

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaïa



Faculté des Sciences Exactes
Département d'Informatique



MEMOIRE

Présentée par :

Farid MECHRAFI

Pour l'Obtention du Diplôme de Master en Informatique

Option :Administration et Sécurité des Réseaux

Thème

Etude du Problème de Transport à la Demande

Devant le jury composé de :

Président : BAADACHE Abderrahmane

Examineur : AISSANI Sofiane

Encadreur : ALOUI Abdelouhab

Année 2013

Remerciements

En tout premier lieu, je remercie Allah le tout puissant, à la sagesse et au savoir infinis, " *Gloire à Toi Nous n'avons de savoir que ce que Tu nous as appris. Certes c'est Toi l'Omniscient, le Sage, le tout miséricordieux le très miséricordieux* " (SOURATE AL-BAQARAH, VERSET 32).

j'exprime ma reconnaissance à Monsieur **Aloui.A** d'avoir joué son rôle de promoteur en étant à mon côté tout au long de l'étude de mon projet, ses conseils et orientations m'ont guidés jusqu'à l'aboutissement de ce travail.

je remercie également **Haroune.L** du département Informatique pour son aide et ses remarques pertinentes qui ont apporté une amélioration certaine à mon travail.

et tous ceux et toutes celles qui ont contribué de près ou de loin à l'aboutissement de mon travail trouvent ici mes sincères remerciements et l'expression de mon reconnaissance.

Un grand merci pour mes collègues du Master pour les bons moments que nous avons passé ensemble.

Dédicaces

je dédie ce modeste travail :

A mes parents,

A mes frères et A ma soeur,

A toute la famille,

A mes amis et collègues, et tout ceux qui m'ont aidé ;

Farid MECHRAFI

Table des matières

Table des matières	i
Liste des figures	v
Liste des tableaux	vi
Introduction générale	1
1 Généralités Sur Transport A La Demande	3
1.1 Introduction	3
1.2 La pratique du TAD	3
1.2.1 Définition du concept de transport à la demande	3
1.2.2 Historique	4
1.2.3 Expériences proches	5
1.2.3.1 Le Car Sharing	5
1.2.3.2 "Com'Urbus" de Dax	6
1.3 Caractéristiques et grands principes du TAD	7
1.3.1 Un nouveau mode de gestion de transport public	7
1.3.2 Fonctionnement des services de TAD	8
1.3.2.1 Cinq grands types de desserte par le TAD	8
1.3.2.2 Les types de véhicules utilisés	10
1.3.2.3 Les systèmes de réservations et de gestion	10
1.3.2.4 Les résultats des activités de tad	12
1.4 Facteurs favorables au développement de TAD	13
1.4.1 L'évolution de mode de vie	13
1.4.2 L'évolution du secteur des transports	14
1.4.3 les nouvelles technologies de communication et d'informaion	15
1.5 Conclusion	16

2	<i>Approche Centralisée Pour Résoudre Le Problème Du TAD</i>	17
2.1	Introduction	17
2.2	Systèmes Multi-Agents	17
2.2.1	Modèles d'Interaction	18
2.2.2	Mécanismes de Coordination	18
2.2.3	Composants du Système-Multi-Agent	19
2.3	Approche Centralisée dynamique	19
2.4	Architecture Centralisée	20
2.4.1	Description du problème	22
2.4.2	Algorithme	23
2.4.3	Offres et choix dynamiques	24
2.5	Conclusion	25
3	<i>Analyse Des Besoins Et Conception</i>	26
3.1	Introduction	26
3.2	Langage de modélisation et démarche adoptée	26
3.2.1	UML	26
3.2.2	Le processus unifié	27
3.2.2.1	Les phases du processus unifié	27
3.2.2.2	Le processus 2TUP	27
3.3	Etude préliminaire	28
3.3.1	Les acteurs du système	28
3.4	Capture des besoins fonctionnels	29
3.5	Les besoins non fonctionnels	30
3.5.1	Identification des cas d'utilisations	30
3.5.2	Relations entre cas d'utilisation	31
3.5.3	Diagramme de cas d'utilisation du système à réaliser	32
3.5.3.1	les cas d'utilisation de l'Utilisateur	33
3.5.3.2	les cas d'utilisation de l'Administrateur	33
3.5.3.3	les cas d'utilisation d'un Véhicule	34
3.6	Analyse	35
3.6.1	Présentation du diagramme de séquence	35
3.6.1.1	Message synchrone et asynchrone	36

3.6.1.2	Fragment d'interaction	36
3.6.2	Elaboration des diagrammes de séquence des cas d'utilisation	37
3.6.2.1	Diagramme de séquence du cas d'utilisation « Authentification »	37
3.6.2.2	Diagramme de séquence du cas d'utilisation « Gestion des véhicule ».	38
3.6.2.3	Diagramme de séquence du cas d'utilisation « Gestion utilisateur ».	39
3.6.2.4	Diagramme de séquence du cas d'utilisation « demande réservation »	40
3.7	Conception	41
3.7.1	Présentation du diagramme de classe	41
3.7.2	Diagramme de classe du système à réaliser	43
3.7.3	Le passage au modèle relationnel	43
3.7.4	Identification des relations entre les classes	44
3.8	Conclusion	45
4	<i>Réalisation</i>	46
4.1	Introduction	46
4.2	Environnement de développement de l'application	46
4.2.1	Environnement logiciels	46
4.2.2	Technologies utilisées	47
4.2.3	Le langage de programmation	49
4.3	Implémentation de la Base de Données	50
4.4	Présentation Interfaces graphiques de l'application	51
4.5	Test	53
4.6	Conclusion	53
Conclusion et perspectives		54
Bibliographie		55

Liste des figures

1.1	Répartition par date de création de 123 systèmes de T.A.D.enquêtés par ADETEC en 2004.	5
1.2	Schéma d'une ligne de TAD de type "virtuel"	9
1.3	Schéma d'une ligne de TAD de type "zonal"	10
1.4	Schéma montre les systemes de gestion de TAD	11
1.5	Les principaux facteurs favorables au developpement de tad	15
2.1	Architecture centralisée	22
3.1	Les étapes de développement suivant le processus 2TUP.	28
3.2	Représentation graphique de la relation d'inclusion.	32
3.3	Représentation graphique de la relation d'extension	32
3.4	Le diagramme de cas d'utilisation de Créer un compte et Demande réservation.	33
3.5	Le diagramme de cas d'utilisation de gérer utilisateur,gérer demande,et gérer véhicule	34
3.6	Le diagramme de cas d'utilisation de recevoir requête et envoyer réponse	35
3.7	Diagramme de séquence du cas d'utilisation « Authentification ».	37
3.8	Diagramme de séquence du cas d'utilisation « Ajout d'un véhicule ».	39
3.9	Diagramme de séquence du cas d'utilisation « Ajout d'un utilisateur ».	40
3.10	Diagramme de séquence du cas d'utilisation « demande réservation ».	41
3.11	Diagramme de Classes de l'application à réaliser	43
3.12	Transformation d'entités/classes.	44
3.13	Transformation d'entités/classes.	44
4.1	Architecture générale d'un pilote JDBC	48
4.2	Schéma générale de la base de données.	50
4.3	la base de données de Véhicule.	50

4.4	la base de données de Administrateur.	50
4.5	Interface utilisateur.	51
4.6	Interface véhicule.	52
4.7	Interface recevoir de message.	52

Liste des tableaux

1.1	Caractéristiques d'offre et de desserte de "Com'Urbus", Dax	7
3.1	Les cas d'utilisations du système à réaliser.	31

Introduction générale

Diverses évolutions sociétales apparaissent aujourd'hui comme défavorables aux transports collectifs classiques, fondés sur des transports de masse, répétitifs, dans lesquels les usagers se déplacent tous en même temps, de la même façon, sur les mêmes parcours et aux mêmes heures.

Par ces raisons vient l'apparition du transport à la demande comme un nouveau mode de transport, souple et flexible, permettant de s'adapter au mieux à la demande.

Pendant longtemps les TAD ont été gérés manuellement. C'est une caractéristique des premiers services. Mais L'introduction des Technologies de l'Information et de la Communication a contribué à relancer les TAD.

Structure du mémoire

Notre mémoire est divisé en quatre chapitres :

Le premier chapitre représente une partie introductive au transport à la demande nous présentons, leur définition, leur historique, leur caractéristiques, leur résultats d'activité, et nous finissons le chapitre par Les facteurs structurels favorables au développement de ce nouveau mode de transport.

Le deuxième chapitre, est structuré comme suit. Nous introduisons les notions d'agents et de SMA et ses caractéristiques. Ensuite, nous faisons la comparaison entre l'approche centralisée dynamique et l'approche centralisée statique, Nous proposons dans ce chapitre un système de Transport à la Demande qui adapte l'offre en temps réel à la demande afin de satisfaire des demandes de transport de clients et d'optimiser leur affectation aux

véhicules en basant sur l'architecture centralisée, à la fin de chapitre, nous proposons un algorithme permettant de résoudre quelques problèmes de transport à la demande.

Le troisième chapitre est structuré comme suit, premièrement nous présentons le cadre général et les objectifs de notre application, nous donnons la spécification de notre application, reposant sur l'analyse des besoins des différents utilisateurs de notre système. Ensuite nous introduisons le modèle de notre application sous la forme de diagrammes (de classe et de séquence).

Le quatrième chapitre, est consacré à la réalisation de l'application et l'implémentation de la solution proposée, nous présentons l'environnement logiciel de l'application, et à la fin de chapitre nous donnons les principales interfaces de notre application.

Généralités Sur Transport A La Demande

1.1 Introduction

Le transport à la demande se développe en tant que nouveau mode de transport ou plutôt comme nouvel outil de gestion du transport public. En réponse à des mobilités spécifiques et plus diffuses, le TAD semble être le moyen le plus approprié pour desservir les zones rurales, peu denses ou excentrées des bourgs et villes-centres.

Développé en milieux urbains, périurbains ou ruraux, le TAD répond aux nouvelles mobilités urbaines liées aux évolutions culturelles, démographiques, économiques ou encore professionnelles.

Ce chapitre comprend donc trois parties :

La première partie est consacrée à la définition du concept de TAD et de son historique et de deux expériences proches de TAD.

La deuxième présente les éléments caractéristiques du TAD, l'ensemble de ses modalités de fonctionnement, et les principaux résultats de ses activités.

La dernière partie est consacrée aux facteurs favorables au développement de TAD.

1.2 La pratique du TAD

1.2.1 Définition du concept de transport à la demande

Le concept de TAD, est défini comme un système de transport de personnes qui implique que le service soit déclenché par la ou les personnes qui souhaitent l'utiliser.

Dans son rapport d'étude effectué pour le compte de la D.A.T.A.R., la D.T.T. et l'A.D.E.M.E., ADETEC propose la définition suivante : « On appelle transport à la demande un service de transport ne fonctionnant que sur appel préalable d'un ou plusieurs clients. Le T.A.D. se distingue du taxi par le fait qu'au moins un des trois aspects suivants est fixé : les jours de fonctionnement, les horaires de dessertes ou les lieux de destination »[1].

Le TAD est une notion qui regroupe tous les services publics de transport dont tout ou partie ne s'effectue qu'à la demande expresse de ceux qui les utilisent. Ils se distinguent des transports réguliers qui s'effectuent en fonction de programmes prédéfinis.

Dans ce contexte, ce transport flexible se situe d'une part entre les transports publics réguliers, et d'autre part les services de transports occasionnels et les taxis.

1.2.2 Historique

Il est difficile d'établir une chronologie exacte de l'apparition et du développement du transport à la demande. Cependant, les diverses sources bibliographiques traitant du sujet permettent d'approcher cet objectif.

Selon le G.A.R.T. il semble que l'apparition du concept de transport à la demande remonte au milieu de la décennie 1970, avec la réalisation en 1974 des premiers schémas régionaux de transport. Ces schémas préconisaient la mise en place de système de ce type pour desservir les espaces ruraux les plus faiblement peuplés.

Nous pouvons également citer le cas du département de l'Aveyron, lui aussi pionnier en la matière. Au total, ce département avait connu, au 1er Janvier 2004, la création de 34 T.A.D. ruraux gérés par des autorités organisatrices de second rang. 25 de ces services ont été créés entre 1980 et 1986.

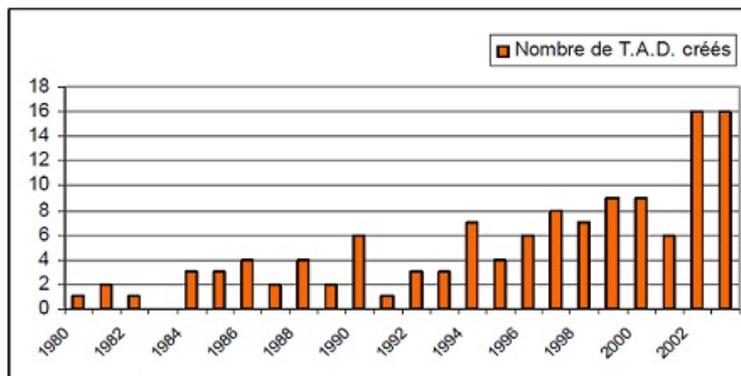


FIGURE 1.1 – Répartition par date de création de 123 systèmes de T.A.D.enquêtés par ADETEC en 2004.

Le graphique précédent présente la répartition de la date de création des 123 services de transport à la demande, qui a fourni cette information lors de l'enquête par questionnaire d'ADETEC en 2004. Les chiffres concernent des T.A.D. ruraux comme périurbains. Nous remarquons, après une augmentation faible du rythme de création sur l'ensemble des années 1980, une nette croissance des créations de T.A.D. à partir de 1992, qui semble s'accélérer dans les années 2000 [2].

