRAVAIL À : *YUCEF*

,*KHALIL* ;

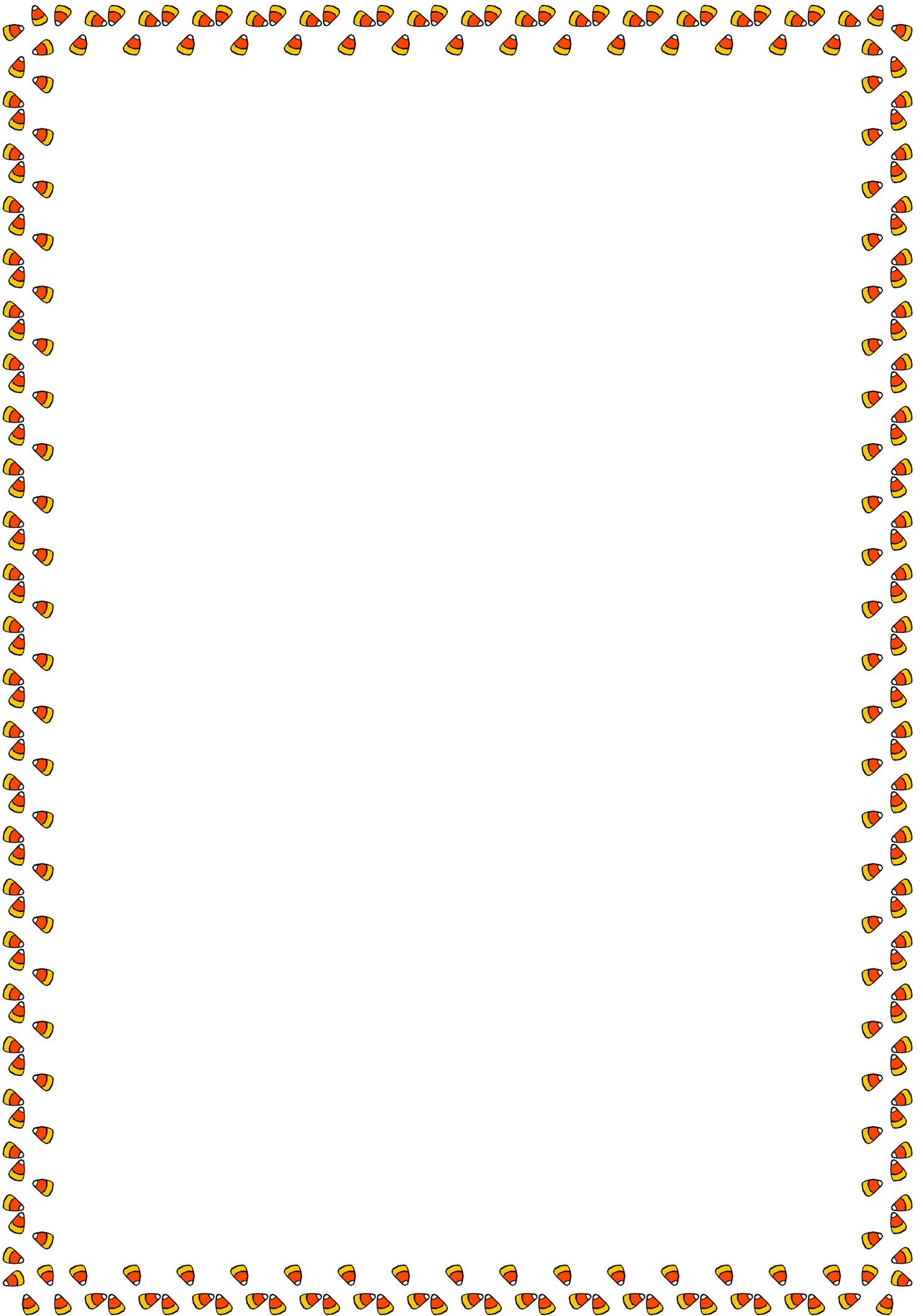
À MES CHÈRE AMIES: *SEIF EDDINE, ALI*

CHIHA, BOUALAM, KHALIL, ;

À TOUS LA FAMILLE *HOCINE* ;

À LA FIN, NOUS REMERCIONS TOUS CEUX QUI ONT AIDÉ
DE PRÈS OU DE LOIN À
RÉALISER NOTRE TRAVAIL.

KHALIL IMAD YUCEF



SOMMAIRE

Remerciements

Introduction générale

CHAPITRE I : Modélisation avec UML

1. Introduction.....	1
2. Modélisation avec UML.....	1
2.1. Définition d'un modèle	1
2.3 Langages de modélisation	2
2.4. Définition d'UML.....	2
2.5. Caractéristiques d'UML.....	2
3. Présentation des vues et diagrammes d'UML.....	3
3.1 Diagrammes structurels ou diagrammes statiques.....	3
3.2 Diagrammes comportementaux ou diagrammes dynamiques.....	4
3.3. Diagramme de cas d'utilisation.....	5
3.4. Diagramme de classe.....	7
3.5. Diagramme d'activité.....	10
3.6. Le diagramme de séquence/séquence système.....	11
3.7. Le diagramme de classe participante	12
4. Le processus.....	13
5. Conclusion.....	14

CHAPITRE II : Le transport à la demande

1. Introduction.....	15
2. Le transport à la demande(TAD).....	15
2.1. Définitions.....	15
2.2. Typologie des problèmes.....	16
2.2.1 Problème du voyageur de commerce(TSP).....	17
2.2.2. Problème de tournées de véhicules(VRP).....	17
2.2.3. Problème de tournées de véhicules multi-dépôt (MDVRP).....	17
2.2.4 Problème de tournées de véhicules avec ramassage et livraison (PDP).....	17
3. Les principaux Avantages du TAD.....	18
4. Les principaux manques des TAD actuels.....	19
5. Exemples des agences de transport à la demande.....	19
6. Conclusion.....	20

CHAPITRE III : modélisation

1. Introduction	21
2. Identification des acteurs.....	21
3. identification des cas d'utilisations.....	21
4. Diagramme de cas d'utilisation.....	22
5. Description textuelle des cas d'utilisations.....	23
5.1. Cas d'utilisation « ajouter client ».....	23
5.2. Cas d'utilisation « ajouter demande de client ».....	23
5.3. Cas d'utilisation «ajouter chauffeurs».....	24

5.4. Cas d'utilisation «ajouter véhicule».....	25
5.5. Cas d'utilisation « modifier client ».....	25
5.6. Cas d'utilisation « modifier véhicule».....	26
5.7. Cas d'utilisation « modifier chauffeur».....	27
5.8. Cas d'utilisation «modifier demande de client.....	27
5.9. Cas d'utilisation «modifier le trajet des tournée».....	27
5.10. Cas d'utilisation « supprimer client ».....	28
5.11. Cas d'utilisation «supprimer demande de client».....	28
5.12. Cas d'utilisation «supprimer plan».....	29
5.13. Cas d'utilisation «supprimer chauffeurs».....	30
5.14. Cas d'utilisation «supprimer véhicule ».....	30
5.15. Cas d'utilisation « chercher client».....	31
5.16. Cas d'utilisation « chercher demande».....	31
5.17. Cas d'utilisation « chercher chauffeur».....	32
5.18. Cas d'utilisation « chercher véhicule».....	32
5.19. Cas d'utilisation « chercher plan».....	33
5.20. Cas d'utilisation « construire un plan».....	34
5.21. Cas d'utilisation «authentification».....	34
5.22. Cas d'utilisation «changer mot de passe».....	34
6. Diagrammes de séquence.....	36
6.1. Cas d'utilisation « ajouter client ».....	36
6.2. Cas d'utilisation « ajouter demande de client ».....	37
6.3. Cas d'utilisation «ajouter chauffeurs».....	38
6.4. Cas d'utilisation «ajouter véhicule».....	39
6.5. Cas d'utilisation « modifier client ».....	40
6.6. Cas d'utilisation « modifier véhicule».....	41
6.7. Cas d'utilisation « modifie chauffeur».....	42
6.8. Cas d'utilisation «modifier demande de client.....	43
6.9. Cas d'utilisation «modifier le trajet des tournée».....	44
6.10. Cas d'utilisation « supprimer client ».....	45
6.11. Cas d'utilisation «supprimer demande de client».....	46
6.12. Cas d'utilisation «supprimer plan».....	47
6.13. Cas d'utilisation «supprimer chauffeurs».....	48
6.14. Cas d'utilisation «supprimer véhicule ».....	49
6.16. Cas d'utilisation « chercher client».....	50
6.16. Cas d'utilisation « chercher demande».....	51
6.17. Cas d'utilisation « chercher chauffeur».....	52
6.18. Cas d'utilisation « chercher véhicule».....	53
6.19. Cas d'utilisation « chercher plan».....	54
6.20. Cas d'utilisation « construire un plan».....	55
6.21. Cas d'utilisation «authentification».....	56
6.22. Cas d'utilisation «changer mot de passe».....	57
7. Diagramme de class.....	58
7.1. Cas d'utilisation « ajouter client ».....	58

7.2. Cas d'utilisation « ajouter demande de client ».....	59
7.3. Cas d'utilisation «ajouter chauffeurs».....	60
7.4. Cas d'utilisation «ajouter véhicule».....	60
7.5. Cas d'utilisation « modifier client ».....	61
7.6. Cas d'utilisation « modifier véhicule».....	61
7.7. Cas d'utilisation « modifie chauffeur».....	62
7.8. Cas d'utilisation «modifier demande de client.....	62
7.9. Cas d'utilisation «modifier le trajet des tournée».....	63
7.10. Cas d'utilisation « supprimer client ».....	63
7.11. Cas d'utilisation «supprimer demande de client».....	64
7.12. Cas d'utilisation «supprimer plan».....	65
7.13. Cas d'utilisation «supprimer chauffeurs».....	66
7.14. Cas d'utilisation «supprimer véhicule ».....	67
7.15. Cas d'utilisation « chercher client».....	68
7.16. Cas d'utilisation « chercher demande».....	68
7.17. Cas d'utilisation « chercher chauffeur».....	69
7.18. Cas d'utilisation « chercher véhicule».....	69
7.19. Cas d'utilisation « chercher plan».....	70
7.20. Cas d'utilisation « construire un plan».....	70
7.21. Cas d'utilisation «authentification».....	71
7.22. Cas d'utilisation «changer mot de passe».....	71
8. Diagramme d'activité.....	72
8.1. Cas d'utilisation « ajouter client ».....	72
8.2. Cas d'utilisation « ajouter demande de client ».....	73
8.3. Cas d'utilisation «ajouter chauffeurs».....	74
8.4. Cas d'utilisation «ajouter véhicule».....	75
8.5. Cas d'utilisation « modifier client ».....	76
8.6. Cas d'utilisation « modifier véhicule».....	77
8.7. Cas d'utilisation « modifie chauffeur».....	78
8.8. Cas d'utilisation «modifier demande de client.....	79
8.9. Cas d'utilisation «modifier le trajet des tournée».....	80
8.10. Cas d'utilisation « supprimer client ».....	81
8.11. Cas d'utilisation «supprimer demande de client».....	82
8.12. Cas d'utilisation «supprimer plan».....	83
8.13. Cas d'utilisation «supprimer chauffeurs».....	84
8.14. Cas d'utilisation «supprimer véhicule ».....	85
8.15. Cas d'utilisation « chercher client».....	86
8.16. Cas d'utilisation « chercher demande».....	87
8.17. Cas d'utilisation « chercher chauffeur».....	88
8.18. Cas d'utilisation « chercher véhicule».....	89
8.19. Cas d'utilisation « chercher plan».....	90
8.20. Cas d'utilisation « construire un plan».....	91
8.21. Cas d'utilisation «authentification».....	92
8.22. Cas d'utilisation «changer mot de passe».....	93

9. Conclusion	93
---------------------	----

CHAPITRE IV: Implémentation

1. Introduction.....	94
2. Outils d'implémentation	94
2.1. Le langage de programmation C#.....	94
2.2. Visual Studio	94
2.3. SQL Server et les bases de données	96
3. Interfaces de l'application développée.....	96
3.1. Page d'accueil	96
3.2. Pages gestion des clients.....	97
3.3. Pages gestion des demandes.....	98
3.4. Pages gestion des véhicules.....	99
3.5. Pages gestion des plans.....	100
4. Conclusion.....	101

Conclusion Générale

LISTES DES FIGURES

Figure 1.1 : Logo UML.....	2
Figure 1.2 : Schéma de la hiérarchie des diagrammes d'UML 2.0.....	5
Figure 1.3 : Représentation d'un acteur.....	6
Figure 1.4 : Représentations de cas d'utilisation.....	6
Figure 1.5 : Représentations d'une note.....	6
Figure 1.6 : Représentation de relation d'association.....	7
Figure 1.7 : Représentations de relation d'inclusion.....	7
Figure 1.8 : Représentations de relation d'extension.....	7
Figure 1.9 : Relation de généralisation.....	7
Figure 1.10 : Représentations d'une classe.....	8
Figure 1.11 : Relation d'association.....	8
Figure 1.12 : Représentation relation d'agrégation.....	8
Figure 1.13 : Représentation relation de composition.....	9
Figure 1.14 : Représentation relation de généralisation.....	9
Figure 1.15 : Fragment alt.....	11
Figure 1.16 : Fragment opt.....	12
Figure 1.17 : Schéma qui présente fragment loop.....	12
Figure 2.1 : Représentation de problème de Transport à la Demande (TAD).....	16
Figure 2.2 : Typologie des problèmes aboutissant au problème du TAD.....	18
Figure 3.1: Diagramme de cas d'utilisation.....	22
Figure 3.2 : Diagramme de séquence ajouter client.....	36
Figure 3.3 : Diagramme de séquence ajouter demande.....	37
Figure 3.4 : Diagramme de séquence ajouter chauffeur.....	38
Figure 3.5 : Diagramme de séquence ajouter véhicule.....	39
Figure 3.6 : Diagramme de séquence modifie client.....	40
Figure 3.7 : Diagramme de séquence modifie véhicule.....	41
Figure 3.8 : Diagramme de séquence modifie chauffeur.....	42
Figure 3.9 : Diagramme de séquence demande de client.....	43
Figure 3.10 : Diagramme de séquence modifie le trajet de tournée.....	44

Figure 3.11 : Diagramme de séquence supprimer client.....	45
Figure 3.12 : Diagramme de séquence supprimer demande de client.....	46
Figure 3.13: Diagramme de séquence supprimer plan.....	47
Figure 3.14 : Diagramme de séquence supprimer chauffeur.....	48
Figure 3.15 : Diagramme de séquence supprimer véhicule.....	49
Figure 3.16 : Diagramme de séquence cherche client.....	50
Figure 3.17 : Diagramme de séquence cherche demande de client.....	51
Figure 3.18 : Diagramme de séquence cherche chauffeur.....	52
Figure 3.19 : Diagramme de séquence chercher véhicule.....	53
Figure 3.20 : Diagramme de séquence cherche plan.....	54
Figure 3.21 : Diagramme de séquence construire un plan.....	55
Figure 3.22 : Diagramme de séquence authentification.....	56
Figure 3.23 : Diagramme de séquence changer mot de passe.....	57
Figure 3.24: Modèle de domaine.....	58
Figure 3.25 : Diagramme de classe participante de cas d'utilisation ajouter client.....	58
Figure 3.26 : Diagramme de classe participante de cas d'utilisation ajouté demande des clients.....	59
Figure 3.27 : Diagramme de classe participante de cas d'utilisation ajouté chauffeur.....	60
Figure 3.28 : Diagramme de classe participante de cas d'utilisation ajouté véhicule.....	60
Figure 3.29 : Diagramme de classe participante de cas d'utilisation modifier client.....	61
Figure 3.30 : Diagramme de classe participante de cas d'utilisation modifier véhicule.....	61
Figure 3.31 : Diagramme de classe participante de cas d'utilisation modifier chauffeur.....	62
Figure 3.32 : Diagramme de classe participante de cas d'utilisation modifier demande de client.....	62
Figure 3.33 : Diagramme de classe participante de cas d'utilisation modifier le trajet de tournée.....	63
Figure 3.34 : Diagramme de classe participante de cas d'utilisation supprimer client.....	63
Figure 3.35 : Diagramme de classe participante de cas d'utilisation supprimer demande de client.....	64
Figure 3.36 : Diagramme de classe participante de cas d'utilisation supprimer plan.....	65
Figure 3.37 : Diagramme de classe participante de cas d'utilisation supprimer chauffeur.....	66
Figure 3.38 : Diagramme de classe participante de cas d'utilisation supprimer véhicule.....	67
Figure 3.39 : Diagramme de classe participante de cas d'utilisation chercher client.....	68
Figure 3.40 : Diagramme de classe participante de cas d'utilisation cherche demande de client.....	68
Figure 3.41 : Diagramme de classe participante de cas d'utilisation cherche chauffeur.....	69
Figure 3.42 : Diagramme de classe participante de cas d'utilisation cherche véhicule.....	69

Figure 3.43 : Diagramme de classe participante de cas d'utilisation cherche plan.....	70
Figure 3.44 : Diagramme de classe participante de cas d'utilisation construire un plan.....	70
Figure 3.45 : Diagramme de classe participante de cas d'utilisation authentification.....	71
Figure 3.46 : Diagramme de classe participante de cas d'utilisation changer le mot de passe.....	71
Figure 3.47: Diagramme d'activité de Cas d'utilisation ajouté client.....	72
Figure 3.48: Diagramme d'activité de cas d'utilisation ajouté demande de client.....	73
Figure 3.49: Diagramme d'activité de cas d'utilisation ajouté chauffeur.....	74
Figure 3.50: Diagramme d'activité de cas d'utilisation ajouté véhicule.....	75
Figure 3.51: Diagramme d'activité de cas d'utilisation modifier client.....	76
Figure 3.52: Diagramme d'activité de cas d'utilisation modifier véhicule.....	77
Figure 3.53: Diagramme d'activité de cas d'utilisation modifier chauffeur.....	78
Figure 3.54: Diagramme d'activité de cas d'utilisation modifier demande de client.....	79
Figure 3.55: Diagramme d'activité de cas d'utilisation modifier le trajet de tournée.....	80
Figure 3.56: Diagramme d'activité de cas d'utilisation supprimer client.....	81
Figure 3.57: Diagramme d'activité de cas d'utilisation supprimer demande de client.....	82
Figure 3.58: Diagramme d'activité de cas d'utilisation supprimer plan.....	83
Figure 3.59: Diagramme d'activité de cas d'utilisation supprimer chauffeur.....	84
Figure 3.60: Diagramme d'activité de cas d'utilisation supprimer véhicule.....	85
Figure 3.61: Diagramme d'activité de cas d'utilisation cherche client.....	86
Figure 3.62: Diagramme d'activité de cas d'utilisation cherche demande de client.....	87
Figure 3.63: Diagramme d'activité de cas d'utilisation cherche chauffeur.....	88
Figure 3.64: Diagramme d'activité de cas d'utilisation cherche véhicule.....	89
Figure 3.65: Diagramme d'activité de cas d'utilisation cherche plan.....	90
Figure 3.66: Diagramme d'activité de cas d'utilisation construire un plan.....	91
Figure 3.67: Diagramme d'activité de cas d'utilisation authentification.....	92

Figure 3.68: Diagramme d'activité de cas d'utilisation changer mot de passe	93
---	----

LISTE DES TABLEAUX

Tableau 3.1 : Fiche descriptive de cas d'utilisation ajouter client.....	23
Tableau 3.2 : Fiche descriptive de cas d'utilisation ajouter demande de client.....	24
Tableau 3.3 : Fiche descriptive de cas d'utilisation ajouter chauffeur.....	24
Tableau 3.4 : Fiche descriptive de cas d'utilisation ajouter véhicule.....	25
Tableau 3.5 : Fiche descriptive de cas d'utilisation modifier client.....	26
Tableau 3.6: Fiche descriptive de cas d'utilisation modifier véhicule.....	26
Tableau 3.7 : Fiche descriptive de cas d'utilisation modifie chauffeur.....	27
Tableau 3.8 : Fiche descriptive de cas d'utilisation modifier demande de client.....	27
Tableau 3.9 : Fiche descriptive de cas d'utilisation modifie le trajet de la tournée.....	28
Tableau 3.10 : Fiche descriptive de cas d'utilisation supprimer client.....	28
Tableau 3.11 : Fiche descriptive de cas d'utilisation supprimer demande de client.....	29
Tableau 3.12 : Fiche descriptive de cas d'utilisation supprimé plan.....	30
Tableau 3.13 : Fiche descriptive de cas d'utilisation supprimer chauffeur.....	30
Tableau 3.14 : Fiche descriptive de cas d'utilisation supprimer véhicule.....	31
Tableau 3.15 : Fiche descriptive de cas d'utilisation chercher client.....	31
Tableau 3.16 : Fiche descriptive de cas d'utilisation chercher demande.....	32
Tableau 3.17 : Fiche descriptive de cas d'utilisation chercher chauffeur.....	32
Tableau 3.18 : Fiche descriptive de cas d'utilisation chercher véhicule.....	33
Tableau 3.19 : Fiche descriptive de cas d'utilisation chercher plan.....	33
Tableau 3.20 : Fiche descriptive de cas d'utilisation construire un plan.....	34
Tableau 3.21 : Fiche descriptive de cas d'utilisation ajouter authentification.....	35
Tableau 3.22 : Fiche descriptive de cas d'utilisation changer mot de passe.....	35

Introduction générale

Introduction générale

L'informatique est un domaine très vaste ayant connu une progression très rapide depuis quelque années. En même temps que l'informatique évolue, l'évolution elle-même contribuera à d'autres progressions futures. L'informatique a couvert quasiment tout autres domaines que ça soit en recherche où en application, ce dernier simplifie les travaux de la gestion, la manipulation et le contrôle des sociétés et administration. A ce stade, elle est considérée comme un outil exploitable par les autres activités afin d'informatiser leurs tâches.

De nos jours, le transport public est une activité très fréquente est importante. Il est suffisamment disponible pour subvenir aux besoins des personnes et est opérationnel la majorité du temps, il couvre la plupart des zones d'agglomérations, et même les plus éloignés. Mais malgré cela, certaines personnes ont besoins de se rendre à un lieu de rendez-vous; qui est évidemment possible d'atteindre en empruntant le transport public; Les transports publics empruntent des trajets fixes avec des arrêts bien définis. Et parfois, le lieu de destination d'une personne ne se situe toujours pas à proximité d'un arrêt, donc la personne est obligé de soit prendre une nouvelle fois le transport où continuer le trajet restant à pieds. Mais dans certaine situation, la contrainte de temps est un facteur très important, et tout temps perdu durant le déplacement où pendant un arrêt à une station intermédiaire pénalise d'un temps non négligeable la personne concerné. Malgré que le transport public fait gagner au voyageur un précieux temps, il aura été encore plus satisfait s'il atteignait le lieu de destination au plus bref délai avec un minimum de risque, d'où l'utilisation du service de TAD.

Les TAD se distinguent des autres services de transports collectifs par le fait que les véhicules n'empruntent pas d'itinéraire fixe et ne respectent pas un horaire précis, sauf pour satisfaire parfois un besoin particulier. Les TAD sont cependant organisés par des professionnels du transport et, à la différence des taxis, les voyages ne sont en général pas individuels. Un opérateur (ou un système automatisé) se charge alors de la réservation, de la planification et de l'organisation afin de prendre en charge l'ensemble des voyageurs.

Dans le cadre de ce travail, nous proposons d'établir un système pour le transport à la demande afin d'obtenir une qualité de service maximal avec un coût minimal.

Ce mémoire est organisé en quatre chapitres :

Chapitre 1 : « langage de modélisation UML» le but de ce chapitre est de donner une description détaillée des diagrammes fondamentaux d'UML (Unified Modeling Language), qui s'est imposé comme une norme standard de modélisation.

Chapitre 2 : « Le transport à la demande » ce chapitre, est dédié à la présentation d'un état de l'art sur le problème de transport à la demande et ses caractéristiques. Et comme notre système est décrit pour une agence virtuelle nous présentons quelques agences existantes pour faciliter l'identification des besoins dans le prochain chapitre.

Chapitre 3 : « Modélisation » Dans le cadre de ce chapitre, nous allons présenter l'ensemble de diagrammes utilisés pour la modélisation de notre système suivant le processus UP et en utilisant le langage UML

Chapitre 4 : « L'implémentation » Dans le cadre de ce chapitre, nous allons définir premièrement, les outils et l'environnement de développement utilisés pour construire l'application. Ensuite, on va présenter les interfaces du système.

Chapitre 01

Chapitre 01 : La modélisation avec UML

1. Introduction

Le développement des applications et des logiciels est une opération complexe. Selon leurs tailles les applications peuvent être développées par une seule personne ou un ensemble d'équipes coordonnées. Le développement de grands logiciels par des grandes équipes posent d'importants problèmes de conception et de coordination. Pour maîtriser la complexité et résoudre ces problèmes plusieurs langages ou méthodes de spécification et outils de modélisation ont été créés (telle MERISE) et autre comme Broch, OMT, UML.

Dans notre projet on va utiliser le langage de modélisation UML. Donc le but de ce chapitre est de donner une description détaillée des diagrammes fondamentaux d'UML (Unifié Mödling Langage), qui s'est imposé comme une norme standard de modélisation.

2. Modélisation avec UML

Le recours à la modélisation est depuis longtemps une pratique indispensable au développement logiciel, car un modèle est prévu pour arriver à anticiper les résultats du codage.

2.1. Définition d'un modèle

Un modèle est une simplification de la réalité qui permet de mieux comprendre le système à développer. Un modèle est une représentation abstraite et simplifiée (i.e. qui exclut certains détails), d'une entité (phénomène, processus, système, etc.) du monde réel en vue de le décrire, de l'expliquer ou de le prévoir. Concrètement, un modèle permet de :

- réduire la complexité d'un phénomène en éliminant les détails qui n'influencent pas son comportement de manière significative.
- De visualiser le système comme il est ou comme il devrait l'être.
- De valider le modèle vis à vis des clients
- De spécifier les structures de données et le comportement du système.
- De fournir un guide pour la construction du système.
- De documenter le système et les décisions prises.

2.3 Langages de modélisation

Un langage de modélisation doit définir la sémantique des concepts et définir une notation pour la représentation de concepts et doit contenir des règles de construction et d'utilisation des concepts.

Les langages de modélisation ont différents niveaux de formalisation :

- **Langages formels (Z, B, VDM)** : le plus souvent mathématiques. ces langages ont un grand pouvoir d'expression et permettent de faire des preuves formelles sur les spécifications ;
- **Langages semi-formels (MERISE, UML...)** : le plus souvent graphiques, ils ont un pouvoir d'expression moins que les langages formels, mais plus faciles à utiliser.

L'industrie du logiciel dispose de nombreux langages de modélisation :

- Adaptés aux systèmes procéduraux (MERISE...);
- Adaptés aux systèmes temps réel (ROOM, SADT...);
- Adaptés aux systèmes à objets (OMT, Broch, UML...).

2.4. Définition d'UML

UML (sigle désignant Unifié Modelling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique à base de pictogramme. Il est apparu dans le monde du génie logiciel. C'est une méthode de modélisation orientée objet développée en réponse à l'appel à propositions lancé par l'OMG (Object Management Group) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets.



Figure1.1: Logo UML.

2.5. Caractéristiques d'UML

UML cadre l'analyse objet en offrant différentes vues (perspectives) complémentaires d'un système, il offre aussi plusieurs niveaux d'abstraction qui permettent de mieux contrôler la complexité dans l'expression des solutions objets.

Les principales caractéristiques d'UML sont les suivantes :

- Sa notation graphique permet d'exprimer visuellement une UML solution objet.

- L'aspect formel de sa notation limite les ambiguïtés et les Incompréhensions.
- Son aspect visuel facilite la comparaison et l'évaluation de Solutions.
- Son indépendance (par rapport aux langages d'implémentation, domaine d'application, processus...) en font un langage Universel.

3. Présentation des vues et diagrammes d'UML

UML dans sa version 2 propose treize diagrammes (9 en UML 1.3) qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles.

3.1 Diagrammes structurels ou diagrammes statiques

Ces diagrammes permettent de visualiser, spécifier, construire et documenter l'aspect statique ou structurel du système informatisé :

- **Diagramme de classes (Class diagramme):** Le but d'un diagramme de classes est d'exprimer de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes. Une classe a des attributs, des opérations et des relations avec d'autres classes.
- **Diagramme d'objets (Object diagramme):** Le diagramme d'objet permet d'éclairer un diagramme de classe en l'illustrant par des exemples. Il montre des objets et des liens entre ces objets (les objets sont des instances de classes dans un état particulier).
- **Diagramme de composants (Component diagramme) :** il montre les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...). Il montre la mise en œuvre physique des modèles de la vue logique avec l'environnement de développement.
- **Diagramme de déploiement (Déploiement diagramme) :** Ce type de diagramme UML montre la disposition physique des matériels qui composent le système (ordinateurs, périphériques, réseaux...) et la répartition des composants sur ces matériels. Les ressources matérielles sont représentées sous forme de nœuds, connectés par un support de communication.
- **Diagramme des paquetages (Package Diagramme) :** un paquetage est un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML, il sert à représenter les dépendances entre paquetages.

- **Diagramme de structure composite (Composite Structure Diagramme) :** Le diagramme de structure composite permet de décrire sous forme de boîte blanche les relations entre les composants d'une seule classe.

3.2 Diagrammes comportementaux ou diagrammes dynamiques

Les diagrammes comportementaux modélisent les aspects dynamiques du système. Ces aspects incluent les interactions entre le système et ses différents acteurs, ainsi que la façon dont les différents objets contenus dans le système communiquent entre eux.

- **Diagramme des cas d'utilisation (Use Case Diagramme) :** Les cas d'utilisation sont une technique de description du système étudié selon le point de vue de l'utilisateur. Ils décrivent sous la forme d'actions et de réactions le comportement d'un système. Donc, le diagramme des cas d'utilisation, permet d'identifier les possibilités d'interaction entre le système et les acteurs. Il permet de clarifier, filtrer et organiser les besoins.
- **Diagramme d'activité (Activité Diagramme) :** Un diagramme d'activité est une variante des diagrammes d'états-transitions. Il permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. dans un diagramme d'activité les états correspondent à l'exécution d'actions ou d'activités et les transitions sont automatiques.
- **Diagramme états-transitions (State Machine Diagramme) :** permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants. Il est composé d'un ensemble d'états, reliés par des arcs orientés qui décrivent les transitions.
- **Diagramme de séquence (Séquence Diagramme) :** Il représente séquentiellement le déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs. Le diagramme de séquence peut servir à illustrer un cas d'utilisation.
- **Diagramme de communication (Communication Diagramme) :** C'est une représentation simplifiée d'un diagramme de séquence, en se concentrant sur les échanges de messages entre les objets.
- **Diagramme global d'interaction (Interaction Ovarien Diagramme) :** permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).
- **Diagramme de temps (Timing Diagramme) :** Le diagramme de temps permet de décrire les variations d'une donnée au cours du temps.

La figure 1.4, montre la hiérarchie des diagrammes d'UML 2.0.

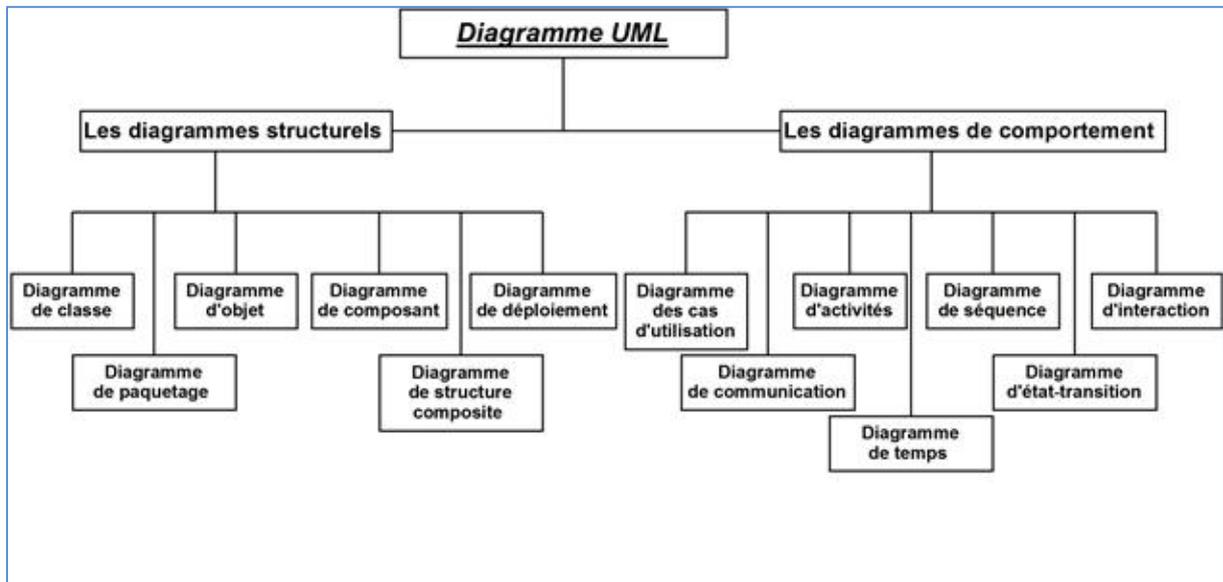


Figure 1.2: Schéma de la hiérarchie des diagrammes d'UML 2.0

Nous décrivons dans ce qui suit les principaux diagrammes d'UML :

3.3. Diagramme de cas d'utilisation.

Le diagramme de cas d'utilisation permet :

- D'exprimer simplement les besoins des utilisateurs
- D'analyser les besoins des utilisateurs
- De déterminer les interfaces du système
- De connaître le comportement du système sans spécifier comment ce comportement est réalisé,
- De définir les limites précises du système,
- De comprendre l'attente des utilisateurs et des experts du domaine.

Les éléments d'un diagramme de cas d'utilisations sont les suivants :

- **Acteur** : Un acteur est l'archétype de l'utilisateur (personne, externe, ...) qui interagit avec le système. Il se représente par un petit bonhomme avec son nom ou par la forme d'un classeur.

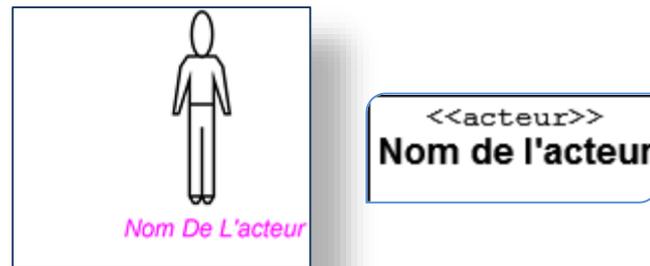


Figure1.3: Représentation d'un acteur

- **Cas d'utilisations:** Les cas d'utilisations modélisent le service rendu par le Système sans en imposer le mode de réalisation. Ils permettent de décrire l'interaction entre l'acteur et le système. Les cas d'utilisations sont représentés par une ellipse sous-titrée par le nom du cas d'utilisation (éventuellement le nom est placé dans l'ellipse). Dans le cas où l'on désire présenter les attributs ou les opérations du cas d'utilisation, il est préférable de le représenter sous la forme d'un classeur stéréotypé.



Figure1.4: représentations de cas d'utilisation.

- **Note:** Une note permet l'ajout d'une information textuelle à un diagramme. Cette information peut être un commentaire, un corps de méthode ou une contrainte. Les notes sont représentées par un rectangle avec le coin supérieur droit replié sur lui-même. On peut relier une note à un élément en utilisant une ligne pointillée.

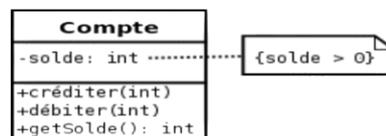


Figure1.5:représentations d'une note

- **Relations**
 - **Relation d'association :** Une relation d'association est un lien de communication entre un acteur et un cas d'utilisation. Elle est représentée par un trait continu.



Figure1.6: représentation de relation d'association.

- **Relation d'association** : La relation d'inclusion spécifie qu'un cas d'utilisation est nécessairement une partie d'un autre cas d'utilisation. Elle est représentée par Une flèche discontinue stéréotypée <<inclusion>>.

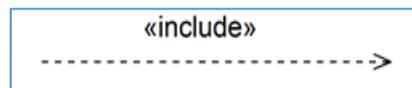


Figure1.7 : représentations de relation d'inclusion.

- **Relation d'extension**: La relation d'extension spécifie qu'un cas d'utilisation est éventuellement une partie d'un autre cas d'utilisation. Elle est représentée par Une flèche discontinue stéréotypée <<extension>>.



Figure1.8 : représentations de relation d'extension.

