



N° Réf :.....

Centre Universitaire de Mila

Institut des Sciences et de la Technologie

Département de Mathématique et Informatique

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

En filière : Informatique Générale

Thème Conception et réalisation d'un système d'information pour la gestion de transport sanitaire

Préparé par : Boucheloukhe ibtissem
Bouzeràa selma
Amimmour souad

Encadré par :
Kimouche Abdelkader

Année universitaire : 2013/2014



Remerciements

*Après avoir terminé ce mémoire de fin d'étude, nous Réserveons ces
Lignes,*

Pour exprimer nos remerciements les plus sincères à notre Dieu.

*Tout puissant de nous avoir donné la santé et la patience Pour terminer
Ce travail Nous remercions tout d'abord ALLAH.*

*Nous tenons à exprimer nos remerciements avec un grand Plaisir
Et un grand respect à notre encadreur Mr kimouche Abdelkader,
Ses conseils, Sa disponibilité et ses encouragements qui nous Ont
Permis de réaliser ce travail dans les meilleures conditions.*

*Nous adressons aussi nos reconnaissances à tous les professeurs qui
depuis Quelques années leurs conseils et leurs connaissances qui Nous
Ont bien servis .*

Souad & ibtissem

& selma



DEDICACE



*Je dédie ce travail A mon père et ma mère Pour l'amour
et l'aide qu'ils m'ont fait entourer A mes frères «
Azzadin» et «Ahssan»,«nabil»,«madjide» et «abdelmalek»
A ma soeur «ratiba»,« soria» , «wanissa» , «rachida» et
«loubna» . Et toute ma famille amimmour a jijel. Et mon
adorable « souhil » et son famille surtout ma chère
lamia, A mes amies «amina »,« nassima »,«ibtissem » et
«khoulod» ,« hazzard»,« amelz» ,« amelk» , «sel ma» , «
Rahma»,«abir»,«safa»,«sana» et « azziza» et «bouchra».
A tous mes amis en particuliers A mes enseignants sans
exceptions.*

Souad.





DEDICACE



*Je dédie ce travail A mon père et ma mère Pour l'amour
et l'aide qu'ils m'ont fait entourer A mes frères «
messaoud», «amar», «abdelkhabir» et «aissa» .A ma soeur
«houda», «fatima » , «aida» , «meriem» et «siham» . Et
toute ma famille boucheloukhe. A mes amies «amina », «
nassima », «souad » , «ines» , «fatima», « amelz» , « amelk»
, «selma» , « Rahma», «abir», «safa», «sana» , « azziza» et
«nadia», «rahmaS» . A tous mes amis en particuliers A
mes enseignants sans exceptions.*

Ibtissem.





DEDICACE



*Je dédie ce travail A mon père «Ali» et ma mère «Dalila»
Pour l'amour et l'aide qu'ils m'ont fait entourer A mes
frères « Faris» et «Aziz» et A ma soeur «Amira», et«
Noussaiba», et« Khadija», Et toute ma famille Bouzeraa
et chaki ro . son famille surtout ma chère Amina A mes
amies : «maraim »,« rabiaa»,« nora »,«fairose » et
«souad» , « ibtissam» « madiha»,« farah» ,« widad» ,
«assia» , «romissa»,«manar», «batol»,«fatima» , «halima
» et «hala». A tous mes amis en particuliers A mes
enseignants sans exceptions.*

Selma.



Table des matières

Introduction générale

Chapitre 1 : UML & 2TUP

1.1 Introduction	1
1.2 Le langage de modélisationUML	1
1.2.1 UML.....	1
1.2.2 Pourquoi UML ?.....	1
1.2.3 Les diagrammes d'UML.....	1
1.2.3.1 les diagrammes structurels ou statiques	1
1.2.3.1.1Diagramme de classes (Class diagarmme).....	1
1.2.3.1.2Diagramme d'objets (Object dia gram).....	2
1.2.3.1.3Diagramme de composants (Component dia gram).....	2
1.2.3.1.4 Diagramme de déploiement (Déploiement dia gram).....	2
1.2.3.1.5 Diagramme des paquetages (Package Dia gram).....	2
1.2.3.1.6 Diagramme de structure composite (Composite Structure Dia gram).....	2
1.2.3.2 Diagrammes comportementaux (diagrammes dynamiques).....	2
1.2.3.2.1. Diagramme des cas d'utilisation (Use Case Dia gra.....	2
1.2.3.2.2 Diagramme d'activité (Activité Dia gram).....	2
1.2.3.2.3. Diagramme états-transitions (State Machine Dia gram).....	2
1.2.3.2.4. Diagramme de séquence (Séquence Dia gram).....	3
1.2.3.2.5 Diagramme de communication (Communication Dia gram).....	3
1.2.3.2.6 Diagramme global d'interaction (Interaction Overview Dia gram).....	3
1.2.3.2.7 Diagramme de temps (Timing Dia gram).....	3
1.2.4 Avantages et inconvénients d'UML.....	3
1.2.4.1 Les Avantages d'UML.....	3
1.2.4.2 Les Inconvénients d'UML.....	3
1.3 La Méthode2TUP.....	3
1.3.1 Définition d'un processus de développement logiciel.....	3
1.3.2 Le Processus unifié(UP).....	4
1.3.3Le Processus 2TUP	5

1.3.4 Un processus piloté par les exigences des utilisateurs.....	7
1.4 Conclusion.....	10

Chapitre 2 : Etude préliminaire & Capture des besoins

2.1 Introduction.....	11
2.2 Etude préliminaire.....	11
2.2.1 Elaboration du cahier des charges.....	11
2.2.2 Recueil des besoins fonctionnel	11
2.2.3 Recueil des besoins opérationnels.	12
2.2.3.1 La sécurité.....	12
2.2.4 Description de contexte du système.....	12
2.2.4.1 Identification des acteurs.....	12
2.2.4.2 Le rôle des acteurs	13
2.3 Capture des besoins.....	13
2.3.1 Capture des besoins fonctionnels.....	13
2.3.1.1 Les cas d'utilisation.....	14
2.3.1.2 Diagramme de cas d'utilisation.....	15
2.3.1.3 La description textuelle.....	17
2.3.1.4 les diagrammes de séquence système.....	29
2.3.1.5 Les diagrammes de classes participantes.....	50
2.3.2 Capture des besoins techniques.....	60
2.3.2.1 Style d'architecture	60
2.3.2.2 La configuration matérielle du système.....	61
2.3.2.3. Spécification de l'architecture en Tiers.....	61
2.4 Conclusion.....	61