Les raisons de l'apparition du transport à la demande sont aisément compréhensibles. L'explication est notamment fournie dans le livre blanc des transports départementaux : Les transports collectifs interurbains sont confrontés au problème de la desserte d'espace ou la faible densité de population est associée à une concurrence forte de la voiture particulière. Le marché que représentent les déplacements susceptibles d'emprunter les transports collectifs y est donc particulièrement étroit, et se limite souvent aux quelques rares personnes non motorisées [3].

1.2.3 Expériences proches

1.2.3.1 Le Car Sharing

Le Car Sharing propose la mise en commun d'une flotte de véhicules. Cette démarche est particulièrement développée en Suisse et en Allemagne, elle remplace efficacement la voiture individuelle, notamment la seconde voiture, souvent sous-utilisée par leurs propriétaires. Les adhérents peuvent faire leurs courses ou organiser une sortie en payant le

véhicule à l'usage. L'adhérent évalue ainsi le coût de ses déplacements et peut choisir le mode de transport le mieux adapté à ses besoins.

Le partage dans le temps de l'usage de véhicules est un concept attirant économiquement grâce à la mutualisation des investissements et à l'optimisation de l'espace urbain.

Les véhicules sont à la disposition des adhérents dans des parkings. L'abonné peut réserver un véhicule 24 heures sur 24, 7 jours sur 7, grâce à un service téléphonique. Avec sa carte d'adhérent, l'abonné récupère la clé du véhicule dans un boîtier. Une puce électronique désactive l'antivol et enregistre les données kilométriques dès la récupération de la clé.

De retour au parking, le système établit la facture qui est débitée automatiquement sur le compte de l'abonné. Des expériences de Car Sharing ont vu le jour à Strasbourg, Lyon, Grenoble, Marseille et Toulouse [4].

1.2.3.2 "Com'Urbus" de Dax

"Com'Urbus" a été créé le 15 octobre 2004 par la Communauté d'Agglomération du Grand Dax suite à l'extension de son PTU (2002).

Avec ce service, les habitants bénéficient d'une nouvelle offre de proximité qui permet de desservir les communes rurales, en dehors des itinéraires des lignes régulières situées au cœur de l'agglomération (Dax et St-Paul-Lès-Dax). Le service de TAD relie les communes excentrées au cœur de l'agglomération.

Pour utiliser le service, une adhésion préalable et gratuite est nécessaire, il suffit de réserver son trajet la veille au plus tard. Le client doit préciser à l'opérateur les points suivants : les arrêts, les heures d'aller et retour du voyage, son identité et le nombre de personnes à transporter, le pôle de destination [5].

	Caractéristiques principales de "Com'Urbus"
Type de desserte	<ul style="list-style-type: none"> - 71 points d'arrêts dans 13 communes - 16 pôles de destination - 2 villes centres : Dax et St-Paul-Lès-Dax - Arrêts et pôles matérialisés - Horaires et itinéraires variables selon réservations
Amplitude du service	Tous les jours sauf les dimanches et jours fériés de 8h30 à 18h30
Fréquence	2 allers/retours par jour (sauf mercredis et samedi : 4 allers :retours)
Véhicules utilisés	4 minibus de 8 places, accessibles aux PMR

TABLE 1.1 – Caractéristiques d'offre et de desserte de "Com'Urbus", Dax

1.3 Caractéristiques et grands principes du TAD

1.3.1 Un nouveau mode de gestion de transport public

La définition la plus explicite du TAD est sans doute celle proposée par François Ascher[6] : "C'est une notion qui englobe tous les services de transport dont tout ou partie n'effectue qu'à la demande expresse de ceux qui l'utilisent".

Le TAD se distingue des services réguliers car il se situe à la fois entre le taxi et les transports collectifs et il répond à une demande non pas a priori, mais bien réelle.

C'est une innovation en matière de transports publics qui repose sur une offre à assurer, à partir d'une demande exprimée par les usagers, qui dicte les besoins auxquels le service doit s'adapter. Ce service de transport permet donc de répondre aux nouvelles mobilités puisqu'il propose une offre souple et flexible.

Aujourd'hui, le TAD répond aux besoins de plus en plus individualisés des usagers. Il permet de gérer, presque au cas par cas, les déplacements des individus, dans des secteurs non desservis par des lignes régulières. C'est également un moyen de favoriser le développement local, de proposer une certaine équité territoriale et sociale et de limiter l'usage de la voiture particulière.

Il existe presque autant de formes et de services de TAD que de territoires : chaque collectivité ou Autorité Organisatrice de Transports (AOT) développe le type de TAD qui lui convient. De ce fait, les TAD remplissent des fonctions diverses et prennent de nombreuses formes, d'où une difficulté à être exhaustif dans leur présentation.

Le TAD se présente comme un service de transport souple, flexible, presque individualisé et répondant à une demande réelle, exprimée par l'utilisateur. Ce n'est pas un "mode" à proprement parler puisque les services de TAD fonctionnent avec des véhicules pouvant être utilisés sur des lignes régulières.

Le TAD est plutôt un mode de gestion du transport public et un outil destiné à compléter l'offre de transport classique, constitue aujourd'hui un moyen de transport alternatif, souple et flexible permettant de s'adapter au mieux à la demande [7].

Le TAD nécessite une réservation préalable. Selon un délai qui peut aller de deux jours à trente minutes, l'utilisateur doit « commander » à l'avance un déplacement.

La réservation préalable est la clé de voûte du transport à la demande. Cette caractéristique le différencie en premier lieu des lignes régulières. Elle présente un certain nombre d'avantages financiers et en matière de gestion. Comme la demande est connue à l'avance, le transporteur peut s'organiser en adaptant l'itinéraire et le véhicule en fonction de ses commandes. Elle permet également d'éviter que les véhicules roulent à vide. D'une manière générale, elle permet de rationaliser les coûts [7].

1.3.2 Fonctionnement des services de TAD

la typologie des services étudiés est très variée et on y retrouve la plupart des modes d'exploitation de TAD en zone périurbaine, puisque ce sont ceux là qui bien évidemment présentent un caractère de complémentarité avec les autres modes de transport collectif opérant sur la même zone géographique [8].

1.3.2.1 Cinq grands types de desserte par le TAD

- **La ligne virtuelle** : Ligne dont tout ou partie de l'itinéraire est desservi uniquement sur réservation. La prise en charge de l'utilisateur se fait à un arrêt matérialisé et défini à l'avance, jamais à domicile. Deux types de lignes virtuelles existent :

- Itinéraires, arrêts et horaires prédéterminés,
- Arrêts et horaires fixes "seulement" (itinéraire en fonction de la demande).

La ligne virtuelle est plutôt efficace en substitution des lignes régulières en heures creuses, vers des secteurs d'activités plus ou moins importants. Elle permet également de compléter un réseau lors d'une extension de PTU ou d'effectuer une desserte spécifique.

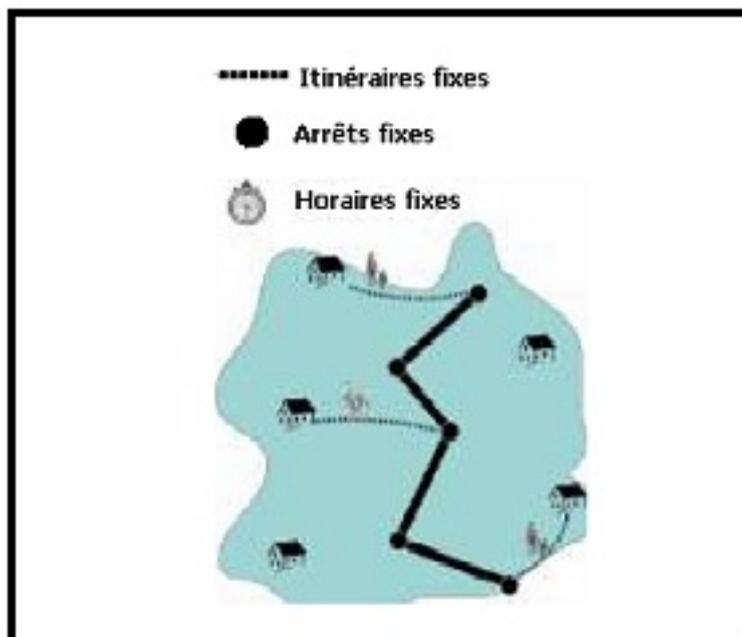


FIGURE 1.2 – Schéma d'une ligne de TAD de type "virtuel"

- **le service collectif en porte-à-porte (horaires libres) :**

L'utilisateur est pris en charge à son domicile et déposé à son lieu de destination. C'est le système le plus proche du taxi. Ce système est souvent utilisé pour les déplacements des personnes âgées ou handicapées et pour les liaisons intercommunales. Cela peut être comme un système zonal.

- **la desserte arrêt-à-arrêt avec horaires libres :**

L'utilisateur définit l'heure de passage en fonction des plages horaires disponibles. Ce type de desserte permet d'assurer un rabattement sur le réseau régulier.

- **La desserte zonale :**

La prise en charge de l'utilisateur se fait au domicile ou à des points d'arrêt prédéterminés ou non, pour des destinations fixées à l'avance. Les itinéraires sont réalisés en fonction des demandes, les horaires sont définis lors de la réservation,

qui active ce type de ligne.

Ce type de desserte semble plus efficace pour les flux de déplacements plus diffus dans des zones peu denses.

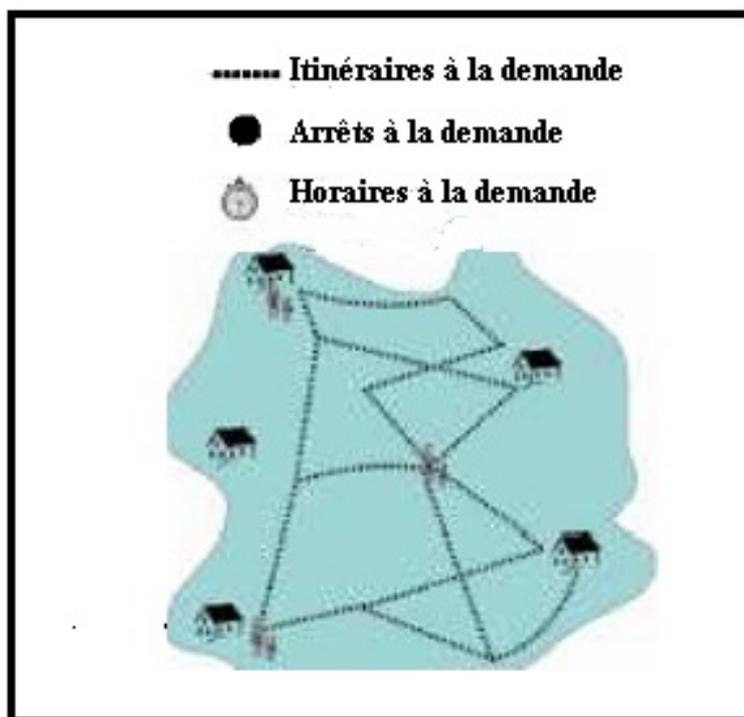


FIGURE 1.3 – Schéma d'une ligne de TAD de type "zonal"

– **la desserte domicile-pôle d'échange :**

Les horaires sont généralement prédéfinis, c'est un système intermédiaire entre la ligne virtuelle et la desserte zonale [5].

1.3.2.2 Les types de véhicules utilisés

Quatre principaux types de véhicules sont utilisés pour les activités de TAD :

1. taxi, avec véhicule standard : au moins 4 places
2. monospaces : de 4 à 8 places,
3. minibus standard : de 4 à 15 places,
4. minibus aménagés, notamment pour les services PMR, en moyenne 8 places 5.

1.3.2.3 Les systèmes de réservations et de gestion

La réservation se fait généralement par téléphone, pour l'ensemble des usagers (même si les services sont plus spécifiques pour les TAD à vocation PMR).

Le délai de réservation est plus ou moins long selon la zone et selon la souplesse souhaitée par l'AOT. En milieu rural par exemple, les usagers peuvent réserver jusqu'à la veille de leur déplacement tandis qu'en urbain ou interurbain, la réservation peut s'effectuer jusqu'à 1h avant le départ. Cependant, plus les systèmes sont souples, plus les contraintes d'exploitation seront fortes et donc plus le coût du service sera élevé, à la fois pour l'exploitant et le client.

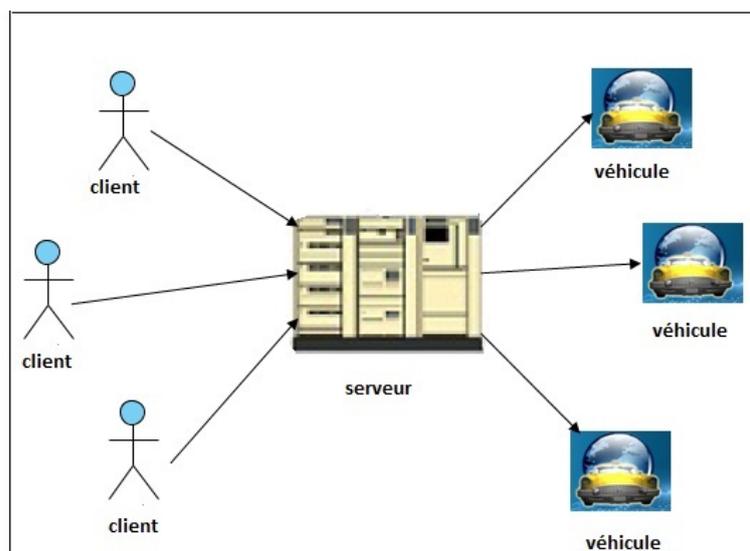


FIGURE 1.4 – Schéma montre les systèmes de gestion de TAD

Trois principaux systèmes de gestion/ réservation existent :

- **Les systèmes d'exploitation "simples"** : Dans ce cas, les exploitants n'utilisent pas de logiciels dédiés aux TAD. C'est une simple gestion des courses avec utilisation de logiciels bureautiques de type tableur (Excel par exemple) et base de données (Access). La prise en charge de la clientèle est effectuée via une centrale de réservation ou par simple appel téléphonique 8.
- **Les systèmes d'exploitation "propriétaires"** : Sont développés spécifiquement pour et/ou par l'exploitant ou le prestataire du service. Ces outils possèdent au moins des fonctions de réservation, de planification et d'optimisation des courses[5].
- **Les logiciels spécifiques** : Permettent d'organiser les réservations, de planifier les courses et de les optimiser. Ils peuvent également assurer le suivi de la clientèle ou réaliser des cartographies. Ce sont également des outils de gestion puisqu'ils permettent d'établir des facturations.