- **Relation de généralisation** : La relation de généralisation/spécialisation est la transposition aux cas d'utilisation de la notion d'héritage dans le paradigme objet. Elle est représentée par Une flèche dont la pointe (un triangle fermé) est dirigée vers l'élément le plus général.

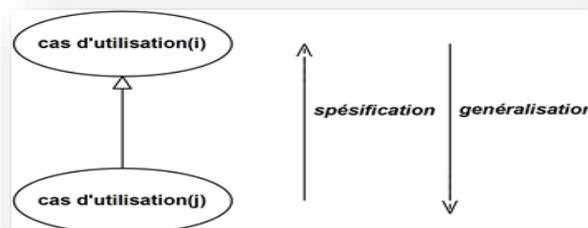


Figure 1.9: relation de généralisation.

3.4. Diagramme de classe

Le diagramme de classe est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Le diagramme de classe est un

diagramme entité association identifie la structure des classe et les interfaces d'un système, y compris les propriétés et les méthodes de chaque classe ainsi que les différentes relations entre celles-ci. Les éléments de diagramme de classe sont les suivants :

- **Classe:** Une classe est une **abstraction** de choses du monde réel possédant des **caractéristiques** (Attributs) et des **comportements communs** (méthodes) qui sont liées ensemble par un champ sémantique. Une classe est représentée par un rectangle séparée en trois parties : la première partie contient le nom de la classe, la seconde contient les attributs de la classe et la dernière contient les méthodes de la classe

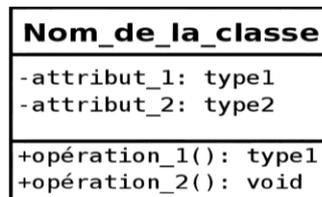


Figure1.10 : représentations d’une classe

- **Relations entre classes**

- **Relation d’association :** Une association est une relation entre deux classes (association binaire) ou plus (association-aire), qui décrit les connexions structurelles entre leurs instances. Une association indique donc qu’il peut y avoir des liens entre des instances des classes associées.



Figure1.11:Relation d’association.

- **Relation d’agrégation :** Une agrégation est une **relation tout-partie** entre un agrégat (le tout) et un composant ou agrégée (la partie). L’agrégation est représentée par un losange blanc du côté de l’agrégat



Figure 1.12: représentation relation d’agrégation.

- **Relation de composition:** La composition, également appelée agrégation composite, décrit une contenance structurelle entre instances. Ainsi, la destruction de l'objet composite implique la destruction de ses Composants. L'agrégation est représentée par un losange noir du côté de l'agrégat composite.



Figure 1.13: représentation relation de composition.

- **Relation de généralisation et héritage:** La **généralisation** est la relation entre une classe et deux autres classes ou plus partageant un sous-ensemble commun d'attributs et/ou d'opérations. La classe qui est affinée s'appelle **superclasse**, les classes affinées s'appellent **sous-classes**. L'opération qui consiste à créer une superclasse à partir de classes s'appelle la généralisation. Inversement la **spécialisation** consiste à créer des sous-classes à partir d'une classe. Elle est représentée par Une flèche dont la pointe (un triangle fermé) est dirigée vers la superclasse.

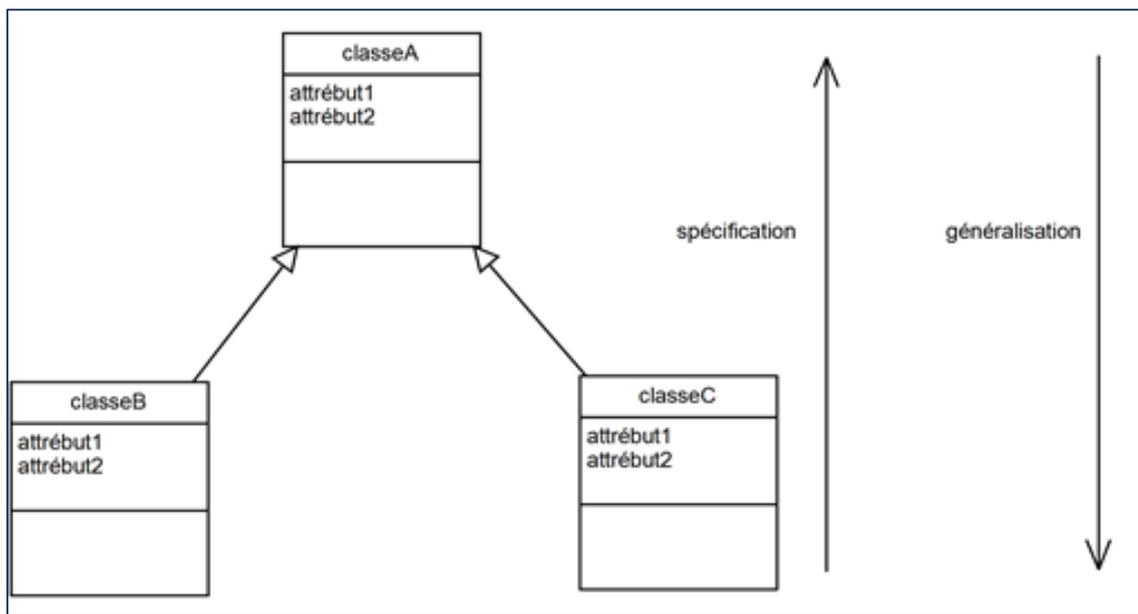


Figure1.14 : représentation relation de généralisation

3.5. Diagramme d'activité

Le diagramme d'activité est un **diagramme états-transitions simplifié** pour lequel les états se réduisent à de simples actions ou activités et dont les transitions se déclenchent **automatiquement** avec éventuellement des gardes. Les diagrammes d'activité sont utilisés pour documenter le déroulement des opérations dans un système, du niveau commercial au niveau opérationnel (de haut en bas). Les éléments de base de diagramme d'activité :

- **Nœud initial** : Un nœud initial est un nœud de contrôle à partir duquel le flot débute lorsque l'activité enveloppante est invoquée. Une activité peut avoir plusieurs nœuds initiaux.
- **Nœud final** : Un nœud final est un nœud de contrôle possédant un ou plusieurs arcs entrants et aucun arc sortant.
- **Nœud de fin d'activité** : Lorsque l'un des arcs d'un nœud de fin d'activité est activé (i.e. lorsqu'un flot d'exécution atteint un nœud de fin d'activité), l'exécution de l'activité enveloppante s'achève et tout nœud ou flot actif au sein de l'activité enveloppante est abandonné.
- **Nœud de fin de flot** : Lorsqu'un flot d'exécution atteint un nœud de fin de flot, le flot en question est terminé, mais cette fin de flot n'a aucune incidence sur les autres flots actifs de l'activité enveloppante.
- **Nœud de décision** : Un nœud de décision est un nœud de contrôle qui permet de faire un choix entre plusieurs flots sortants. Il possède un arc entrant et plusieurs arcs sortants. C
- **Nœud de fusion** : Un nœud de fusion est un nœud de contrôle qui rassemble plusieurs flots alternatifs entrants en un seul flot sortant. Il n'est pas utilisé pour synchroniser des flots concurrents (c'est le rôle du nœud d'union) mais pour accepter un flot parmi plusieurs.
- **Le nœud d'action** : Un nœud d'action est un nœud d'activité exécutable qui constitue l'unité fondamentale de fonctionnalité exécutable dans une activité.
- **Le nœud d'objet** : Un nœud d'objet permet de définir un flot d'objet (i.e. un flot de données) dans un diagramme d'activités. Ce nœud représente l'existence d'un objet généré par une action dans une activité et utilisé par d'autres actions.
- **La transition** : Quand un état d'activité est accompli, le traitement passe à un autre état d'activité. Les transitions sont utilisées pour marquer ce passage. Les transitions sont modélisées par des flèches.

3.6. Le diagramme de séquence/séquence système

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur, à savoir : saisir une donnée, consulter une donnée, lancer un traitement...etc. Il indique les objets que l'acteur va manipuler, et les opérations qui font passer d'un objet à l'autre. Un diagramme de séquence se représente globalement dans un grand rectangle avec indication du nom du diagramme en haut à gauche. Les éléments d'un diagramme de séquences sont les suivants :

- **Messages :** Un message définit une communication particulière entre des lignes de vie. Ainsi, un message est une communication d'un objet vers un autre objet. La réception d'un message est considérée par l'objet récepteur comme un événement qu'il faut traiter (ou pas). Plusieurs types de messages existent, les plus communs sont :
 - L'invocation d'une opération : **message synchrone** (appel d'une méthode de l'objet cible).
 - L'envoi d'un signal : **message asynchrone** (Typiquement utilisé dans la gestion événementielle).
 - La création ou la destruction d'une instance de classe au cours du cycle principal.
- **Fragment d'interaction :** Un fragment d'interaction dit combiné correspond à un ensemble d'interaction auquel on applique un opérateur. Un fragment combiné se représente globalement comme un diagramme de séquence avec indication dans le coin à gauche du nom de l'opérateur.
 - **Fragment d'interaction avec opérateur alt :** L'opérateur alt correspond à une instruction de test avec une ou plusieurs alternatives possibles possédant plusieurs opérandes séparés par des pointillés. Il est aussi permis d'utiliser les clauses de type sinon.



Figure1.15 : fragment alt

- **Fragment d'interaction avec operateur opt** : L'opérateur *option* (**opt**) comporte un opérande et une condition de garde associée. Le sous-fragment s'exécute si la condition de garde est vraie et ne s'exécute pas dans le cas contraire.

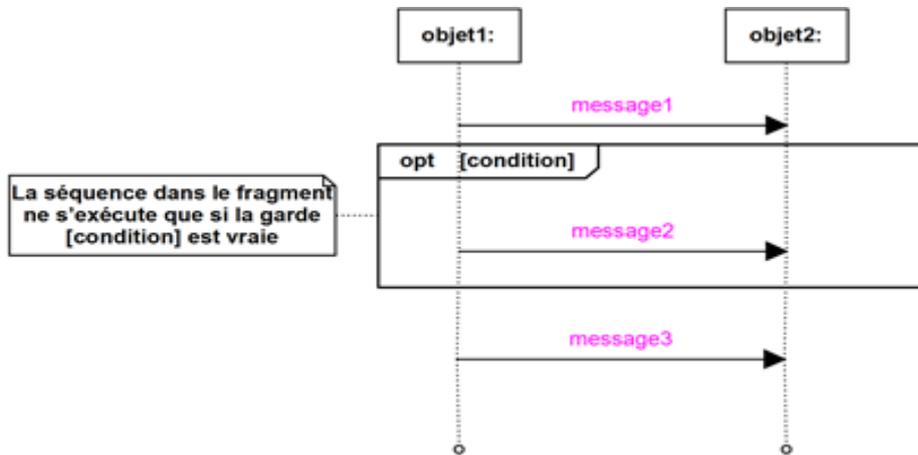


Figure1.16 : fragment opt

- **Fragment d'interaction avec operateur loop**: L'opérateur loop correspond à une instruction de boucle qui permet d'exécuter une Séquence d'interaction tant qu'une condition est satisfaite.

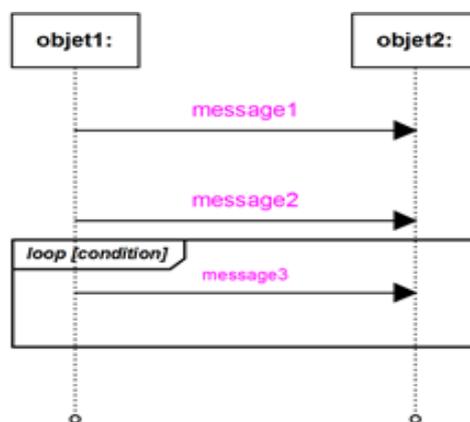


Figure1.17 : schéma qui présente fragment loop

- **Fragment d'interaction avec operateur ref** : permet de faire appel à un autre diagramme de séquence.

3.7. Le diagramme de classe participante

Le diagramme de classes participantes est un diagramme de classes UML qui décrit cas d'utilisation par cas, les trois principales classes d'analyse et leurs relations.

- **Les classes dialogues** : possèdent des attributs et des opérations. Les attributs représentent des champs de saisie ou des résultats. Les opérations elles, représentent des actions de l'utilisation sur l'Interface Homme Machine (IHM)
- **Les classes contrôles** : contiennent des opérations.ces opérations représentent la logique applicative de l'application, les règles métiers ou les comportements du système informatique.
- **Les classes entités** : possèdent en général des informations persistantes de l'application.

4. Le processus :

Un processus de développement définit une séquence d'étapes, en partie ordonnée, qui concoure à l'obtention d'un système logiciel ou à l'évolution d'un système existant ; pour produire des logiciels de qualité, qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles.

Processus s'appuyant sur UML :

Les processus unifiés :

Un processus unifié est un processus construit sur UML (Unified Modeling Language). Les processus unifiés sont le résultat de l'unification, non pas des processus, mais plus exactement les meilleures pratiques du développement objet.

Le processus 2TUP :

Le processus 2TUP (Two Trak Unified Processus) est un processus unifié. Il gère la complexité technologique en donnant part à la technologie dans son processus de développement.

Une méthode simple et générique :

Dans l'étude de notre projet, nous allons suivre une démarche simple et générique. Cette démarche inspirée du processus UP est la mieux adaptée pour notre projet vu sa taille. Cette méthode contient trois phases structurées comme suit :

1. Identification des besoins :

- A. Diagramme de cas d'utilisation
- B. Diagramme de séquence système
- C. Maquette de l'IHM

2. Phase d'analyse :

- A. modèle du domaine ;
- B. Diagramme de classes participantes ;
- C. Diagramme d'activités.

3. Phase de conception :

- A. Diagrammes d'interaction ;
- B. Diagramme de classes de conception ;

5. Conclusion

Dans ce chapitre nous avons présenté brièvement la modélisation orientée objet, la cause de modélisation et son objectif. Ensuite, nous avons donné une explication détaillée de langage de modélisation UML et ses diagrammes.

En fin, UML est un outil précieux. Mais, pour développer une application il faut l'accompagner avec une démarche pour obtenir une vue claire et globale de l'architecture du système, et pour cela nous appliquons dans le reste du mémoire une démarche simple pour élaborer notre système.

Chapitre 02

Chapitre 02 : Le transport à la demande

1. Introduction

La mobilité tient une place importante dans nos vies quotidiennes. Il s'agit de modes de transport individuel (vélo, marche à pied,...), de modes de transport en communs (bus, train,...) ou de modes de transport flexibles qui s'adaptent à des zones moins densément peuplées (transport à la demande, covoiturage,...) [1]. Dans le cadre de notre travail nous nous intéressons au transport à la demande (TAD). Le problème de transport à la demande consiste à déterminer les tournées et horaires des véhicules qui effectuent le transport d'usagers à leurs demandes. Ce problème se pose, par exemple, dans le transport adapté de personnes handicapées, âgées et des patients. Actuellement les recherches dans ce domaine sont en perpétuelle évolution vue l'importance de ce mode de transport dans l'amélioration de la vie des citoyens.

Le présent chapitre est dédié à la présentation d'un état de l'art sur le problème de transport à la demande et ses caractéristiques. Et comme notre système est décrit pour une agence virtuelle nous présentons quelques agences existantes pour faciliter l'identification des besoins dans le prochain chapitre.

2. Le transport à la demande(TAD)

2.1. Définitions

Les définitions officielles de TAD varient d'un continent à un autre. Néanmoins, au-delà de ces définitions officielles s'imposent également les définitions usuelles du TAD, qui font écho à l'usage réservé au TAD en question et sont donc une question de points de vue différents, quand bien même le principe demeure relativement inchangé. Néanmoins, quelle que soit l'acception retenue, toutes s'accordent sur le point que le TAD est une forme de transport public. Dénommé « *Demande Responsive Transport* » (DRT) chez les Anglo-Saxons et dans la littérature scientifique associée.

Le problème de transport à la demande (**TAD**) ou Demande Responsive Transport (**DRT**) est un problème générique qui englobe a priori tous les services de transports dont tout ou partie ne s'effectue qu'à la demande expresse de ceux qui les utilisent. Ce problème générique peut se présenter de la manière suivante : dans un espace donné (un réseau), on cherche à

acheminer des demandes de clients à l'aide de moyens de transport pouvant combiner différents types de véhicules. Chaque demande du client est définie par un lieu d'origine, un lieu de destination, un nombre de passagers et des contraintes temporelles. Ces demandes peuvent être déterministes (connues à l'avance) ou stochastiques (non connues à l'avance) [1].

- **Un service** : Le service peut être accessible à tous publics, ou réservé à des abonnés au profil libre ou spécifique. La réservation s'effectue pour certains TAD à l'avance. Le service comporte parfois un point d'arrêt avec prise en charge sans réservation.
- **Une requête** : est une demande de services pour une opération de transport, un ramassage et une livraison en général, formulée par des passagers. Une demande ou requête du client consiste à le lieu d'origine pour le chargement des passagers et le lieu de destination pour le déchargement des passagers.
- **Une tournée** : est le plan de route exact attribuable à un véhicule qui passer à un ou plusieurs services.

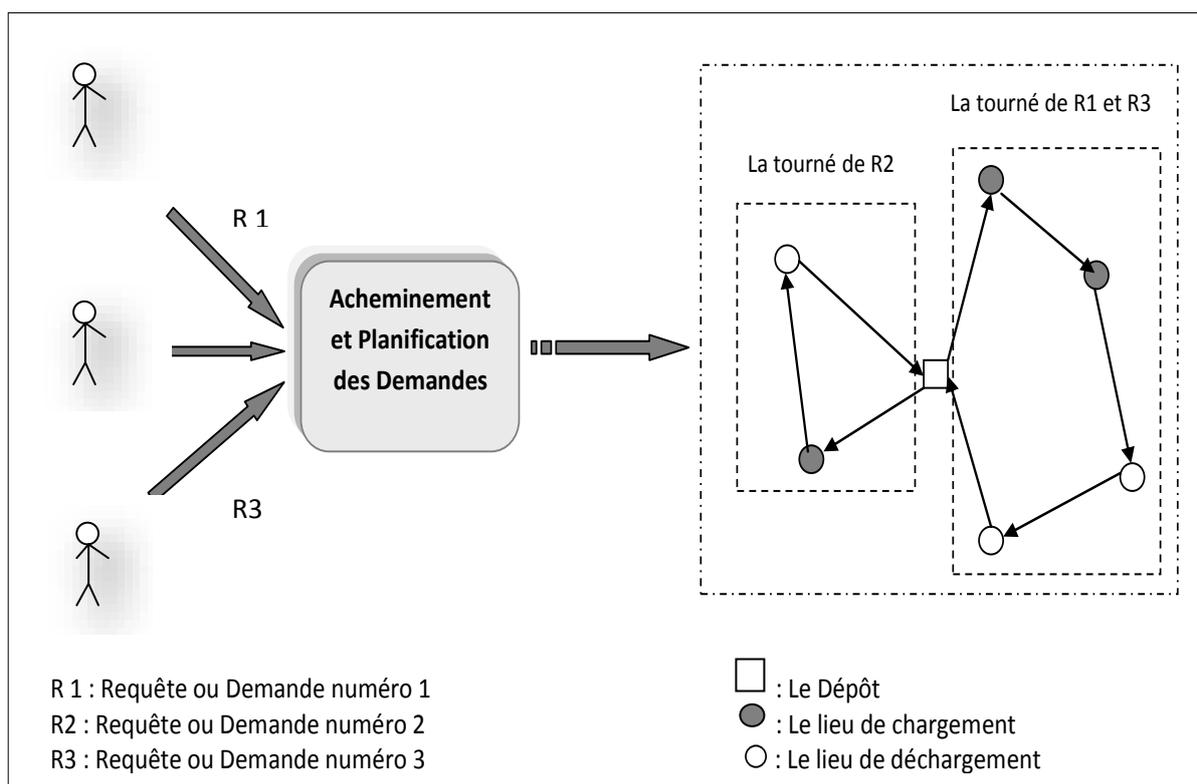


Figure 2.1 : Représentation de problème de Transport à la Demande (TAD).

2.2. Typologie des problèmes

Dans ce paragraphe, nous donnons une typologie des problèmes aboutissant au problème du TAD. Nous détaillons également les différences importantes entre les problèmes considérés dans la littérature. Le problème du TAD fait partie de la famille

des problèmes de tournées de véhicules, qui ont des extensions du problème de voyageur de commerce.

2.2.1. Problème du voyageur de commerce(TSP)

Dans le problème de voyageur de commerce (TSP : Travelling Sales man Problème), des nœuds doivent être visités par un voyageur de commerce qui doit revenir au nœud depuis lequel il est parti. La tournée doit être construite de telle manière que la distance parcourue est minimale.

2.2.2. Problème de tournées de véhicules(VRP)

Le problème de tournées de véhicules (VRP : Véhicule Routine Problème) est un TSP multi-véhicules dans lequel toutes les origines ou toutes les destinations se confondent avec le dépôt central et à chaque nœud est associé une quantité et chaque véhicule a une capacité limitée. La somme des quantités associées aux nœuds parcourus par chaque véhicule ne doit pas dépasser sa capacité. Le problème n'est plus seulement géographique puisque la contrainte de capacité rend des solutions non faisables malgré leur minimisation des distances parcourues.

2.2.3. Problème de tournées de véhicules multi-dépôt (MDVRP)

La version à plusieurs dépôts (MDVRP : Multiple Dépôts VRP) est un VRP où les nœuds à visiter doivent être desservis par un véhicule depuis l'un des dépôts disponibles, et chaque véhicule part et revient à son dépôt initial [2].

2.2.4. Problème de tournées de véhicules avec ramassage et livraison (PDP)

Le problème général de ramassage et livraison ou chargement et déchargement (PDP : Pickup and délivre problème) constitue une classe importante de problème de tournées de véhicules où chaque requête de transport spécifie une origine et une destination uniques et où tous les véhicules partent et reviennent en un dépôt central [9]. Il peut être décrit comme un problème de conception de tournées mettant en jeu des lieux géographiquement dispersés.

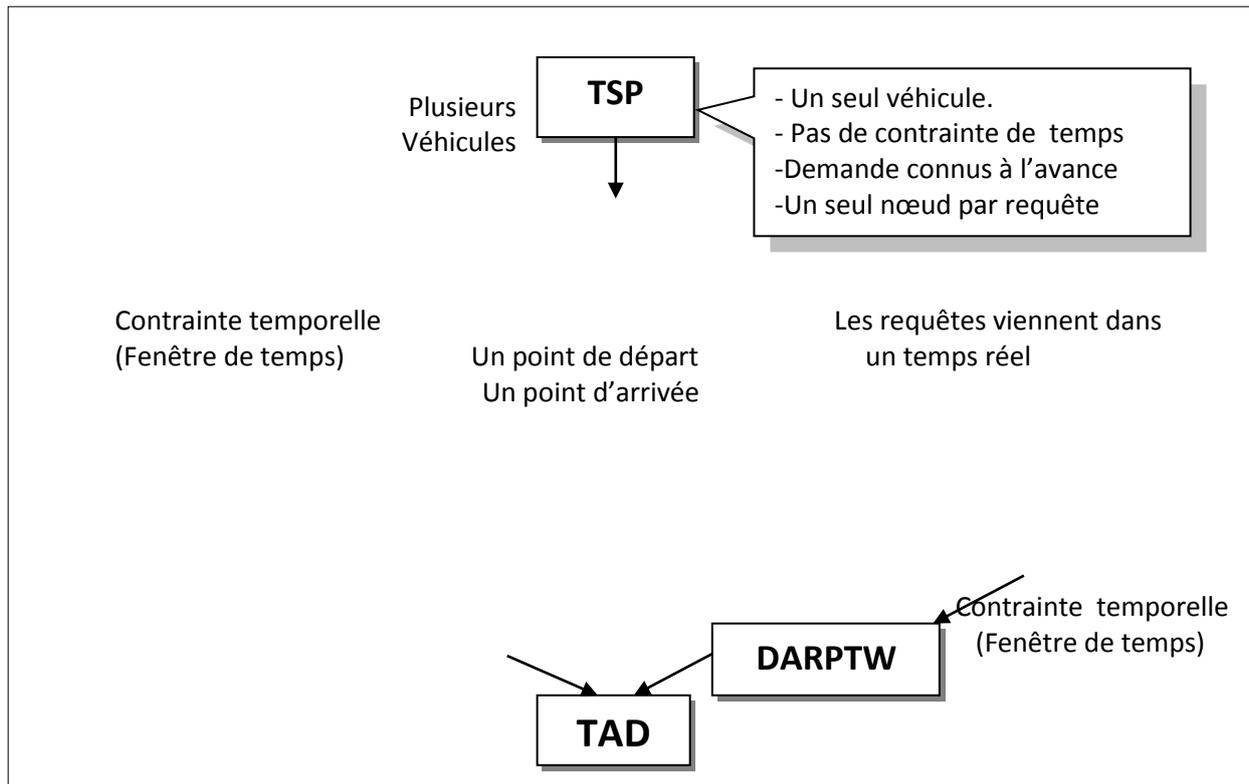


Figure 2.2 : typologie des problèmes aboutissant au problème du TAD

3. Les principaux Avantages du TAD

Le fait de situer le TAD à mi-chemin entre le taxi et le bus tend à montrer que ce service s'adapte aux souhaits des usagers. La flexibilité ou souplesse peut s'exprimer à différents niveaux qui peuvent être :

- la desserte spatiale : le TAD n'est pas à ligne fixe (du moins dans ses versions les plus évoluées). Les itinéraires empruntés par les véhicules assurant la desserte sont libres et définis manuellement ou optimisés de manière informatique, avec une résolution de tournées ;
- les horaires : tout comme les trajets ne sont pas fixes, les horaires de prise en charge et de desserte sont, au mieux, optimisés selon les souhaits des usagers ;
- la tarification : celle-ci peut être fixe (un tarif unique pour le voyage) ou définie selon d'autres modes de calcul.

Une des raisons du succès des TAD auprès des collectivités réside dans son moindre coût de fonctionnement. En effet, comme le TAD n'est pas régulier, celui-ci n'est déclenché que s'il y a besoin. Autrement dit, un TAD ne roule jamais à vide, contrairement à certains véhicules sur des lignes fixes. Les TAD peuvent être sous-traités auprès de compagnies privées ou des

artisans taxis pour réduire les charges liées aux frais de fonctionnement et ne payer que les déplacements effectués.

4. Les principaux manques des TAD actuels

Il existe nous plusieurs freins majeurs à l'utilisation massive du TAD. Le premier réside dans les délais de réservation. En effet, les routes empruntées par les véhicules pour prendre en charge les clients nécessitent des optimisations informatiques préalables. Plus la demande n'est forte, plus les temps de calcul des tournées sont élevés

Le deuxième frein réside dans les capacités de desserte. Les TAD sont souvent hélas pensés comme des substituts de lignes fixes qui suivent des tracés préétablis n'offrant pas l'ubiquité souhaitée par l'utilisateur quand bien même celle-ci est à la base même d'un TAD. Notons que cette insuffisance venait aussi de la difficulté à optimiser la desserte quand le système subissait une forte montée en charge. Toutefois, les progrès réalisés permettent d'envisager une forte évolution des services, alliant qualité et efficacité économique. Mais le principal point noir de la difficulté de développement du TAD reste le simple fait qu'ils sont déployés sur des territoires possédant généralement une offre de TC préalable. Ils n'interviennent alors qu'en complément et ne peuvent capter que des flux de faible effectif et de faible amplitude.

5. Exemples des agences de transport à la demande

Depuis une décennie, des nouveaux services et systèmes de transport à la demande sont mis en œuvre en milieux urbain, périurbain ou rural. Nous en présentons quelques uns:

- **TELEBUS** : Un système de transport à la demande pour les handicapés.
- **Lilas** : Un système de réservation de véhicules par abonnement en utilisant le téléphone ou internet.
- **Mobilité**: Un système de réservation de véhicules pour 55500 clients, les 1700 véhicules de 8 catégories différentes sont répartis pas sur 1000 emplacements dans toute la Suisse.
- **CityCarClub** : Un système de réservation de véhicules créé en 2000, avec plus de 350 véhicules et plus de 40 stations réparties.
- **Modulauto** : Un système de location de voiture en libre service dans l'agglomération. Ce n'est pas un système de covoiturage mais de location simplifiée.

- **PRAXITELE**: Un service public de transport, complémentaire des transports en commun et des taxis qui visent à offrir une alternative aux déplacements en voiture individuelle privée.
- **LISELEC**: Un service par abonnement de location de voitures électriques. Créé en 1999 par PSA Peugeot Citroën, VIA GTI et Alcatel CGA.
- **Vé lib, Auto lib**: un système de vélo (voitures propres électriques) en libre service dans certaines grandes villes de France.
- **Agence iben seina**: est une agence qui fournit un service de transport pour les malades entre leurs domiciles et les hôpitaux ou les cabines médicales.

6. Conclusion

Dans ce chapitre, nous avons présenté le problème de TAD comme variante du problème de tournée de véhicules. Après la définition du problème, nous avons présenté une classification de ce dernier. Ensuite, nous avons énuméré quelques avantages et points faibles qui freinent la mise en place de ce mode de transport. Le chapitre est clôturé par la présentation de quelques agences qui fournissent un service de transport à la demande.

Chapitre 03

Chapitre 03 : Modélisation

1. Introduction

Dans le cadre de ce chapitre, nous allons présenter l'ensemble de diagrammes utilisés pour la modélisation de notre système suivant le processus UP et en utilisant le langage UML selon les étapes suivantes: identification des acteurs de ce système, identification des cas d'utilisation, diagramme de cas d'utilisation, description textuelle du cas d'utilisation, diagrammes de séquences, diagrammes de class, diagrammes d'activités.

2. Identification des acteurs

Nous allons à présent identifier tous les acteurs qui vont interagir avec le système.

- Administrateur : La fonctionnalité principale d'un administrateur est de gérer la totalité du système via la construction et le validation des plans selon les critères d'optimisation. L'administrateur s'occupe aussi de la gestion des ressources (chauffeurs et véhicules).
- Agent : La fonctionnalité principale d'un agent est d'interagir avec les clients selon un système de réservation où il s'occupe de l'enregistrement de requêtes des clients.

3. identification des cas d'utilisations

- Gestion des chauffeurs : Le logiciel assure la gestion des informations personnelles des chauffeurs de l'agence afin de permettre à l'administrateur d'ajouter, modifier, Supprimer ou chercher un chauffeur.
- Gestion des véhicules : Le logiciel supervise le parking de véhicules de l'agence afin de permettre à l'administrateur d'ajouter, modifier ou supprime les informations caractérisant un véhicule.
- Gestion des plans : Le logiciel supervise la gestion des plans de l'agence afin de permettre à l'administrateur construire, modifier, supprimer ou chercher des tournées.

- Gestion des clients: Le logiciel assure la gestion des informations personnelles des clients de l'agence afin de permettre à l'administrateur d'ajouter, modifier, supprimer ou chercher un client.
- Gestion des demandes: Le logiciel supervise la gestion des demandes de l'agence afin de permettre à l'agent d'ajouter une nouvelle demande, modifier, supprimer ou chercher cette dernière.
- Authentification : Le logiciel doit demander au cours de démarrage à l'administrateur ou l'agent de faire l'authentification avant l'accès à l'application pour assurer la confidentialité et l'intégrité des données.

4. Diagramme de cas d'utilisation.

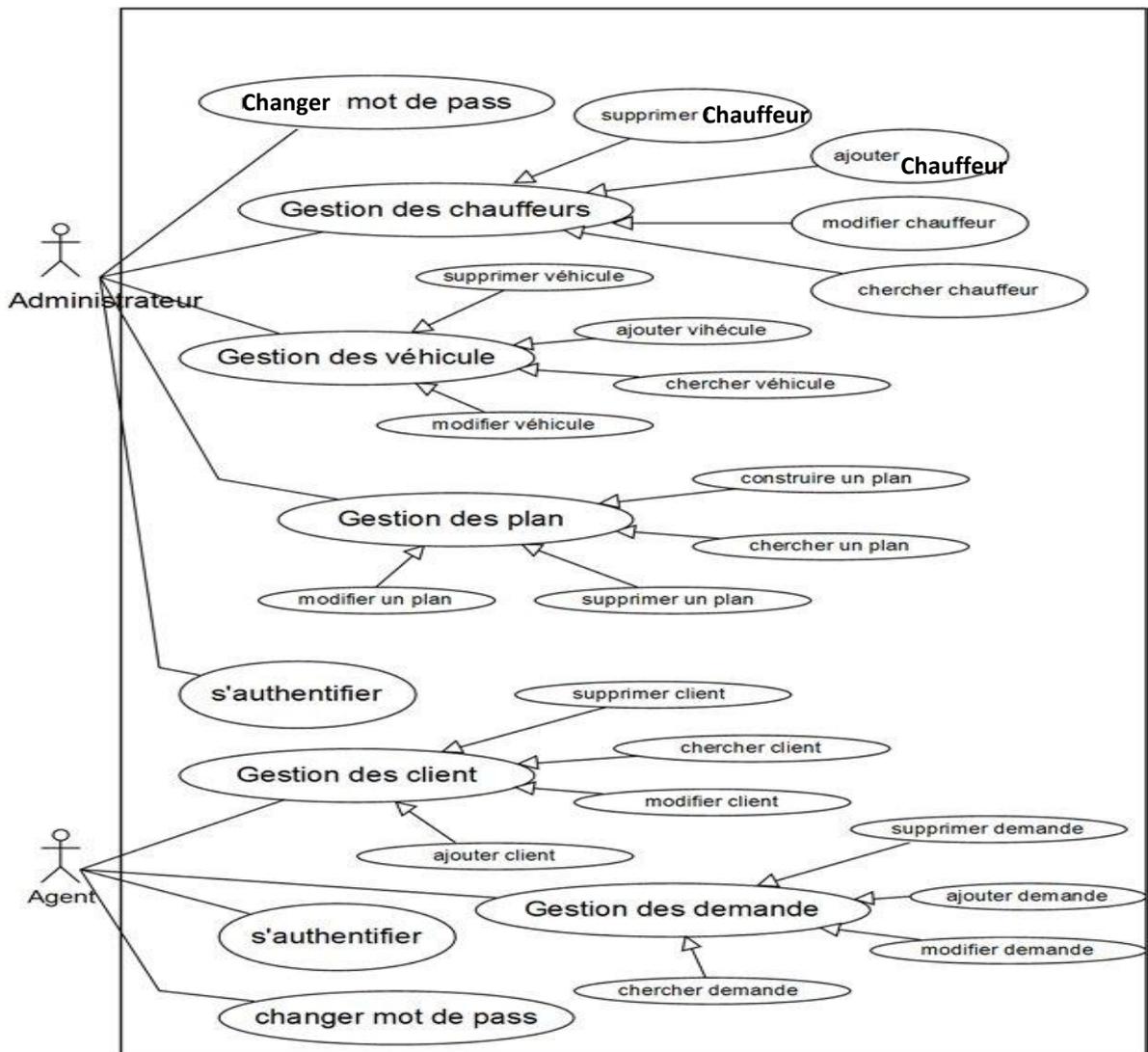


Figure 3.1: diagramme de cas d'utilisation

5. Description textuelle des cas d'utilisations

5.1. Cas d'utilisation « ajouter client »

<i>Acteur :</i>	Agent
<i>But :</i>	ajouter un nouveau client
<i>Pré condition :</i>	agent s'authentifier.
Scénario nominale :	1- agent lance l'ajout d'un nouveau client. 2-le système affiche un formulaire pour saisir les informations d'un client. 3-agent saisie tous les informations demandées par le système et enregistre. 4-Le système vérifie la cohérence et la validité des informations saisies et enregistre l'ajout.
Scénario alternatif	1. Le client existe déjà et/ou le formulaire incorrecte ou incomplet. 2. Retourne à l'étape2.
Scénario exceptionnel :	Néant
Post conditions :	L'ajout est effectué avec succès.

Tableau 3.1 : Fiche descriptive de cas d'utilisation ajouter client.

5.2. Cas d'utilisation « ajouter demande de client »

<i>Acteur :</i>	Agent
<i>But :</i>	ajouter une nouvelle demande
<i>Pré condition :</i>	agent doit être authentifié.
Scénario nominale :	1- agent lance l'ajout d'une nouvelle demande. 2-le système affiche un formulaire pour saisir les informations d'une

	<p>demande.</p> <p>3-agent saisie tous les informations demandées par le système et enregistre la saisie.</p> <p>4-Le système vérifie la cohérence et la validité des informations saisies et enregistre l'ajout.</p>
Scénario alternatif	<p>1. la demande existe déjà et/ou le formulaire incorrecte ou incomplet.</p> <p>2. Retourne à l'étape2.</p>
Scénario exceptionnel :	Néant
Post conditions :	L'ajout est effectué avec succès.

Tableau 3.2 : Fiche descriptive de cas d'utilisation ajouter demande de client.