Chapitre 3 : Analyse

3.1 Introduction	62
3.2 Découpage en catégorie.....	62
3.2.1 Définition d'une catégorie	62
3.3 Répartition des classes candidates en catégorie.....	62
3.4 Dépendances entre les catégories.....	63
3.5 Les diagrammes de classe préliminaire.....	63
3.5.1. Le diagramme de classe préliminaire de la catégorie « identification du patient»..	63
3.5.2. Le diagramme de classes préliminaire de la catégorie « Parc ».....	64

3.5.3. Diagramme de classes préliminaire de la catégorie « Transport du patient ».....	64
3.6. Développement du modèle statique	65
3.6.1. Les diagrammes détaillés pour chaque catégorie.....	65
3.6.1.1 Diagramme détaillés de catégorie patient.....	65
3.6.1.2 Diagramme détaillés de catégorie transport.....	66
3.6.1.3 Diagramme détaillés de catégorie parc.....	66
3.6.2 Diagramme de classe optimisé du système.....	67
3.7.Conclusion.....	68

Chapitre 4 : Conception

4.1 Introduction	69
4.2 Conception préliminaire	69
4.2.1 Concevoir le modèle de déploiement.....	69
4.3 Conception détaillée.....	69
4.3.1 Typage des attributs.....	70
4.3.2 Le passage du modèle objet au modèle relationnel.....	70
4.3.2.1 Règles de passage.....	70
1. Transformation des classes (R1).....	71
2. Transformation des associations.....	71
4.3.2.2 Traduction des associations.....	72
4.3.2.3 Les classes modifiées	73
4.3.3 Le modèle relationnel du système.....	74
4.4 Conclusion	74

Chapitre 5 :Réalisation

5.1 Introduction	75
5.2 Environnement de développement de l'application.....	75
5.2.1. Pourquoi Delphi ?.....	75
5.3 Interfaces graphique.....	75
5.3.1 Interface authentification	75
5.3.2 Interface de menu principale <<assurer planning>>.....	76
5.3.3 Interface de menu principale <<gérer les véhicules>>.....	77
5.3.4 Interface de menu principale <<gérer des patients>>.....	77
5.3.5 Interface de menu principale <<Etablir les factures>>.....	78
5.3.6 Interface <<Ajouter Patient>>.....	78

5.3.7	Interface <<Rechercher pour supprimer un Patient>>	79
5.3.8	Interface <<supprimer Patient>>	79
5.3.9	Interface <<Modifier un Patient>>	80
5.3.10	Interface <<Consulter un Patient>>	80
5.4	Conclusion.....	81

Liste des figures

Chapitre 1

Fig. 1.1 Le processus de développement en Y.....6

Fig. 1.2 Influence des cas d'utilisation dans le processus en Y.....9

Chapitre 2

Fig. 2.1 diagramme de cas d'utilisation.....15

Fig. 2.2 Diagramme de séquence système de cas d'utilisation Authentifier..... 30

Fig. 2.3 Diagramme de séquence système de cas d'utilisation Ajouter patient.....31

Fig. 2.4 Diagramme de séquence système de cas d'utilisation Consulter patient.....32

Fig. 2.5 Diagramme de séquence système de cas d'utilisation Modifier patient.....33

Fig. 2.6 Diagramme de séquence système de cas d'utilisation Supprimer patient.....34

Fig. 2.7 Diagramme de séquence système de cas d'utilisation Ajouter véhicule35

Fig. 2.8 Diagramme de séquence système de cas d'utilisation Consulter véhicule..... 36

Fig. 2.9 Diagramme de séquence système de cas d'utilisation Modifier véhicule..... 37

Fig. 2.10 Diagramme de séquence système de cas d'utilisation Supprimer véhicule.....38

Fig. 2.11 Diagramme de séquence système de cas d'utilisation Créer planning.....39

Fig. 2.12 Diagramme de séquence système de cas d'utilisation Consulter planning.....40

Fig. 2.13 Diagramme de s séquence système de cas d'utilisation Modifié planning.....41

Fig. 2.14 Diagramme de séquence système de cas d'utilisation Créer rapport..... 42

Fig. 2.15 Diagramme de séquence système de cas d'utilisation Consulter rapport.....43

Fig. 2.16 Diagramme de séquence système de cas d'utilisation Modifier rapport.....44

Fig.2.17 Diagramme de séquence système de cas d'utilisation Etablir le dossier du patient.....45

Fig. 2.18. Diagramme de séquence système de cas d'utilisation Accorder l'admission.....46

Fig. 2.19 Diagramme de séquence de cas d'utilisation Consulter les statistiques..... .47

Fig. 2.20 Diagramme de séquence système de cas d'utilisation Etablit la facture.....48

Fig. 2.21 Diagramme de séquence système de cas d'utilisation Editer la facture49

Fig. 2.22 Diagramme de classe participant du cas d'utilisation Authentifier.....50

Fig. 2.23 Diagramme de classe participant du cas d'utilisation Ajouter patient..... 51

Fig. 2.24 Diagramme de classe participant du cas d'utilisation Modifier patient.....51

Fig. 2.25 Diagramme de classe participant du cas d'utilisation Consulter patient.....52

Fig. 2.26 Diagramme de classe participant du cas d'utilisation Supprimer patient.....52

Fig. 2.27 Diagramme de classe participant de cas d'utilisation Ajouter véhicule.....53