Les fonctionnalités les plus demandées consistent à : saisir les réservations, assurer un serveur vocal, gérer les fichiers des clients, gérer l'organisation des courses : disponibilité des conducteurs et des véhicules, réaliser la facturation, le suivi, les statistiques, effectuer des cartographies et des localisations.

Le choix d'utiliser un des trois systèmes dépend de plusieurs facteurs comme le prix d'acquisition ou de réalisation du logiciel, les besoins de l'exploitant, les fonctionnalités disponibles, les coûts techniques et l'ampleur du service de TAD.[5].

1.3.2.4 Les résultats des activités de tad

Les constructeurs automobiles se sont emparés de ce créneau en proposant des véhicules toujours moins polluants et plus économes en carburant. Si des avancées ont été effectivement réalisées dans le domaine et ont apporté des solutions au problème de pollution ou de montée des prix du carburant, celles-ci ont été limitées jusqu'ici par l'augmentation du nombre de déplacements et de distances parcourues.

De plus, elles ne résolvent toujours pas les problèmes de congestion et de circulations constatées dans les grandes agglomérations.

Le transport à la demande est un mode de transport qui présente la particularité de nécessiter une réservation préalable. Celle-ci en fait un mode économe en carburant (le service ne fonctionne qu'en cas de réservation), optimisé (le trajet est calculé par rapport aux réservations passées) et aussi peu coûteux (les véhicules utilisés sont généralement de petite taille, i.e. minibus, monospaces et parfois taxis associés en sous-traitance). Ainsi, à service équivalent, les TAD sont plus rationnels que des lignes de bus circulant à horaires réguliers.

De ce fait, le transport à la demande a connu ces dernières années une croissance assez importante. En effet, les autorités organisatrices de transport, quels que soient leur taille et leur statut (communes, communautés de communes, communautés d'agglomération, communautés urbaines, conseils généraux) ont vu dans le TAD une solution rationnelle et relativement facile à mettre en oeuvre pour résoudre leurs problèmes.

Cette croissance s'est manifestée par l'apparition de nouveaux services de TAD, tantôt dédiés à la desserte des périphéries des villes ou des espaces ruraux, tantôt voués à assurer des déplacements nocturnes ou spécifiques au coeur des villes [7].

Atouts et Limites du TAD

Avantages :

- Souplesse dans le temps.
- Desserte de zones d'activités difficiles d'accès.
- Amélioration de l'image du réseau.
- Gagner un nouveau marché.

Limites :

- Accès pour les personnes extérieures du réseau.
- Obligation de programmer ses déplacements à l'avance : pas toujours possible.
- Systèmes parfois rigides.

1.4 Facteurs favorables au développement de TAD

Diverses évolutions sociétales apparaissent aujourd'hui comme défavorables aux transports collectifs classiques, fondés sur des transports de masse, répétitifs, dans lesquels les usagers se déplacent tous en même temps, de la même façon, sur les mêmes parcours et aux mêmes heures.

Mais les exigences environnementales comme la lutte contre les inégalités appellent au développement de services publics de mobilité permettant à tous de se déplacer dans l'ensemble des agglomérations urbaines, tout en économisant les ressources naturelles non renouvelables.

1.4.1 L'évolution de mode de vie

Notre société est animée par une tendance lourde, multiséculaire, d'autonomie croissante des individus. Peu à peu, cette dynamique d'individuation pénètre tous les domaines de la vie sociale, modifiant les territoires, les temporalités, la nature et la forme des liens sociaux.

Les individus privilégient les modes de transport et les localisations d'activités qui leur semblent offrir à la fois les potentiels de choix les plus larges, et le plus d'autonomie dans la construction de leurs espaces-temps et de leurs programmes d'action. Les zones urbaines denses et les transports collectifs disposent évidemment de certains atouts dans

cette perspective d'un citoyen rationnel, mais ils se heurtent à des limites de densité et d'accessibilité.

Celles-ci réclament donc des modes de transports plus polyvalents, ou la possibilité de combiner plusieurs modes de transport.

Les citoyens ont également des exigences de confort qui leur font choisir tel mode de déplacement en raison du temps météorologique, ou en fonction des objets qu'ils transportent ou encore des personnes avec lesquelles ils peuvent voyager. Ces diverses possibilités de se déplacer accroissent encore plus la variété des déplacements et l'irrégularité des flux.

L'offre de transports est de plus en plus confrontée à cette évolution qui ouvre la possibilité d'inventer ou de réinventer de nouveaux produits et services. De fait, on voit fleurir aujourd'hui des modes nouveaux, des utilisations nouvelles de modes anciens (bicyclette et multimodalité par exemple), des modes d'accès nouveaux à certains modes (les coopératives automobiles, le covoiturage, le libre-service), le remplacement du déplacement de personnes par des livraisons d'objets, et des services de transports collectifs à la demande. Chaque individu tend à élargir la gamme des moyens qu'il utilise pour bouger dans la ville ou pour faire bouger les personnes et les objets [9].

1.4.2 L'évolution du secteur des transports

Les cloisonnements traditionnels entre transports urbains et non urbains, transports privés et transports publics, transports collectifs ferrés et transports collectifs routiers, transports réguliers et transports à la demande, perdent sensiblement de leur importance.

Certes, les évolutions sont plus ou moins rapides, car elles sont freinées notamment par des dispositions juridiques, par des logiques économiques différentes, par des obstacles institutionnels et politiques.

Toutefois, le contexte entrepreneurial actuel est de plus en plus favorable au dépassement de ces cloisonnements. Cela pourrait contribuer à promouvoir l'intermodalité au sein même de ces sociétés.

Par ailleurs, on peut aussi faire l'hypothèse que l'ouverture à la concurrence au niveau international obligera les sociétés de transports publics à innover. Ces évolutions

pourraient participer à créer les conditions d'une approche nouvelle des TAD dans les prochaines années [9].

1.4.3 les nouvelles technologies de communication et d'informaion

Pendant longtemps les TAD ont été gérés manuellement. C'est une caractéristique des premiers services. L'introduction des Technologies de l'Information et de la Communication (TIC) à partir des années 90 a contribué à relancer les TAD.

Les TIC ont également ouvert de nouvelles perspectives pour développer de services plus flexibles et plus en adéquation avec les besoins des usagers.

Le TAD en tant que mode de service public n'a pas à assurer les déplacements individuels de type taxi, c'est un mode collectif, qui a vocation à transporter plusieurs personnes à la fois. Or, cette tâche est difficilement réalisable manuellement dès que les flux deviennent importants. Le regroupement des passagers nécessite l'emploi d'un logiciel d'optimisation capable de gérer les détours, les plus courts chemins... Le regroupement peut être incité par la tarification ou le type de desserte choisi. Mais sur des territoires vastes, denses ou lorsque le nombre de voyages par jour dépasse une certaine limite, l'emploi de solutions automatisées devient obligatoire.

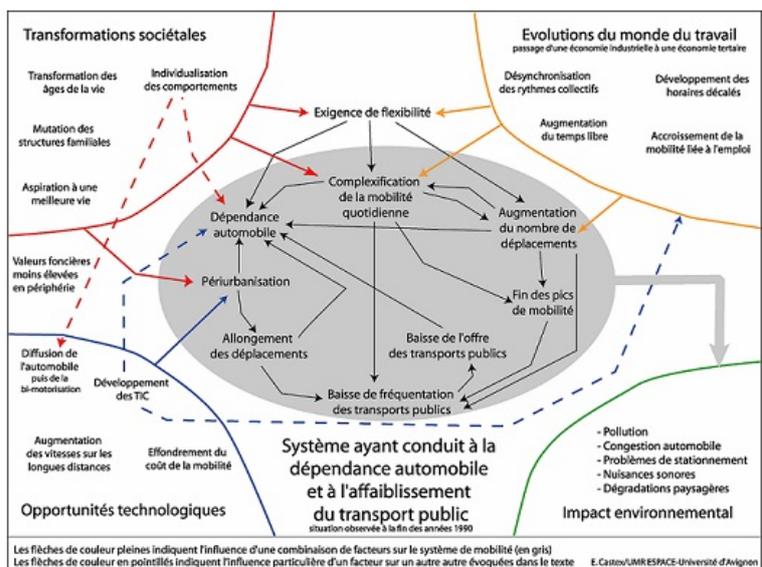


FIGURE 1.5 – Les principaux facteurs favorables au développement de tad

Les principaux facteurs évoqués ont été regroupés dans les 3 angles de la figure. Ils ont été rassemblés par catégories. Le premier groupe en haut à droite désigne les transformations qui ont affecté la société, le deuxième, celles qui ont touché le monde du travail et enfin le troisième, en bas à gauche, les opportunités techniques. L'ensemble de ces facteurs a engendré un système favorable à la dépendance automobile au détriment des transports publics (en gris au centre) aux lourdes conséquences environnementales représentées dans le quatrième angle en bas à droite[7].

1.5 Conclusion

Le transport à la demande est un mode de transport public (bien qu'il existe d'autres formes de TAD) soumis à une réservation préalable, qui doit être passée plus ou moins longtemps à l'avance. Il est un mode collectif, économe, réservé au départ à des « niches commerciales » mais dernièrement, gagne chaque année de nouveaux territoires et tend à se développer sous des formes de plus en plus flexibles.

Ce premier chapitre s'est attaché à présenter des généralités sur le TAD, de la définition du concept, et ses caractéristiques aux facteurs favorables à son développement.

On a vu, dans ce chapitre les systèmes de transports publics actuels sont déterministes et basés sur des itinéraires fixés à l'avance, c'est la demande qui s'adapte à l'offre. Nous proposons un système de transport à la demande qui doit à l'opposé adapter l'offre à la demande, et ce que nous verrons en détail dans le chapitre suivant.

Approche Centralisée Pour Résoudre Le Problème Du TAD

2.1 Introduction

Les systèmes de transports publics actuels sont déterministes et basés sur des itinéraires fixés à l'avance, et sont peu adaptatifs à une demande très changeante, c'est la demande qui s'adapte à l'offre. Nous proposons un système de transport à la demande qui doit à l'opposé adapter l'offre à la demande. Il doit permettre à terme de générer une forte diminution du trafic et d'offrir une qualité de service maximale à un moindre coût d'exploitation. ce qui n'est pas offert par les systèmes de transport classiques. Dans ce chapitre, nous utiliserons l'architecture centralisée et système multiagent.

2.2 Systèmes Multi-Agents

Un SMA est un ensemble d'agents autonomes, situés dans un environnement, et interagissant entre eux.

Un SMA est un réseau d'agents couplés qui interagissent afin de résoudre des problèmes qui dépassent les capacités ou les connaissances de chacun[10].

Le TAD se prête bien à une modélisation multi-agent pour les raisons suivantes. D'abord, Il s'agit d'un problème difficile, pour lequel les méthodes exactes ne traitent que des versions de petite taille, qui ne peuvent représenter des applications réelles. Le choix d'une modélisation permettant une distribution des traitements peut être une solution afin de proposer des temps de réponses courts face à des demandes clients. Ensuite, il s'agit d'un problème qui suppose une gestion dynamique des données. Par définition, toutes les données ne sont pas disponibles avant le démarrage de l'exécution et de plus, le système doit pouvoir s'adapter à la réalité du réseau de transport : perturbation, pannes,

etc. Par ailleurs, avec le développement technologique, il est raisonnable de considérer des véhicules avec des capacités calculatoires embarquées. Dans ce contexte, le problème est, de fait, distribué et nécessite une modélisation adaptée afin de tirer profit des capacités embarquées des véhicules.

2.2.1 Modèles d'Interaction

Une interaction est une mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques.

Les agents interagissent à travers un ensemble d'événements pendant lesquels les agents sont en relation les uns avec les autres soit directement soit par le biais de l'environnement.

La notion d'interaction suppose :

- La présence d'agents capables d'agir et/ou de communiquer.
- Un certain jeu dans les relations entre les agents leur permettant à la fois d'être en relation, mais aussi de pouvoir se séparer de cette relation, c'est-à-dire de disposer d'une certaine autonomie.

2.2.2 Mécanismes de Coordination

Plusieurs approches existent pour la coordination des agents dans un SMA, que nous présentons dans ce qui suit :

- Planification multi-agent.
- Résolution distribuée de problèmes.
- Structuration organisationnelle.
- Coordination par protocoles.
- Négociation.

– Planification Multi-Agent

Pour éviter des actions conflictuelles, une approche majeure en coordination multi-agent est la planification multi-agent. Un plan détaille les actions et interactions futures des agents du système, nécessaires à la réalisation de leurs buts.

– Résolution Distribuée de Problèmes

Dans la résolution distribuée de problèmes, les agents sont coopératifs et essaient d'aboutir collectivement à une solution au problème global.

– **Structuration Organisationnelle**

La structuration organisationnelle exploite le fait que les structures organisationnelles définissent les responsabilités et les capacités de chaque agent.