5.3. Cas d'utilisation «ajouter chauffeurs»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	ajouter un nouveau chauffeur
<i>Pré condition :</i>	Administrateur s'authentifier
Scénario nominale :	<p>1- l'administrateur lance l'ajout d'un nouveau chauffeur.</p> <p>2-le système affiche un formulaire pour saisir les informations d'un chauffeur.</p> <p>3- Administrateur saisie tous les informations demandées par le système et enregistre la saisie.</p> <p>4-Le système vérifie la cohérence et la validité des informations saisies et enregistre l'ajout.</p>
Scénario alternatif	<p>1. Le chauffeur existe déjà et/ou le formulaire incorrecte ou incomplet.</p> <p>2. Retourne à l'étape 2.</p>
Scénario exceptionnel :	Néant
Post conditions :	L'ajout est effectué avec succès.

Tableau 3.3 : Fiche descriptive de cas d'utilisation ajouter chauffeur

5.4. Cas d'utilisation «ajouter véhicule»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	Permettre à administrateur d'ajouter un véhicule
<i>Pré condition :</i>	Administrateur s'authentifier
Scénario nominale :	<ol style="list-style-type: none"> 1- Administrateur lance l'ajout d'un nouveau véhicule 2-le système affiche un formulaire pour saisir les informations d'un véhicule 3- Administrateur saisie les informations demandées par le système et enregistre la saisie. 4-Le système vérifie la cohérence et la validité des informations saisies et enregistre l'ajout.
Scénario alternatif	<ol style="list-style-type: none"> 1. Information invalide ou redondante 2. Le système affiche un message d'erreur 3. Retourne à l'étape 2.
Scénario erreurs :	Néant
Post conditions:	Le véhicule a été créé

Tableau 3.4 : Fiche descriptive de cas d'utilisation ajouter véhicule.

5.5. Cas d'utilisation « modifier client »:

<i>Acteur :</i>	Agent
<i>But :</i>	Modifier les informations d'un client
<i>Pré condition :</i>	Agent s'authentifier
Scénario nominale :	<ol style="list-style-type: none"> 1. agent demande de voir la liste des clients 2-le système affiche la liste de tous les clients

	<p>3-agent sélectionne le client à modifier.</p> <p>4-le système affiche les informations concernant ce client.</p> <p>5- agent fait la modification et valide l'opération.</p> <p>6- le system contrôle et enregistre la modification.</p>
Scénario alternatif	Néant
Scénario exceptionnel :	Néant
Post conditions :	La modification effectuée avec succès.

Tableau 3.5 : Fiche descriptive de cas d'utilisation modifier client.

5.6. Cas d'utilisation « modifier véhicule»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	Modifier les informations d'un véhicule
<i>Pré condition :</i>	administrateur s'authentifier
Scénario nominale :	<p>1. administrateur demande de voir la liste des véhicules</p> <p>2-le système affiche la liste de tous les véhicules</p> <p>3- administrateur sélectionne le véhicule à modifier.</p> <p>4-le système affiche les informations concernant ce véhicule.</p> <p>5- administrateur fait la modification et valide l'opération.</p> <p>6- le system contrôle et enregistre la modification.</p>
Scénario alternatif	Néant
Scénario exceptionnel :	Néant
Post conditions :	La modification effectuée avec succès.

Tableau 3.6: Fiche descriptive de cas d'utilisation modifier véhicule.

5.7. Cas d'utilisation « modifie chauffeur»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	Permet modifie un chauffeur

<i>Pré condition :</i>	administrateur s'authentifier
Scénario nominale :	1. administrateur demande de voir la liste du chauffeur 2-le système affiche la liste du chauffeur 3- administrateur sélectionne le chauffeur qu'à modifier. 4-le système affiche les informations concernant ce chauffeur. 5- administrateur fait la modification et valide l'opération. 6- le system contrôle et enregistre la modification.
Scénario alternatif	Néant
Scénario exceptionnel :	Néant
Post conditions :	La modification effectuée avec succès.

Tableau 3.7 : Fiche descriptive de cas d'utilisation modifie chauffeur

5.8. Cas d'utilisation «modifier demande de client

<i>Acteur :</i>	agent
<i>But :</i>	Permettre la modification d'une demande
<i>Pré condition :</i>	Agent s'authentifier
Scénario nominale :	1. agent demande de voir la liste des demandes 2-le système affiche la liste de tous les demandes 3-agent sélectionne la demande à modifier. 4-le système affiche les informations concernant cette demande. 5- agent fait la modification et valide l'opération. 6- le system contrôle et enregistre la modification.
Scénario alternatif	Néant
Scénario erreur :	Néant
Post conditions :	La modification est effectuée avec succès.

Tableau 3.8 : Fiche descriptive de cas d'utilisation modifier demande de client.

5.9. Cas d'utilisation «modifier le trajet des tournée»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	Permettre la modification d'une tournée

<i>Pré condition :</i>	Administrateur s'authentifier
Scénario nominale :	<ol style="list-style-type: none"> 1. Administrateur demande de voir la liste des trajets des tournées 2-le système affiche la liste de tous les trajets 3- Administrateur sélectionne le trajet qu'il veut modifier. 4-le système affiche les informations concernant ce trajet. 5- Administrateur fait la modification et valide l'opération. 6- le system contrôle et enregistre la modification.
Scénario alternatif	Néant
Scénario exceptionnel :	Néant
Post conditions :	La modification est effectuée avec succès.

Tableau 3.9 : Fiche descriptive de cas d'utilisation modifie le trajet de la tournée.

5.10. Cas d'utilisation « supprimer client »:

<i>Acteur :</i>	Agent
<i>But :</i>	Supprimer un client
<i>Pré condition :</i>	agent s'authentifier.
Scénario nominale :	<ol style="list-style-type: none"> 1. agent demande de voir la liste des clients 2. le système affiche la liste des clients 3. agent sélectionne le client à supprimer. 4. Le système demande une confirmation de l'action 5. agent confirme la demande 7. Le système affiche la nouvelle liste des clients
Scénario alternatif	Néant
Scénario erreur:	<ol style="list-style-type: none"> 7.1. agent décline la confirmation <p>* Le système informe agent que la suppression a été annulée</p>
Post conditions :	Un client est supprimé de la liste des clients

Tableau 3.10 : Fiche descriptive de cas d'utilisation supprimer client.

5.11. Cas d'utilisation «supprimer demande de client»:

<i>Acteur :</i>	agent
<i>But :</i>	Permet de supprimer une demande d'un client
<i>Pré condition :</i>	Agent s'authentifier
<i>Scénario nominale :</i>	<ol style="list-style-type: none"> 1. agent demande de voir la liste des demandes 2. le système affiche la liste des demandes 3. agent sélectionne demande à supprimer. 4. agent choisit la fonction supprimer la demande 5. Le système demande une confirmation de l'action 6. agent confirme la demande 7. Le système affiche la nouvelle liste des demandes
<i>Scénario alternatif</i>	Néant
<i>Scénario erreur :</i>	<ol style="list-style-type: none"> 7.1 agent décline la confirmation <p>* Le système informe agent que la suppression a été annulée</p>
<i>Post conditions :</i>	Une demande est supprimée de la liste des demandes

Tableau 3.11 : Fiche descriptive de cas d'utilisation supprimer demande de client.

5.12. Cas d'utilisation «supprimer plan»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	Permettre la suppression d'un plan
<i>Pré condition :</i>	Administrateur s'authentifier
<i>Scénario nominale :</i>	<ol style="list-style-type: none"> 1. Administrateur demande de voir la liste des plans 2. le système affiche la liste des plans 3. Administrateur sélectionne le plan qu'il veut supprimer. 4. Administrateur choisit la fonction de supprimer le plan 5. Le système demande une confirmation de l'action 6. Administrateur confirme la demande 7. Le système affiche la nouvelle liste des plans
<i>Scénario alternatif</i>	<ol style="list-style-type: none"> 7.1 l'administrateur décline la confirmation <p>* Le système informe l'administrateur que la suppression a été annulée</p>

Scénario exceptionnel :	Néant
Post conditions :	Un plan est supprimé de la liste des plans

Tableau 3.12 : Fiche descriptive de cas d'utilisation supprimé plan

5.13. Cas d'utilisation «supprimer chauffeurs»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	supprimer un chauffeur
<i>Pré condition :</i>	Administrateur s'authentifier
Scénario nominale :	<ol style="list-style-type: none"> 1. Administrateur demande de voir la liste des chauffeurs. 2. le système affiche la liste des chauffeurs 3. Administrateur sélectionne le chauffeur qu'il veut supprimer. 4. Administrateur choisit la fonction de supprimer le chauffeur 5. Le système demande une confirmation de l'action 6. Administrateur confirme la demande 7. Le système affiche la nouvelle liste des chauffeurs
Scénario alternatif	Néant
Scénario erreur	<p>7.1 l'administrateur décline la confirmation</p> <p>* Le système informe l'administrateur que la suppression a été annulée</p>
Post conditions:	Un chauffeur est supprimé de la liste des chauffeurs

Tableau 3.13 : Fiche descriptive de cas d'utilisation supprimer chauffeur

5.14. Cas d'utilisation «supprimer véhicule »:

<i>Acteur :</i>	Administrateur
<i>But :</i>	Permettre la suppression d'un véhicule
<i>Pré condition :</i>	Administrateur est s'authentifier
Scénario nominale :	<ol style="list-style-type: none"> 1. Administrateur demande de voir la liste des véhicules 2. le système affiche la liste des véhicules 3. Administrateur sélectionne les véhicules qu'il veut supprimer. 4. Administrateur choisit la fonction de supprimer le véhicule

	5. Le système demande une confirmation de l'action 6. l'administrateur confirme la demande 7. Le système affiche la nouvelle liste des véhicules
Scénario alternatif	Néant
Scénario erreurs :	7. 1 l'administrateur décline la confirmation *Le système informe l'administrateur que la suppression a été annulée
Post conditions:	Le véhicule est supprimé

Tableau 3.14 : Fiche descriptive de cas d'utilisation supprimer véhicule.

5.15. Cas d'utilisation « chercher client »:

<i>Acteur :</i>	Agent
<i>But :</i>	chercher un client
<i>Pré condition :</i>	agente s'authentifier
Scénario nominale :	1- agent lance la recherche d'un client. 2. Le système affiche la page de recherche 3. agent choisie le critère de recherche. 4. Le système affiche le résultat de recherche selon le critère de la recherche.
Scénario alternatif	Néant
Scénario exceptionnel :	Néant
Post conditions :	La recherche effectuée avec succès.

Tableau 3.15 : Fiche descriptive de cas d'utilisation chercher client.

5.16. Cas d'utilisation « chercher demande »:

<i>Acteur :</i>	Agent
<i>But :</i>	Recherche une demande
<i>Pré condition :</i>	agent s'authentifier
Scénario nominale :	1- agent lance la recherche d'une demande. 2. Le système affiche la page de recherche

	3. agent choisie le critère de recherche. 4. Le système affiche le résultat de recherche selon le critère de la recherche.
Scénario alternatif	Néant
Scénario exceptionnel :	Néant
Post conditions :	La recherche effectuée avec succès.

Tableau 3.16 : Fiche descriptive de cas d'utilisation chercher demande.

5.17. Cas d'utilisation « chercher chauffeur»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	chercher un chauffeur
<i>Pré condition :</i>	Administrateur s'authentifier
Scénario nominale :	1- Administrateur lance la recherche d'un chauffeur. 2. Le système affiche la page de recherche 3. Administrateur choisie le critère de recherche. 4. Le système affiche le résultat de recherche selon le critère de la recherche.
Scénario alternatif	Néant
Scénario exceptionnel :	Néant
Post conditions :	La recherche effectuée avec succès.

Tableau 3.17 : Fiche descriptive de cas d'utilisation chercher chauffeur.

5.18. Cas d'utilisation « chercher véhicule»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	Recherche un véhicule
<i>Pré condition :</i>	Administrateur est s'authentifier
Scénario nominale :	<ol style="list-style-type: none"> 1- Administrateur lance la recherche d'un véhicule. 2. Le système affiche la page de recherche 3. Administrateur choisie le critère de recherche. 4. Le système affiche le résultat de recherche selon le critère de la recherche.
Scénario alternatif	Néant
Scénario exceptionnel :	Néant
Post conditions :	La recherche effectuée avec succès.

Tableau 3.18 : Fiche descriptive de cas d'utilisation chercher véhicule.

5.19. Cas d'utilisation « chercher plan»:

<i>Acteur :</i>	Administrateur
<i>But :</i>	chercher un plan
<i>Pré condition :</i>	Administrateur s'authentifier
Scénario nominale :	<ol style="list-style-type: none"> 1- Administrateur lance la recherche d'un plan. 2. Le système affiche la page de recherche 3. Administrateur choisie le critère de recherche. 4. Le système affiche le résultat de recherche selon le critère de la recherche.
Scénario alternatif	Néant
Scénario exceptionnel :	Néant
Post conditions :	La recherche effectuée avec succès.

Tableau 3.19 : Fiche descriptive de cas d'utilisation chercher plan

5.20. Cas d'utilisation « construire un plan »:

<i>Acteur :</i>	Administrateur
<i>But :</i>	Permettre à l'administrateur de construire les tournées pour chaque véhicule.
<i>Pré condition :</i>	Administrateur s'authentifier
Scénario nominale :	<ol style="list-style-type: none"> 1- Administrateur lance la construction du plan. 2. Le système affiche un formulaire pour saisir l'information nécessaire à la construction des tournées. 3. Administrateur choisie les informations nécessaire. 4. Le système vérifier la cohérence des informations saisies et construire le plan et afficher ce dernier à l'administrateur. 5. l'administrateur control le plan afficher et demande l'enregistrement de ce dernier
Scénario alternatif	Les informations sont incomplètes ou erronées
Scénario exceptionnel :	Néant
Post conditions :	La construction du plan est effectuée avec succès.

Tableau 3.20 : Fiche descriptive de cas d'utilisation construire un plan.

5.21. Cas d'utilisation « authentication »:

<i>Acteur :</i>	Administrateur et agent
<i>But :</i>	vérifier l'autorisation d'accès au system
<i>Pré condition :</i>	L'acteur possède un compte.
Scénario nominale :	<ol style="list-style-type: none"> 1- L'acteur lance l'application. 2- le system affiche le formulaire pour saisir les informations d'accès (mot de passe ; nom d'utilisateur). 3- L'acteur saisis le mot de passe et nom d'utilisateur. 4- le system vérifie la validation du mot de passe et nom d'utilisateur et ouvre la page d'accueil d'application

Scénario alternatif	4. 1- le mot de passe et \ou nom d'utilisateur erronés *le system affiche un message d'erreur Reprendre à l'étape 2
Scénario exceptionnel :	L'acteur utilise les max tentative d'accès à l'application donc elle doit être fermée
Post conditions :	Le menu de l'application est ouvert.

Tableau 3.21 : Fiche descriptive de cas d'utilisation ajouter authentification.

5.22. Cas d'utilisation «changer mot de passe»:

<i>Acteur :</i>	Administrateur et agent
<i>But :</i>	changer le mot de passe
<i>Pré condition :</i>	L'acteur s'authentifier
Scénario nominale :	<ol style="list-style-type: none"> 1. L'acteur lance l'application pour modifier le mot de passe. 2. le système affiche un formulaire pour saisir les informations concernant le nouveau mot de passe 3. L'acteur saisie l'ancien mot de passe et remplir les champs de nouveau mot de passe. 4. Le système vérifie la cohérence et la validité des informations saisies et enregistre le changement
Scénario alternatif	les informations sont incomplètes ou erronées.
Scénario exceptionnel :	Le mot de passe et le nom d'utilisateur ancien qui saisit par l'acteur et erroné.
Post conditions :	Le cas d'utilisation effectué avec succès.

Tableau 3.22 : Fiche descriptive de cas d'utilisation changer mot de passe.