Fig. 2.28	Diagramme de classe participant de cas d'utilisation Consulter véhicule.....	53
Fig. 2.29	Diagramme de classe participant de cas d'utilisation Modifier véhicule.....	54
Fig. 2.30	Diagramme de classe participant de cas d'utilisation Supprimer véhicule....	54
Fig. 2.31	Diagramme de classe participant de cas d'utilisation Créer planning	55
Fig. 2.32	Diagramme de classe participant de cas d'utilisation Consulter planning.....	55
Fig. 2.33	Diagramme de classe participant de cas d'utilisation Modifier planning.....	56
Fig. 2.34	Diagramme de classe participant de cas d'utilisation Créer rapport.....	56
Fig. 2.35	Diagramme de classe participant de cas d'utilisation Consulter rapport.....	57
Fig. 2.36	Diagramme de classe participant de cas d'utilisation Modifier rapport.....	57
Fig. 2.37	Diagramme de classe participant de cas d'utilisation Etablit le dossier.....	58
Fig. 2.38	Diagramme de classe participant de cas d'utilisation Accorder l'admission....	58
Fig. 2.39	Diagramme de classe participant de cas d'utilisation Etablit la facture.....	59
Fig. 2.40	Diagramme de classe participant de cas d'utilisation Editer la facture.....	59
Fig. 2.41	Diagramme de classe participant de cas d'utilisation Consulter les statistiques.	60
Fig. 2.42	architecture client serveur du système.....	60
Fig. 2.43	configuration matérielle du système.....	61
Fig. 2.44	Architecture à deux niveaux (client/serveur).....	61

Chapitre 3

Fig.3.1	Modèle structurel d'analyse.....	63
Fig. 3.2	Diagramme préliminaire de la catégorie patient.....	63
Fig. 3.3	Diagramme préliminaire de catégorie parc.....	64
Fig.3.4	. Diagramme préliminaire de catégorie transport.....	64
Fig. 3.5	Diagramme de classe détaillée de catégorie patient	65
Fig. 3.6	Diagramme de classe détaillée de catégorie transport.....	66
Fig. 3.7	Diagramme de classe détaillée de catégorie parc.....	66
Fig. 3.8	Diagramme de classes optimisé du système.....	67

Chapitre 4

Fig.4.1	concevoir le modèle de déploiement.....	69
Fig.4.2	le modèle relationnel du système	74

Chapitre 5

Fig.5.1	interface <<authentification>>.....	76
Fig. 5.2	interface <<assurer planning>>.....	76
Fig. 5.3	interface <<gérer les véhicules>>.....	77
Fig.5.4	interface <<gérer les patients>>.....	77
Fig. 5.5	interface <<établit la facture>>.....	78

Fig. 5.6	interface <<ajouter patient>>.....	78
Fig. 5.7	interface <<rechercher pour supprimer un patient>>.....	79
Fig. 5.8	interface <<supprimer un patient>>.....	79
Fig.5.9	interface <<modifier un patient>>.....	80
Fig.5.10	interface <<consulter un patient>>.....	80

INTRODUCTION GENERALE

A l'heure ou la multiplicité des domaines à gérer et la complexité des procédures de travail à provoquer les responsables et les utilisateurs à penser à des solutions informatiques.

L'informatique est la science rationnelle de l'information, elle est considérée n'est plus seulement comme une ressource stratégique pour l'entreprise, c'est l'outil très efficace permettant la manipulation rapide de l'information pour l'analyse des résultats élaborés par le traitement effectuée sur ces informations entraînent ainsi un gain de temps.

Elle se trouve dans tous les domaines d'activité « Scientifique, Gestion, Contrôle de processus ».

Son utilisation dans la gestion de transport sanitaire permettra d'avoir une meilleure organisation des activités au sein des centres d'hémodialyse et de donner des bons résultats.

Elle a pour but :

- D'améliorer la qualité des services offerts.
- De minimiser les erreurs dues de l'être humain.
- D'assurer une bonne sécurité des données.

C'est pour cette raison que l'administration du centre souhaite l'automatisation de toutes les tâches du travail afin d'atteindre ces buts.

1.1 Introduction

Dans ce chapitre, nous allons faire une présentation générale du langage UML et le processus de développement que nous allons adopter pour la conception de notre logiciel, à savoir 2TUP, leur objectif et leur domaine d'application.

1.2 Le langage de modélisation UML

1.2.1 UML

UML (UNIFIED MODELING LANGUAGE) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, Esquisser des architectures logicielles, concevoir des solutions et Communiquer des points de vue.

1.2.2 Pourquoi UML ?

UML pour :

- Obtenir une modélisation de très haut niveau indépendante des langages et des Environnements.
- Faire collaborer des participants de tous horizons autour d'un même document de Synthèse.
- Faire des simulations avant de construire un système.
- Exprimer dans un seul modèle tous les aspects statiques, dynamiques, juridique, Spécifications, etc.
- Documenter un projet.
- Générer automatiquement la partie logicielle d'un système.

1.2.3 Les diagrammes d'UML

UML dans sa version 2.0 propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles.

1.2.3.1 les diagrammes structurels ou statiques :

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

1.2.3.1.1 Diagramme de classes (Class Diagram): 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.

1.2.3.1.2 Diagramme d'objets (Object Diagram): Il montre des objets et des liens entre ces objets (les objets sont des instances de classes dans un état particulier). Il montre des objets et des liens entre ces objets (les objets sont des instances de classes dans un état particulier)

1.2.3.1.3 Diagramme de composants (Component Diagram) : 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.

1.2.3.1.4 Diagramme de déploiement (Déploiement Diagram) : 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.

1.2.3.1.5 Diagramme des paquetages (Package Diagram) : 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.

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

1.2.3.2 Diagrammes comportementaux (Dynamics Diagram) : 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.

1.2.3.2.1 Diagramme des cas d'utilisation (Use Case Diagram) : 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.

1.2.3.2.2 Diagramme d'activité (Activity Diagram) : 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.

1.2.3.2.3 Diagramme états-transitions (State Machine Diagram) : 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.

1.2.3.2.4 Diagramme de séquence (Sequence Diagram) : 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.

1.2.3.2.5 Diagramme de communication (Communication Diagram): C'est une représentation simplifiée d'un diagramme de séquence, en se concentrant sur les échanges de messages entre les objets.