– **Coordination par Protocoles**

Le principe de la coordination par protocoles est de définir un ordonnancement des messages échangés, qui doit être respecté par les agents participants.

– **Négociation**

La négociation, est le processus de communication d'un groupe d'agents afin d'atteindre un accord mutuellement accepté sur un certain sujet [11]

2.2.3 Composants du Système-Multi-Agent

Notre système est composé d'un ensemble dynamique d'agents interagissant pour aboutir à une solution au problème.

Nous définissons trois catégories d'agents :

- **L'agent Véhicule** représente chaque véhicule dans le système, il gère sa tournée et exécute des modifications de sa tournée en cas de changement.
- **L'agent Client** modélise chaque client du système. Il envoie sa demande et communique avec l'agent Administrateur. L'agent Client et l'agent véhicule représentent les acteurs du monde réel dans le système.
- **L'agent Administrateur** gère la relation entre le SMA. En particulier, il est responsable de l'interaction avec les usagers du système TAD.

2.3 Approche Centralisée dynamique

Les problèmes opérationnels de tournées de véhicules sont rarement statiques, et nous pouvons dire qu'aujourd'hui, un système statique n'a pas de chance de satisfaire les besoins en mobilité des utilisateurs. En effet, dans un contexte opérationnel, et même si l'ensemble des requêtes est connu avant l'exécution, il existe toujours quelques éléments qui rendent le problème dynamique. Ces éléments englobent les pannes, les retards, les voyageurs ne se présentant pas, etc. Il est ainsi toujours utile de considérer un problème qui n'est pas totalement statique.

Quelques différences entre le cas dynamique et le cas statique :

- **La dimension temps est essentielle** Dans le cas statique, la dimension temps peut être ou ne pas être importante; dans le cas dynamique, le temps l'est toujours; le système doit connaître en permanence la position de tous les véhicules, spécialement lors de la réception de requêtes.
- **Le problème peut être ouvert** Le processus de résolution est temporellement borné dans le problème statique. Un véhicule quitte son dépôt avec un plan complet, partant et revenant au dépôt. Dans une configuration dynamique, le processus peut très bien être non borné, les véhicules pouvant disposer de chemins à suivre, qui changent dynamiquement au lieu d'avoir un plan complet.
- **L'information future peut être imprécise ou inconnue** Dans le problème statique, l'information est connue. Le futur dans le problème dynamique n'est jamais connu avec assurance. Au meilleur des cas, on peut en avoir une information probabiliste.
- **La mise à jour des informations est essentielle** Toutes les données utilisées par un problème dynamique sont théoriquement sujettes à des changements durant l'exécution. Il est par conséquent essentiel que les mécanismes de mise à jour des informations soient intégrés dans la méthode de résolution. Naturellement, la mise à jour des informations est sans objet dans le cas statique.
- **La gestion de la file d'attente des clients à traiter est importante** Si le nombre de clients demandant un service devient important, le système devient congestionné et les algorithmes utilisés condamnés à donner des résultats de moindre qualité[12]

2.4 Architecture Centralisée

Le scénario d'exécution commence avec la première demande d'un usager qui apparaît aléatoirement dans un endroit de la ville. Il envoie une requête en désignant sa position, et le nombre de places désiré. La résolution consiste à choisir le véhicule le mieux placé pour satisfaire les demandes.

Dans cette architecture, le véhicule qui correspond à un client sera celui dont le coût pour transporter le client est minimal.

Pour connaître ce coût (distance), le véhicule envoie à l'agent Administrateur sa po-

sition, et le client aussi envoie sa position à l'administrateur, ce dernier calcule le coût (distance) entre le client et chaque véhicule. Afin de traiter des demandes avec une méthode centralisée, différentes approches peuvent être envisagées. D'abord, tirer profit des méthodes d'optimisation, tout en leur permettant de répondre à des demandes dynamiques. Le principe est de discrétiser les temps de traitement et de faire tourner le système pendant un certain temps défini en considérant les clients connus jusque-là. La deuxième alternative la plus prometteuse est celle suivie par la majorité des approches distribuées traitant des problèmes dynamiques : Traitant chaque nouveau client indépendamment de ses successeurs, cette approche traite les nouveaux clients d'une manière naturelle, qui consiste à leur trouver un véhicule approprié.

Dans l'architecture centralisée (voir Figure 2.1), l'ensemble des demandes des clients sont traitées par un même « agent ». Il dispose de toutes les informations nécessaires sur chaque véhicule et sur chaque client : l'occupation des véhicules, ainsi que leurs positions actuelles et la demande du client et son état actuel. Ayant toutes ces informations, il attribue à chaque client le véhicule le plus approprié pour le desservir.

Le scénario utilisé est le suivant : à un instant quelconque, un utilisateur entre en interaction avec un agent Administrateur, l'agent Utilisateur envoie sa demande à l'agent Administrateur, qui essaie de l'insérer dans la tournée de chaque véhicule du SMA. Pour ce faire, il exécute séquentiellement, pour chacun des véhicules, l'algorithme d'ordonnancement, et retient celui qui a le coût minimal. Enfin, l'agent Administrateur envoie la tournée courante à l'agent Véhicule concerné et informe l'agent Utilisateur de son véhicule. Les agents Véhicule n'effectuent donc aucun calcul et se contentent d'accuser réception de leur tournée courante, et mettent à jour leurs informations.

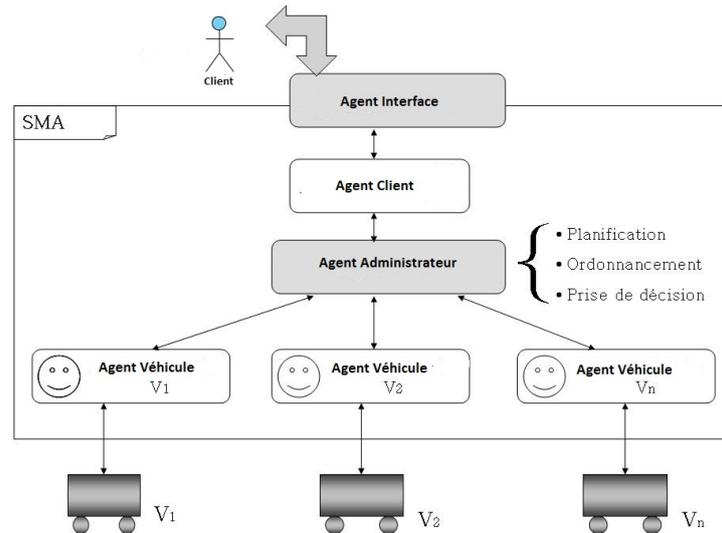


FIGURE 2.1 – Architecture centralisée

2.4.1 Description du problème

Le modèle qu'on va proposer essaie d'arbitrer entre différentes contraintes. Chaque usager souhaite :

- Être pris rapidement en charge une fois sa demande acceptée.
- Être amené à destination dans des délais n'excédant "pas trop" le temps minimal pour aller de leur provenance à leur destination.
- Ne pas voir son temps de transport augmenter trop brutalement (écarts trop manifestes par rapport à sa perception d'un bon itinéraire, demi-tours . . .).

Pour chaque véhicule, les transporteurs essaient :

- D'optimiser le taux de remplissage en dérivant d'un itinéraire déjà planifié pour prendre en charge une nouvelle demande.
- De négocier avec l'agent Administrateur afin de savoir celui qui est le mieux positionné pour servir une nouvelle demande.

Le système est composé des agents suivants : agent Véhicule, agent Administrateur et agent Utilisateur.

Un usager se connecte au système via un support donné (serveur), il est alors instancié par un agent Utilisateur qui a pour fonction de le représenter au sein du système. L'utilisateur désigne sa position comme indiqué précédemment. Ainsi, l'agent Utilisateur entre en interaction avec l'agent Administrateur. Ce dernier, diffuse la demande de l'utilisateur aux agents Véhicules.

2.4.2 Algorithme

Algorithme : Comportement de système

Tant que client demande **faire**

envoyer position client ;

envoyer nombre place demandé ;

Tant que demande à l'administrateur != vide **faire**

selectionne les véhicules qui ont le nombre de place demandé ;

diffuser la demande ;

Pour chaque vehicule de la flotte **faire**

Tant que non fin de tournée **faire**

message = consultation de boîte à lettre ;

Si message! = vide **alors**

Si message appartient demande client **alors**

si nombre place libre existe **alors**

envoyer nombre place ;

envoyer position véhicule ;

Fin Si

sinon

envoyer nombre place libre=nul ;

Fin Si Fin tant que Fin pour

Si message appartient réponse véhicule **alors**

calculer les coûts ;

Si tous les réponses sont arrivées **alors**

classer les coût ;

Fin si

vainqueur = tete (liste des coûts) ;

diffuser le vainqueur ;

Fin si

Pour chaque véhicule **faire**

Si message appartient un vainqueur **alors**

modifier itinéraire ;

Fin si Fin pour Fin pour Fin tant que

2.4.3 Offres et choix dynamiques

Notre modèle est basé sur deux phases simultanées, une phase d'offres et une phase de choix. On cherche à établir un accord entre les propositions de transport et les intérêts du client. Un élément clef du système est l'appariement entre transporteur et usager. Quel est le meilleur transporteur pour une demande donnée ? Qui le détermine et comment ? Comment le transporteur sait-il qu'il a été choisi ? Ces questions ne sont d'ailleurs pas indépendantes.

Le transporteur qui correspondra à l'utilisateur sera choisi en essayant de minimiser le coût pour le transporteur.

Déterminer le véhicule choisi consiste alors à sélectionner celui de moindre coût parmi ceux qui peuvent prendre en charge le nouveau client.

A chaque apparition d'un nouveau client, il diffuse sa demande à l'Administrateur contenant sa position et nombre de place désiré. puis l'administrateur, diffuse la demande de client à tous les véhicules qui ont le nombre de place demandé . Chaque agent Véhicule envoyer ensuite au système un message contenant leur identifiant ainsi que leur position, et le nombre de place libre, s'il a de nombre de place libre satisfait la demande de client, mais dans le cas contraire (pas de place libre), l'agent véhicule envoi un message à l'Administrateur pour l'informer que y a de place libre qui satisfait la demande. Pour le traitement de ces messages et la d'éducation de l'agent Véhicule vainqueur pqui desservir le client, nous proposons le processus suivant. Au bout d'un certain temps, le système compare entre les coûts des agents Véhicule. Puis il distingue le petit coût, ensuite il envoie un message à L'agent Véhicule vainqueur (celui qui a le petit coût), pour aller amener le client. Tous les autres agents Véhicule ayant découverts ce vainqueur, le système s'annonce à l'agent Client pour l'informer du résultat de la négociation.

2.5 Conclusion

Dans ce chapitre, nous avons présenté un système de transport à la demande dynamique dans un environnement en changement perpétuel. Pour cela, nous avons adopté une approche centralisée basée sur l'optimisation.

L'objectif de notre projet, consiste à réaliser une application pour gestion de transport à la demande, et implémentation d'une solution, Comme tout projet logiciel, il est primordial de faire une analyse et une conception bien détaillées. C'est l'objet du chapitre suivant.

Analyse Des Besoins Et Conception

3.1 Introduction

Après avoir achevé l'étude faite sur le transport à la demande, où nous avons pu déterminer les problèmes posés dans ce genre de transport, nous allons nous intéresser dans ce chapitre à l'analyse et à la conception de la solution proposée.

3.2 Langage de modélisation et démarche adoptée

Le génie logiciel et la méthodologie s'efforcent de couvrir tous les aspects de la vie du logiciel. Issus de l'expérience des développeurs, concepteurs et chefs de projets, ils sont en constante évolution, parallèlement à l'évolution des techniques informatiques et du savoir-faire des équipes.

Comme toutes les tentatives de mise à plat d'une expérience et d'un savoir-faire, les méthodologies ont parfois souffert d'une formalisation excessive, imposant aux développeurs des contraintes parfois contre-productives sur leur façon de travailler.

Avec la mise en commun de l'expérience et la maturation des savoir-faire, on voit se développer à présent des méthodes de travail à la fois plus proches de la pratique réelle des experts et moins contraignantes. UML, qui se veut un instrument de capitalisation des savoir-faire puisqu'il propose un langage commun à tous les experts du logiciel, va dans le sens de cet assouplissement des contraintes méthodologiques [14].

3.2.1 UML

UML (Unified Modeling Language) est une norme du langage de modélisation objet qui a été publiée, dans sa première version, en novembre 1997 par l'OMG (Object Management Group) [15].

UML est un ensemble de notations et de règles permettant de modéliser de manière claire et précise la structure ainsi que le comportement d'un système indépendamment de toute méthode ou de tout langage de programmation. Les créateurs de l'UML insistent tout particulièrement sur le fait qu'UML soit un langage de modélisation et non une méthode [16].

3.2.2 Le processus unifié

Un processus unifié (Unified Process) est un processus construit sur l'UML, il est le résultat de l'unification, non pas des processus mais plus exactement les meilleures pratiques du développement objet [13].

3.2.2.1 Les phases du processus unifié

La phase d'initialisation : elle conduit à définir la « vision » du projet, sa portée, et sa fiabilité afin de pouvoir décider au mieux de sa poursuite ou de son arrêt.

La phase d'élaboration (analyse des besoins) : elle poursuit trois objectifs principaux en parallèle. Le premier consiste à identifier et à décrire la majeure partie des besoins des utilisateurs, tandis que le deuxième a pour but de construire l'architecture de base du système. Le troisième, quant à lui, tente de lever les risques majeurs du projet.

La phase de construction (conception) : elle consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base).

La phase de transition : elle permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux.