6. Diagrammes de séquence

6.1 Cas d'utilisation «ajouter client»:

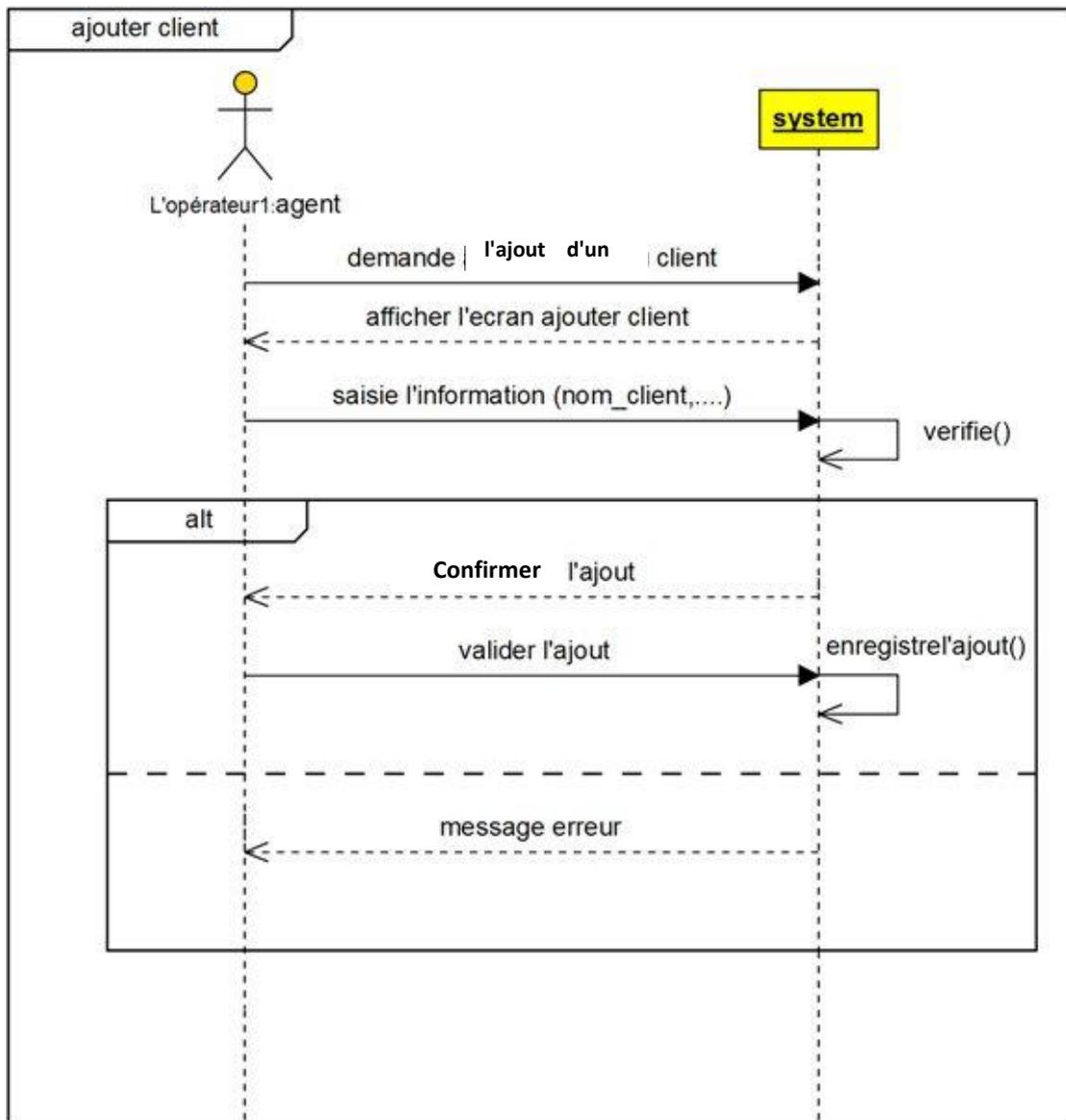


Figure 3.2 : diagramme de séquence ajouter client

6.2 Cas d'utilisation «ajouter demande d'un client»:

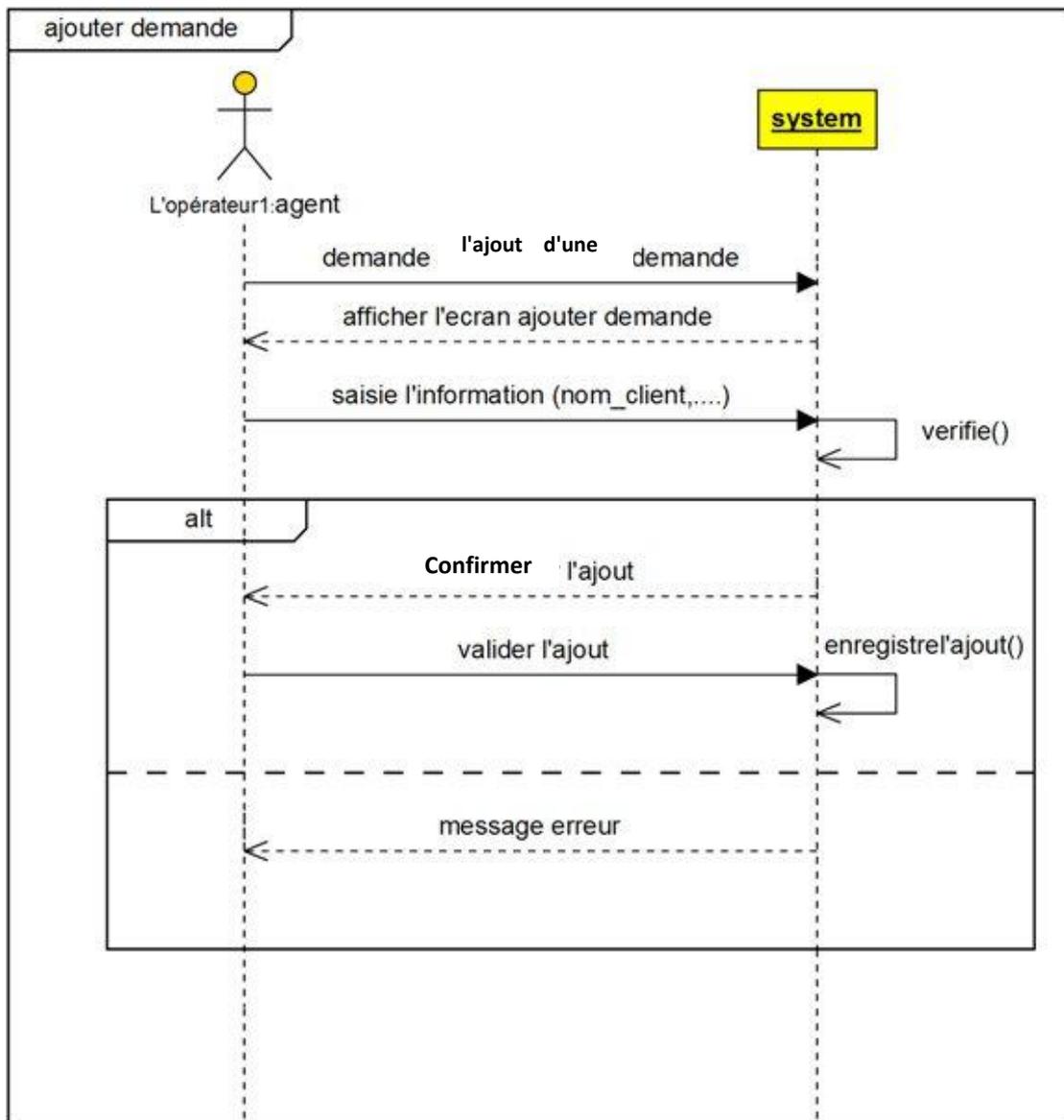


Figure 3.3 : diagramme de séquence ajouter demande

6.3 Cas d'utilisation «ajouter chauffeur»:

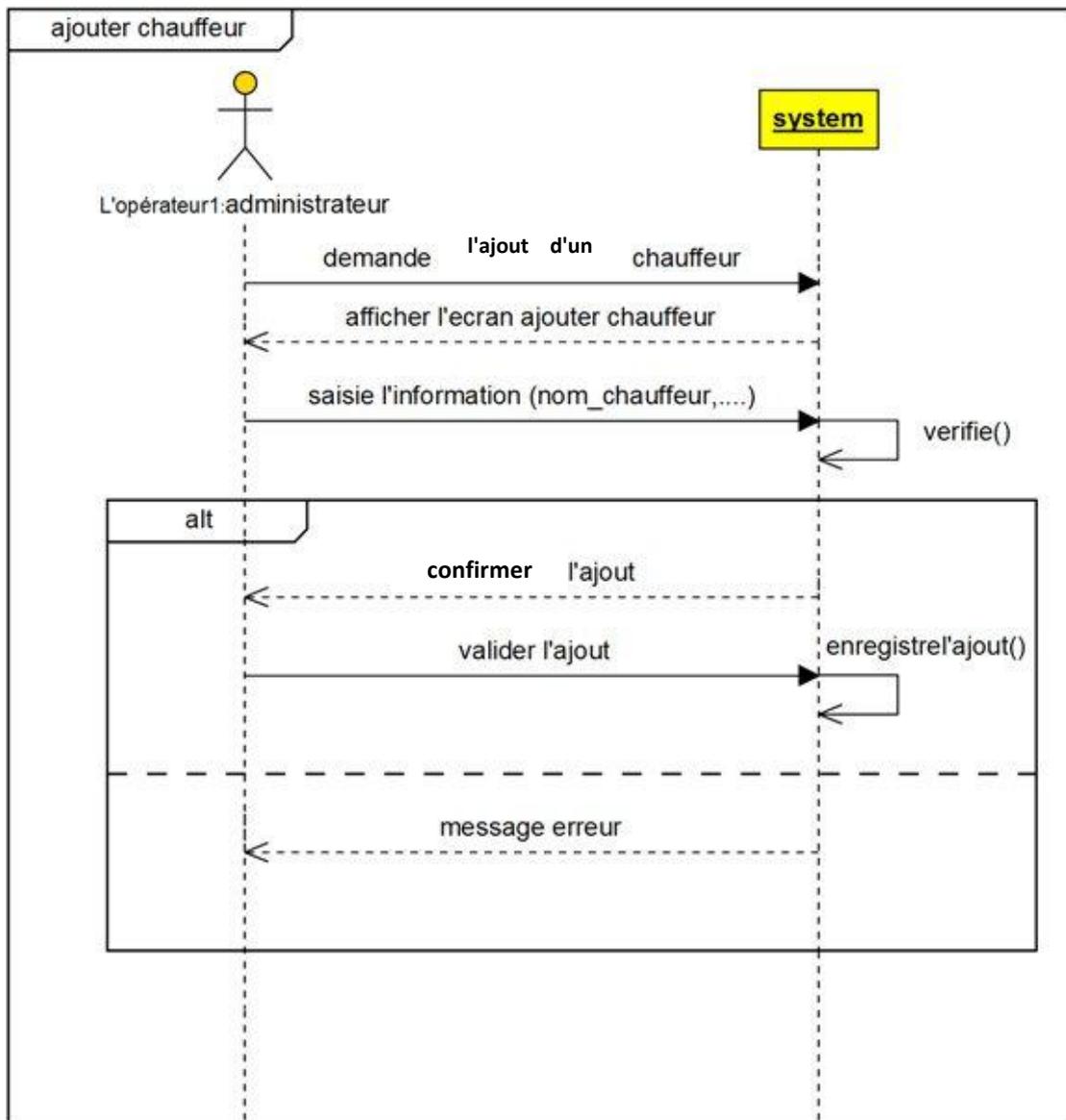


Figure 3.4 : diagramme de séquence ajouter chauffeur

6.4 Cas d'utilisation «ajouter véhicule»:

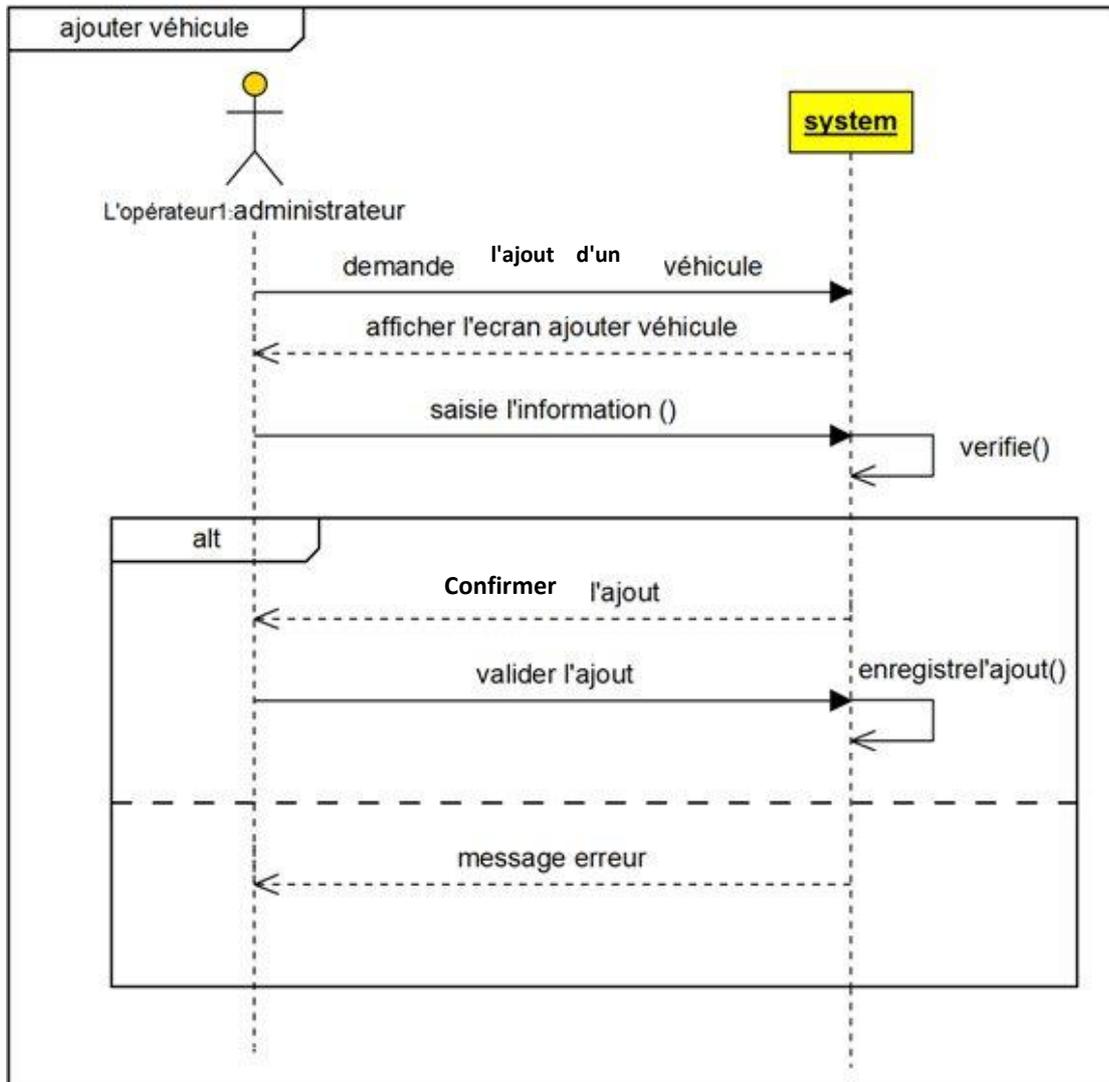


Figure 3.5 : diagramme de séquence ajouter véhicule

6.5 Cas d'utilisation «modifier client »:

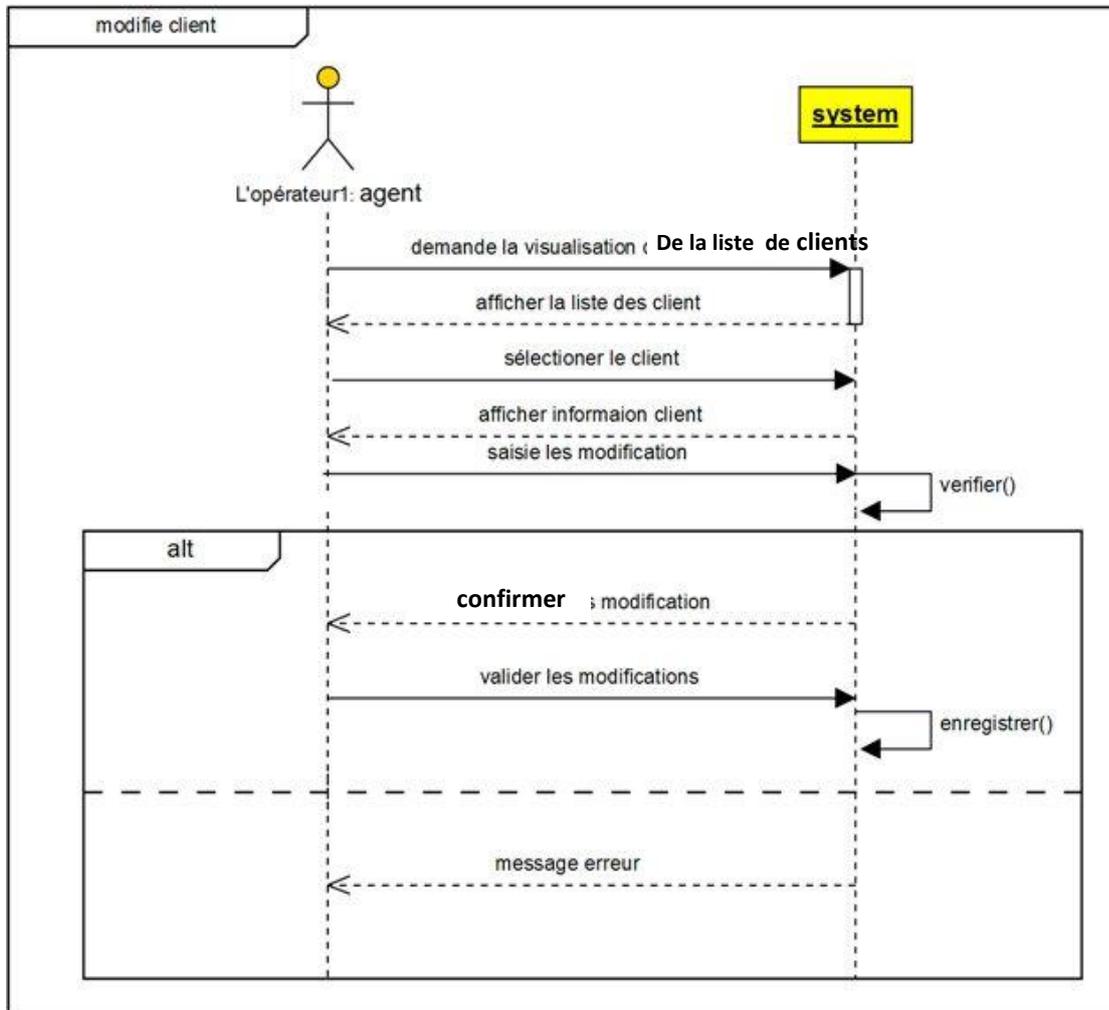


Figure 3.6 : diagramme de séquence modifier un client

6.6 Cas d'utilisation «modifier véhicule»:

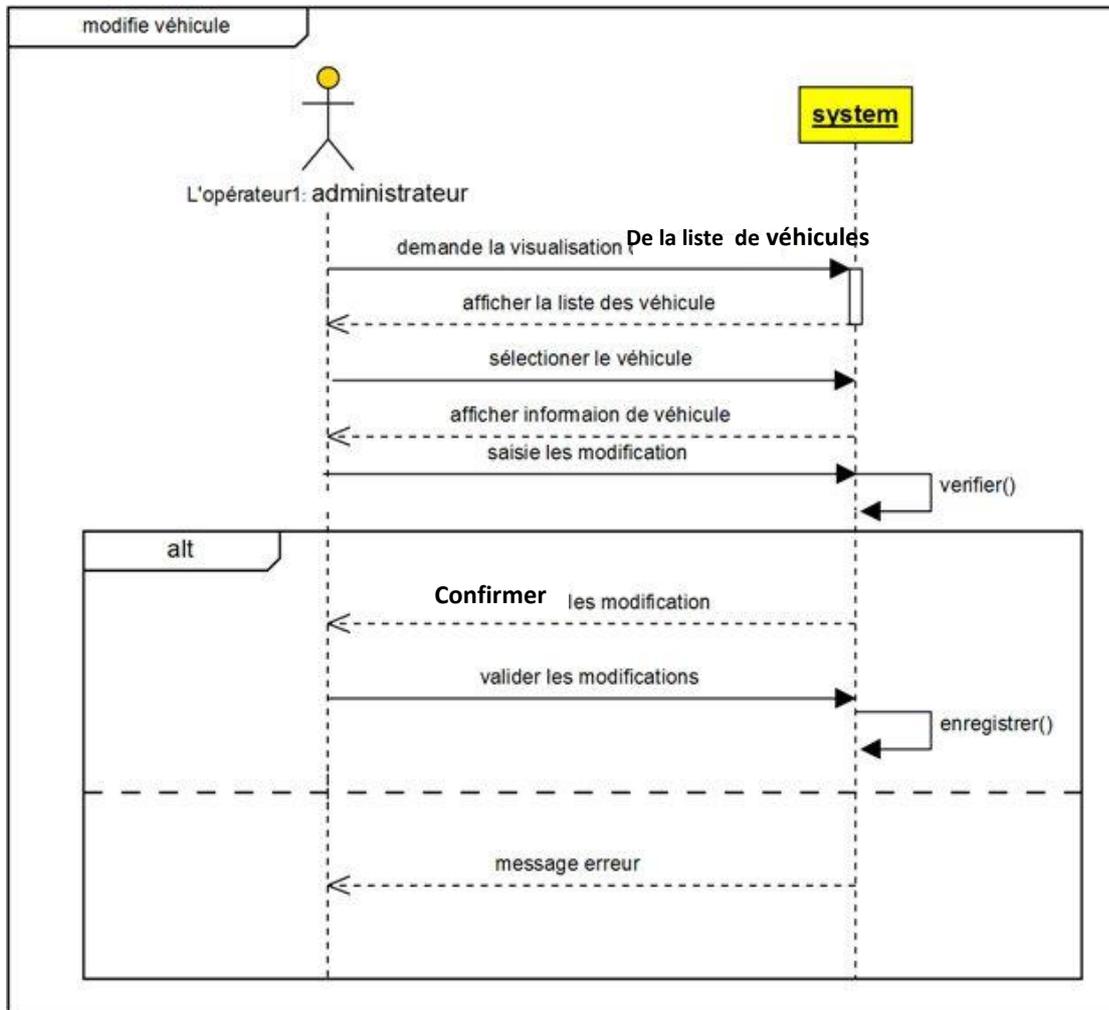


Figure 3.7 : diagramme de séquence modifier véhicule

6.7 Cas d'utilisation «modifier chauffeur»:

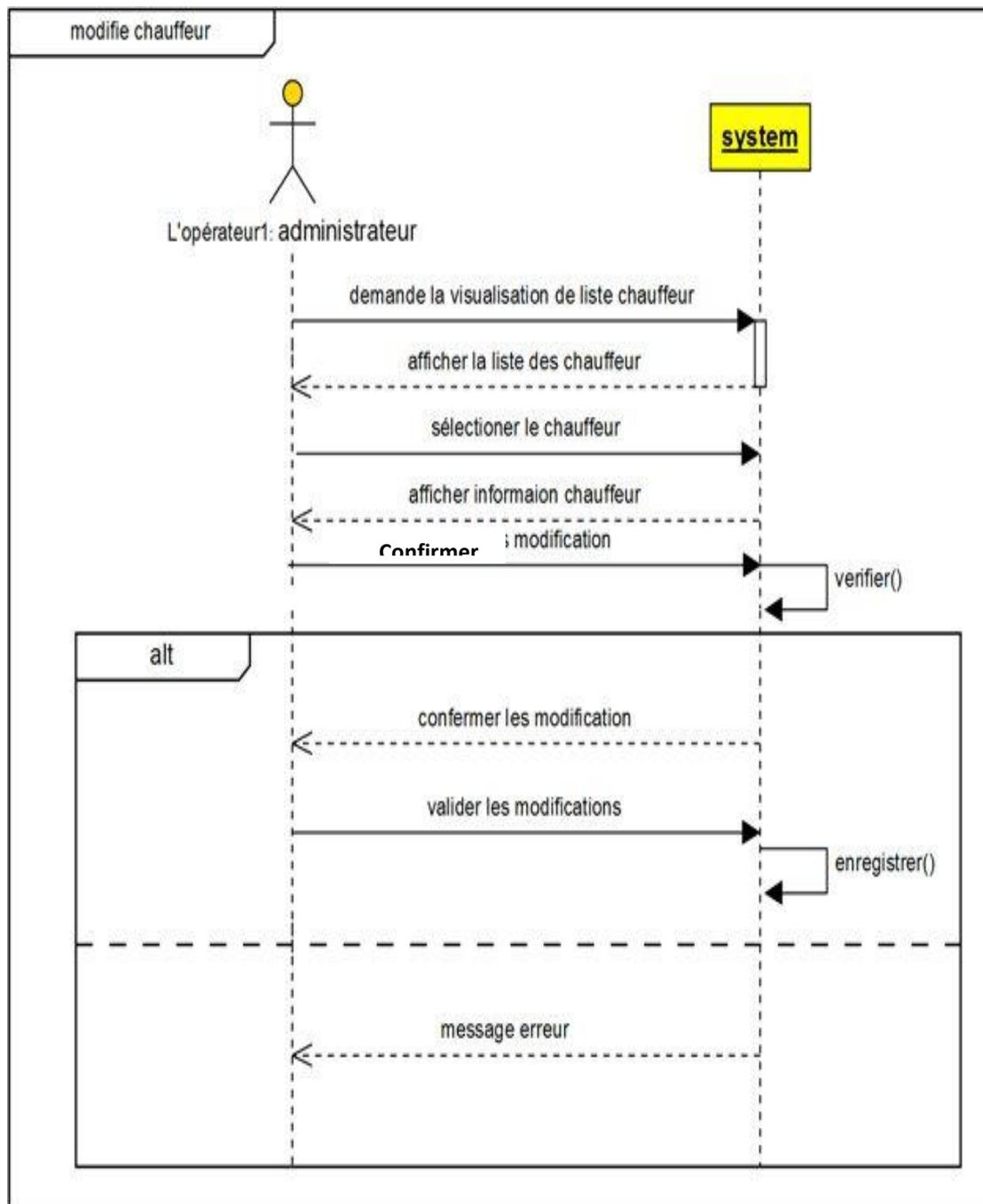


Figure 3.8 : diagramme de séquence modifie chauffeur

6.8 Cas d'utilisation «modifier demande »:

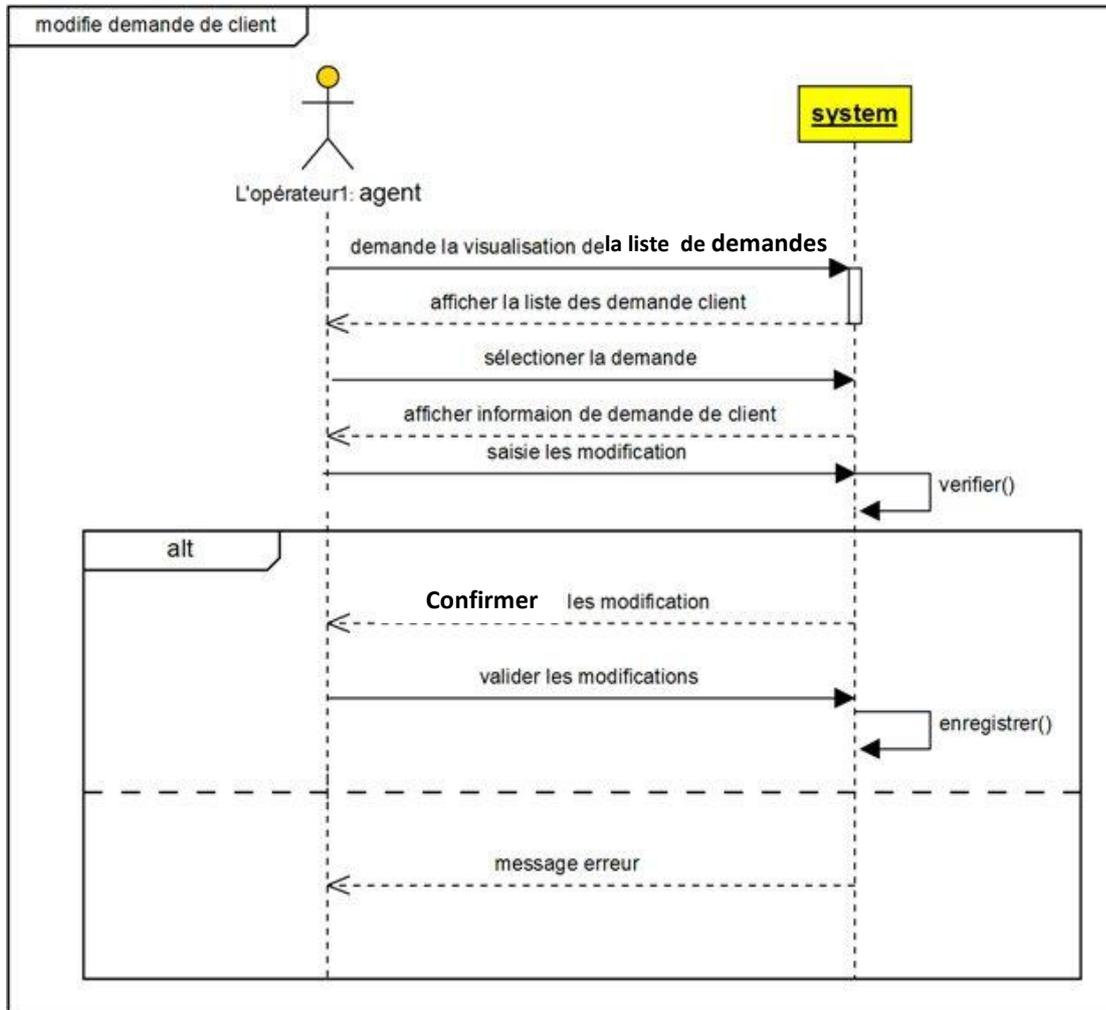


Figure 3.9 : diagramme de séquence modifier demande de client

6.9 Cas d'utilisation «modifier le trajet de tournée»:

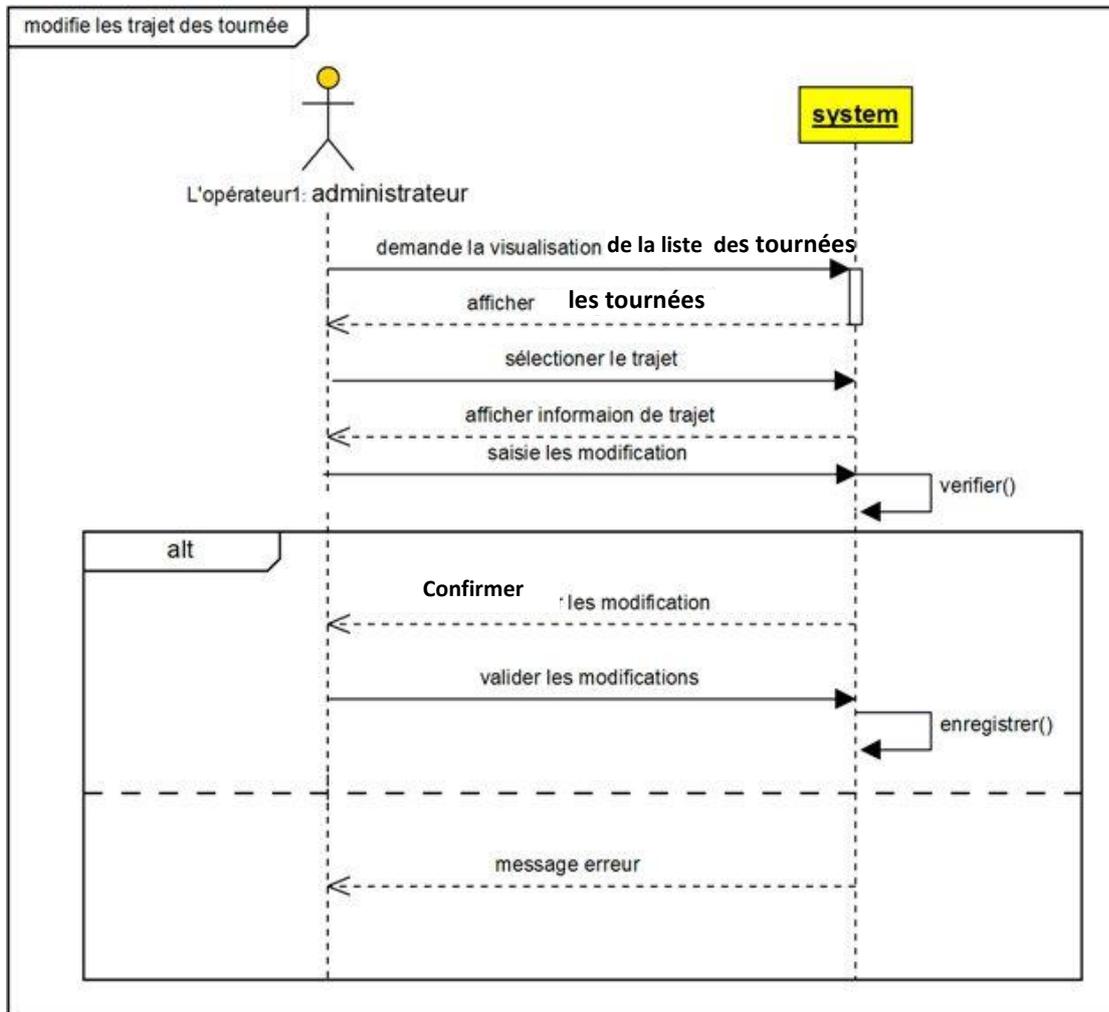


Figure 3.10 : diagramme de séquence modifie le trajet de tournée

6.10 Cas d'utilisation «supprimer client »:

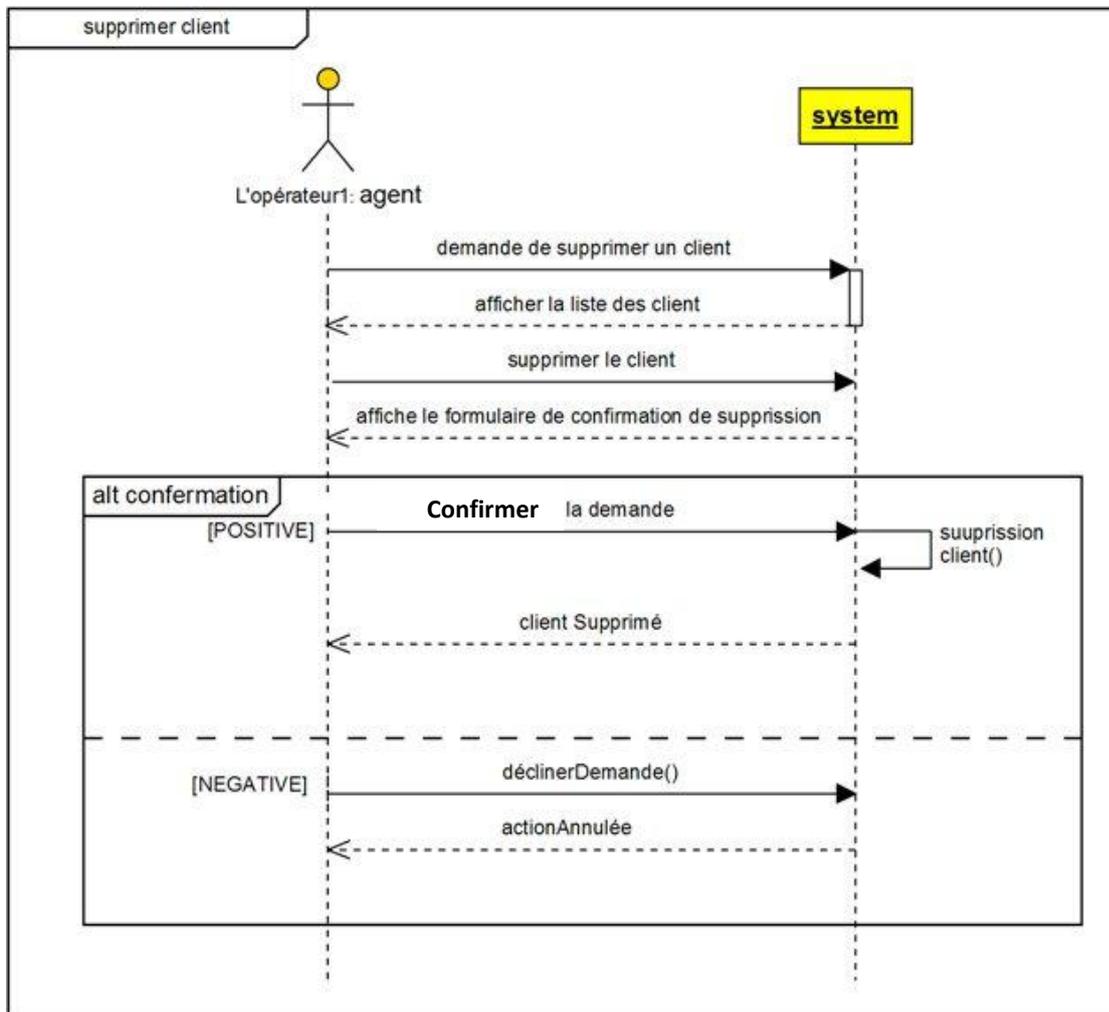


Figure 3.11 : diagramme de séquence supprimé client

6.11 Cas d'utilisation «supprimer demande de client»:

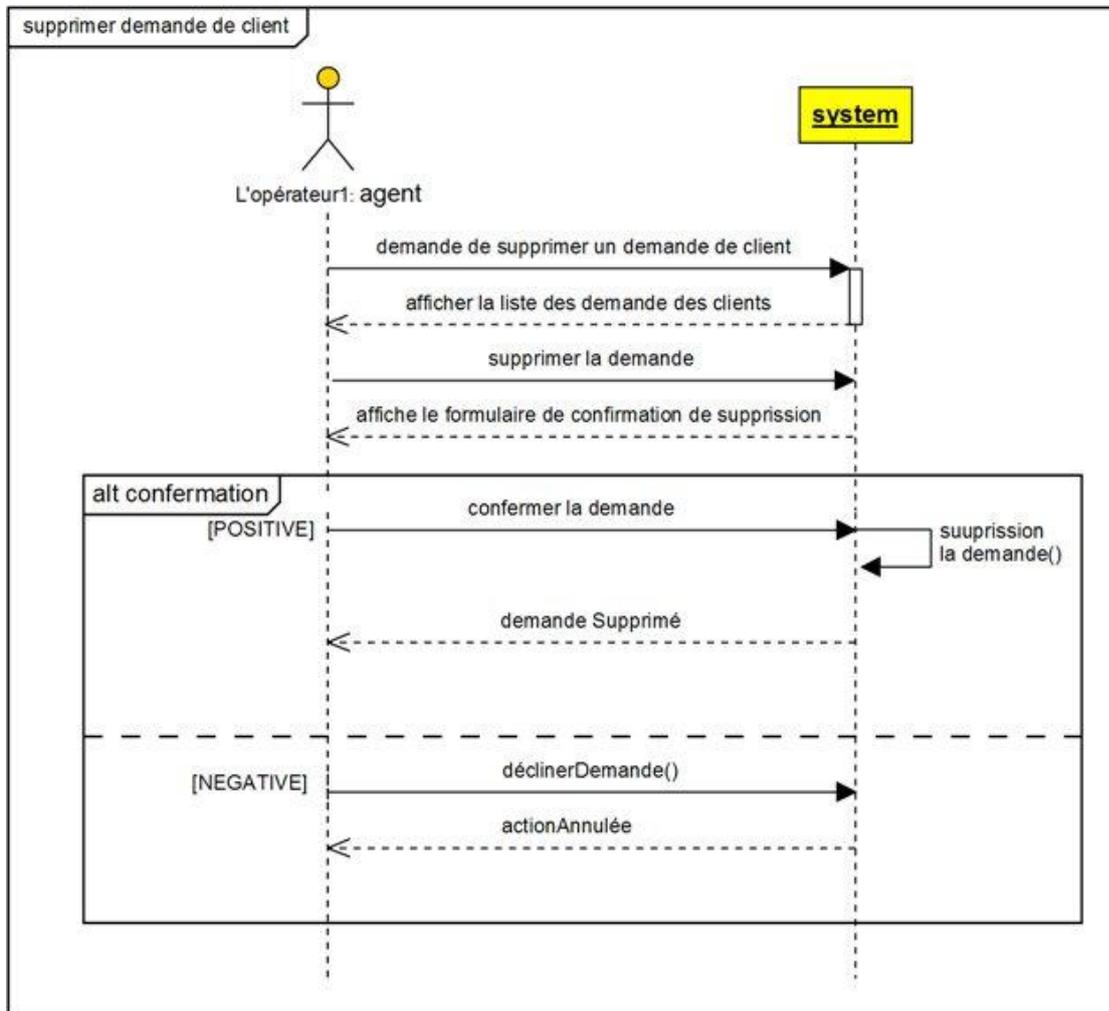


Figure 3.12 : diagramme de séquence supprimer demande de client

6.12 Cas d'utilisation «supprimer plan »:

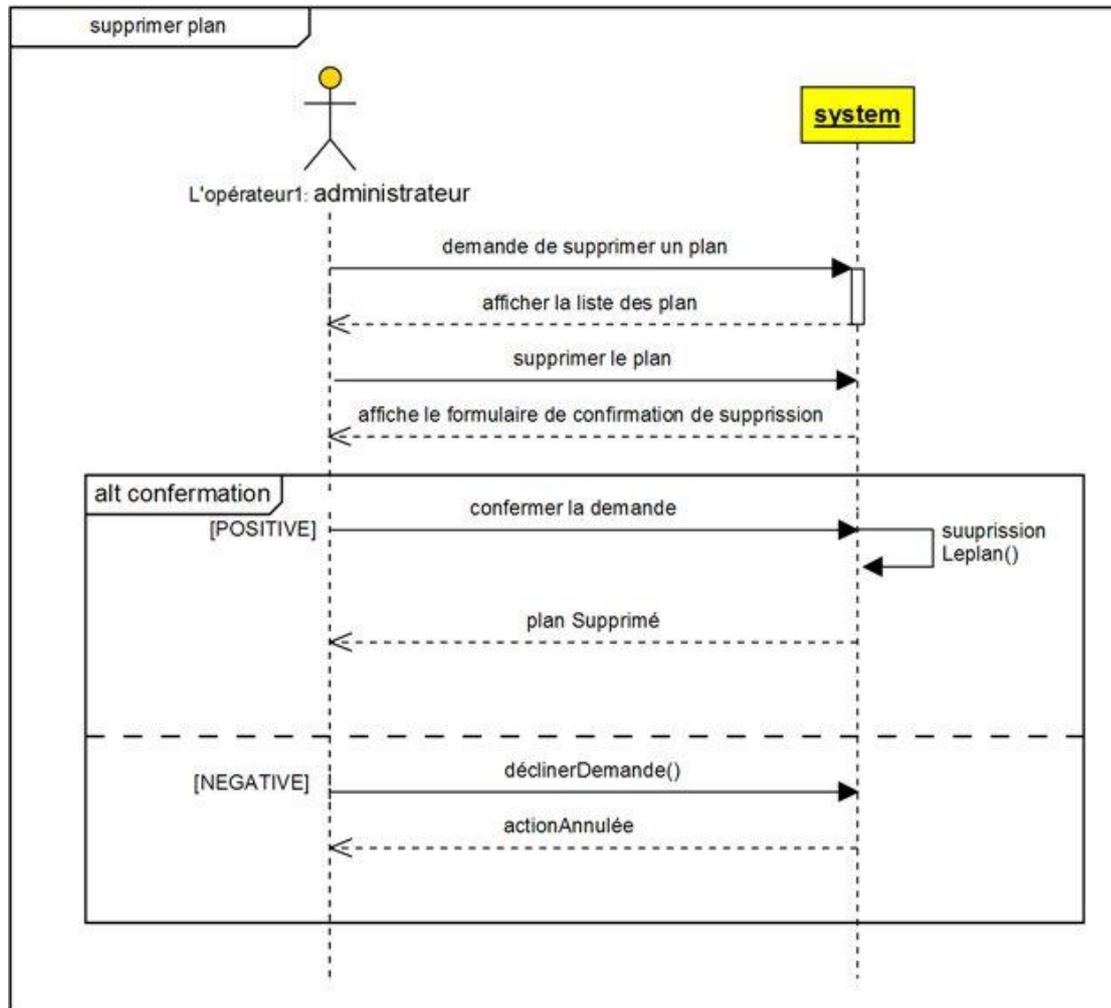


Figure 3.13: diagramme de séquence supprimé plan

6.13 Cas d'utilisation «supprimer chauffeur»:

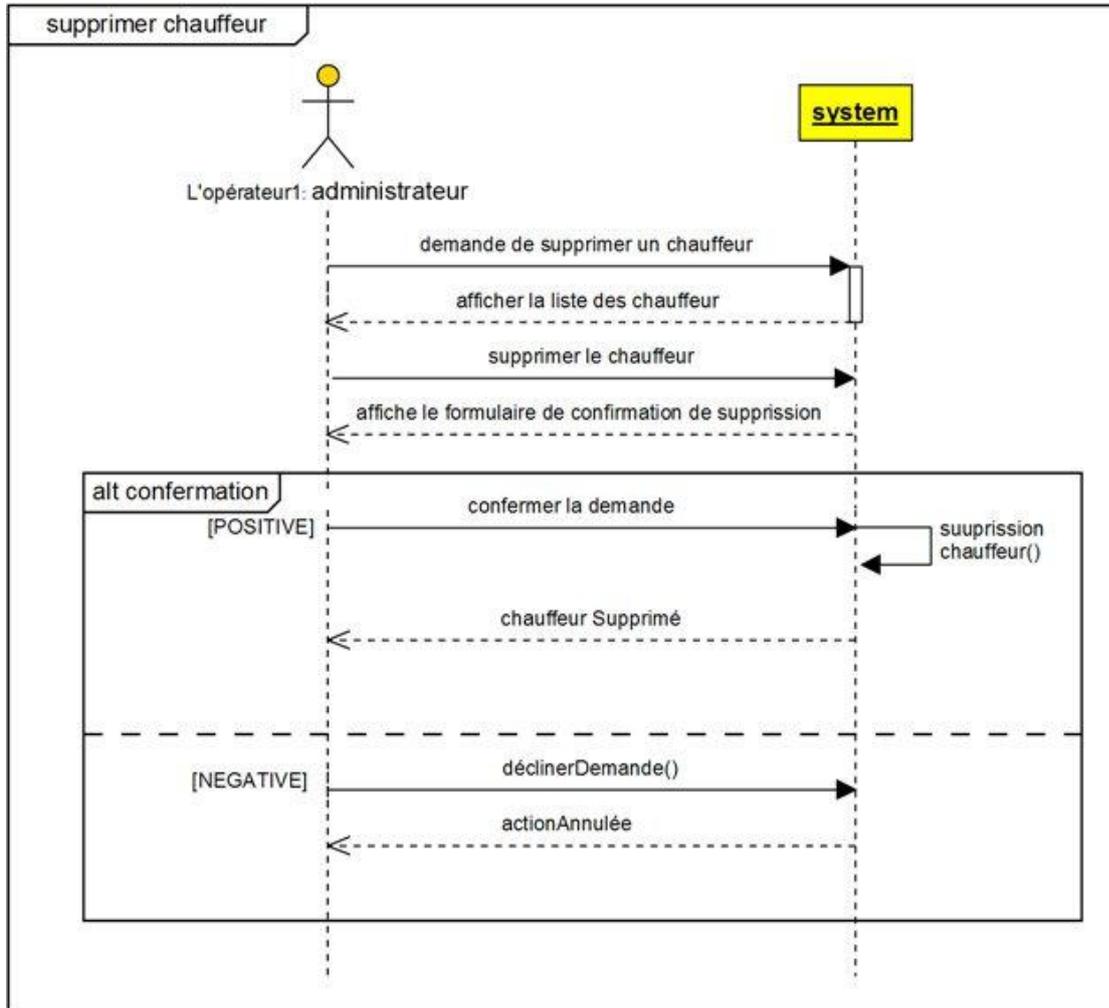


Figure 3.14 : diagramme de séquence supprimé chauffeur

6.14 Cas d'utilisation «supprimer véhicule»:

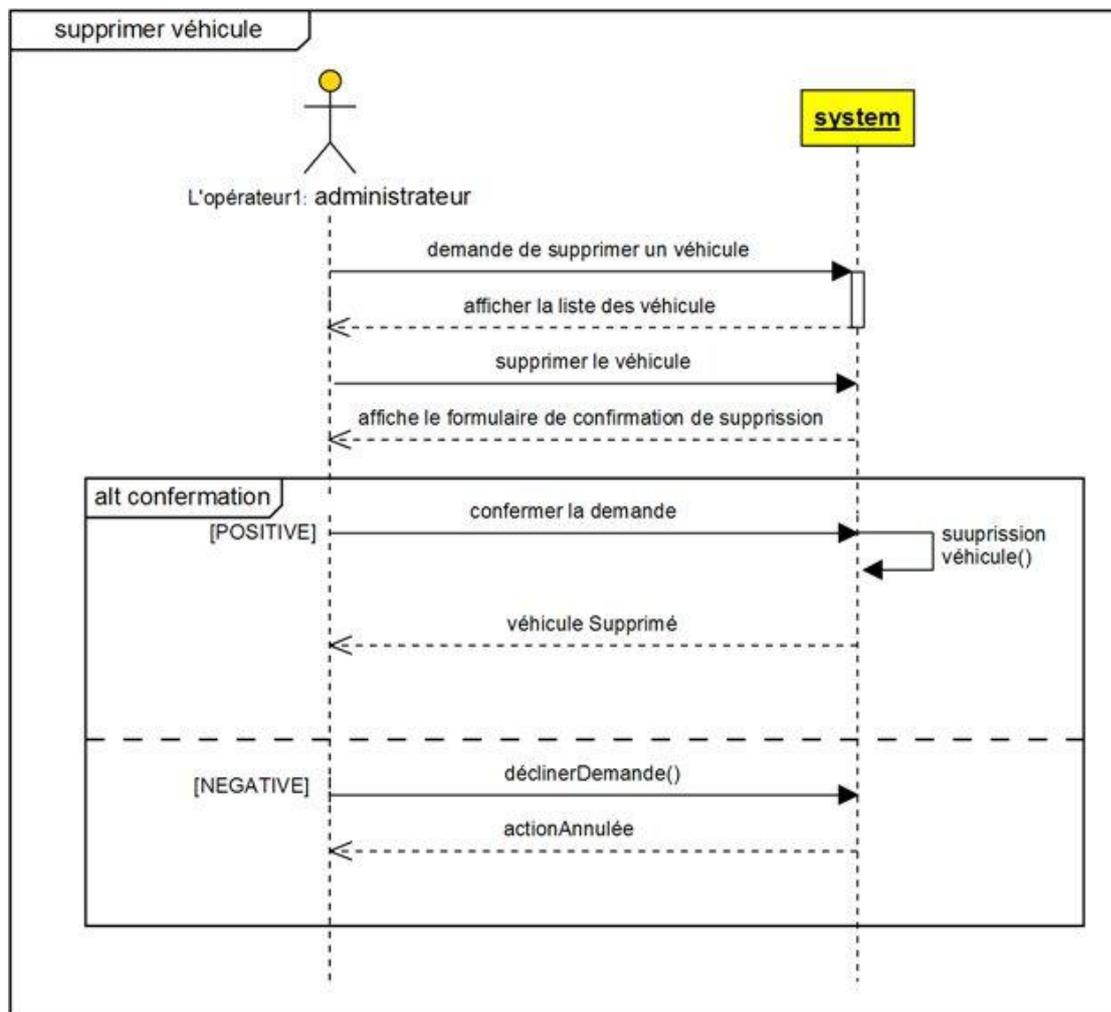