1.2.3.2.6 Diagramme global d'interaction (Interaction Overview Diagram) : Permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité)

1.2.3.2.7 Diagramme de temps (Timing Diagram) : Le diagramme de temps permet de décrire les variations d'une donnée au cours du temps.

1.2.4 Avantages et inconvénients d'UML :

1.2.4.1 Les Avantages d'UML :

- UML est un langage formel et normalisé, il permet un gain de précision et un gain de
- Stabilité Ce qui encourage l'utilisation d'outils.
- UML est un support de communication performant
- compréhension des représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

1.2.4.2 Les inconvénients d'UML :

- La mise en pratique d'UML nécessite un apprentissage et passe par une période D'adaptation.
- UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure
Car il unifie les différentes approches et en donne une définition plus formelle.
- Le processus (non couvert par UML) est une autre clé de la réussite d'un projet.
- l'intégration d'UML dans un processus n'est pas triviale et améliorer un processus
Une tâche complexe et longue.

1.3 La méthode 2TUP

1.3.1 Définition d'un processus de développement logiciel

Un processus définit une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins

de leurs utilisateurs dans des temps et des coûts prévisibles. En conséquence, le processus peut se décomposer suivant deux axes de contrôle sur le développement :

- l'axe de développement technique, qui se concentre principalement sur la qualité de la production ;
- l'axe de gestion du développement, qui permet la mesure et la prévision des coûts et des délais. Nous ne traitons pas cet aspect dans « UML en *Action*... ».

1.3.2 Le Processus unifié(UP)

Le processus unifié est un processus de développement logiciel construit sur UML, il est itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques.

La gestion d'un tel processus est organisée suivant les 4 phases suivantes: Pré étude, élaboration, construction et transition.

Ses activités de développement sont définies par 5 disciplines fondamentales qui décrivent la capture des besoins, l'analyse et la conception, l'implémentation, le test et le déploiement.

Le processus unifié doit donc être compris comme une trame commune des meilleures pratiques de développement, et non comme l'ultime tentative d'élaborer un processus universel. La définition d'un processus UP est donc constituée de plusieurs disciplines d'activité de production et de contrôle de cette production. Tout processus UP répond aux caractéristiques ci-après.

- Il est itératif et incrémental. La définition d'itérations de réalisation est en effet la meilleure pratique de gestion des risques d'ordre à la fois technique et fonctionnel. On peut estimer qu'un projet qui ne produit rien d'exécutable dans les 9 mois court un risque majeur d'échec. Chaque itération garantit que les équipes sont capables d'intégrer l'environnement technique pour développer un produit final et fournit aux utilisateurs un résultat tangible de leurs spécifications. Le suivi des itérations constitue par ailleurs un excellent contrôle des coûts et des délais.

- Il est piloté par les risques. Dans ce cadre, les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité. Nous identifions une première cause provenant de l'incapacité de l'architecture technique à répondre aux contraintes opérationnelles, et une seconde cause liée à l'inadéquation du développement aux besoins des utilisateurs.
- Il est construit autour de la création et de la maintenance d'un modèle, plutôt que de la production de montagnes de documents. Le volume d'informations de ce modèle nécessite une organisation stricte qui présente les différents points de vue du logiciel à différents degrés d'abstraction. L'obtention de métriques sur le modèle fournit par ailleurs des moyens objectifs d'estimation.
- Il est orienté composant. Tant au niveau modélisation que production, c'est une garantie de souplesse pour le modèle lui-même et le logiciel qu'il représente. Cette pratique constitue le support nécessaire à la réutilisation logicielle et offre des perspectives de gains non négligeables.
- Il est orienté utilisateur, car la spécification et la conception sont construites à partir des modes d'utilisation attendus par les acteurs du système.

1.3.3 Le Processus 2TUP

2TUP signifie « 2 Track Unified Process ». C'est un processus UP qui répond aux caractéristiques que nous venons de citer. Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tel système « 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des chemins « fonctionnels » et « d'architecture technique », qui correspondent aux deux axes des changements imposés au système informatique.

L'axiome fondateur du 2TUP consiste à constater que toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement suivant un axe fonctionnel et un axe technique.

À l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des deux branches. Cette fusion

Conduit à l'obtention d'un processus de développement en forme de Y, comme illustré par la figure 1