3.2.2.2 Le processus 2TUP

Le processus 2TUP (Two Track Unified Process) est une variante du processus unifié. Il gère la complexité technologique en donnant part à la technologie dans son processus de développement. Le 2TUP propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels et propose une étude parallèle des deux branches (fonctionnelle pour l'étude de l'application et technique pour l'étude de l'implémentation). Le processus 2TUP s'articule autour de trois branches : une branche technique, une branche fonctionnelle et une branche de conception réalisation.

La figure suivante détaille les étapes de développement des trois branches du processus 2TUP [14].

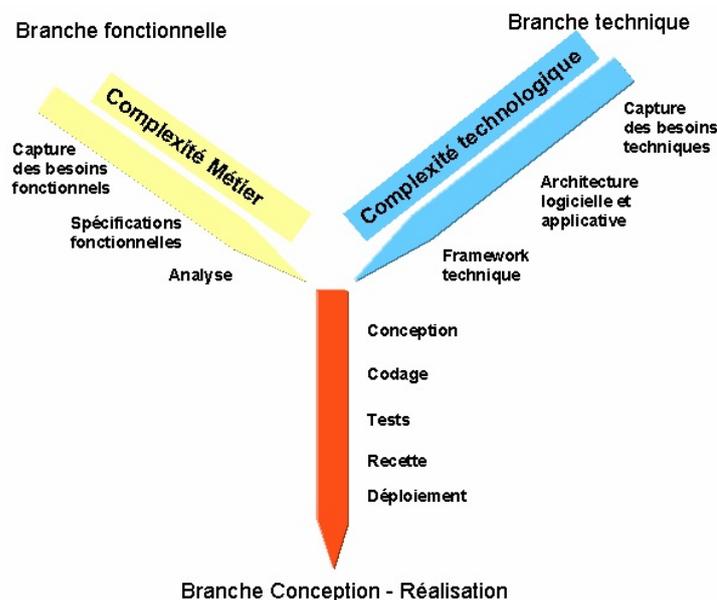


FIGURE 3.1 – Les étapes de développement suivant le processus 2TUP.

3.3 Etude préliminaire

L'étude préliminaire a pour objectifs principaux de [15] :

- Etablir un recueil initial des besoins fonctionnels et opérationnels.
- Modéliser le contexte du système considéré comme une boîte noire en identifiant les entités externes qui interagissent directement avec les acteurs, et en répertoriant les interactions (messages) entre ces acteurs et le système.

3.3.1 Les acteurs du système

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Il peut consulter et/ou modifier directement l'état du système, en émettant ou en recevant des messages susceptibles d'être porteurs de données [16].

Les acteurs du système :

- **Administrateur**
- **Utilisateur**
- **Véhicule**

Les acteurs du système que nous allons mettre en place sont :

L'administrateur, qui accède à toutes les fonctionnalités de système. Il gère les différentes entités (véhicules, utilisateurs), diffuse ou reçoit des données aux véhicules ou aux utilisateurs, et calcule les coûts, et à partir de ce calcul il décide d'associer les véhicules les mieux placés pour les demandeurs (utilisateurs). En utilisant l'algorithme déjà cité dans le chapitre 2.

L'Utilisateur, qui peut accéder à l'espace inscription, après son inscription, l'utilisateur peut envoyer des demandes de Réservation, ces demandes contiennent principalement sa position de départ et d'autres données. Comme il peut recevoir les réponses de système qui confirme ou refuse ses demandes.

Le Véhicule, cette entité de système reçoit des requêtes de la part de l'Administrateur, et il répond avec une réponse qui contient sa position actuelle, comme ça l'Administrateur effectue le calcul nécessaire et renvoie un autre message, là où il choisit un véhicule. Le véhicule choisi exécute la demande de l'administrateur.

3.4 Capture des besoins fonctionnels

L'étape de capture des besoins fonctionnels, est la première phase de la branche gauche, dite branche fonctionnelle, du processus en Y. Elle produit le modèle des besoins en se basant sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs.

Cette phase a pour objectif de définir [13] :

- La frontière fonctionnelle entre le système considéré comme une boîte noire et son environnement. Il s'agit du niveau contexte.
- Les activités attendues des différents utilisateurs par rapport au système. C'est le niveau cas d'utilisation.

Ce projet est soumis aux contraintes suivantes :

- Le développement des applications doit se baser sur l'emploi des technologies Open Source.
- La convivialité : l'application doit avoir des interfaces conviviales et faciles à utiliser. Elles doivent combiner des données textuelles bien claires et des données graphiques bien structurées.
- L'efficacité : les transactions effectuées par l'utilisateur doivent se faire en un temps de réponse raisonnable.
- La maintenance : les différents modules de l'application doivent être bien compréhensibles pour qu'elle puisse être maintenue facilement et rapidement.

Pour mener à bien ce projet, nous utilisons les outils et les technologies suivants :

- Existence d'un réseau local
- Existence d'un serveur de données avec le SGBD MYSQL

3.5 Les besoins non fonctionnels

- Répondre à des demandes des Utilisateur à tout moment et s'adapter chaque nouvelle offre.
- Saisir et enregistrer les nouvelles inscriptions des Utilisateur TAD.
- Saisir, modifier, et enregistrer les Demandes de réservation TAD.
- Valider les demandes TAD.
- Consulter l'historique des demandes TAD.
- Gérer et administrer des véhicules.

3.5.1 Identification des cas d'utilisations

Un cas d'utilisation représente un ensemble de séquences d'actions réalisées par le système et qui produisent un résultat observable pour un acteur particulier. Chaque cas d'utilisation spécifie un comportement attendu du système sans imposer la façon dont il est réalisé. Les cas d'utilisation permettent de décrire ce que le futur système devra faire sans spécifier comment le faire. L'ensemble des cas d'utilisations doit décrire exhaustivement les exigences fonctionnelles du système[17].

Les cas d'utilisation relatifs à l'application à réaliser sont représentés dans le tableau suivant :

Numéro	Acteurs	Cas d'utilisation
01	<ul style="list-style-type: none"> ●Administrateur ●Utilisateur ●Véhicule 	Authentification
02	Administrateur	<ul style="list-style-type: none"> ●Gérer utilisateur ●Gérer la demande réservation. ●calculer le cout et choix de véhicule. ●Gérer véhicules
03	Utilisateur	<ul style="list-style-type: none"> ●Créer un nouveau compte tad. ● Demande réservation .
04	Véhicule	<ul style="list-style-type: none"> ●Recevoir demande. ●calculer la position actuelle,et nombre place ● Envoyer réponse ● Exécuter requête.

TABLE 3.1 – Les cas d'utilisations du système à réaliser.

3.5.2 Relations entre cas d'utilisation

Afin d'optimiser la formalisation des besoins en ayant recours, notamment à la réutilisation de cas d'utilisation, trois relations peuvent être décrites entre cas d'utilisation [18].

Relation d'inclusion « include » : une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B. Elle est représentée graphiquement comme suit :



FIGURE 3.2 – Représentation graphique de la relation d'inclusion.

Relation d'extension « extend » : une relation d'extension d'un cas d'utilisation A par un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B. Elle est représentée de la façon suivante :

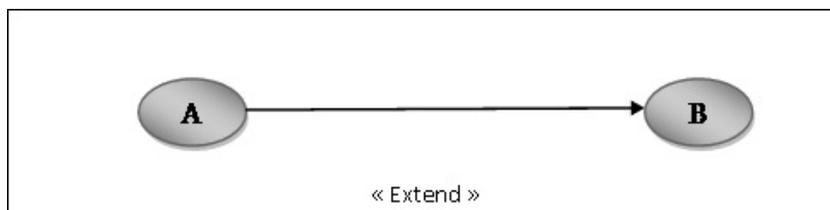


FIGURE 3.3 – Représentation graphique de la relation d'extension

3.5.3 Diagramme de cas d'utilisation du système à réaliser

Un diagramme de cas d'utilisation est un graphe d'acteurs, un ensemble de cas d'utilisation englobé par la limite du système, des associations de communication entre les acteurs et les cas d'utilisations, des généralisations entre cas d'utilisation. Il est destiné à représenter les besoins des utilisateurs par rapport au système [19].

3.5.3.1 les cas d'utilisation de l'Utilisateur

Les cas d'utilisation d'un Utilisateur consistent en demande de réservation, cela après une authentification qui est la raison pour laquelle la relation « include » existe et création d'un compte.

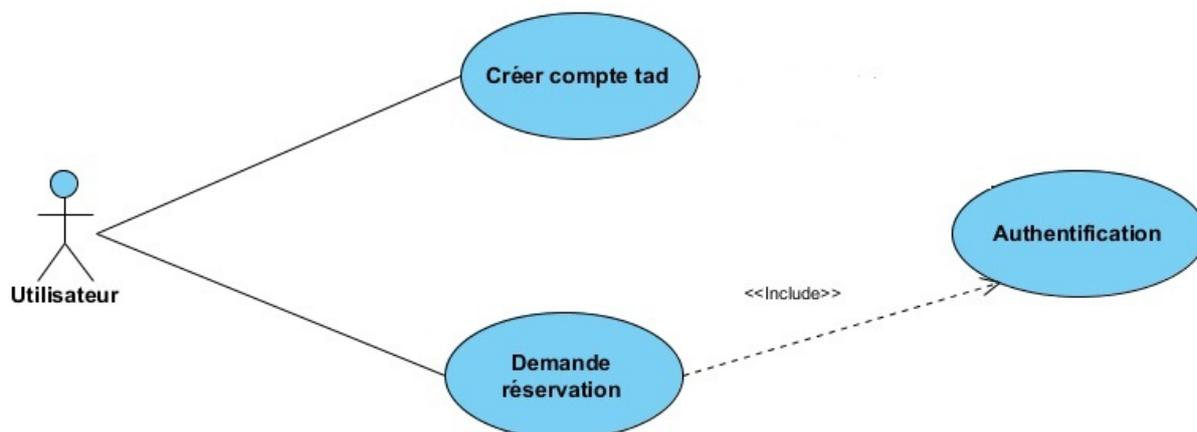


FIGURE 3.4 – Le diagramme de cas d'utilisation de Créer un compte et Demande réservation.

3.5.3.2 les cas d'utilisation de l'Administrateur

Les cas d'utilisation d'un Administrateur consistent en gestion des utilisateurs, gestion de demande, et gestion des véhicules, et calcule le cout, et cela après une authentification qui est la raison pour laquelle la relation « include » existe.

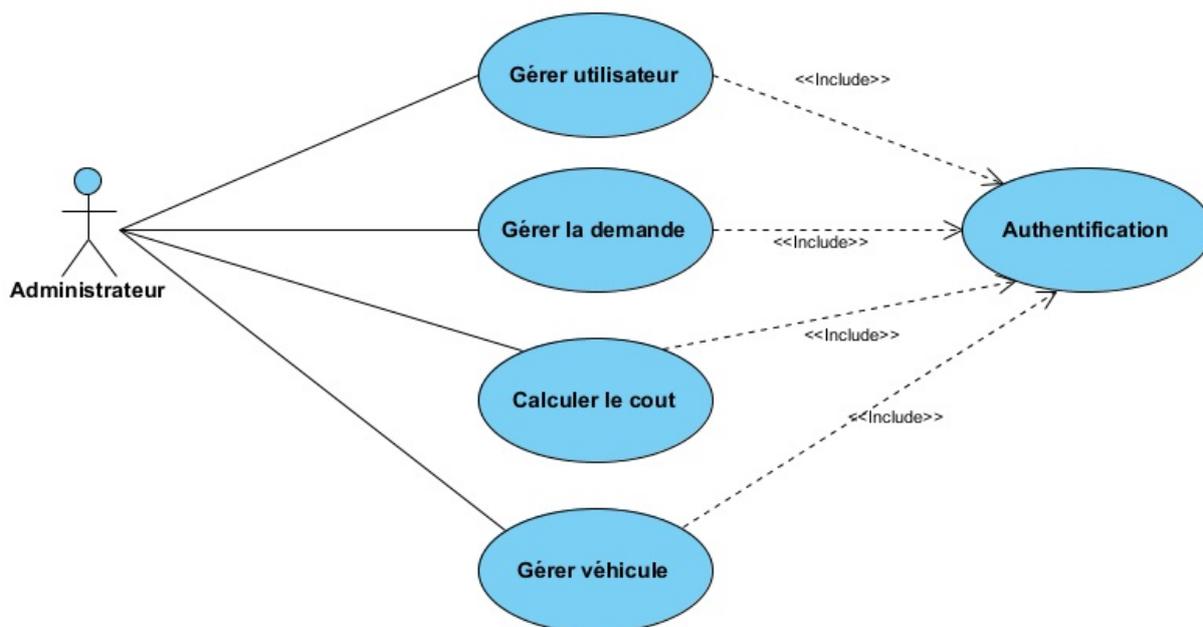


FIGURE 3.5 – Le diagramme de cas d'utilisation de gérer utilisateur, gérer demande, et gérer véhicule .

3.5.3.3 les cas d'utilisation d'un Véhicule

Les cas d'utilisation d'un Véhicule consistent en recevoir la demande, calculer la position, réponse à la demande ,et exécuter requête, et cela après une authentification qui est la raison pour laquelle la relation « include » existe.