Figure 3.15 : diagramme de séquence supprimer véhicule

6.15 Cas d'utilisation «cherche client»:

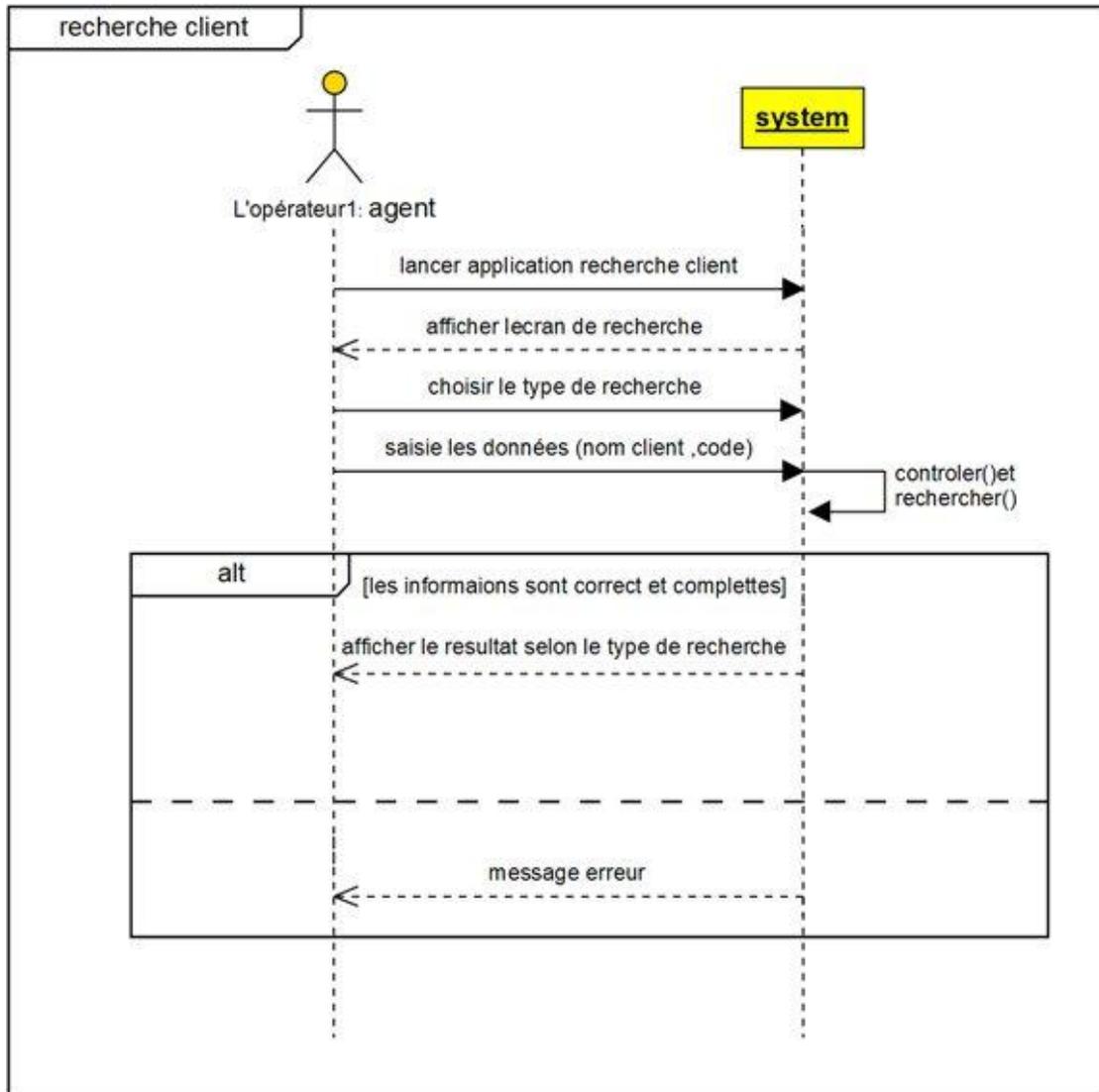


Figure 3.16 : diagramme de séquence cherche client

6.16 Cas d'utilisation «cherche la demande de client »:

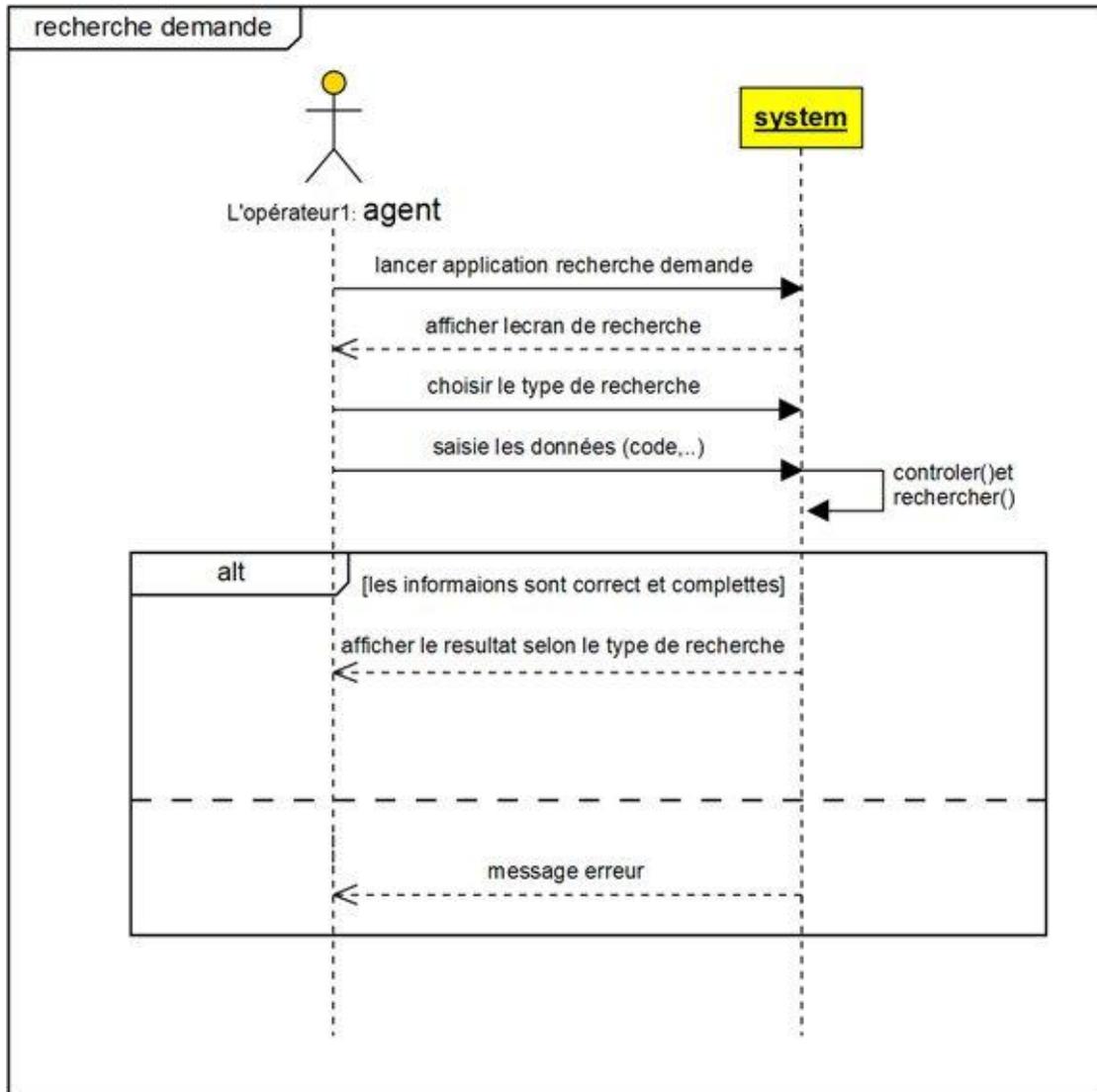


Figure 3.17 : diagramme de séquence cherche demande de client

6.17 Cas d'utilisation «cherche chauffeur»:

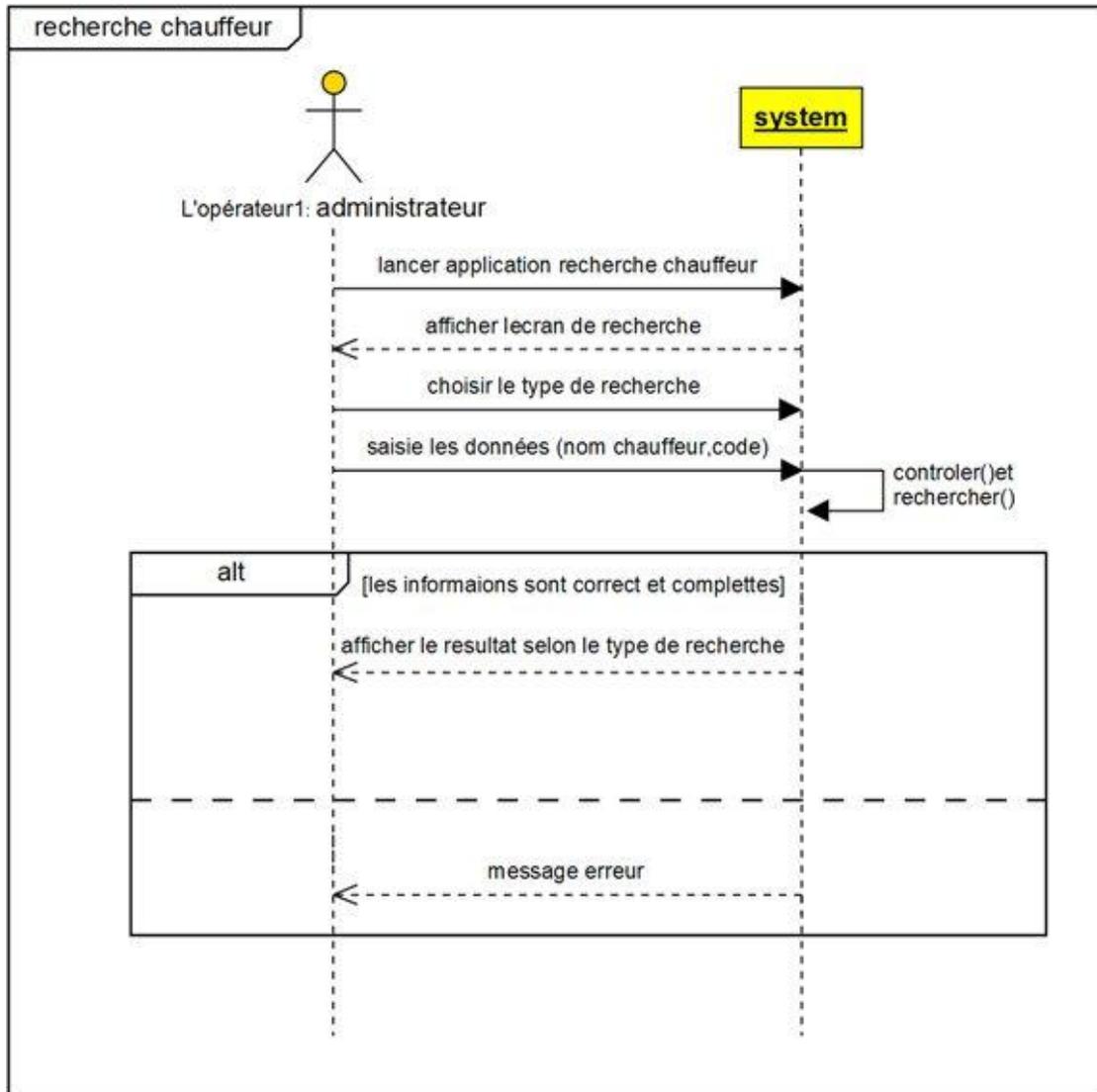


Figure 3.18 : diagramme de séquence cherche chauffeur

6.18 Cas d'utilisation «cherche véhicule»:

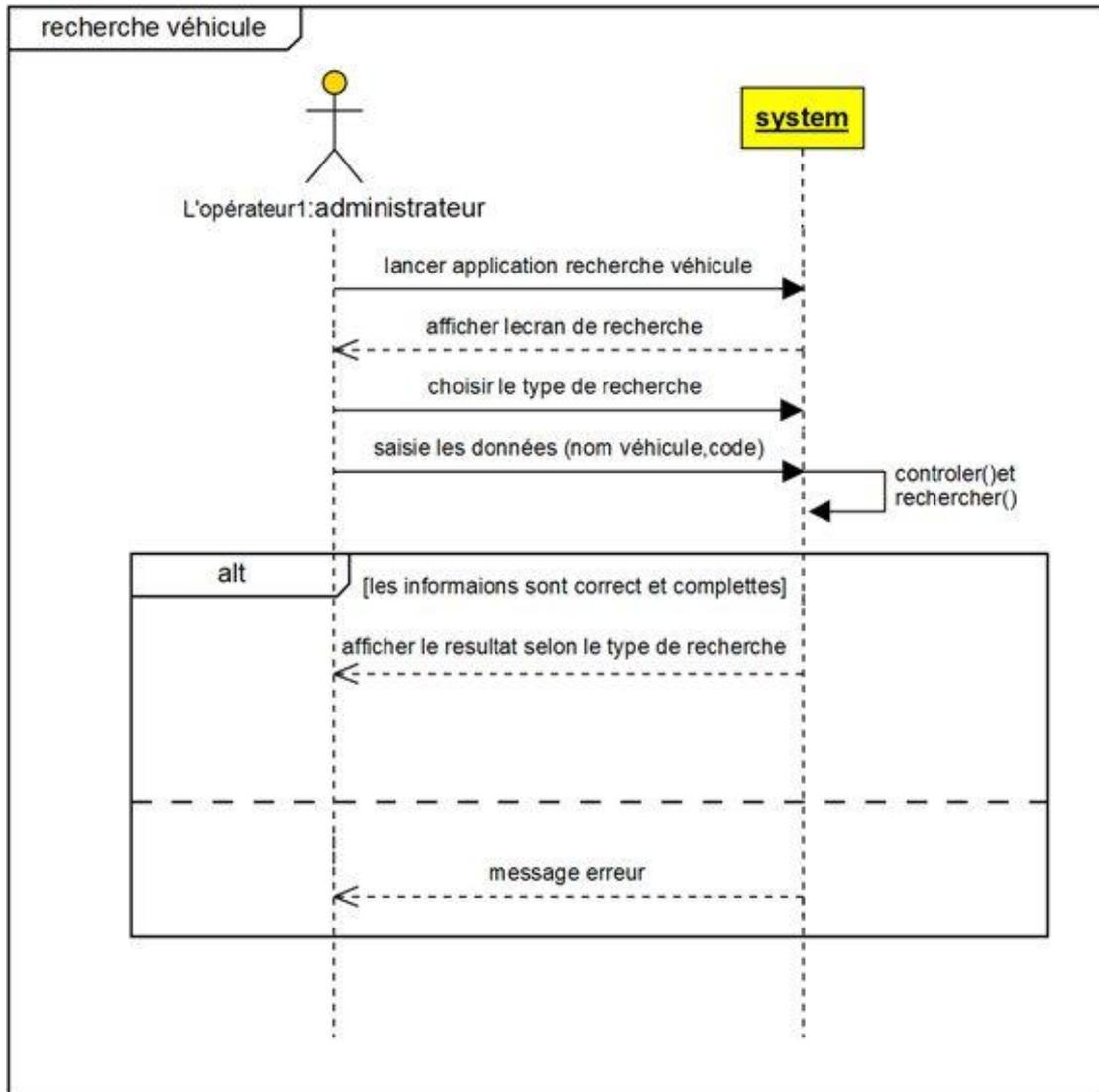


Figure 3.19 : diagramme de séquence chercher véhicule

6.19 Cas d'utilisation «chercher plan»:

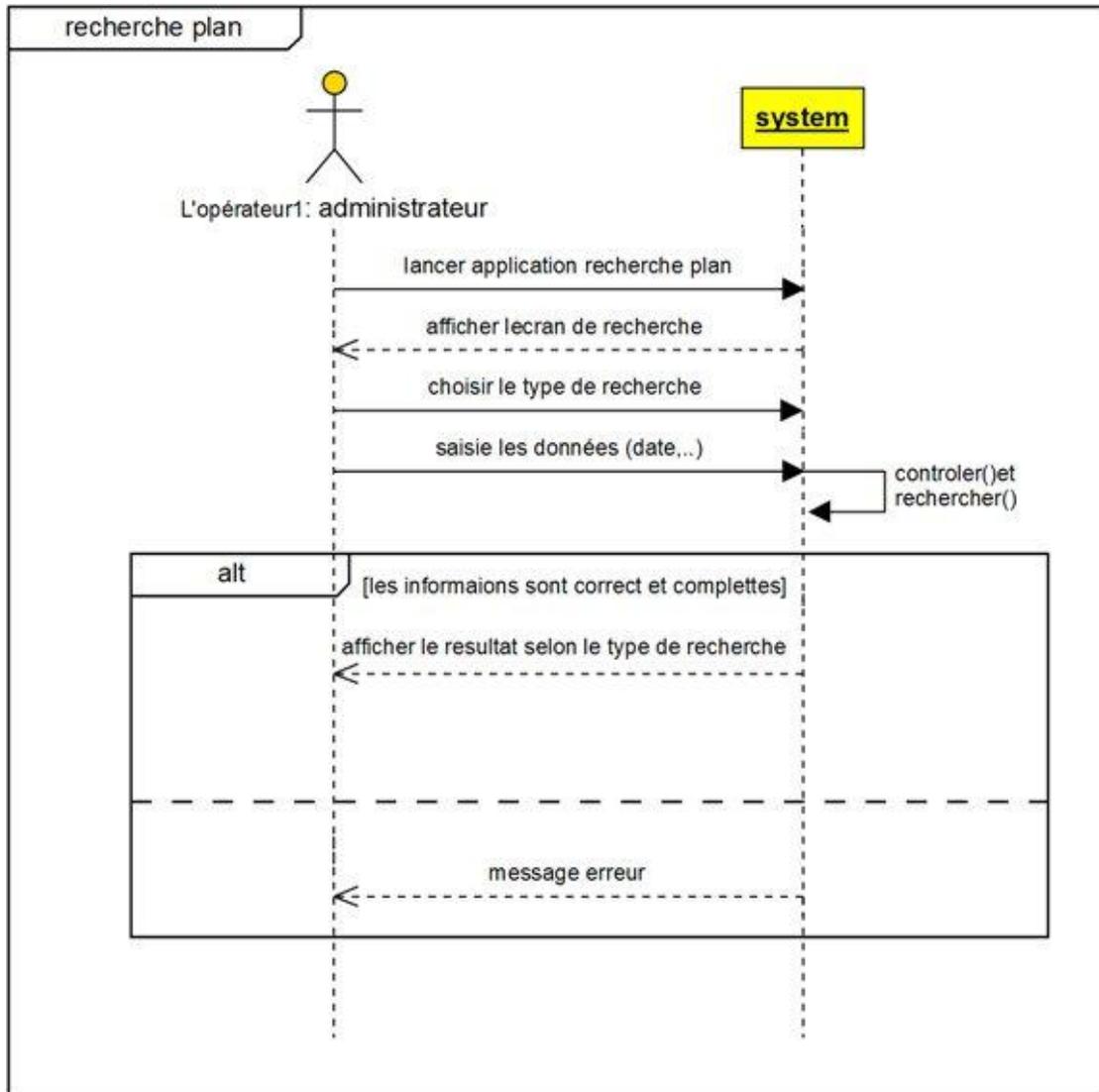


Figure 3.20 : diagramme de séquence cherche plan

6.20 Cas d'utilisation «construire un plan»:

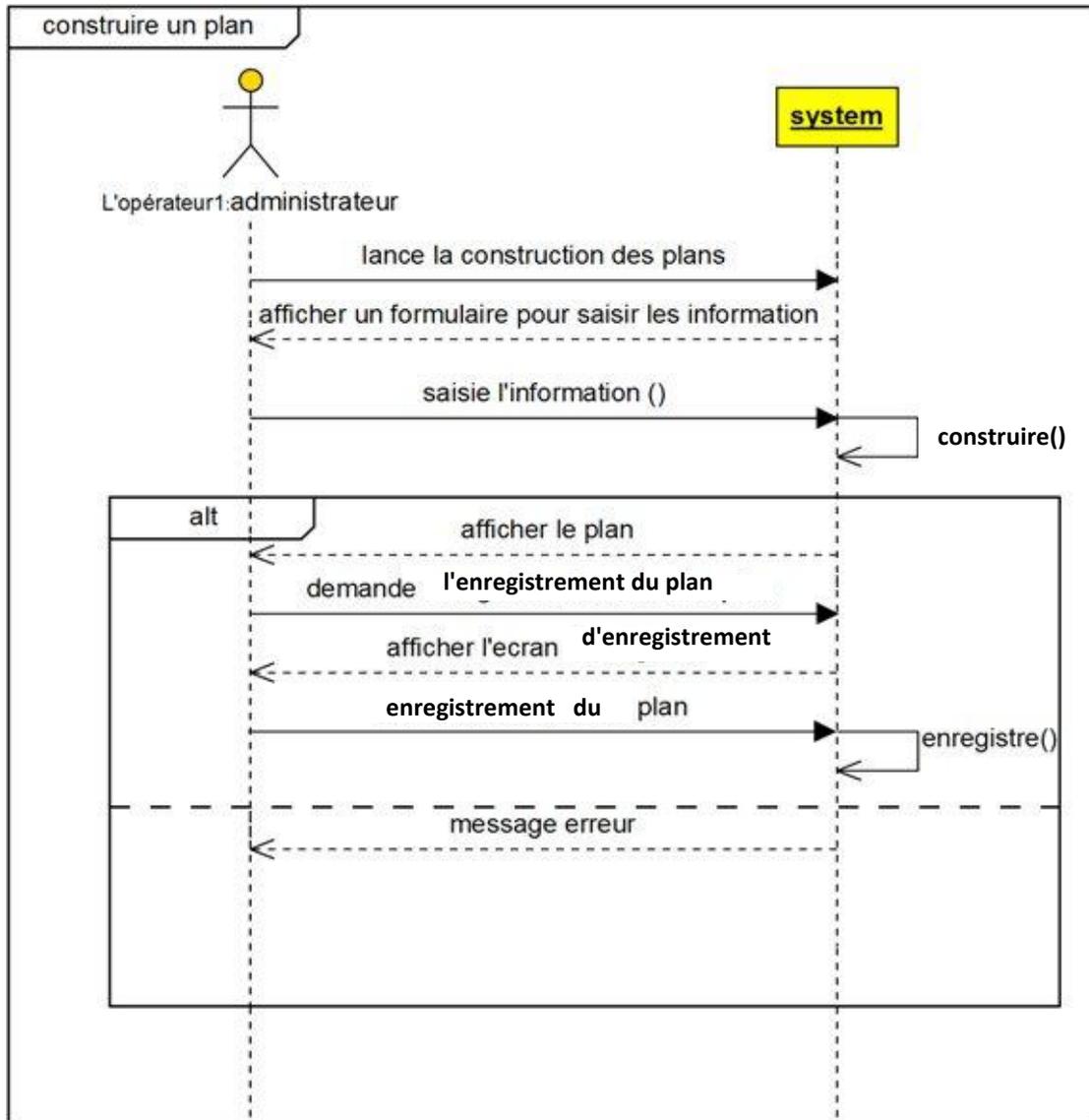


Figure 3.21 : diagramme de séquence construire un plan

6.21 Cas d'utilisation «authentification»:

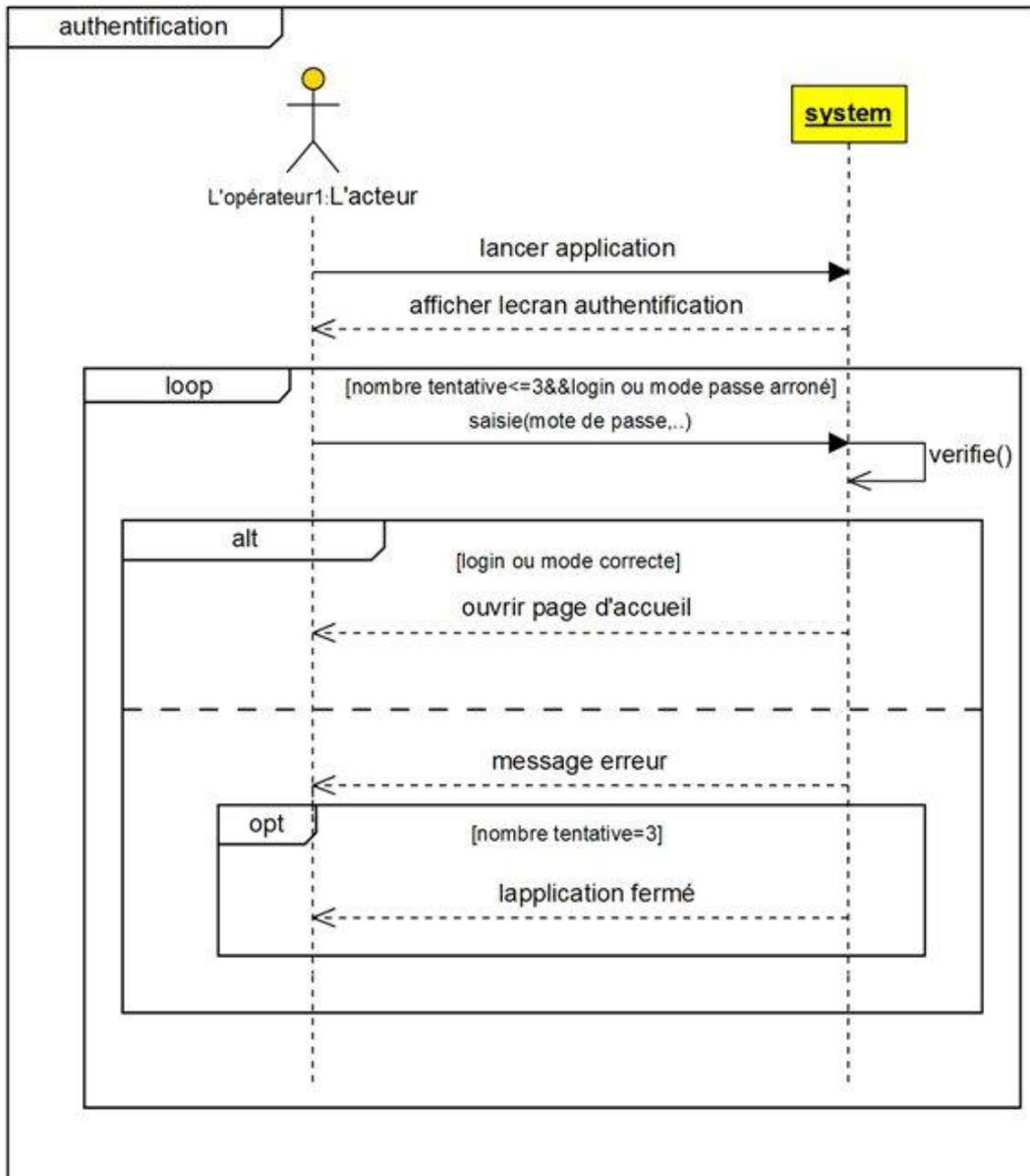


Figure 3.22 : diagramme de séquence authentification

6.22 Cas d'utilisation «changer le mot de passe»:

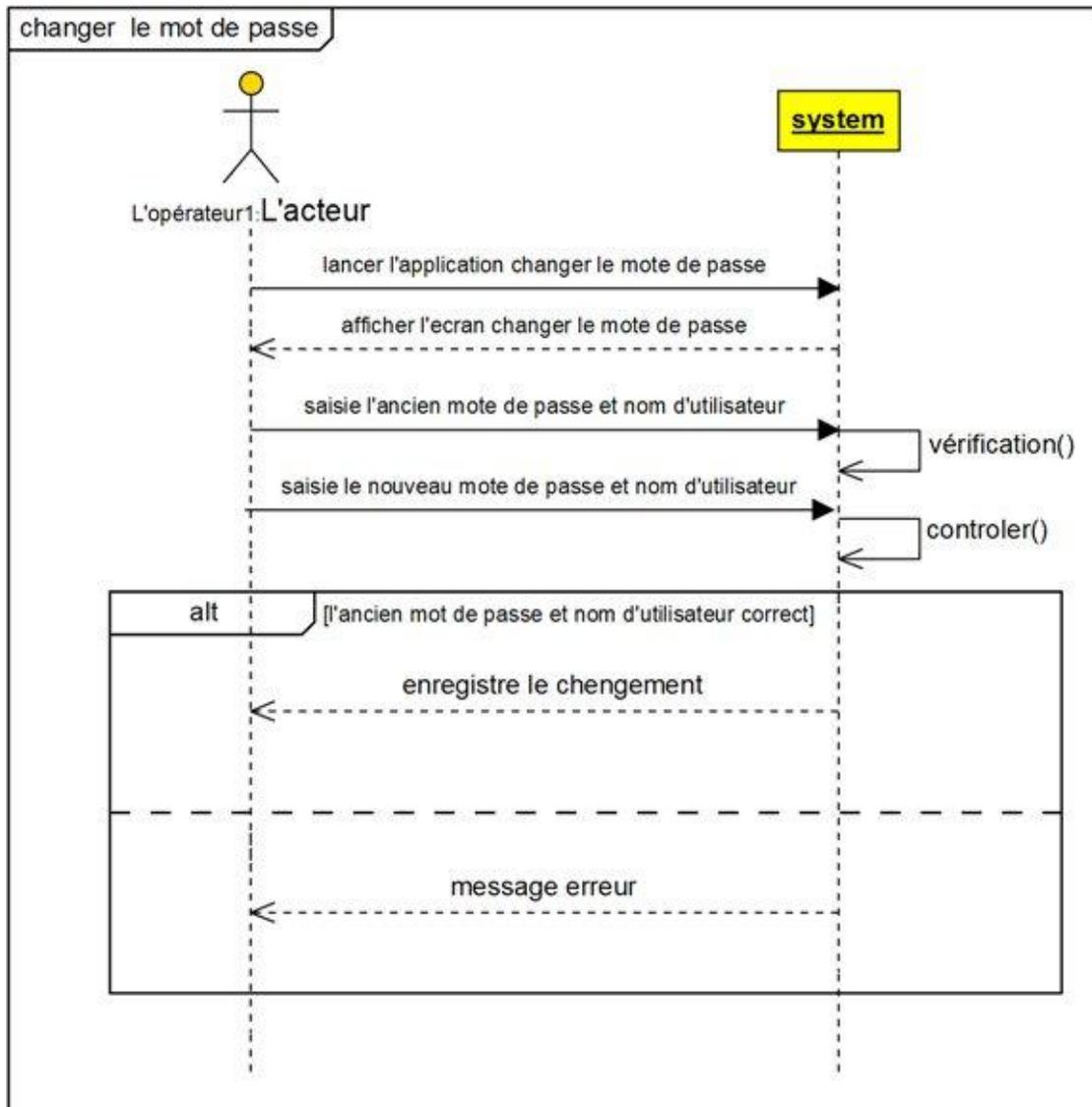


Figure 3.23 : diagramme de séquence changé mot de passe

7. Diagramme de class

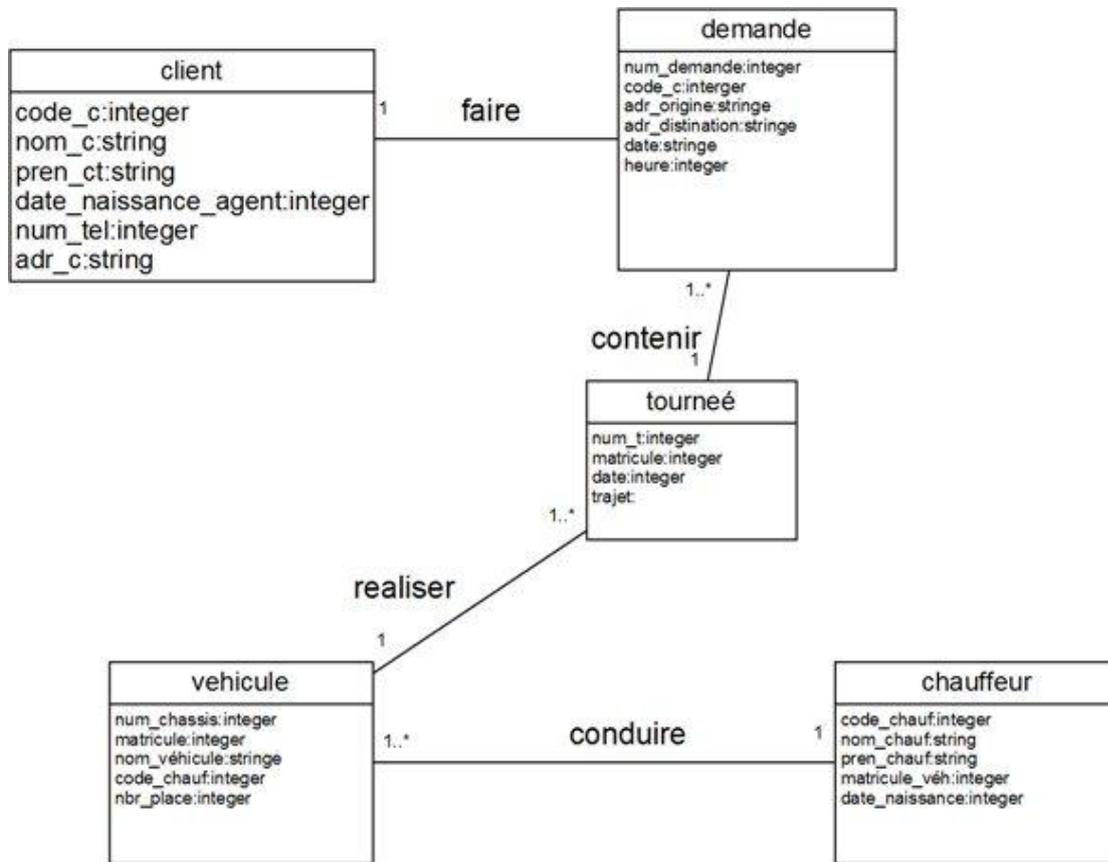


Figure 3.24: Modèle de domaine

7.1 : Cas d'utilisation «ajouter client»:

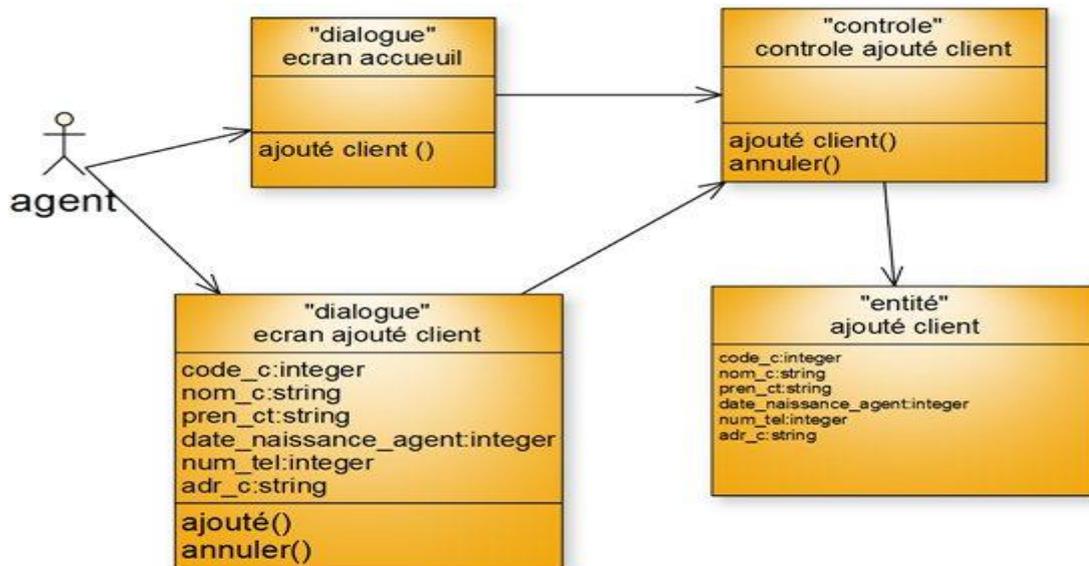


Figure 3.25 : diagramme de classe participante de cas d'utilisation ajouter client

7.2 : Cas d'utilisation «ajouter demande des clients»:

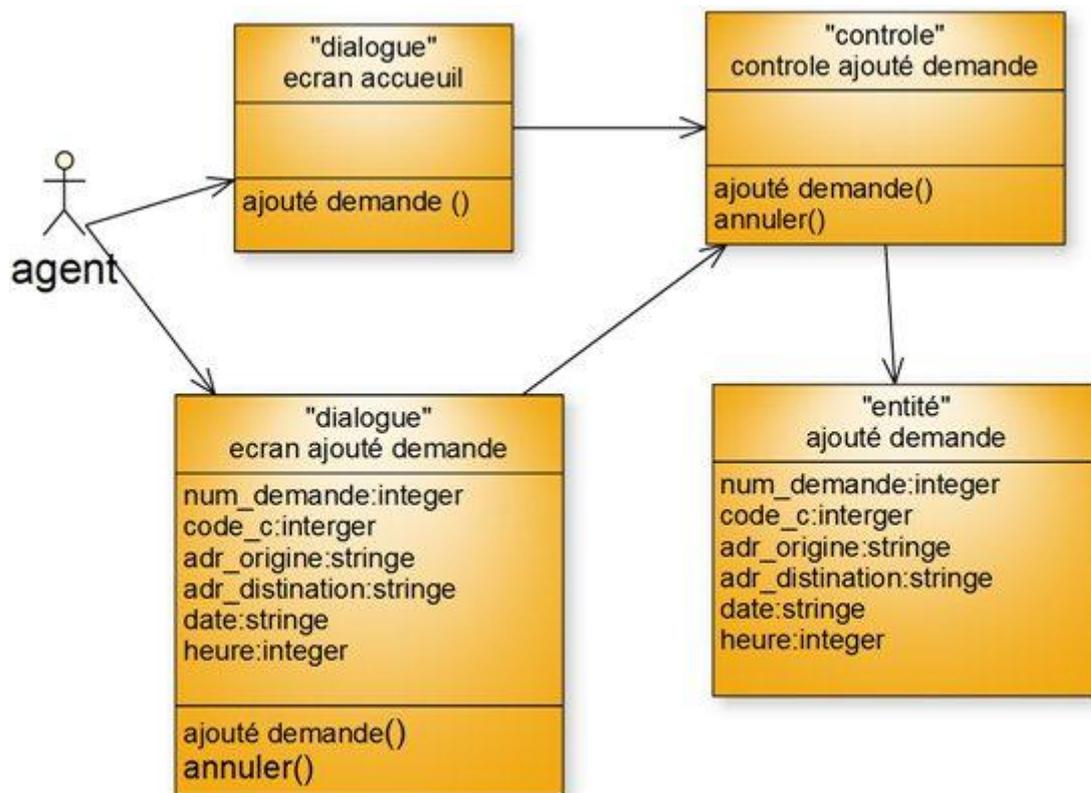


Figure 3.26 : diagramme de classe participante de cas d'utilisation ajouter demande des clients

7.3 : Cas d'utilisation «ajouter chauffeur»:

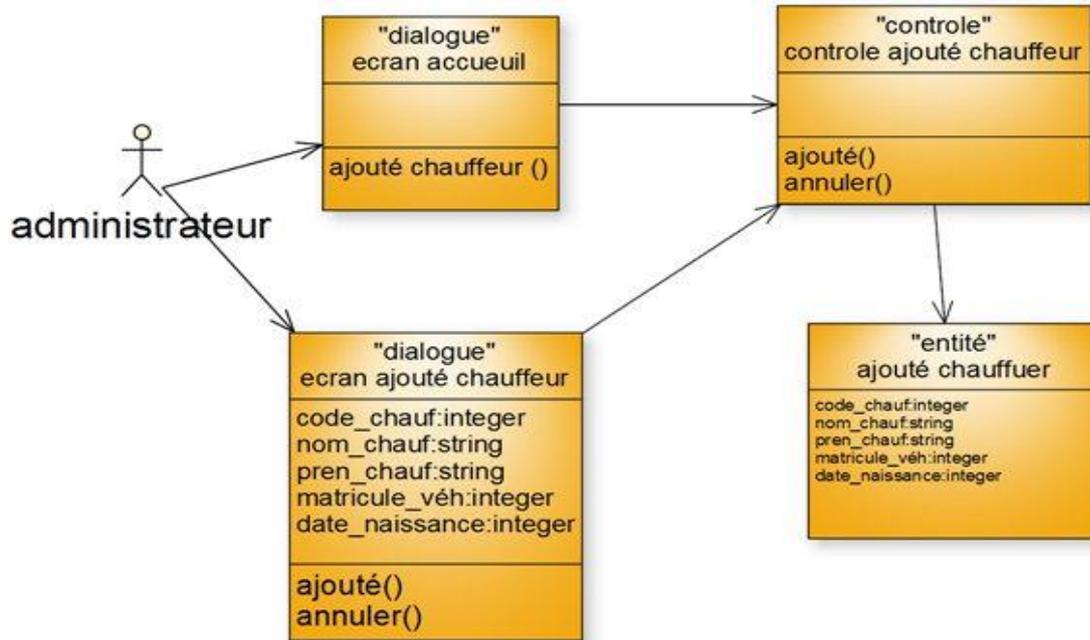


Figure 3.27 : diagramme de classe participante de cas d'utilisation ajouter chauffeur

7.4 : Cas d'utilisation «ajouter véhicule»:

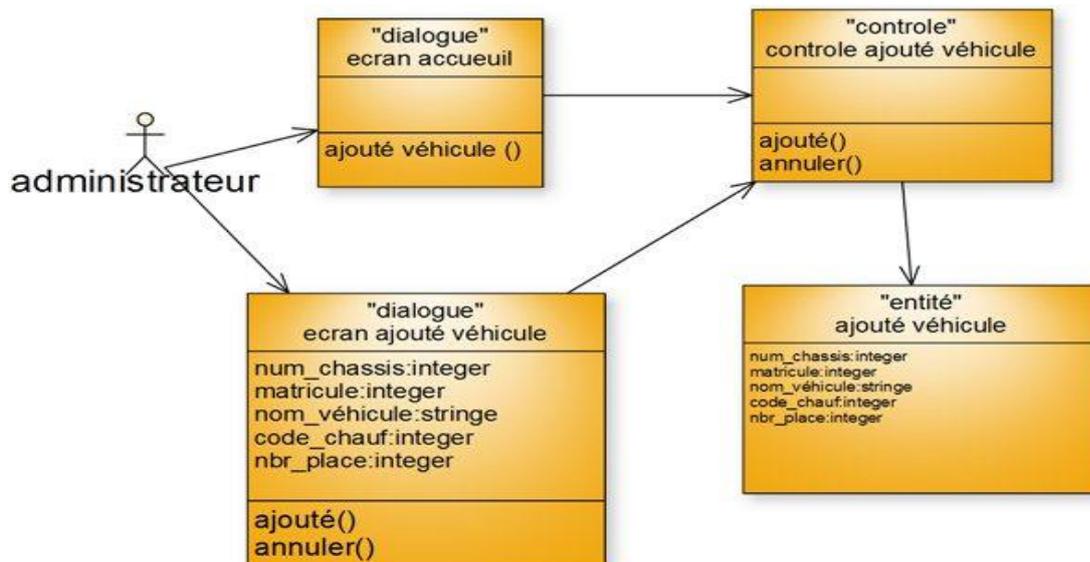


Figure 3.28 : diagramme de classe participante de cas d'utilisation ajouter véhicule.