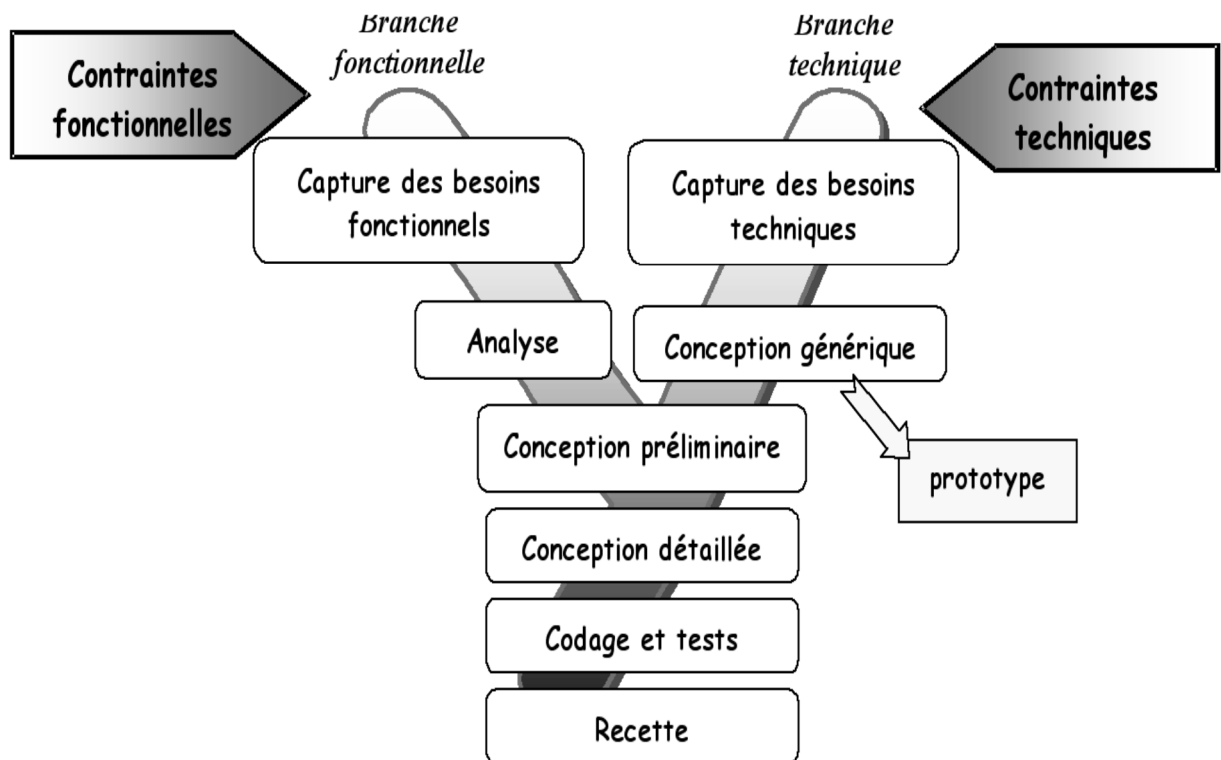


Figure 1.1 Le processus de développement en Y

La branche gauche (fonctionnelle) comporte :

- **la capture des besoins fonctionnels**

Qui produit un modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie au plus tôt le risque de produire un système inadapté aux utilisateurs. De son côté, la maîtrise d'œuvre consolide les spécifications et en vérifie la cohérence et l'exhaustivité.

- **L'analyse**

Qui consiste à étudier précisément la spécification fonctionnels de manière à obtenir une idée de ce que va réaliser le système en termes de métier. Les résultats de l'analyse ne dépendent d'aucune technologie particulière.

La branche droite (architecture technique) comporte :

- **La capture des besoins techniques**

Qui recense toutes les contraintes et les choix dimensionnant la conception du système. Les outils et les matériels sélectionnés ainsi que la prise en compte de contraintes d'intégration avec l'existant conditionnent généralement des pré requis d'architecture technique.

- **La conception générique**

Qui définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est la moins dépendante possible des aspects fonctionnels. Elle a pour objectif d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système informatique et écarte la plupart des risques de niveau technique. L'importance de sa réussite est telle qu'il est conseillé de réaliser un prototype pour assurer sa validité.

La branche du milieu comporte :

- **La conception préliminaire**

Qui représente une étape délicate, car elle intègre le modèle d'analyse dans l'architecture technique de manière à tracer la cartographie des composants du système à développer ;

La conception détaillée, qui étudie ensuite comment réaliser chaque composant;

- **L'étape de codage**

Qui produit ces composants et teste au fur et à mesure les unités de code réalisées.

- **L'étape de recette**

Qui consiste enfin à valider les fonctions du système Développé.

1.3.4 Un processus piloté par les exigences des utilisateurs:

Comme nous l'avons souligné précédemment, un bon nombre de risques Proviennent de la non-adéquation technique et fonctionnelle du système aux Besoins des utilisateurs. Les exigences des utilisateurs donc prioritairement traitées dans les deux branches du processus en Y en considérant deux types d'acteurs différents du système informatique :

- l'utilisateur consommateur des fonctions du système, qui correspond généralement à un poste, un rôle ou un métier dans l'entreprise. Il convient dans ce cadre de se focaliser sur la plus-value que lui apporte le système de l'exercice de sa profession.
- l'utilisateur exploitant le système, qui correspond plutôt à un rôle technique et opérationnel commun à la plupart des systèmes informatiques. Dans le domaine Client /serveur, les utilisateurs, considérés au sens large, attendent des performances, une tenue à la charge, une sécurité d'accès, etc. L'axe technique permet également d'introduire le point de vue des exploitants et des administrateurs souvent oubliés dans la livraison finale d'un produit.

- L'enjeu du processus en Y est donc de développer le point de vue utilisateur et de construire la spécification puis la conception objet à partir des concepts maniés par les acteurs du système. Les cas d'utilisation sont justement des outils construits pour définir les besoins, développant de surcroît le point de vue des utilisateurs. Il convient par la suite de montrer comment s'établit la traçabilité entre les cas d'utilisation et le modèle de conception. montre comment les cas d'utilisation influencent l'analyse et la conception d'architecture du système. Sur la branche gauche, pour la capture des besoins fonctionnels, les cas d'utilisation portent uniquement sur la plus-value métier des fonctions du système. Chaque cas d'utilisation met en évidence des classes d'analyse qui sont les concepts utilisés par l'utilisateur et des scénarios qui établissent les comportements attendus du système. Sur la branche droite, pour la capture des besoins techniques, la nature des cas d'utilisation a été quelque peu adaptée en fonction des plus-values opérationnelles du système pour ses exploitants. Chaque cas d'utilisation technique structure des spécifications d'architecture que l'on peut par la suite décomposer par couche logicielle. Les cas d'utilisation techniques permettent de concevoir les classes qui vont offrir une réponse aux contraintes opérationnelles du système. Les interactions entre ces classes permettent par ailleurs d'étudier les échanges entre classes, de consolider et de vérifier la conception des cas d'utilisation techniques. Lors de la conception préliminaire, les classes obtenues naissent de la distribution des classes d'analyse sur les couches logicielles. Les interactions entre classes de conception permettent de consolider et de vérifier à terme la conception des cas d'utilisation fonctionnelle tenant compte des contraintes opérationnelles.