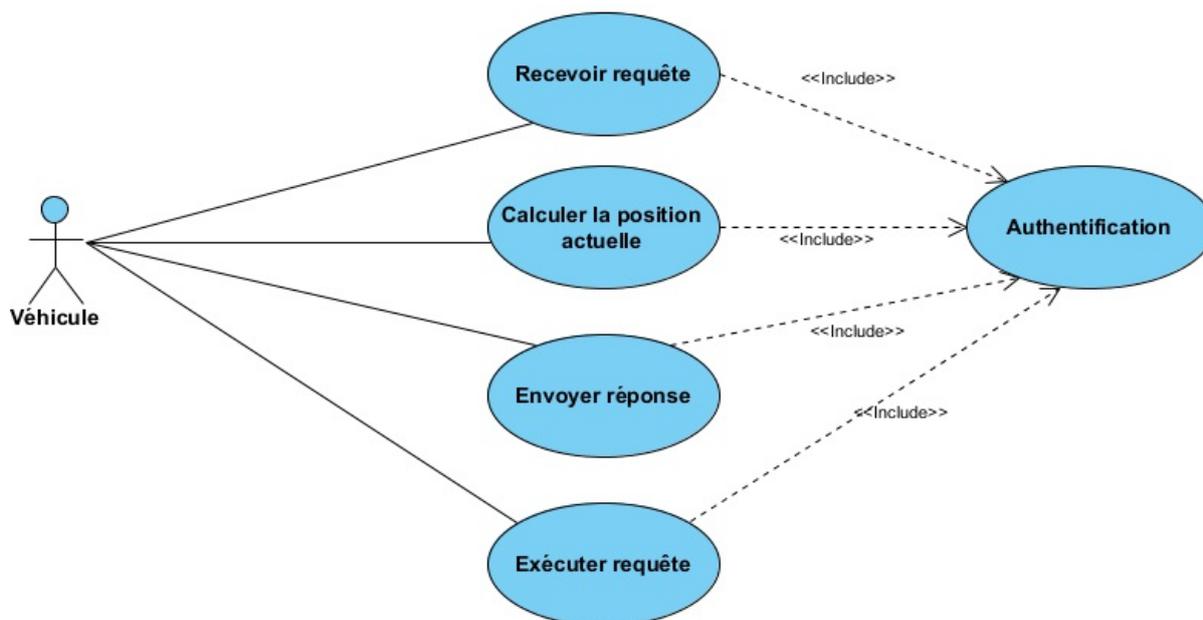


FIGURE 3.6 – Le diagramme de cas d'utilisation de recevoir requête et envoyer réponse .

3.6 Analyse

La phase d'analyse a pour objectif de décrire de manière précise, concise, correcte et compréhensible les besoins et les exigences du client. Il s'agit de livrer des spécifications pour permettre le choix de la conception de la solution [20].

Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système.

La phase d'analyse permet de s'accorder sur « ce que doit faire le système ? » [20].

3.6.1 Présentation du diagramme de séquence

Le diagramme de séquence permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets. L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés[21].

3.6.1.1 Message synchrone et asynchrone

Dans un diagramme de séquence, deux types de messages peuvent être distingués[21].

Les messages synchrones : dans ce cas, l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message. Le message retour peut ne pas être représenté car il est inclus dans la fin d'exécution de l'opération de l'objet destinataire du message.

Les messages asynchrones : dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message.

3.6.1.2 Fragment d'interaction

Dans un diagramme de séquence, il est possible de distinguer des sous-ensembles d'interactions qui constituent des fragments. 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 du nom de l'opérateur [21].

Il existe 13 opérateurs définis dans la notation UML 2. Nous citons, dans ce qui suit, les opérateurs utilisés pour l'analyse de notre projet : **alt**, **opt**, **loop**, **par** et **ref**.

Opérateur alt (alternative) : correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser les clauses de type sinon.

Opérateur opt : correspond à une instruction de test sans alternative.

Opérateur loop : correspond à une instruction de boucle qui permet d'exécuter une séquence d'interaction tant qu'une condition est satisfaite.

Opérateur par (parallèle) : permet de représenter deux séries d'interactions qui se déroulent en parallèle.

Opérateur ref : permet d'appeler une séquence d'interaction décrite par ailleurs constituant ainsi une sorte de sous-diagramme de séquence.

3.6.2 Elaboration des diagrammes de séquence des cas d'utilisation

En se basant sur les éléments précédents, nous allons présenter les diagrammes de séquence des cas d'utilisations les plus importants.

3.6.2.1 Diagramme de séquence du cas d'utilisation « Authentification »

Lorsqu'un utilisateur (Utilisateur, Administrateur, Véhicule) veut s'authentifier, deux cas peuvent se présenter : données correctes ou données incorrectes. C'est pourquoi on a utilisé l'opérateur « alt ». En effet, si les données d'authentification fournies par l'utilisateur sont correctes alors le système accorde l'accès à l'interface appropriée. Dans le cas contraire, un message d'erreur est généré et la page d'authentification est réaffichée.

Ce procédé est exécuté à chaque fois que l'utilisateur tente de s'authentifier, ce qui justifie l'utilisation de l'opérateur « loop ».

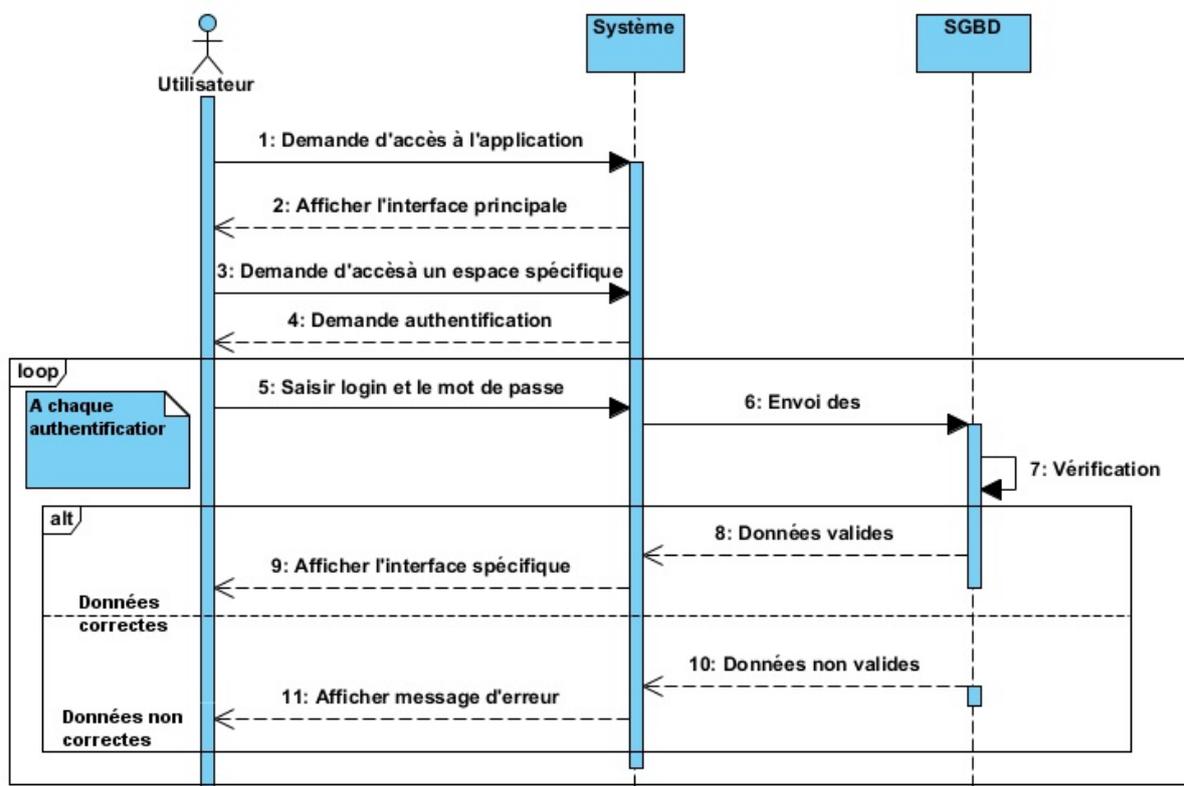


FIGURE 3.7 – Diagramme de séquence du cas d'utilisation « Authentification ».

3.6.2.2 Diagramme de séquence du cas d'utilisation « Gestion des véhicule ».

Après l'authentification, l'administrateur, peut gérer les véhicules, Les scénarios du cas d'utilisation « Gestion des véhicules » sont : ajout, modification, suppression.

Ajout d'un véhicule : l'administrateur demande d'ajouter un véhicule, le système affiche le formulaire d'ajout véhicule, ensuite l'administrateur le remplit l'enregistré dans la base de données .

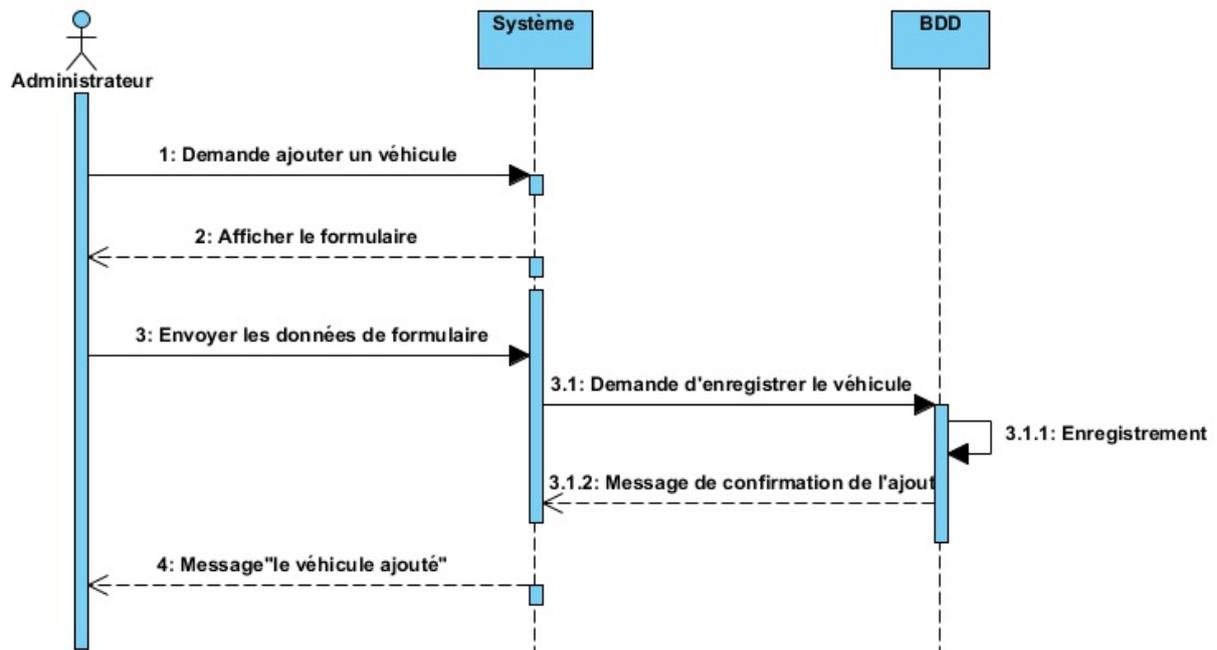


FIGURE 3.8 – Diagramme de séquence du cas d'utilisation « Ajout d'un véhicule ».

3.6.2.3 Diagramme de séquence du cas d'utilisation « Gestion utilisateur ».

Après l'authentification, l'administrateur peut gérer les utilisateurs. Les scénarios du cas d'utilisation « Gestion des utilisateurs » sont : ajout, modification, suppression.

Ajout d'un utilisateur : l'administrateur demande d'ajouter un utilisateur, le système affiche le formulaire d'ajout utilisateur, ensuite l'administrateur le remplit et l'enregistre dans la base de données.

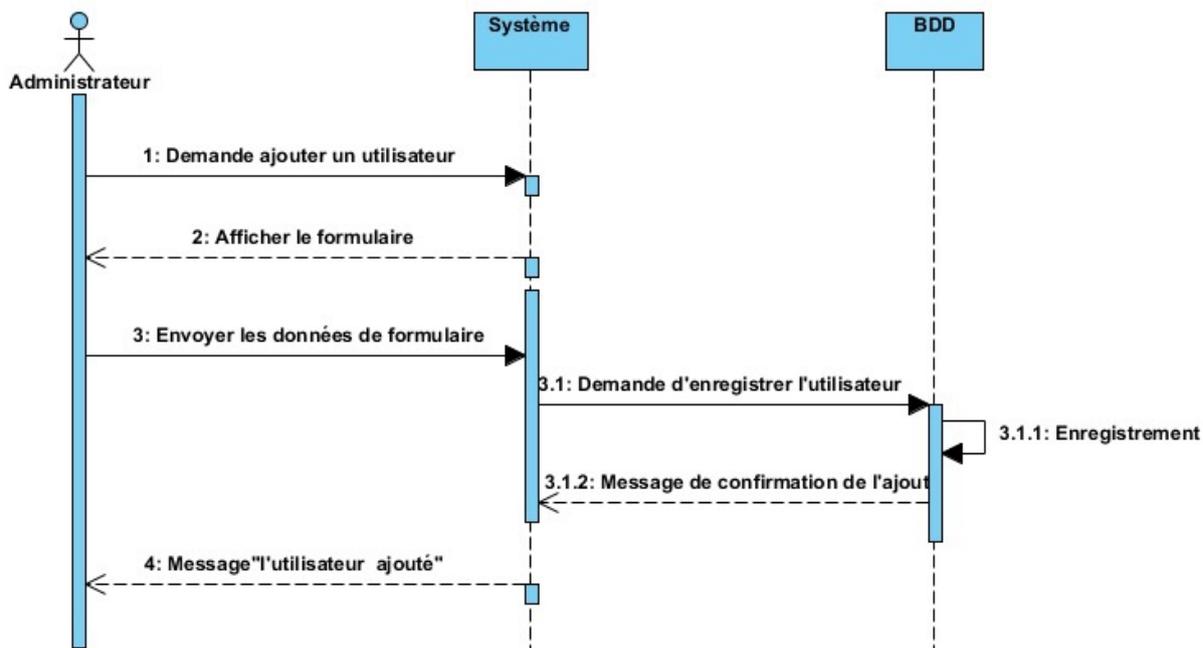


FIGURE 3.9 – Diagramme de séquence du cas d’utilisation « Ajout d’un utilisateur »

3.6.2.4 Diagramme de séquence du cas d’utilisation «demande réservation »

l'utilisateur demande une réservation qui contient sa position actuelle et nombre de place désiré ,puis l'Administrateur enregistre ces données dans la base de données.