7.5 : Cas d'utilisation «modifier client»:

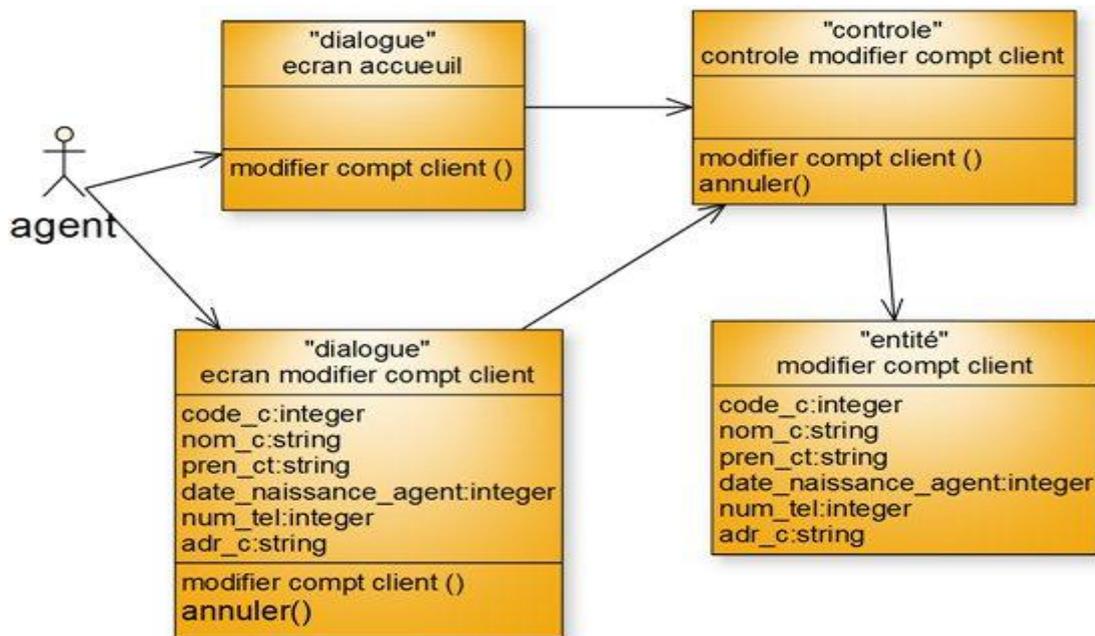


Figure 3.29 : diagramme de classe participante de cas d'utilisation modifier client

7.6 : Cas d'utilisation «modifier véhicule»:

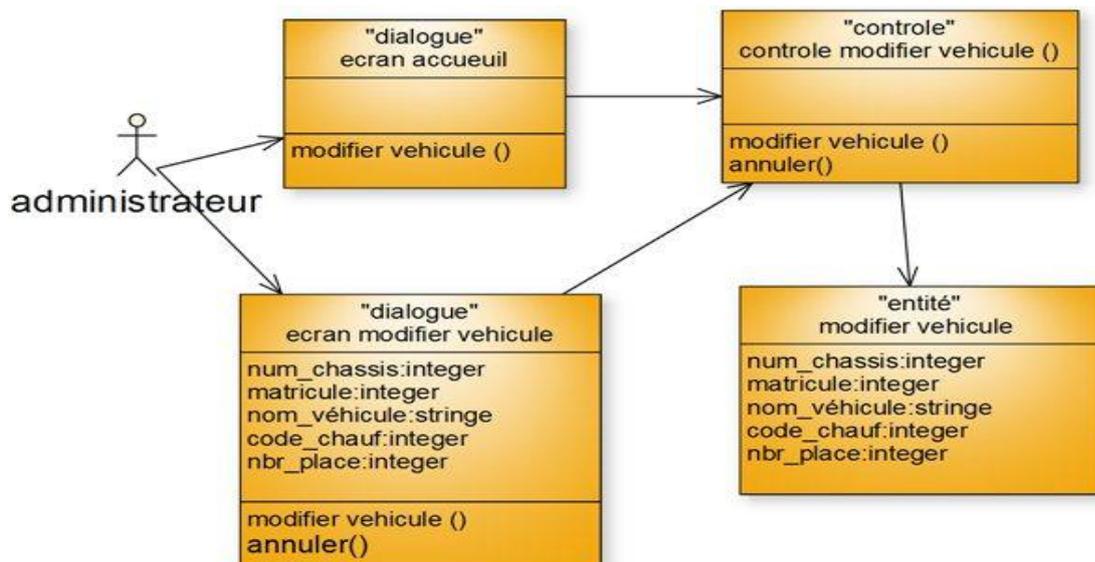


Figure 3.30 : diagramme de classe participante de cas d'utilisation modifier véhicule.

7.7 : Cas d'utilisation «modifier chauffeur»:

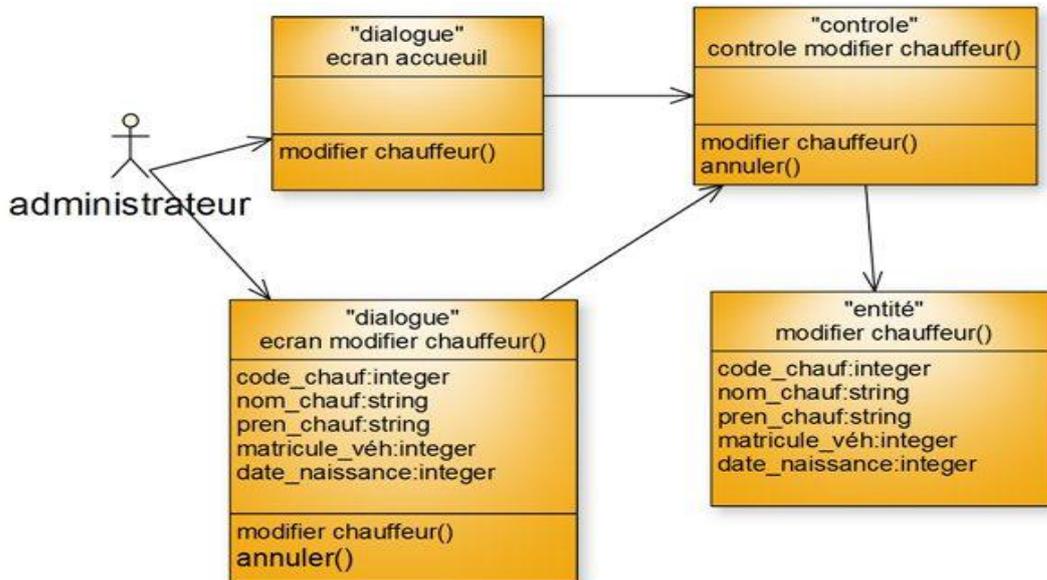


Figure 3.31 : diagramme de classe participante de cas d'utilisation modifier chauffeur

7.8 : Cas d'utilisation «modifier demande de client»:

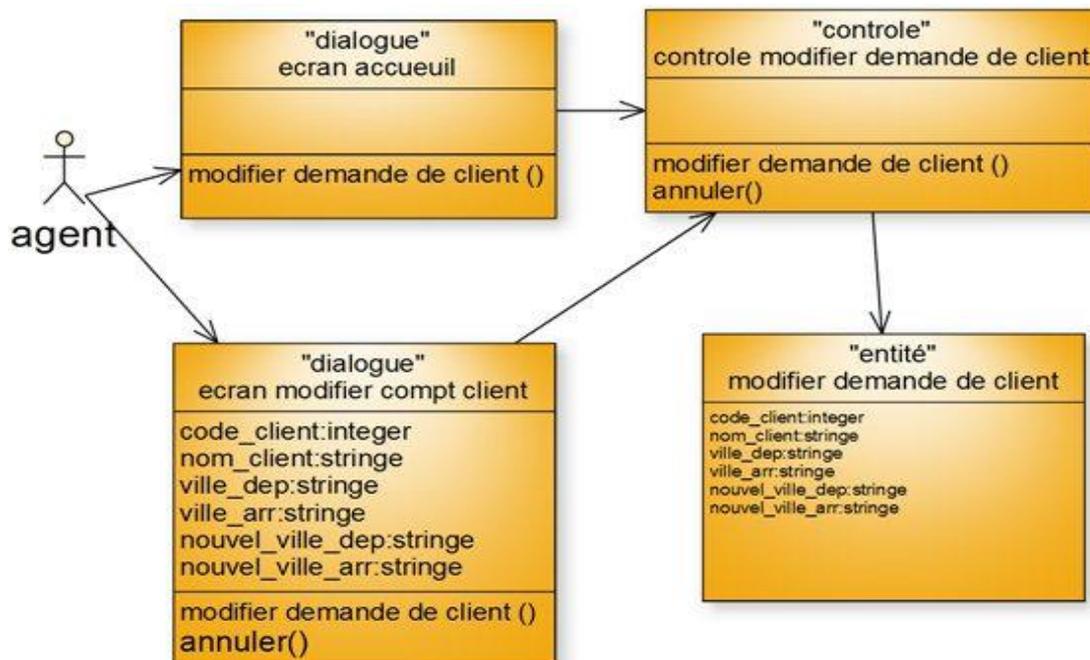


Figure 3.32 : diagramme de classe participante de cas d'utilisation modifier demande de client

7.9 : Cas d'utilisation «modifier trajet de tournée»:

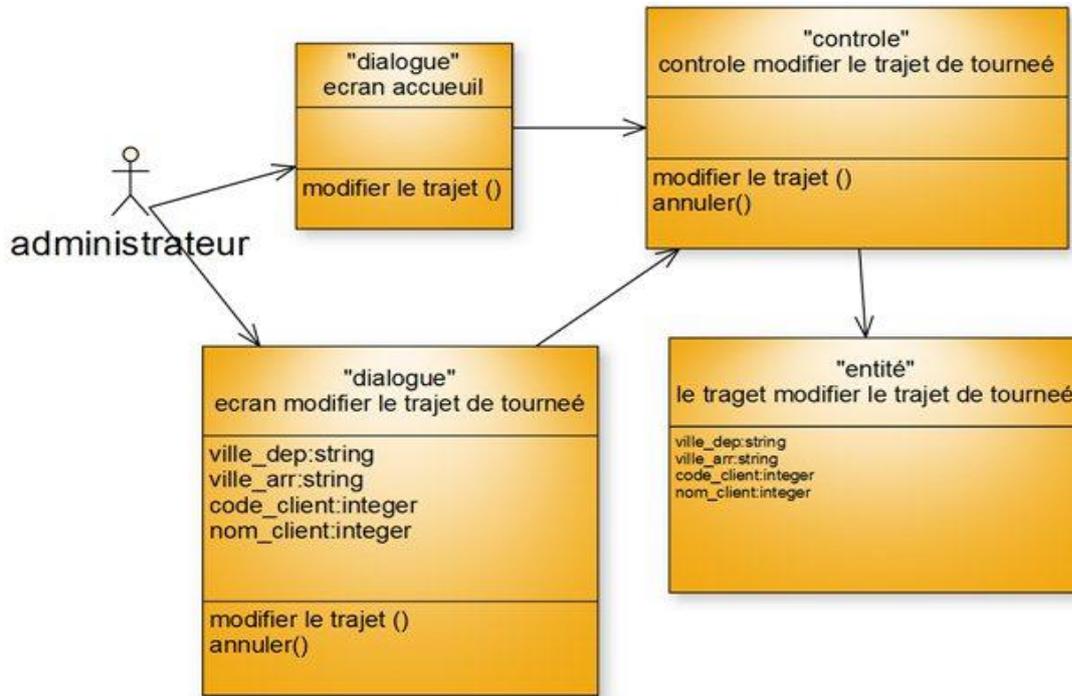


Figure 3.33 : diagramme de classe participante de cas d'utilisation modifier trajet de tournée

7.10 : Cas d'utilisation «supprimer client»:

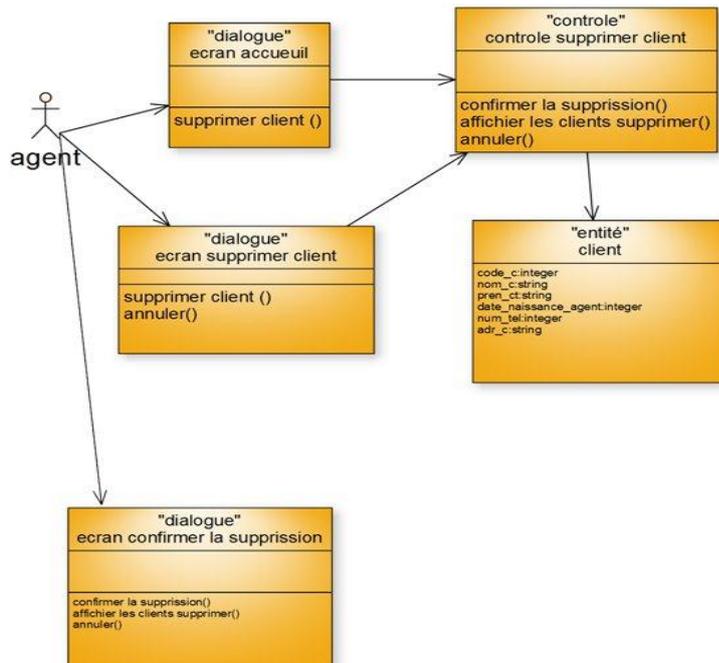


Figure 3.34 : diagramme de classe participante de cas d'utilisation supprimer client

7.11 : Cas d'utilisation «supprimer demande de client»:

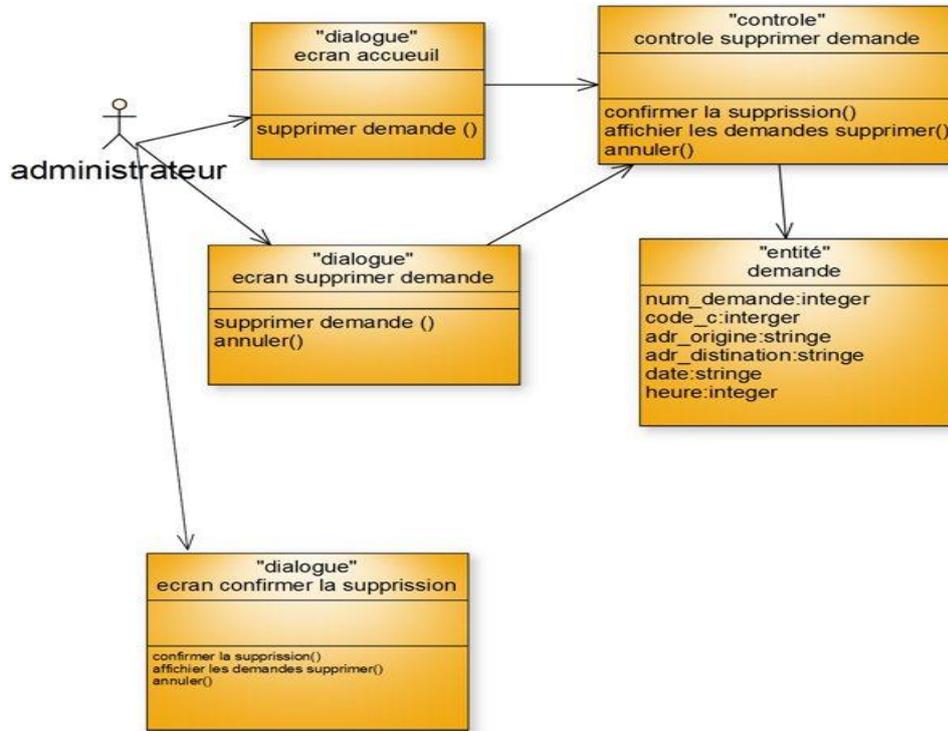


Figure 3.35 : diagramme de classe participante de cas d'utilisation supprimer demande de client.

7.12 : Cas d'utilisation «supprimer plan»:

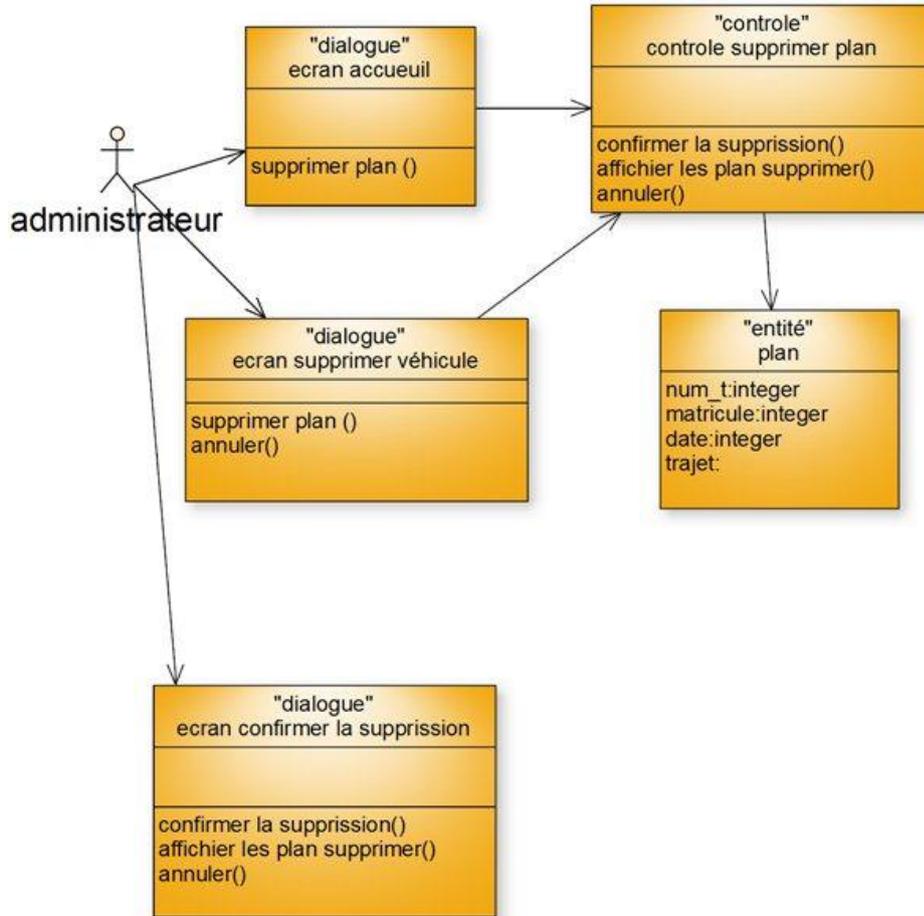


Figure 3.36 : diagramme de classe participante de cas d'utilisation supprimer plan

7.13 : Cas d'utilisation «supprimer chauffeur»:

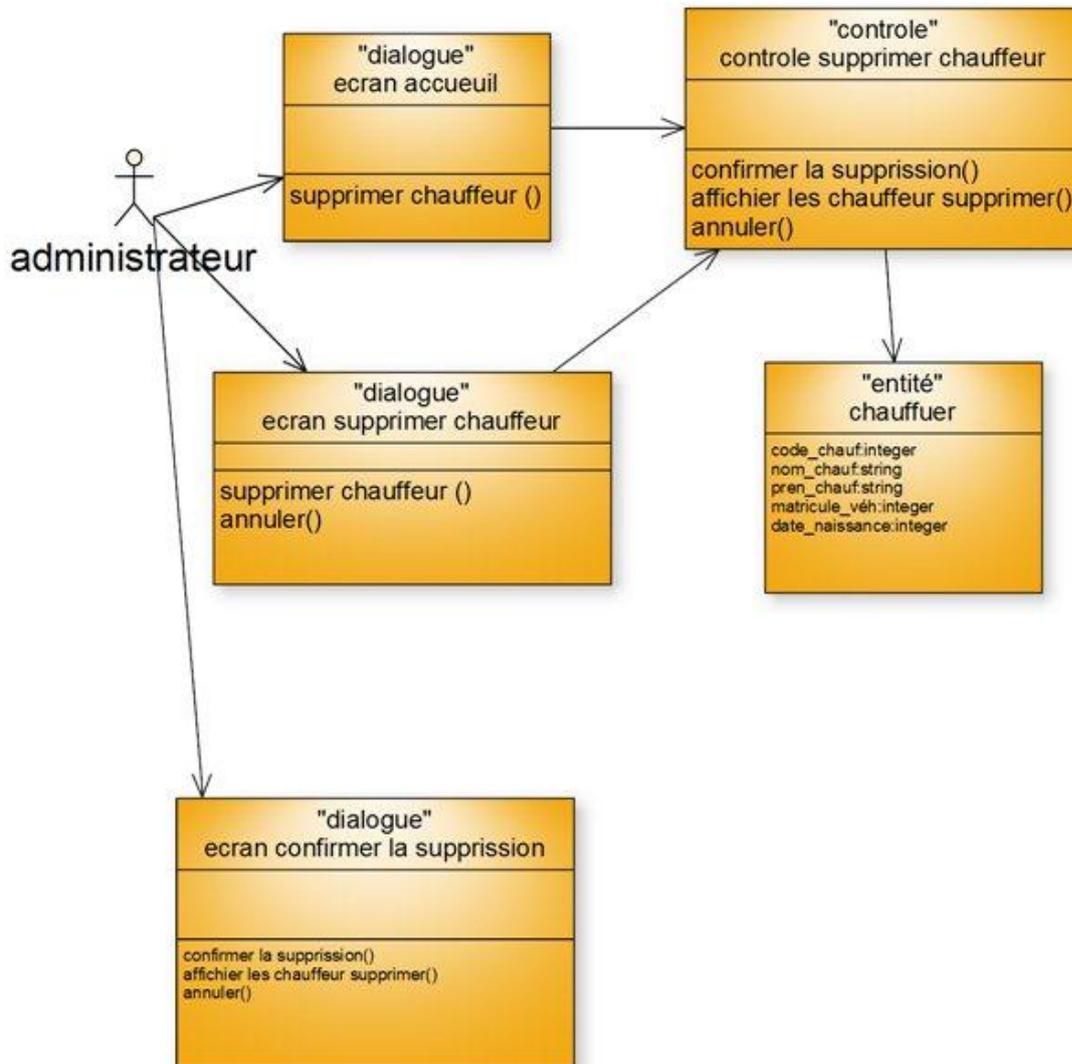


Figure 3.37 : diagramme de classe participante de cas d'utilisation supprimer chauffeur

7.14 : Cas d'utilisation «supprimer véhicule»:

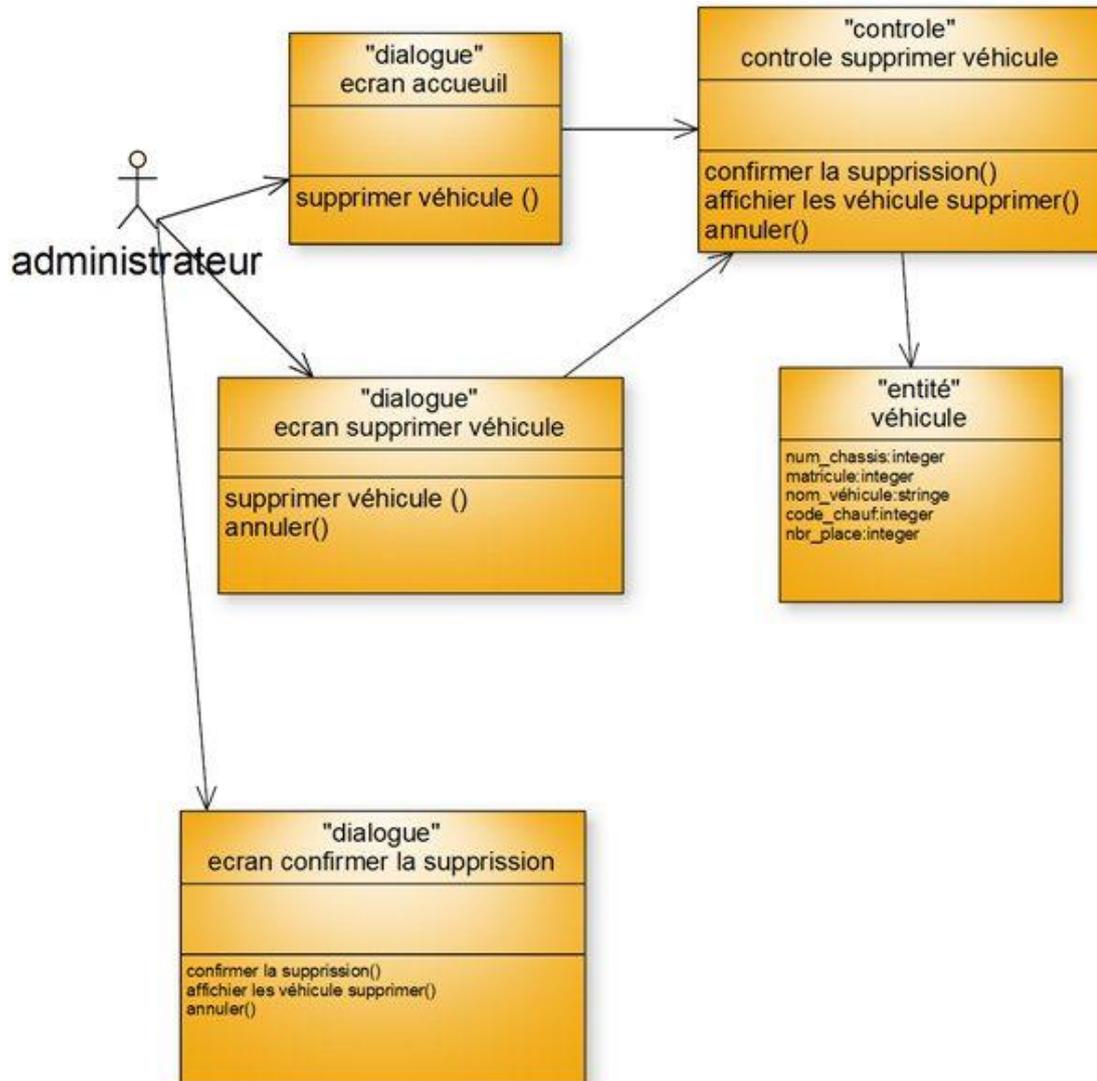


Figure 3.38 : diagramme de classe participante de cas d'utilisation supprimer véhicule

7.15 : Cas d'utilisation «chercher client»:

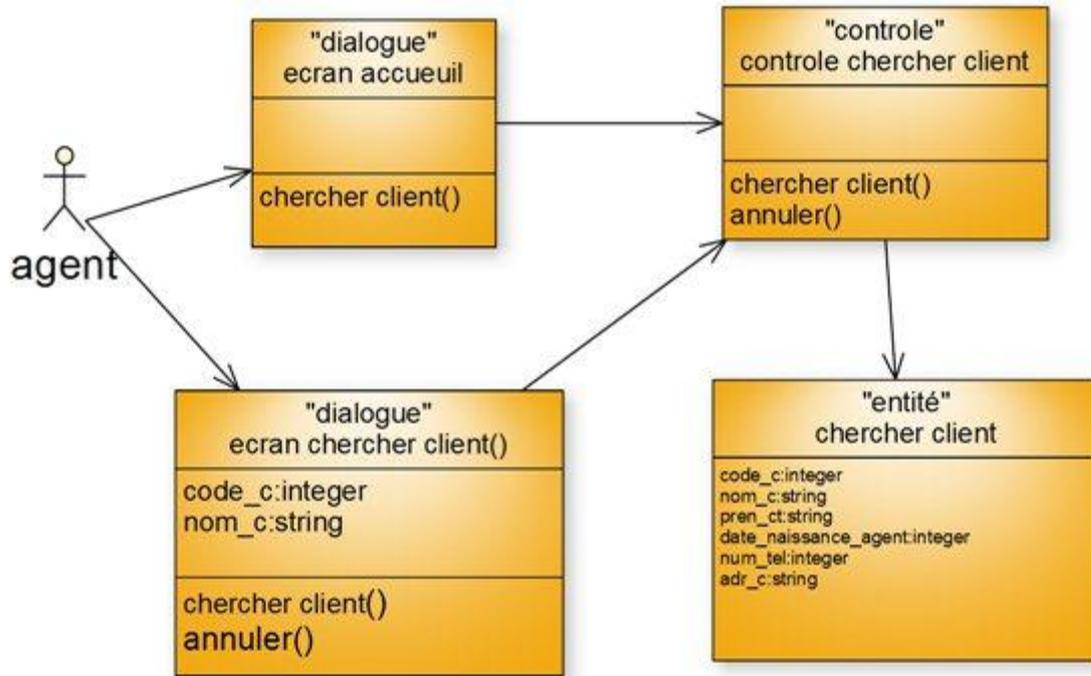


Figure 3.39 : diagramme de classe participante de cas d'utilisation chercher client

7.16 : Cas d'utilisation «chercher demande de client» :

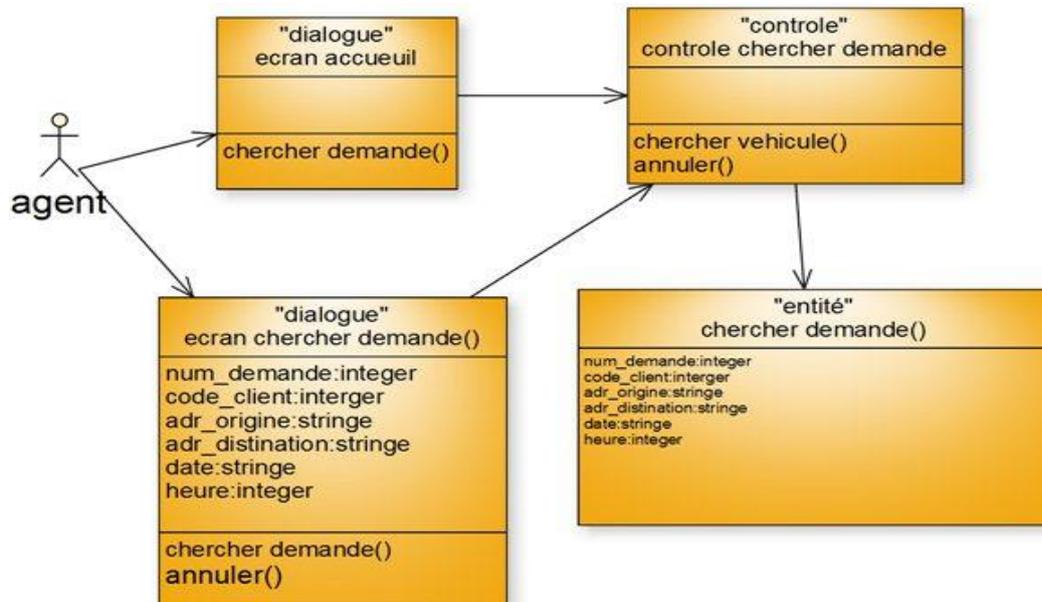


Figure 3.40 : diagramme de classe participante de cas d'utilisation cherche demande de client.

7.17 : Cas d'utilisation «chercher chauffeur»:

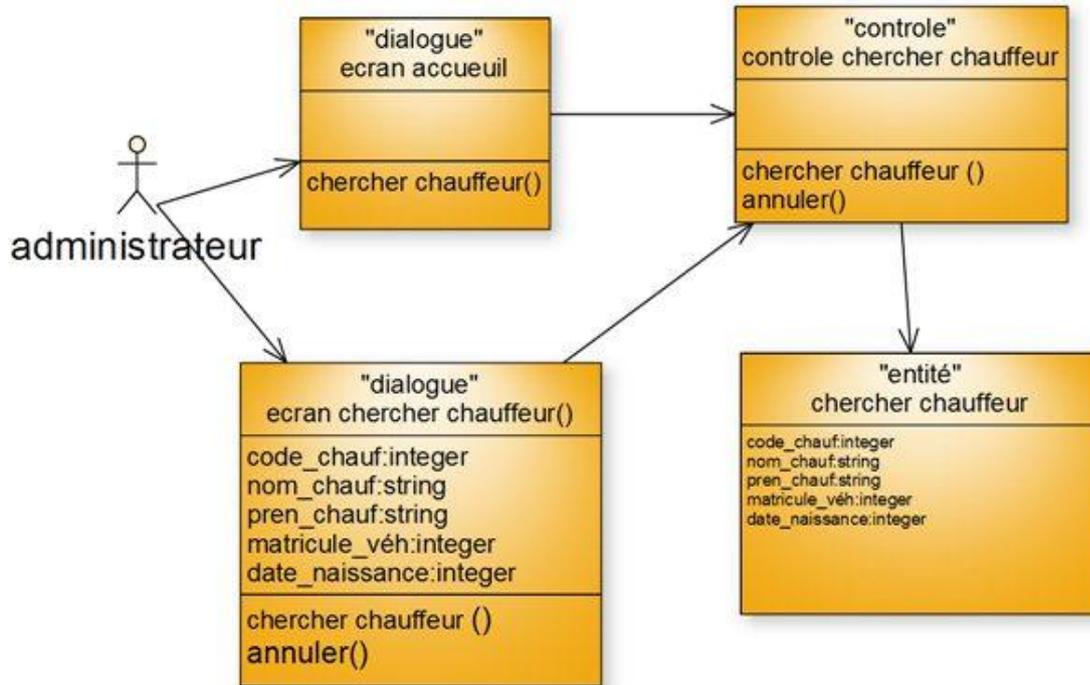


Figure 3.41 : diagramme de classe participante de cas d'utilisation cherche chauffeur.

7.18 : Cas d'utilisation «chercher véhicule»:

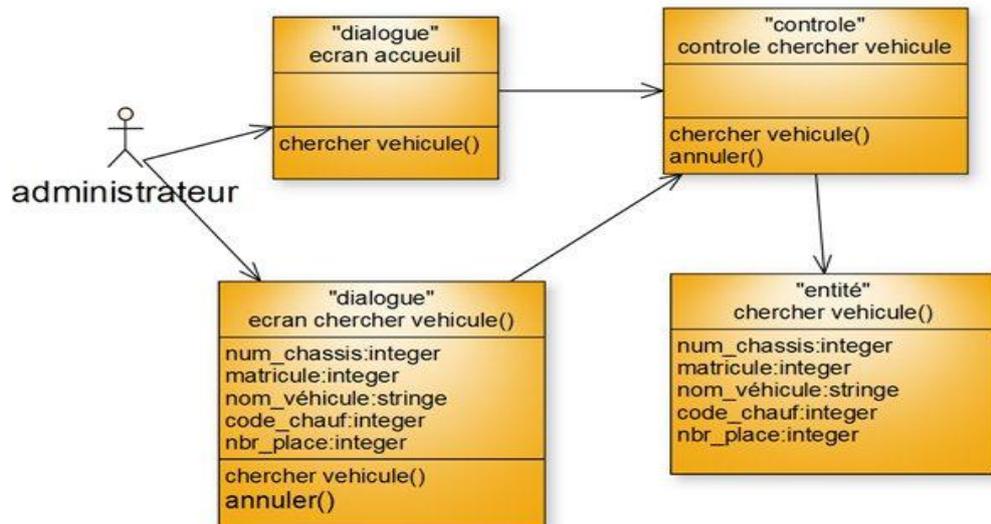


Figure 3.42 : diagramme de classe participante de cas d'utilisation cherche véhicule.