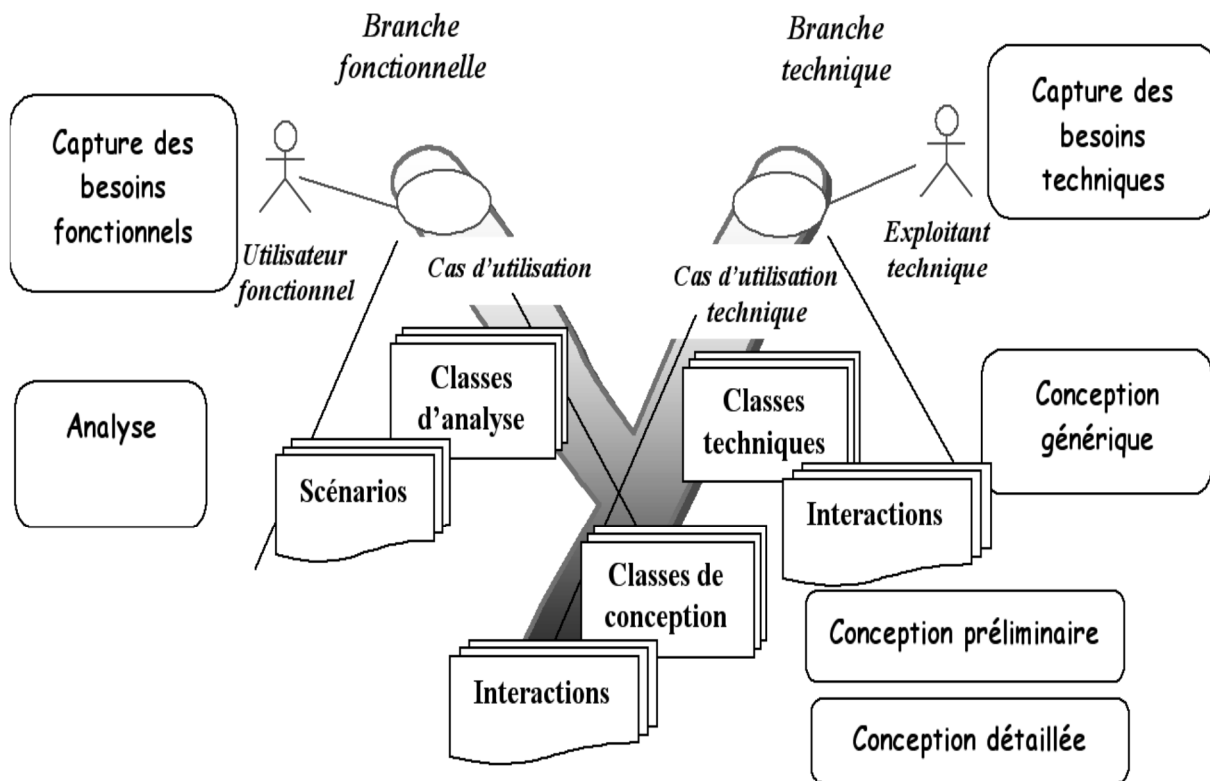


Figure 1.2 Influence des cas d'utilisation dans le processus en Y

Au travers des cas d'utilisation, le point de vue utilisateur a donc bien le rôle d'initiation souhaité pour une conception. Vous noterez par ailleurs que l'orientation métier imposée par les cas d'utilisation de la branche gauche renforce la plus-value apportée par le système et introduit une dimension d'analyse de la valeur.

Le pilotage par les cas d'utilisation consiste justement à ordonner les cas d'utilisation par priorité, de manière à organiser les itérations par valeur ajoutée. En phase de construction notamment, c'est une bonne pratique d'inclure les cas d'utilisation les plus prioritaires en réalisation des premières itérations, puis de continuer par ordre de priorité. En apportant au plus tôt le maximum de valeurs ajoutées au système, on rentabilise plus rapidement le développement, ce qui est encore une manière de réduire les risques.

1.4 Conclusion

Dans ce chapitre, nous avons passé en revue le langage de modélisation UML. D'abord, Nous avons présenté les différents diagrammes définis par UML, puis nous avons présenté le processus de développement 2TUP pour modéliser et analyser le problème de notre application. UML est un langage riche, il couvre toutes les phases d'un cycle de développement. Il est également ouvert (indépendant du domaine d'application et des langages d'implémentation).

2.1 Introduction

Dans ce chapitre on va commencer les premières étapes du processus de développement 2TUP qui sont l'étude préliminaire puis la capture des besoins, la principale tâche dans cette partie est détermination des cas d'utilisation, parce que toutes les autres étapes se basent sur lui.

2.2 Etude préliminaire.

2.2.1 Elaboration du cahier des charges.

- **Problématique :**

- Volume important des informations traitées manuellement.
- Difficulté de la communication entre les différents services.
- Recherche difficile sur les registres qui engendre une perte du temps
- Une faible sécurité des informations au niveau des documents
- Possibilité d'erreur dans les calculs des statistiques
- Nombre important des archives qui engendre une difficulté de stockage

- **Objectifs**

- Faciliter la recherche et l'accès aux informations.
- Améliorer la qualité de stockage des informations.
- Automatiser les tâches qui se traitent manuellement.
- Rapidité dans l'établissement des différents documents.
- Stocker les informations dans des supports informatique ce qui assurera leurs sécurité.
- Gain du temps dans le calcul des statistiques

2.2.2 Recueil des besoins fonctionnel

Le bureau de gestion des patients ayant pour mission d'établir et de suivre toutes les procédures administratives réglementaires relatives à la prise en charge des patients dès leurs admissions.

- A l'arrivée du patient au niveau du bureau de gestion des patients, il dépose leur certificat médicale avec le numéro d'assurance pour faire une demande de participation sur notre service.
- L'agent de gestion des patients envoie la demande du nouveau patient avec son certificat médical vers la CNAS

- Le médecin de la CNAS étudie les demandes des malades et envoie la décision à la société de transport sanitaire
- L'agent de gestion des patients accorde l'admission du patient
- L'agent de gestion des patients remplit la fiche d'admission du patient par les informations sur le patient (information d'identification du patient).
- L'agent de gestion des patients enregistre et fait un dossier (la prise en charge) pour le patient.
- Le patient devient un membre.
- Le régulateur va programmer les missions du nouveau patient et les ajouter dans le planning.