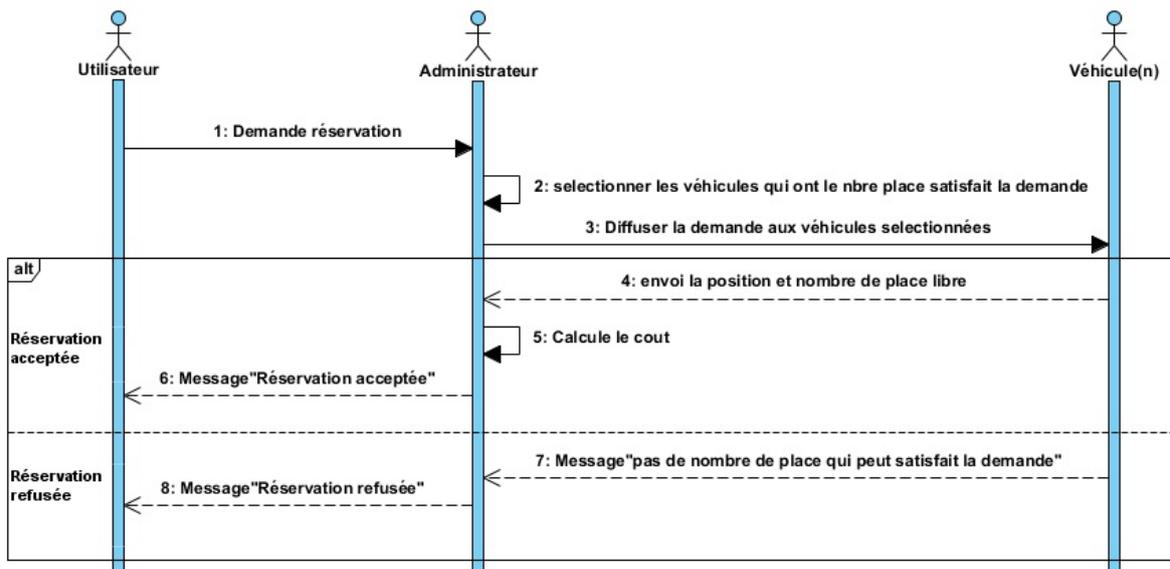


FIGURE 3.10 – Diagramme de séquence du cas d'utilisation « demande réservation ».

3.7 Conception

La conception détaillée est la phase ultime de la modélisation qui consiste à construire et à documenter précisément les classes, les tables et les méthodes qui constituent le codage de la solution.

3.7.1 Présentation du diagramme de classe

Le diagramme de classes est sans doute le diagramme le plus important à représenter pour les méthodes d'analyse orientées objet, comme il présente le point central de tout développement orienté objet. C'est une collection d'éléments de modélisation statique qui montre la structure d'un modèle. Un diagramme de classes fait abstraction des aspects dynamiques et temporels du système [22].

Un diagramme de classe est composé des éléments suivants [22].

Objet : représente une entité du monde réel (ou du monde virtuel pour les objets immatériels) qui se caractérise par un ensemble de propriétés (attributs), des états significatifs et un comportement.

Classe : est l'abstraction d'un ensemble d'objets qui possèdent une structure identique (liste des attributs) et un même comportement (liste des opérations).

Méthode (ou opération de la classe) : les méthodes décrivent les opérations qui sont applicables aux instances de la classe.

Attribut : est une propriété élémentaire d'une classe. Pour chaque objet d'une classe, l'attribut prend une valeur.

Association : une relation entre deux classes 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

Classe d'association : il s'agit d'une classe qui réalise la navigation entre les instances d'autres classes. Elle sert à connecter les classes entre elles.

Cardinalité : elle permet de définir les conditions de participation d'une entité à une relation. Toutefois, une entité peut participer à plusieurs relations.

3.7.2 Diagramme de classe du système à réaliser

Le diagramme de classes associé à l'application que nous allons développer pour la gestion de transport à la demande.

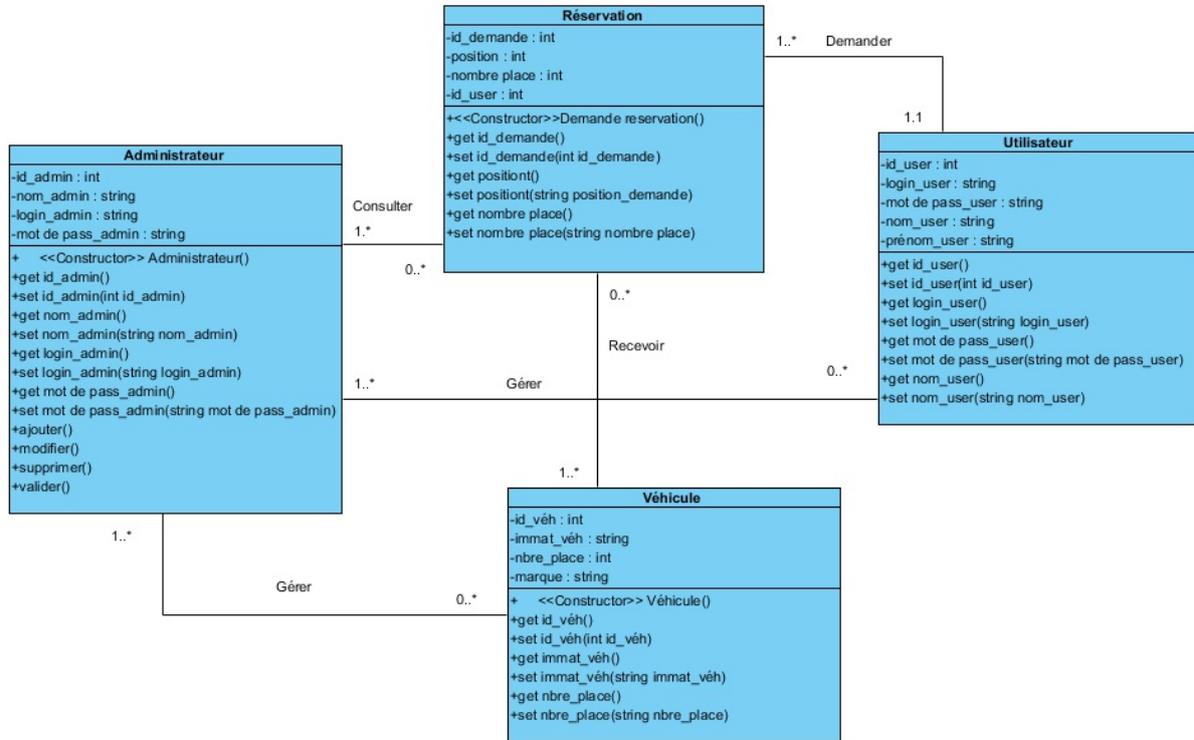


FIGURE 3.11 – Diagramme de Classes de l'application à réaliser

3.7.3 Le passage au modèle relationnel

A partir du modèle de classes de notre application et pour obtenir le modèle relationnel, nous avons appliqué les règles de passage décrites dans le livre[23].

Transformation des entités/classes

Règle R1 : chaque classe du diagramme UML devient une relation. Il faut choisir un attribut de la classe pouvant jouer le rôle d'identifiant. Si aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire.

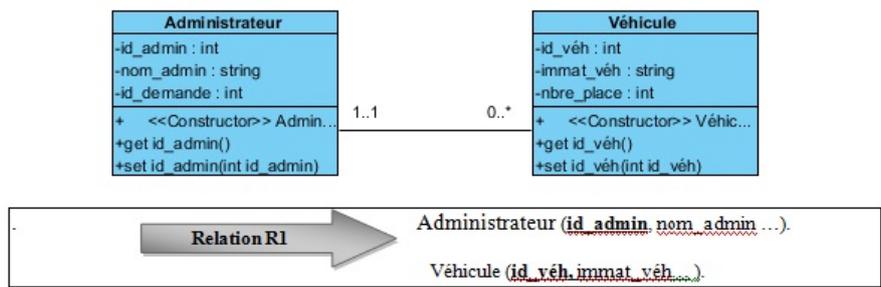


FIGURE 3.12 – Transformation d’entités/classes.

Transformation des associations

Les règles de transformation que nous allons voir dépendent des multiplicités maximales des associations.

Règle R2 : un-à-plusieurs.

Il faut ajouter un attribut de type clé étrangère dans la relation fils. L’attribut porte le nom de la clé primaire de la relation père de l’association.

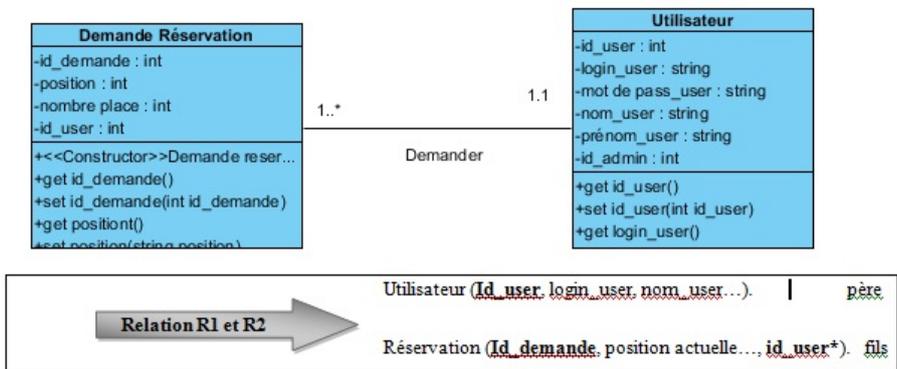


FIGURE 3.13 – Transformation d’entités/classes.

3.7.4 Identification des relations entre les classes

Administrateur (id admin, nom admin, login admin, motdepass login).

Utilisateur (id user, id admin* ,nom user ,prénom user, login user, motdepass user,).

Réserve (id demande, position,nombre place, id user*).

Véhicule (id véh, immat véh,position,nbr place,marque).

3.8 Conclusion

Au cours de l'étape de l'analyse des besoins et conception, nous avons donné une synthèse sur l'axe fonctionnel du système à réaliser. Nous avons d'abord déterminé les acteurs qui interagissent avec le système, puis nous avons décrit les cas d'utilisations associés à chaque acteur. Pour chaque cas d'utilisation, nous avons élaboré un diagramme de séquence décrivant la collaboration entre objets de point de vue temporel. Enfin, nous avons tiré le schéma de la base de données de notre système à partir du diagramme de classes.

Ce chapitre a permis de cerner la solution proposée et mieux comprendre le fonctionnement du système et ses différentes fonctionnalités. Il a également permis de préparer la phase de réalisation qui concrétisera tout ce qui a été présenté jusque-là.

Réalisation

4.1 Introduction

Après avoir conçu une solution qui répondrait au mieux aux objectifs fixés préalablement, nous entamons l'étape de la réalisation de l'application que nous avons développée pour la gestion de transport à la demande.

Dans ce chapitre nous présenterons les technologies utilisées pour le développement du système et nous donnerons un aperçu des interfaces réalisées.

4.2 Environnement de développement de l'application

4.2.1 Environnement logiciels

- **Visual Paradigm for UML 10.1**



Permet la création des diagrammes UML et des modèles qui en sont à l'origine. Ceux-ci peuvent alors générer du code dans un langage de programmation déterminé. Il propose également la création d'autres types de diagrammes, comme celui qui permet la modélisation des bases de données pouvant, lui aussi, générer des canevas d'applications basé sur des Framework et Pattern mais en plus, générer du code SQL qu'il peut ensuite déployer automatiquement dans différents environnements.

- **NetBeans 7.0**

Netbeans Est un environnement de développement intégré(EDI), placé en open source par Sun en juin 2000. En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et

HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, éditeur graphique d'interfaces et de pages Web ...). Conçu en Java, NetBeans est disponible sous Windows, Linux, Mac OS X etc ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit (JDK) est requis pour les développements en Java.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE Netbeans s'appuie sur cette plate forme Ainsi L'IDE Netbeans s'enrichit à l'aide de plugins.

4.2.2 Technologies utilisées

- **IDE Eclipse**



Eclipse IDE est un environnement de développement intégré libre, extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

- **Java DataBase Connectivity (JDBC)**

JDBC est l'acronyme de Java DataBase Connectivity et désigne une API Java (ensemble de classes et d'interfaces défini par SUN et les acteurs du domaine des Bases de Données) permettant d'accéder aux bases de données à l'aide du langage Java via des requêtes SQL. Cette API permet d'atteindre de manière quasi-transparente des bases : My Sql, Sybase, Oracle, Informix, ... avec le même programme Java JDBC.

Cette API permet de faire tout types de requêtes tel que :

- La sélection de données dans des tables
- La création de tables et l'insertion d'éléments dans les tables ;
- La gestion des transactions

Dans un système Client/Serveur, l'accès aux bases de données avec JDBC peut s'effectuer selon une architecture à deux couches ou bien une architecture à trois couches.