7.19 : Cas d'utilisation «chercher plan»:

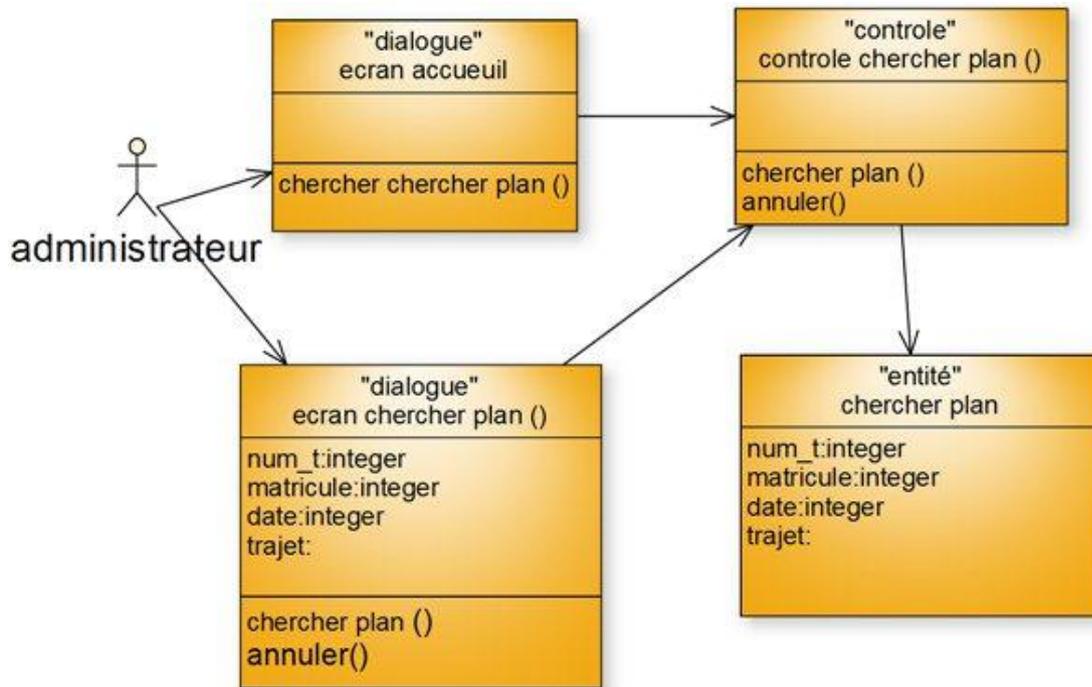


Figure 3.43 : diagramme de classe participante de cas d'utilisation chercher plan.

7.20 : Cas d'utilisation «construire un plan»:

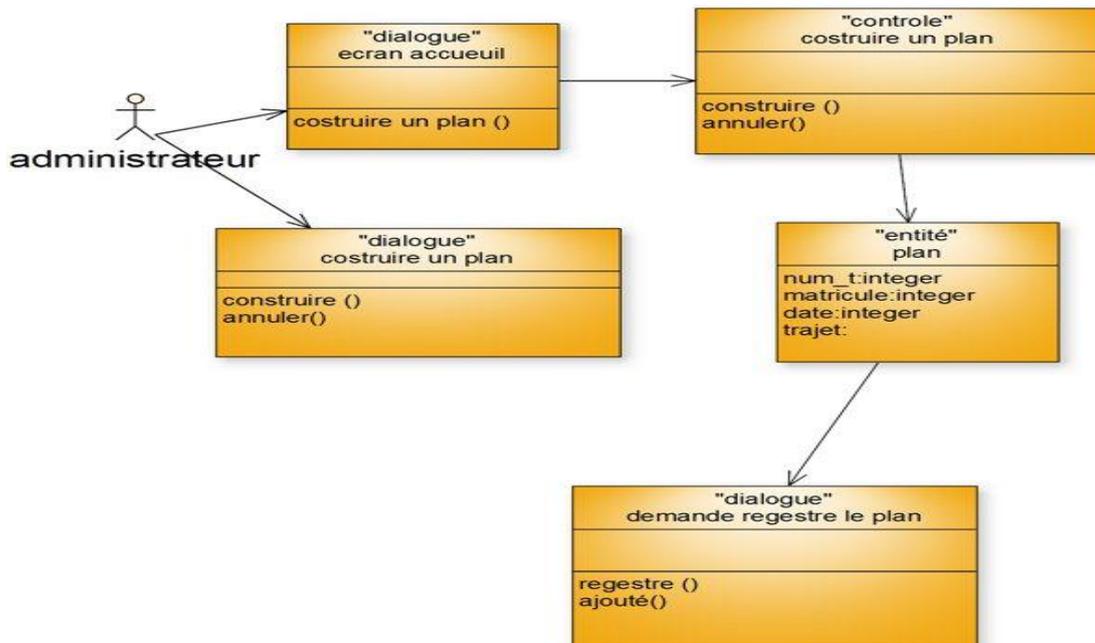


Figure 3.44 : diagramme de classe participante de cas d'utilisation construire un plan

7.21 : Cas d'utilisation «authentification»:

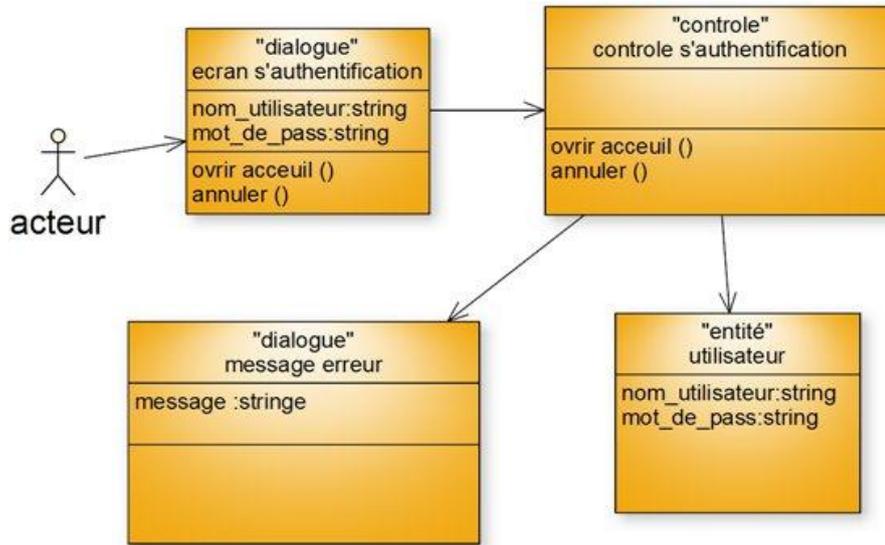


Figure 3.45 : diagramme de classe participante de cas d'utilisation authentification

3.4.22 : Cas d'utilisation «changer mot de passe»:

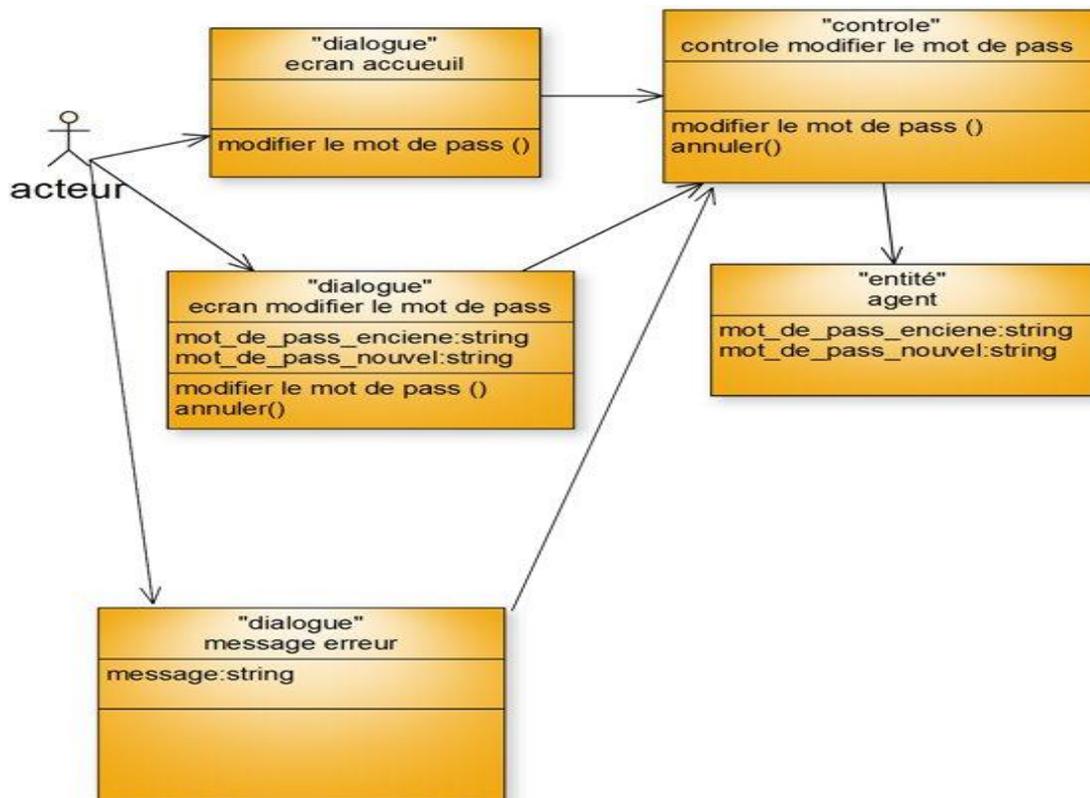


Figure 3.46 : diagramme de classe participante de cas d'utilisation changer le mot de passe

8. Diagramme d'activité

8.1 : Cas d'utilisation «ajouter client»:

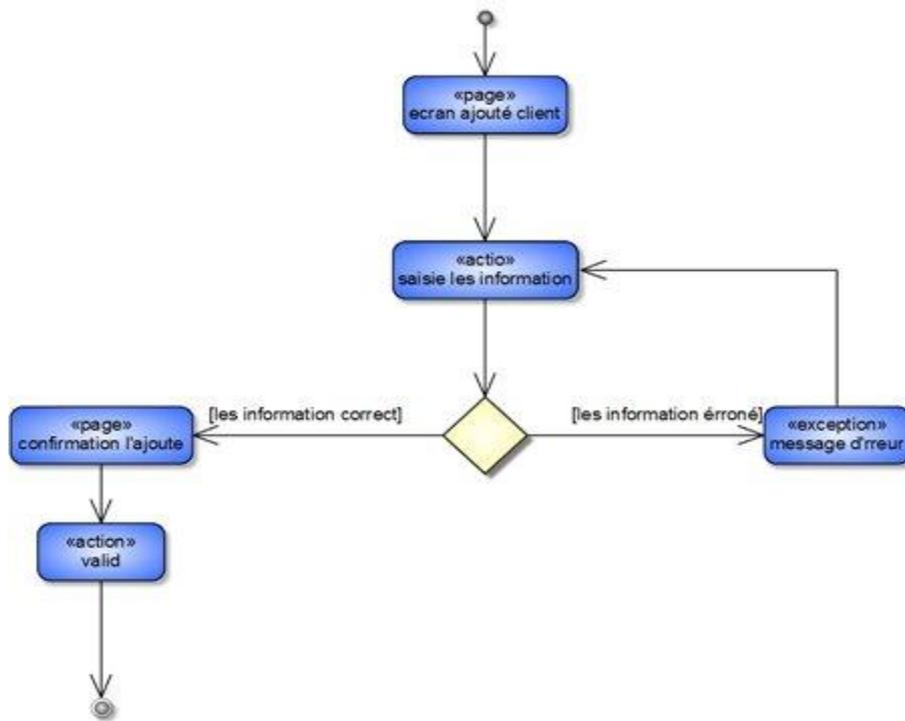


Figure 3.47: diagramme d'activité de Cas d'utilisation ajouter client

8.2 : Cas d'utilisation «ajouter demande de client »:

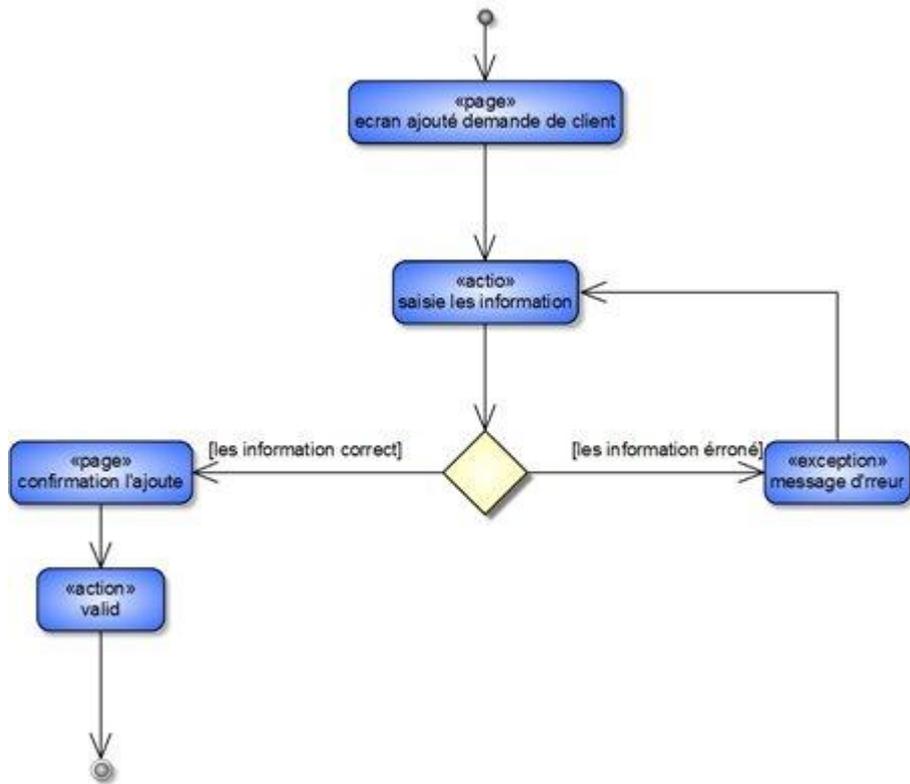


Figure 3.48: diagramme d'activité de Cas d'utilisation ajouté demande de client

8.3 : Cas d'utilisation «ajouter chauffeur»:

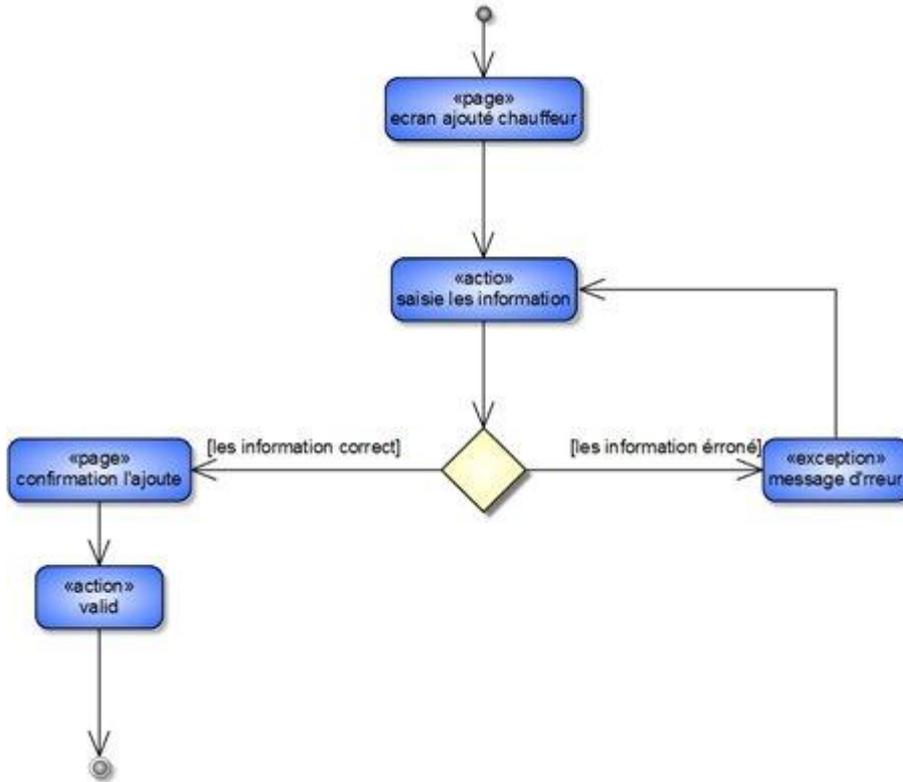


Figure 3.49: diagramme d'activité de cas d'utilisation ajouter chauffeur

8.4 : Cas d'utilisation «ajouter véhicule»:

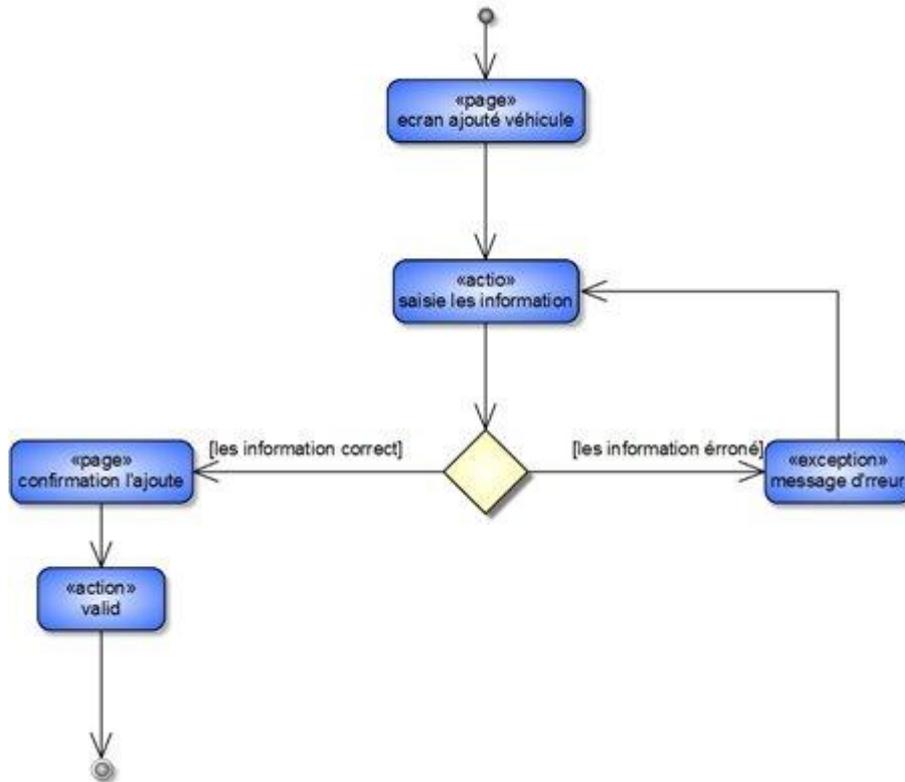


Figure 3.50: diagramme d'activité de cas d'utilisation ajouter véhicule

8.5 : Cas d'utilisation «modifier client»:

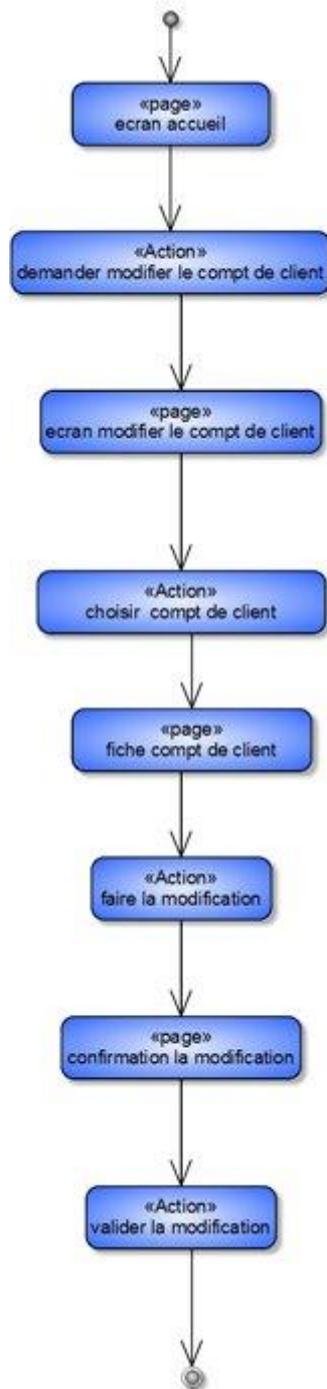


Figure 3.51: diagramme d'activité de cas d'utilisation modifier client

8.6 : Cas d'utilisation «modifier véhicule»:

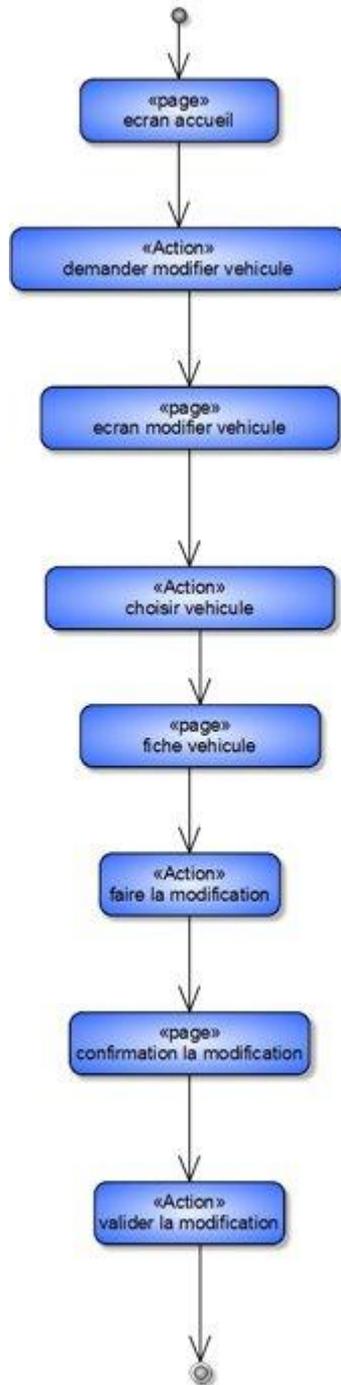


Figure 3.52: diagramme d'activité de cas d'utilisation modifier véhicule

8.7 : Cas d'utilisation «modifier chauffeur»:

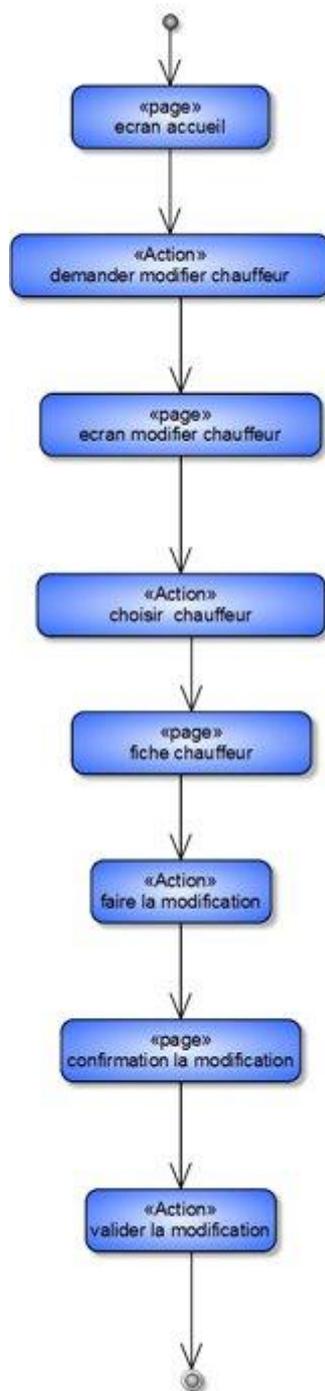


Figure 3.53: diagramme d'activité de cas d'utilisation modifier chauffeur

8.8 : Cas d'utilisation «modifier demande de client»:



Figure 3.54: diagramme d'activité de cas d'utilisation modifier demande de client

8.9 : Cas d'utilisation «modifier le trajet de tournée»:



Figure 3.55: diagramme d'activité de cas d'utilisation modifier le trajet de tournée

8.10 : Cas d'utilisation «supprimer client»:

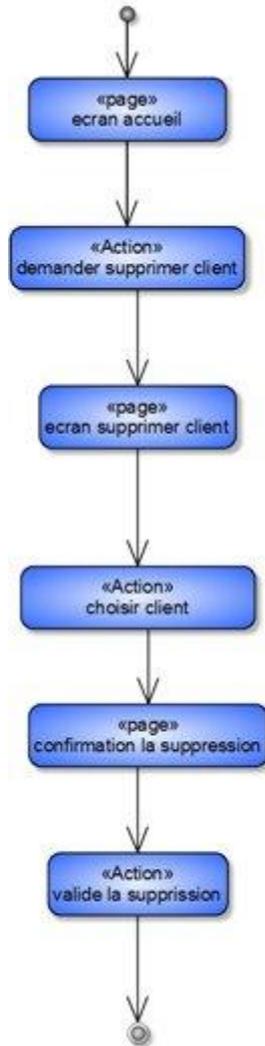


Figure 3.56: diagramme d'activité de cas d'utilisation supprimer client

8.11 : Cas d'utilisation «supprimer demande de client»:



Figure 3.57: diagramme d'activité de cas d'utilisation supprimer demande de client

8.12 : Cas d'utilisation «supprimer plan»:

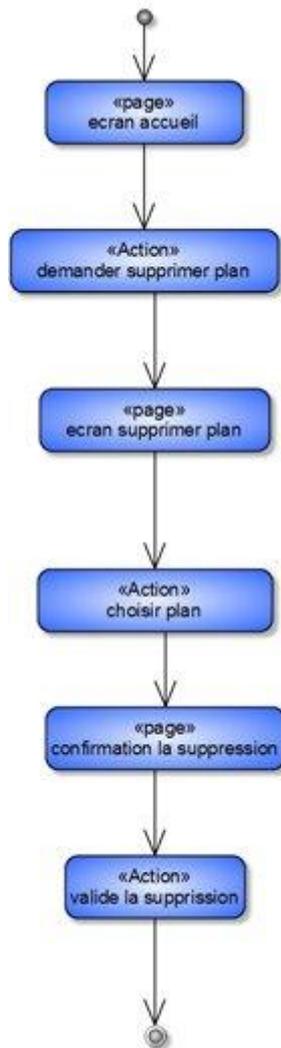


Figure 3.58: diagramme d'activité de cas d'utilisation supprimer plan

8.13 : Cas d'utilisation «supprimer chauffeur»:

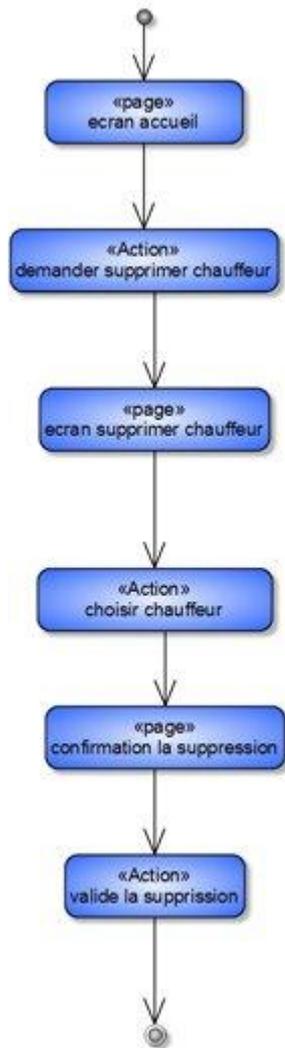


Figure 3.59: diagramme d'activité de cas d'utilisation supprimer chauffeur

8.14 : Cas d'utilisation «supprimer véhicule»:

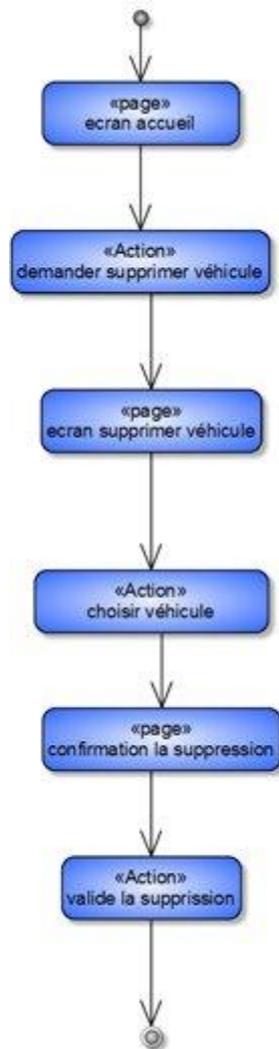


Figure 3.60: diagramme d'activité de cas d'utilisation supprimer véhicule

8.15 : Cas d'utilisation «chercher client»:

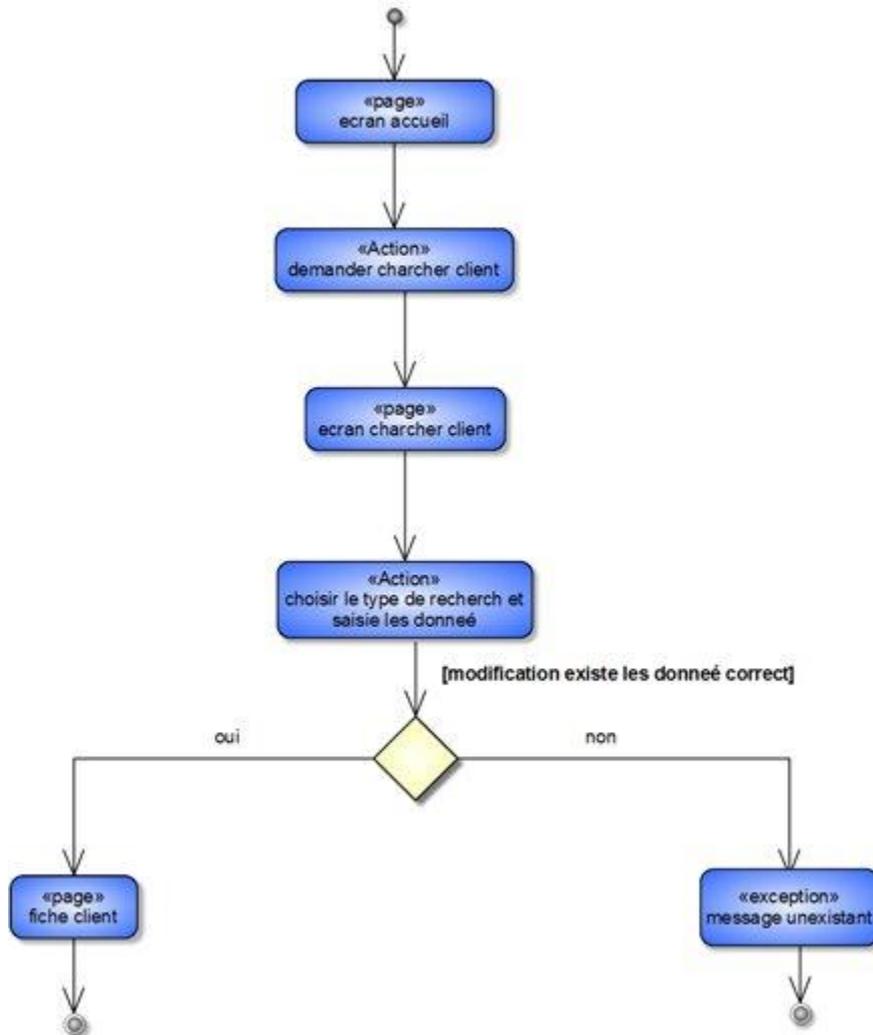


Figure 3.61: diagramme d'activité de cas d'utilisation cherche client

8.16 : Cas d'utilisation «chercher demande de client»:

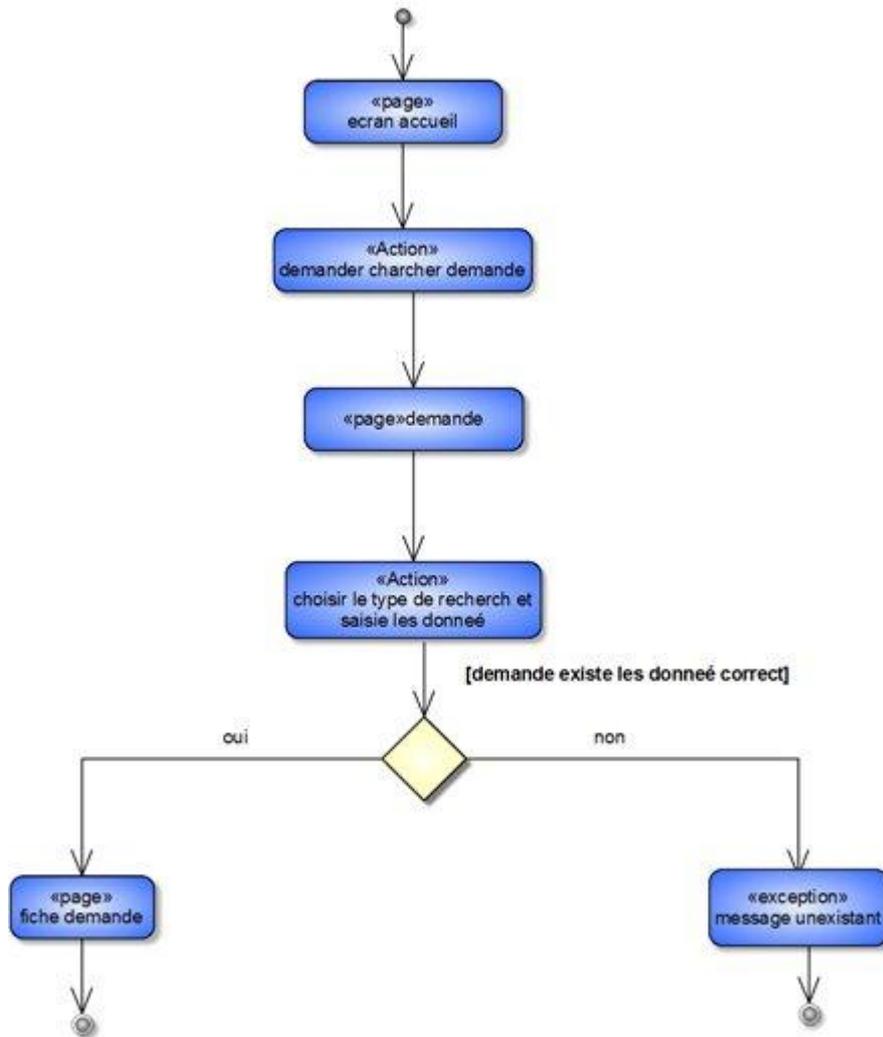


Figure 3.62: diagramme d'activité de cas d'utilisation cherche demande de client

8.17 : Cas d'utilisation «chercher chauffeur»:

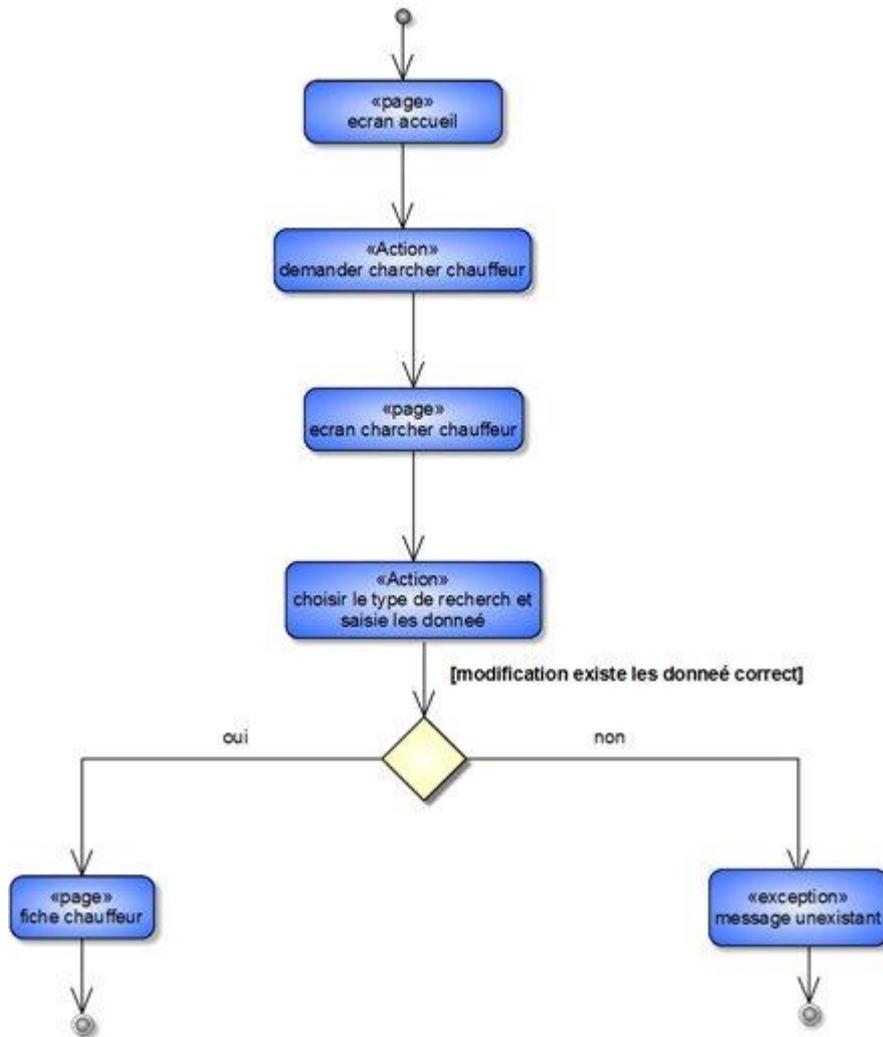


Figure 3.63: diagramme d'activité de cas d'utilisation recherche chauffeur

8.18 : Cas d'utilisation «chercher véhicule»:

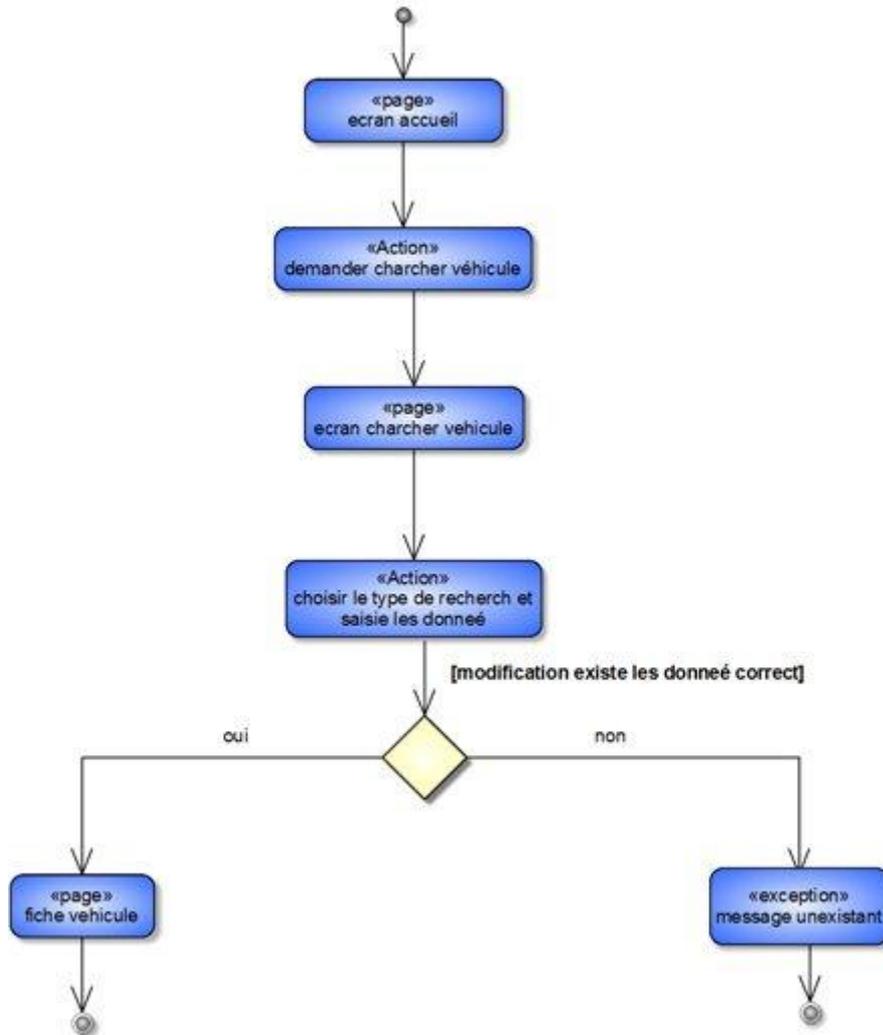


Figure 3.64: diagramme d'activité de cas d'utilisation recherche véhicule

8.19 : Cas d'utilisation «chercher plan»:

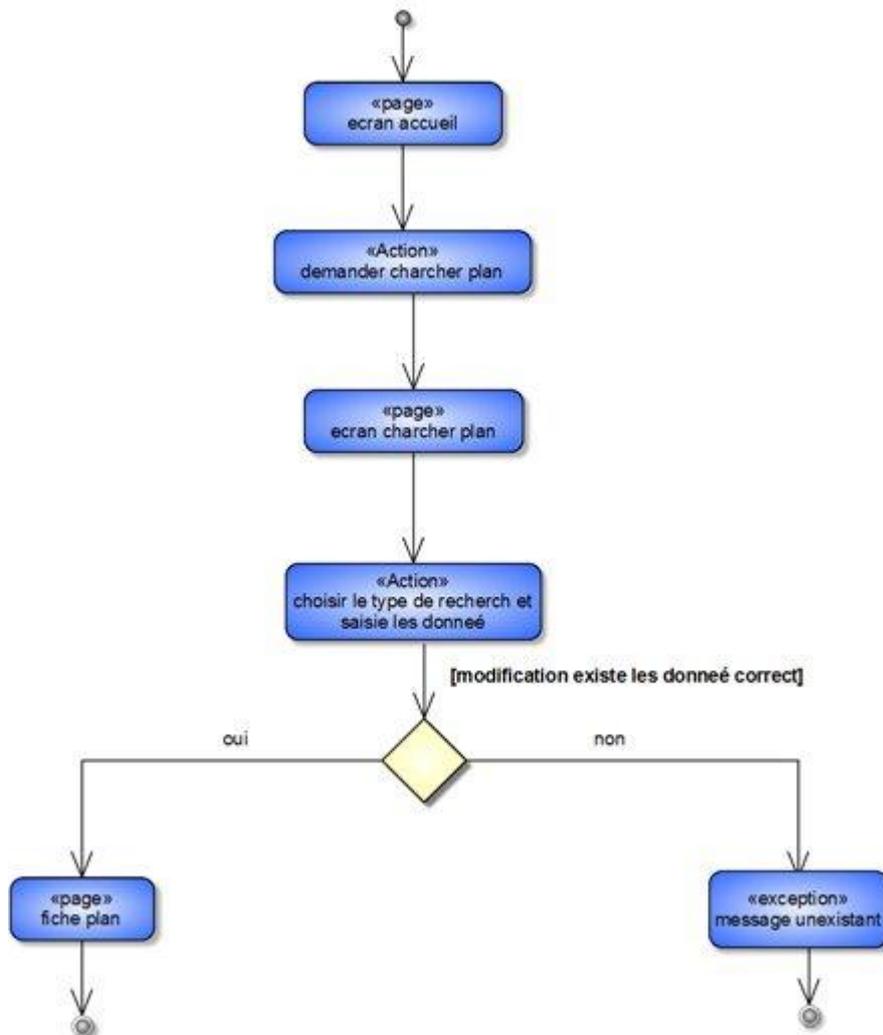


Figure 3.65: diagramme d'activité de cas d'utilisation cherche plan

8.20 : Cas d'utilisation «construire un plan»:

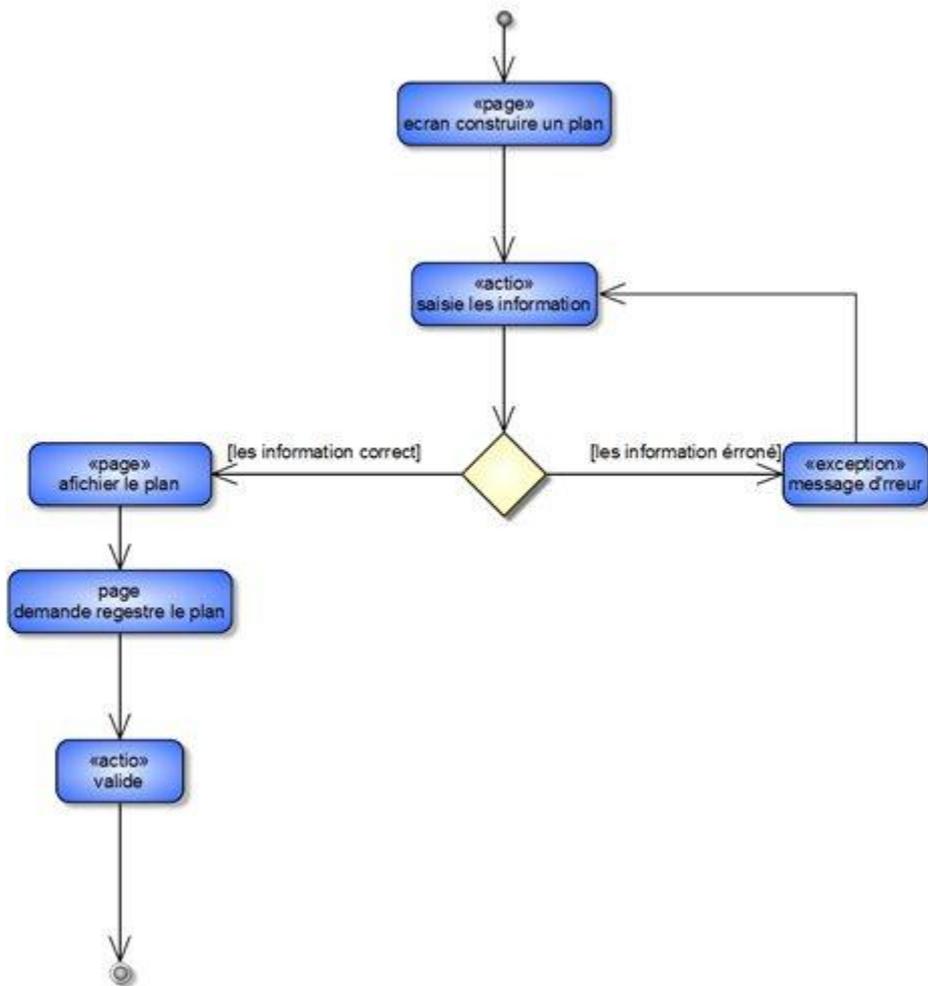


Figure 3.66: diagramme d'activité de cas d'utilisation construire un plan

8.20 : Cas d'utilisation «authentification»:

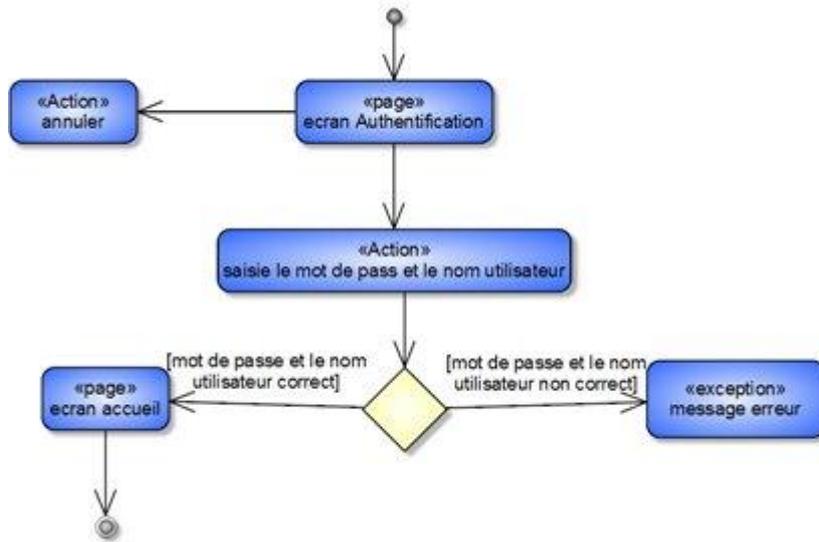


Figure 3.67: diagramme d'activité de cas d'utilisation authentification

8.20 : Cas d'utilisation «changer le mot de passe»:

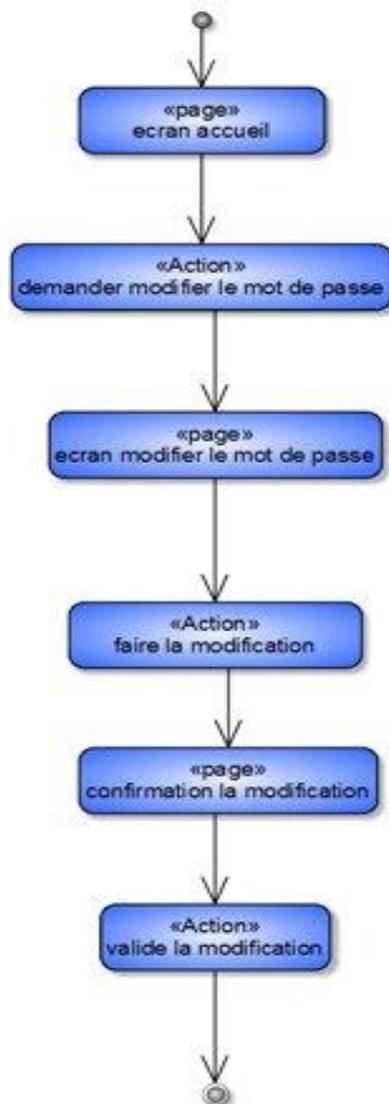


Figure 3.68: diagramme d'activité de cas d'utilisation changer mot de passe

9. Conclusion

Au cours de ce chapitre nous avons donné une description de tous les diagrammes correspondants à l'étape de modélisation. Dans le chapitre suivant nous donnons une description de l'application développée.

Chapitre 04

Chapitre 04 : Implémentation

1. Introduction

Après la phase de modélisation nous passons dans ce dernier chapitre à l'étape final de notre démarche, c'est la phase d'implémentation. Dans cette dernière nous allons construire notre application on se basant sur plusieurs outils de programmation. Dans ce qui suit, nous allons présenter premièrement les outils utilisés puis on passe à la description des différentes interfaces de notre application.

2. Outils d'implémentation.

2.1.Le langage de programmation C#.

Pour le développement de notre application nous avons choisi le langage de programmation C#. C'est un langage de Microsoft récent (2002) par rapport aux autres langages existants. Ce dernier offre des éléments et des outils simples et faciles à utiliser pour construire des applications.

2.2.Visual Studio.

C'est l'application (l'environnement) utilisée pour le développement de nos programmes en C#. Visual studio est un environnement open source est sont téléchargement est facile à partir de sa page officielle.

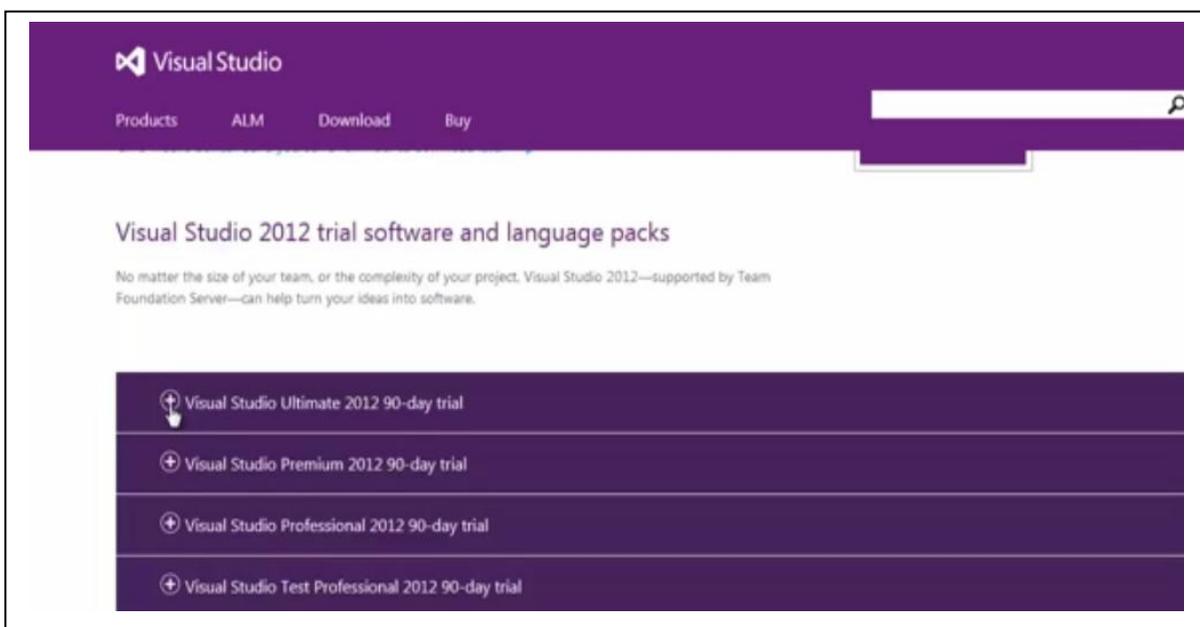


Figure 1 : Page officielle pour télécharger Visual Studio

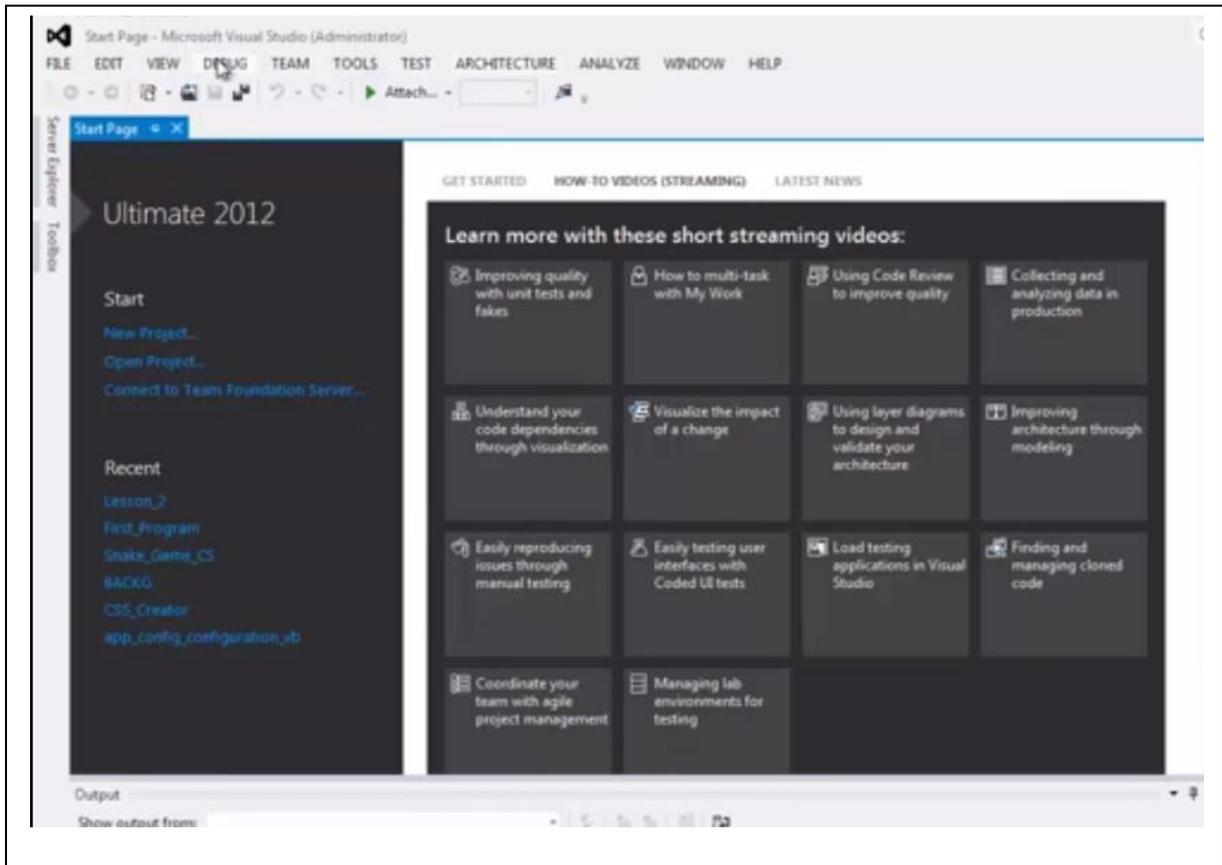


Figure : Fenêtre de départ de Visual Studio.

Visual studio propose plusieurs modes de programmation pour C#. Pour le développement de notre application nous utilisons le « Windows Forms Applications » car ce dernier propose des outils et des composants riches pour construire des interfaces interactives.

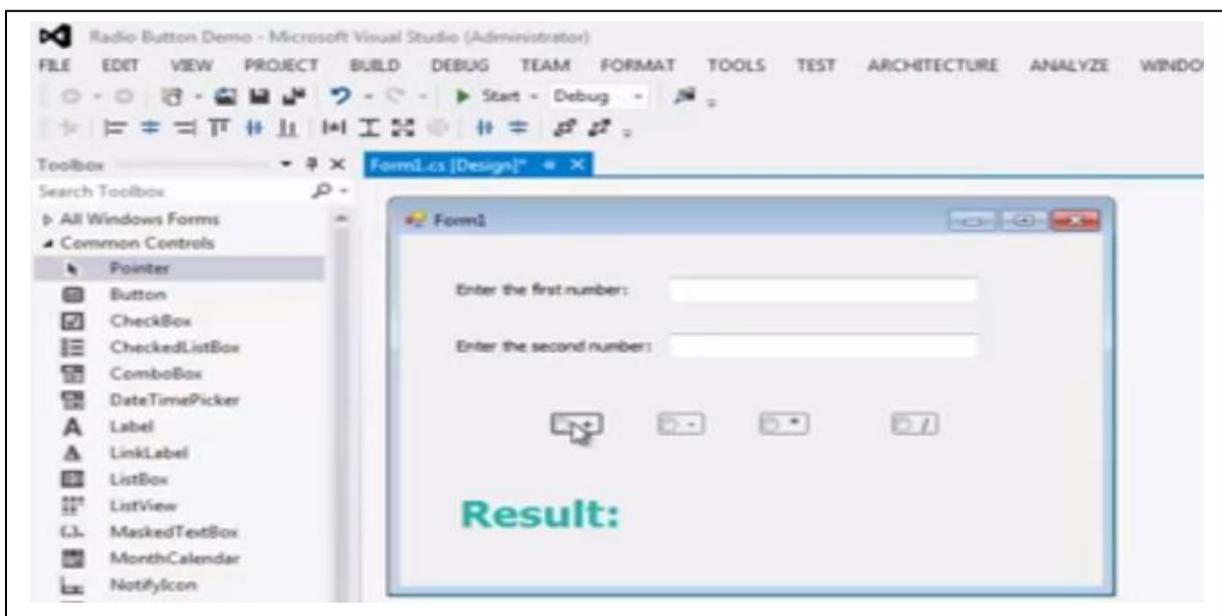


Figure3 : Le mode « Windows Forms Applications».

2.3. SQL Server et les bases de données.

Pour la création et l'interrogation de nos bases de données nous avons utilisé « SQL Server » qui est un autre produit de la société Microsoft. SQL Server propose un environnement complet pour la création et la manipulation des bases de données.

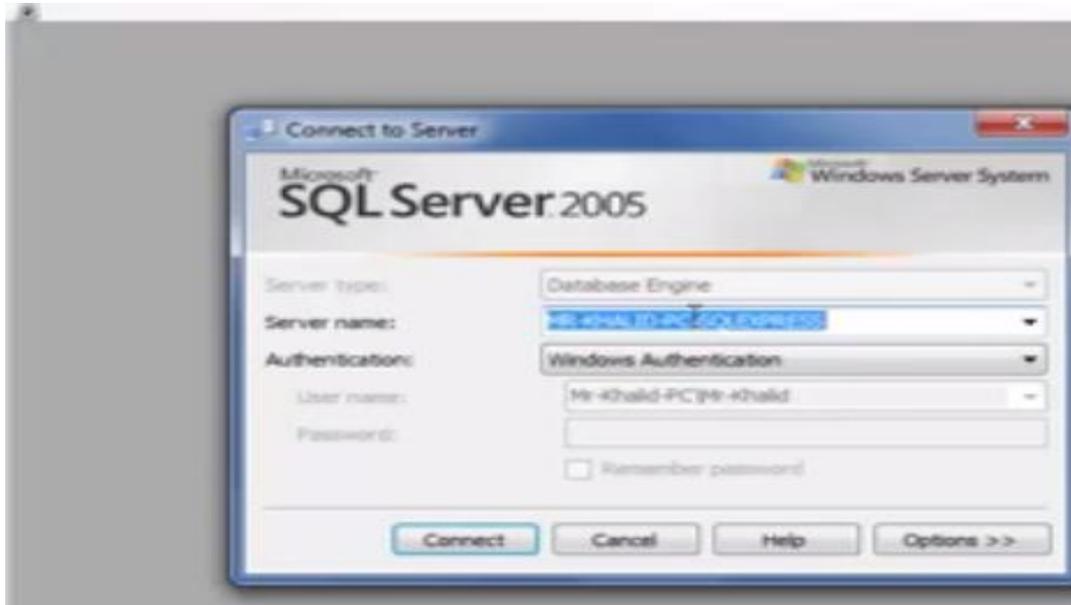


Figure 4 : SQL serveur.

3. Interfaces de l'application développée

3.1. Page d'accueil.

L'entrée à notre application est sécurisée par un mot de passe. L'administrateur ou bien l'agent doit s'authentifier pour avoir le droit d'utiliser à l'application.



Figure5 : Page d'authentification.

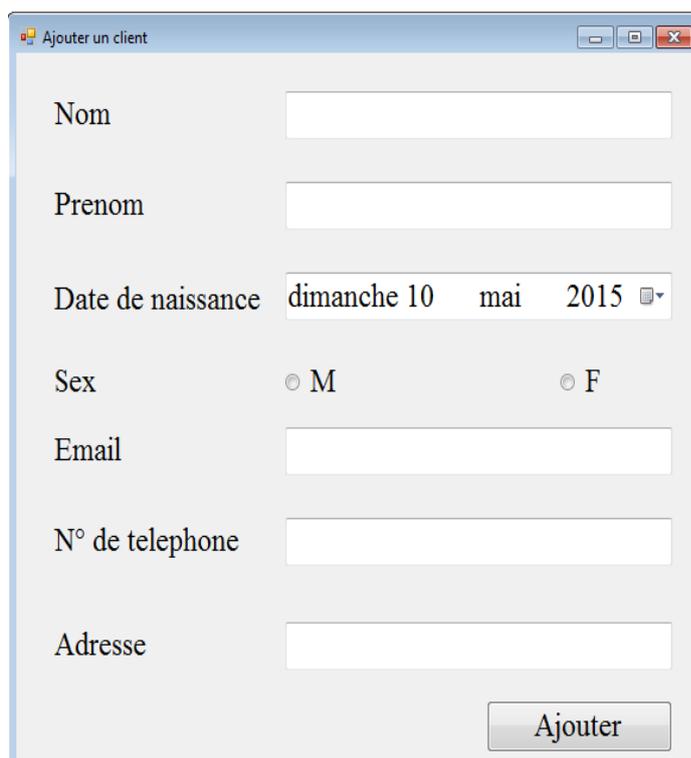
Après l'authentification la fenêtre suivantes s'affiche, cette fenêtre permet à l'utilisateur (agent et administrateur) de compléter ses taches selon ses responsabilités.



Figure 6 : Page d'accueil.

3.2.Pages gestion des clients.

- Ajouter un nouveau client



Nom	<input type="text"/>
Prenom	<input type="text"/>
Date de naissance	dimanche 10 mai 2015
Sex	<input type="radio"/> M <input type="radio"/> F
Email	<input type="text"/>
N° de telephone	<input type="text"/>
Adresse	<input type="text"/>
	<input type="button" value="Ajouter"/>

- Modifier les informations d'un client

Changer les information d'un client

Nom Prenom

Voici les informations du client cherché

Nom

Prenom

Email

N° de telephone

Adresse

3.3.Pages gestion des demandes.

- Ajouter une demande

Ajouter une demande de transport

Adresse de depart

Adresse d'arrivée

La date de la demande dimanche 10 mai 2015

L'heure de la demande 0 : 0

Nom de client

Prenom de client

- Modifier les informations d'une demande

Changer les information d'une demande

Prenom de client Nom de client

La date de la demande dimanche 10 mai 2015

Voici les informations du demande cherché

Adresse de depart

Adresse d'arrivée

La date de la demande dimanche 10 mai 2015

L'heure de la demande 0 : 0

3.4.Pages gestion des véhicules.

- Ajouter un véhicule

Ajouter un véhicule

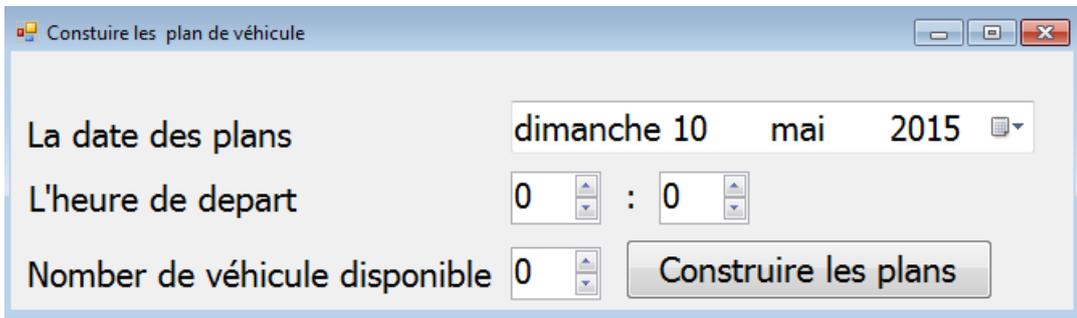
Type de véhicule

immatriculation

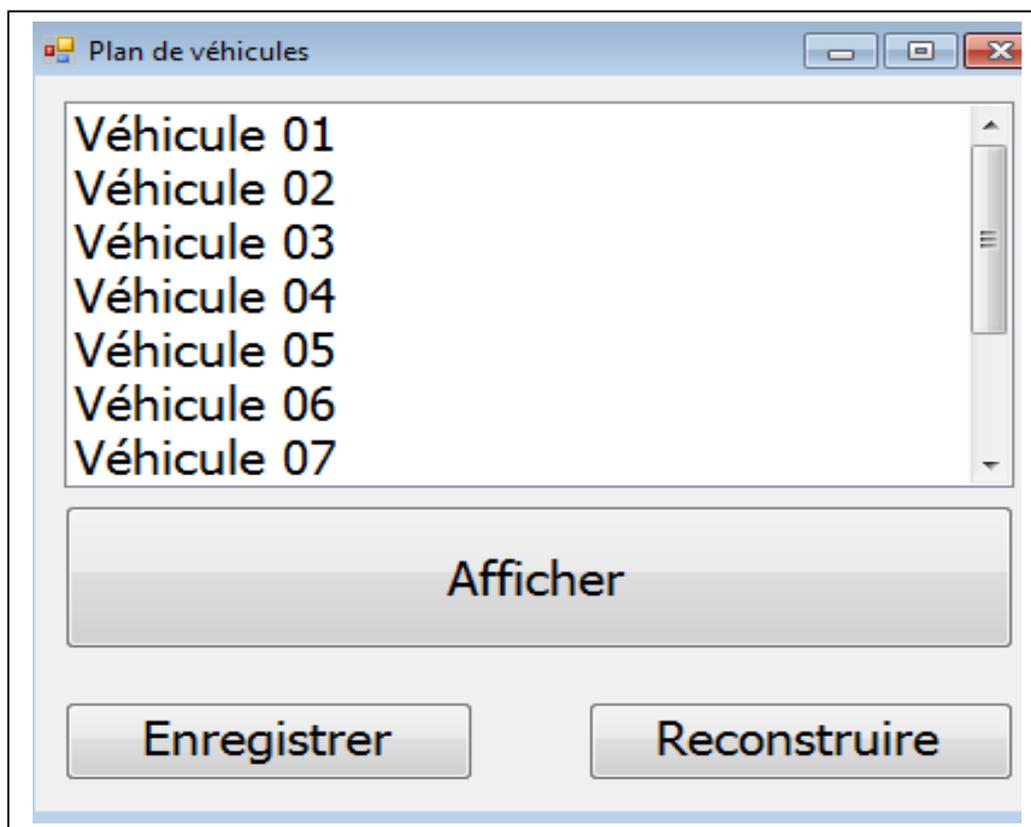
Number de place 0

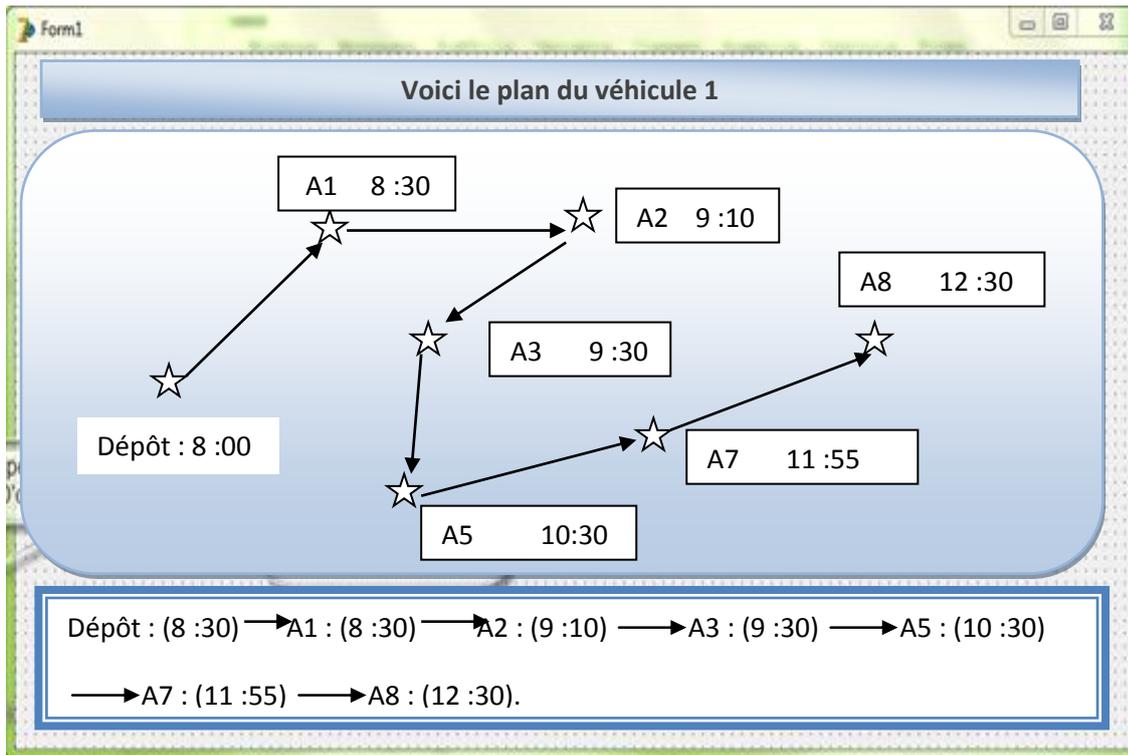
3.5.Pages gestion des plans.

- Construire les Plans de véhicules



- Afficher les plans de véhicules.





4. Conclusion.

Dans ce chapitre, nous avons décrit brièvement le processus de réalisation de notre application en spécifiant l'environnement de développement, l'implémentation et la démarche suivie pour la réalisation, ainsi nous avons présenté les interfaces essentiels de l'application en effet, nous avons achevé l'implémentation et les tests de tous les cas d'utilisation, tout en respectant la conception élaborée

Conclusion Générale

Conclusion générale

Dans ce projet, nous avons développé un système pour la gestion d'une agence de transport à la demande (TAD). Ce problème préoccupe de plus en plus les compagnies de transport, car il a un impact direct sur la qualité des services offerts aux clients. Nous nous sommes intéressés à une variante de ce problème qui est le problème de transport à la demande à plusieurs véhicules de capacité limités. Ce problème, consiste à déterminer les tournées et les horaires pour les véhicules qui effectuent le transport des clients d'une origine vers une destination en minimisant le coût total des tournées.

Avant d'aborder la construction de notre application, nous avons présenté une étude bibliographique relative au domaine du TAD. Nous avons présenté aussi les caractéristiques des diagrammes UML utilisé comme outils de modélisation dans le deuxième chapitre. Enfin, nous avons exposé les interfaces de notre application.

Ce projet a été avant tout une synthèse de notions acquises durant les trois années d'étude, il nous a permis d'acquérir de nouvelles connaissances.

Finalement nous espérons que ce modeste mémoire aura une utilité dans le cadre de La formation des future étudiants du centre universitaire de Mila.

Bibliographie :

[1]: **Le champion de la programmation Delphi 5,6 et 7, M.c BELAID.**

[2]: **Pascal ROQUES, UML 2 par la pratique étude de cas et exercices corrigés, ÉDITIONS EYROLLES, Septembre 2006.**

[3]: **Laurent AUDIBERT, UML 2-de l'apprentissage à la pratique (cour et exercice)**

[4]: **UML 2 Mise en oeuvre guidée avec études de cas ANALYSE ET CONCEPTION Joseph Gabay, David Gabay.**

Site Web : <http://www.wikipedia.org>