2.2.3 Recueil des besoins opérationnels.

2.2.3.1 La sécurité.

La sécurité des systèmes d'information (SSI) est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité du système d'information. Il consiste à garantir que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu. La sécurité informatique vise généralement cinq principaux objectifs :

- **L'intégrité**, c'est-à-dire garantir que les données sont bien celles que l'on croit être.
- **La confidentialité**, consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées.
- **La disponibilité**, permettant de maintenir le bon fonctionnement du système d'information.
- **La non répudiation**, permettant de garantir qu'une transaction ne peut être niée ;
- **L'authentification**, consistant à assurer que seules les personnes autorisées aient accès aux ressources.

2.2.4 Description de contexte du système

2.2.4.1 Identification des acteurs.

Un acteur représente l'abstraction d'un rôle joué par des entités externes au système étudié (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec lui. Il peut consulter et/ou modifier directement l'état du système en émettant et/ou recevant des

messages éventuellement porteurs de données Nous avons, dans ce système, les acteurs suivants :

- Régulateur
- L'agent de gestion des patients
- L'agent de facturation
- Personnel du parc

2.2.4.2 Le rôle des acteurs :

-Régulateur :

- Adapter le planning
- Organiser les missions de transport des patient d'un lieu à un autre (clinique ou hôpital) vers le domicile et vice-versa.
- Afficher le tableau (planning) pour que chacun puisse connaître son emploi du temps
- Affecter à chaque patient le véhicule correspondant.

- Personnel du parc :

- Mise à jour des véhicules
- Gestion du parc
- Etablir les rapport concernant les chauffeurs et états des véhicules

-L'agent des patients :

- Traiter les demandes des patients
- La réception des dossiers des patients
- Accorder l'admission des patients
- Etablir les dossiers des patients
- La mis à jour des patients

-L'agent de facturation :

- Etablir les factures
- Editer les factures des patients (pour les envoyer vers la CNAS)

2.3 Capture des besoins.

2.3.1 Capture des besoins fonctionnels

2.3.1.1 Les cas d'utilisation :

Un cas d'utilisation (une case) représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier.

Après l'étude de problème de transport sanitaire nous avons trouvé plusieurs cas d'utilisation :

- Assurer le planning
- Gérer les véhicules
- Établir le rapport
- Consulter les statistiques
- Accorder l'admission
- Gérer les patients
- Établir la facture
- Éditer la facture.

2.3.1.2 Diagramme de cas d'utilisation.

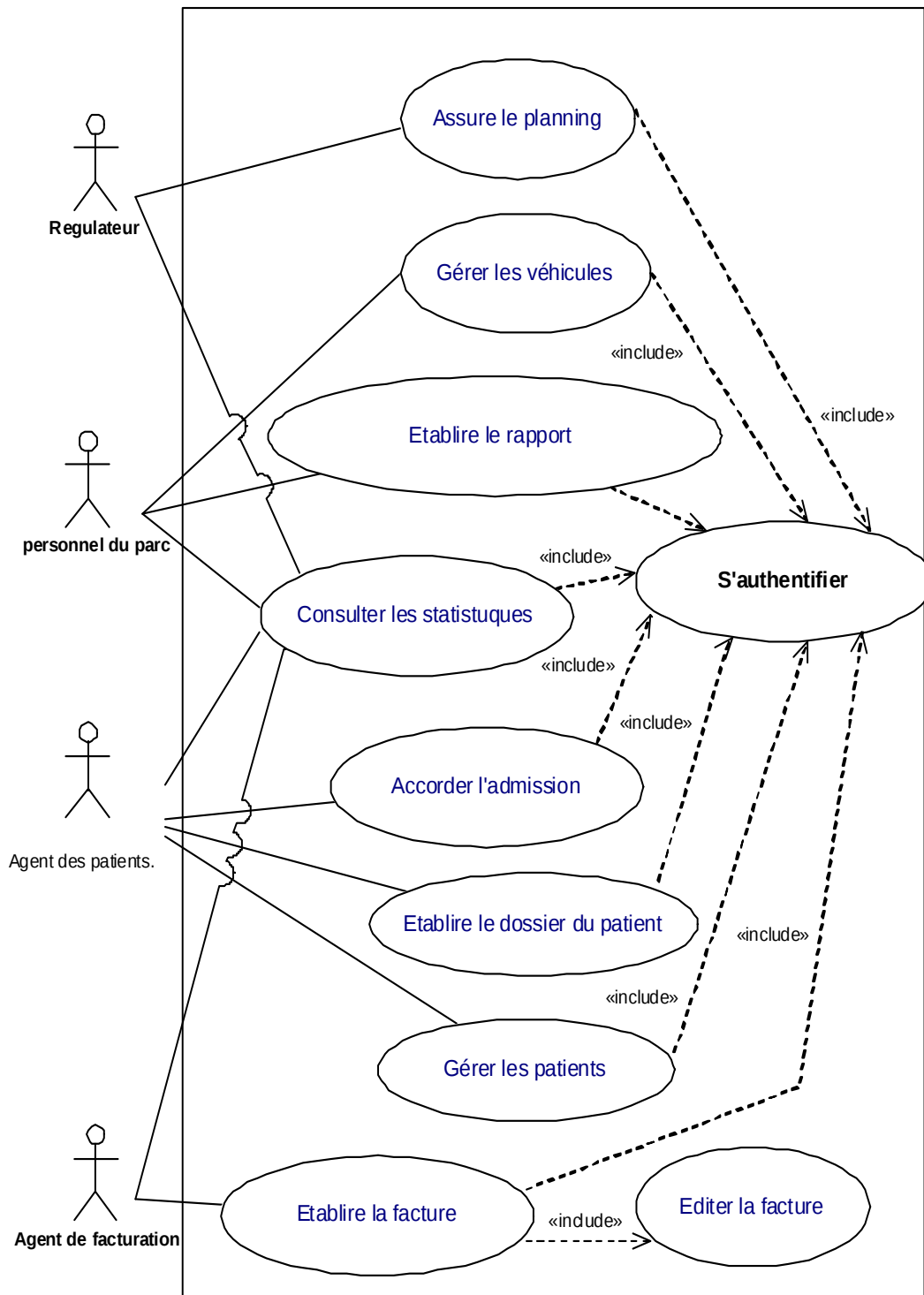
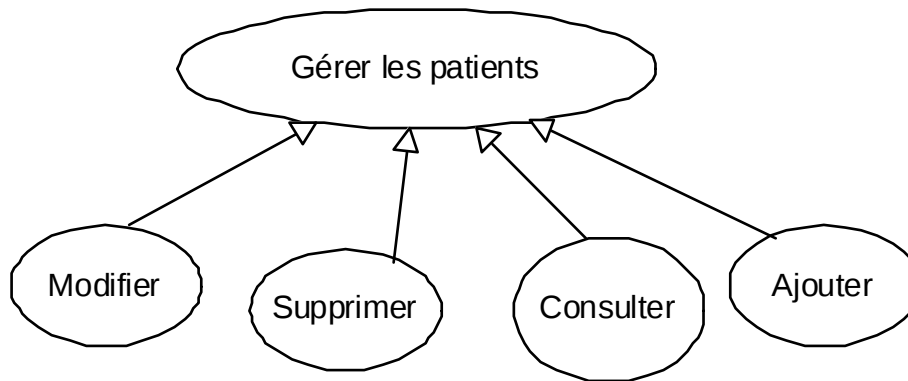
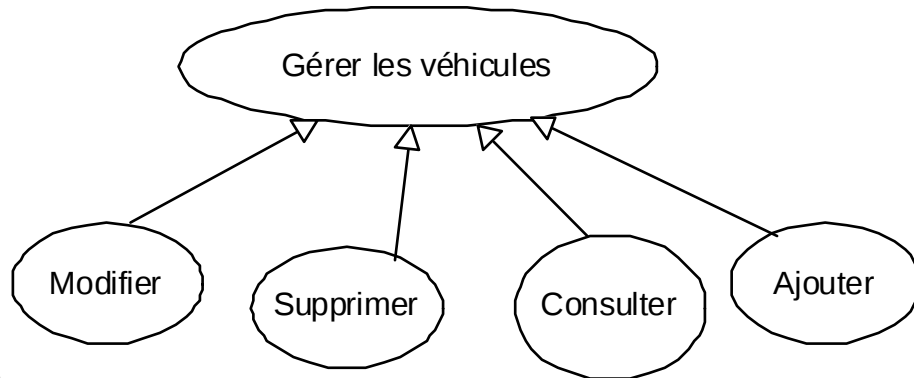