La figure suivante -figure 01- spécifie l'architecture générale d'un pilote JDBC :

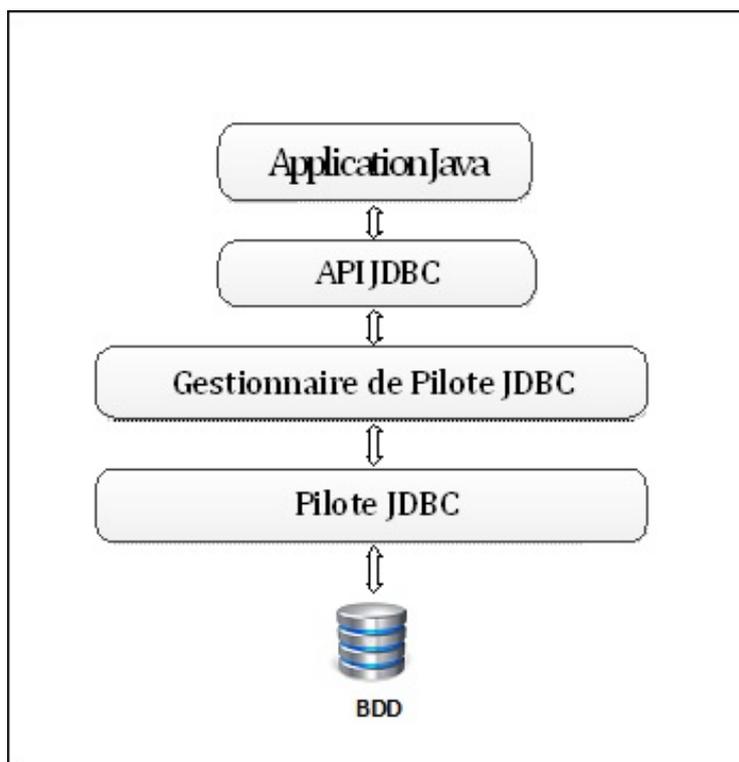


FIGURE 4.1 – Architecture générale d'un pilote JDBC

- MySQL



MySQL est un système de gestion de base de données relationnel, un langage de requêtes vers les bases de données exploitant le modèle relationnel et utilise le langage SQL comme langage de requête. SQL est un langage de manipulation de bases de données mis au point dans la années 70 par IBM, il permet d'effectuer trois types de manipulations :

1. La manipulation des tables : Création, suppression, modification de la structure.
2. Les manipulations des données de la base : Sélection, modification, suppression d'enregistrement.
3. La gestion des droits d'accès aux tables : Contrôle des données, droit d'accès, validation des modifications.

- **RMI**

RMI (Remote Methode Invocation) permet l'utilisation d'objet sur des JVM distantes. Il va plus loin que RPC puisqu'il permet non seulement l'envoi des données d'un objet, mais aussi de ses méthodes. RMI est un ensemble de classes permettant de manipuler des objets sur des machines distantes (objets distants) de manière similaire aux objets sur la machine locale (objet locaux). Ces manipulations sont "relativement" transparentes. Pour cela RMI propose :

- un ramasse-miettes distribué. la gestion des représentants locaux d'objets distants et leur activation.
- la liaison avec les couches transport et l'ouverture de sockets appropriés.
- la syntaxe d'invocation, d'utilisation d'un objet distant est la même qu'un objet local[24].

4.2.3 Le langage de programmation

JAVA



C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut être utilisé sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp).

Enfin, nous rappelons que le Java, étant un langage de programmation orienté objet utilisable sur divers systèmes d'exploitation, est un langage assez robuste, portable et à hautes performances.

4.3 Implémentation de la Base de Données

Pour implémenter notre base des données, nous avons utilisé l’environnement de création de base des données PHPMyAdmin et le système de gestion de base des données MySQL. Le tableau ci-dessous présente la table de la base de données générale :

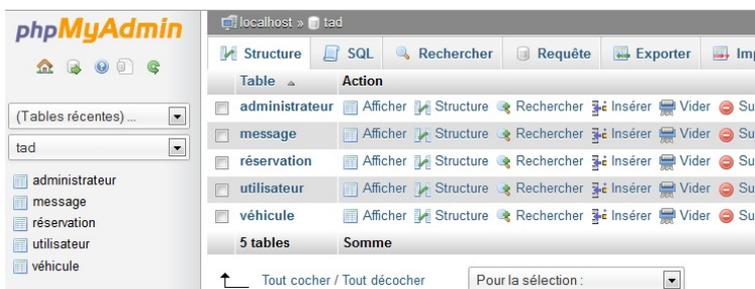


FIGURE 4.2 – Schéma générale de la base de données.

Voilà la table de la base de données de Véhicule

	immatricule_véhicule	nombre place	marque	@ip
<input type="checkbox"/>	05678097275	5	mercèdes	123.6.6.1
<input type="checkbox"/>	08765451432	3	ciio	
<input type="checkbox"/>	2135412736	0		
<input type="checkbox"/>	255232743285943	0		117.0.8.45

FIGURE 4.3 – la base de données de Véhicule.

Voilà la table de la base de données de l’Administrateur

	id_administrateur	nom	login	mot de passe
<input type="checkbox"/>	1	freud	farid	mechrafi
<input type="checkbox"/>	3	sa	sal	entre
<input type="checkbox"/>	4	soufianne	sofi	yes

FIGURE 4.4 – la base de données de Administrateur.

4.4 Présentation Interfaces graphiques de l'application

Dans ce qui suit, nous allons présenter quelques interfaces de l'application réalisée :

- **Interfaces utilisateur**

Dans cette interface l'utilisateur peut insérer le nombre de place désiré et sa position (dans les champs X et Y).

cette interface représente la demande de réservation de l'utilisateur.

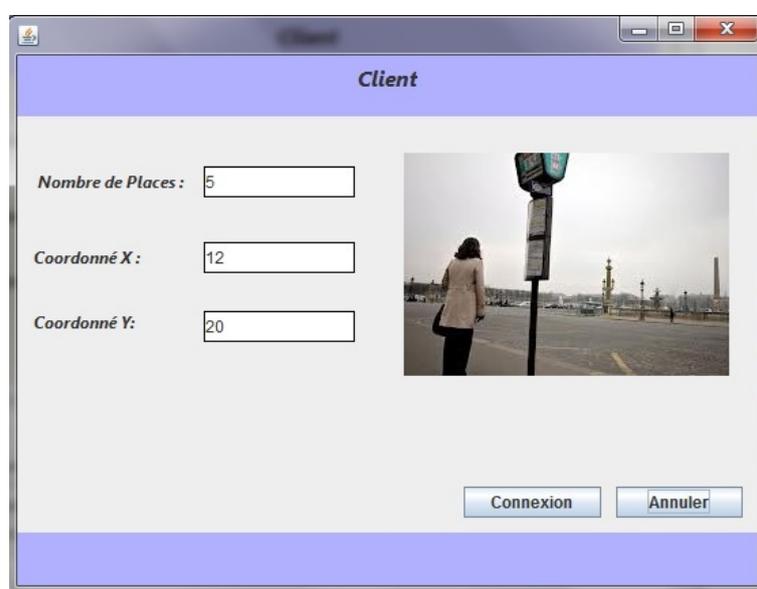


FIGURE 4.5 – Interface utilisateur.

- **Interface véhicule**

Dans cette interface le véhicule peut envoyer le nombre de place libre.

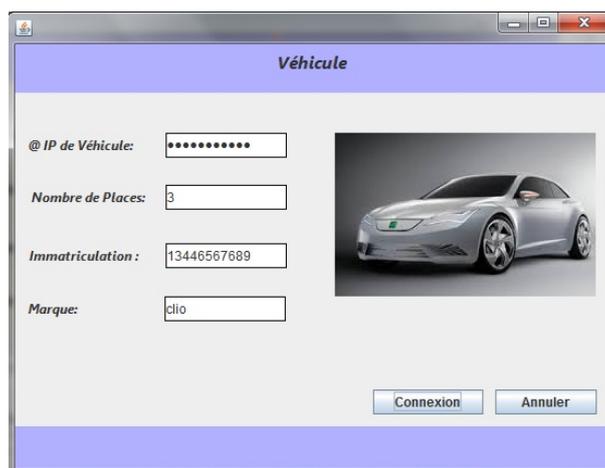


FIGURE 4.6 – Interface véhicule.

- **Interface recevoir de message**

Dans cette interface le véhicule peut recevoir les messages, et aussi la demande de réservation de l'utilisateur diffusé par l'administrateur.



FIGURE 4.7 – Interface recevoir de message.

4.5 Test

Cette phase contient les différents tests qu'a subit notre l'application afin de détecter les anomalies et les défauts : tester l'ergonomie, la rapidité et le déroulement des fonctionnalités.

Pour cela nous avons choisis un exemple d'envoyer la position de client à l'administrateur(serveur),l'administrateur diffuse la demande et recupère la position de tous les véhicules qui ont le nombre de place désiré, le serveur calcule le cout minimal,et prend la décision.d'envoyer un message à le véhicule choisit pour desservir le client,et cette étape ça marche bien.

4.6 Conclusion

La phase de réalisation est une étape très importante dans le cycle de vie de l'application, car à la fin de cette phase nous avons vu le résultat de toutes les étapes ultérieures.

Au cours de ce chapitre, nous avons présenté l'environnement matériel et logiciel ainsi que les différents choix techniques utilisés. Enfin, nous avons présenté les différentes interfaces offertes par notre application.

Conclusion et perspectives

Conclusion

A travers les recherches que nous avons effectuées durant ce projet et les travaux qui ont été réalisés par les différents chercheurs et sociétés, nous avons constaté l'importance de ce nouveau mode de transport, qui présente la particularité de nécessiter une réservation préalable. Celle-ci en fait un mode économe en carburant (le service ne fonctionne qu'en cas de réservation), optimisé (le trajet est calculé par rapport aux réservations) et aussi peu coûteux (les véhicules utilisés sont généralement de petite taille, (minibus, monospaces), les TAD sont plus rationnels que des lignes de bus circulant à horaires réguliers. De ce fait, le transport à la demande a connu ces dernières années une croissance assez importante. En effet, les autorités organisatrices de transport, quels que soient leur taille et leur statut (communes, communautés de communes, communautés d'agglomération, communautés urbaines, conseils généraux) ont vu dans le TAD une solution rationnelle et relativement facile à mettre en oeuvre pour résoudre leurs problèmes. Dans notre mémoire, on a donné une idée générale sur le TAD, et on a proposé une solution pour résoudre le problème de transport, en prenant en considération deux contraintes, la distance et le nombre de places. En fin, on a implémenté cette solution.

Perspectives

Pour perfectionner notre système, on envisage de lui adapter des dispositifs de localisation (GPS).

Elargir l'application et la perfectionner en lui ajoutant un mécanisme hybride (centralisé, décentralisé).

Bibliographie

- [1] ADETEC. "*Services à la demande et transports innovants en milieu rural, de l'inventaire à la valorisation des expériences -2004*", rapport d'étude édité par la D.A.T.A.R., la D.T.T. et l'ADEME, 209 pages.
- [2] Certu. "*Systèmes de transports à la demande, Enquête sur les caractéristiques et les modes d'exploitation* ", décembre 2002.
- [3] A.P.C.G., C.D.C., Transdev. " « *Le livre blanc des transports départementaux : bilan et perspectives d'avenir* " , 1998 ouvrage édité par l'Assemblée des Présidents des Conseils Généraux de France.273 pages.
- [4] La thèse de Besma Zeddini, Moncef Temani, et Adnan Yassine. "*Modélisation multi-agent pour une résolution dynamique du problème de transport à la demande* ". en 2008.
- [5] ALLEMAND Ludivine. "*Le Transport à la Demande en France : pistes de réflexion pour favoriser son développement et optimiser son exploitation Etude des cas du Conseil Général du Rhône, Toulouse, Dax et Evreux*". 11 septembre 2007, (réalisé dans le cadre du Master TURP, ENTPE, Université Lumière Lyon 2),106 pages.
- [6] François Ascher. "*le transport à la demande, un nouveau mode de gestion des mobilités urbaines* ",juillet 2000. Page 72.
- [7] Élodie CASTEX. "*Le Transport À la Demande (TAD) en France : de l'état des lieux à l'anticipation, Modélisation des caractéristiques fonctionnelles des TAD pour développer les modes flexibles de demain* ". Thèse soutenue le 22 novembre 2007.
- [8] Certu. "*Systèmes de transport à la demande : enquête sur les caractéristiques et les modes d'exploitation.* ".Technical report, Ministère de l'équipement, des transports et du logement, Paris (France), décembre 2002.
- [9] ASCHER F, BOURDIN A, CHARREL N. "*Le transport à la demande, un nouveau mode de gestion des mobilités urbaines* ".PREDIT, 219 pages.

- [10] Katia P. Sycara. "*Multiagent Systems*",1998.
- [11] Michael P. Georgeff. "*Communication and interaction in multi-agent planning.*", 1983
- [12] Bisma Zeddini, Moncef Temani, Adnan Yassine, and Khaled Ghedira. "*Du collectif pour le problème de transport à la demande : un modèle d'auto-organisation.*",2008.
- [13] www.freewebs.com/fresma.
- [14] Olivier Sigaud. "*Introduction à la modélisation orientée avec UML*".Edition 2005-2006.
- [15] Bruno Bouzy. "*Unified modeling language (uml)*"2001.
- [16] pierre-alain muller. "*Modélisation objet avec uml*".deuxieme édition 2000.
- [17] MIGNET. "*De Merise à UML*",édition, Octobre 2001.
- [18] Jean-Marie Ottelé. "*Analyse et conception d'une application informatique de gestion*".LTECG Luxembourg, Version 9.2009.
- [19] M. Grimaldi. "*Modélisation UML diagramme comportementaux*".février 2007.
- [20] Daniel CHOLLEY , Patrick GENDRE. "*INFORMATIQUE POUR LES SYSTEMES D'AIDE A LA GESTION DU TRAFIC*".Octobre 1999
- [21] Joseph Gabay, DUNOD "*UML2 Analyse et conception*".2008.
- [22] Pascal Roques, "*UML2 par la pratique*", Eyrolles 2005.
- [23] Christian Soutou. "*UML 2 pour les bases de données*".édition EYROLLES ,15 mars 2007.
- [24] [http ://sardes.inrialpes.fr/people/krakowia](http://sardes.inrialpes.fr/people/krakowia)