Figure 2.1. Diagramme de cas d'utilisation.

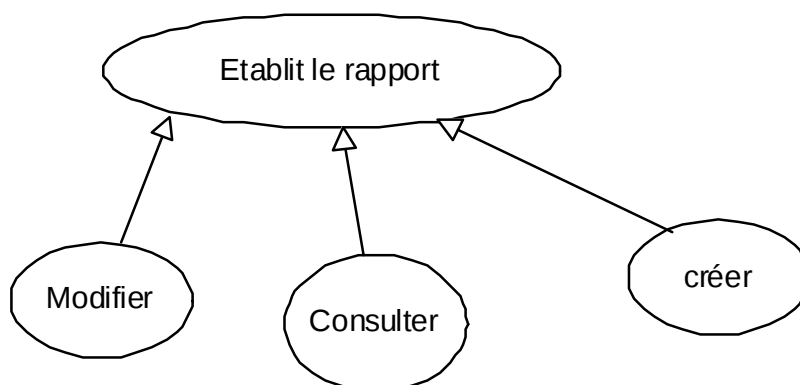
- Cas d'utilisation gérer les véhicules.



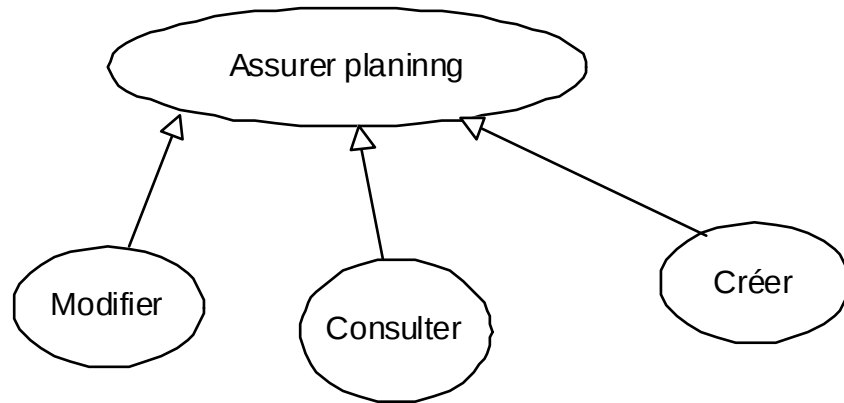
- Cas d'utilisation assurer le planning.



- Cas d'utilisation établit rapport.



- Cas d'utilisation gérer les patients.



2.3.1.3 la description textuelle des cas d'utilisation

➤ Cas d'utilisation <<s'authentifier>> :

➤ Cas d'utilisations : **S'authentifier.**

-**Acteur** : Régulateur, personnel du parc, Agent des patients, Agent de facturation.

-**But** : vérifier l'autorisation d'accéder au system

- **Pré condition** : l'utilisateur possède les informations d'accès.

-**Post condition** : l'utilisateur est authentifié par le système.

➤ **Scénario nominal**

1_ l'utilisateur choisit d'accéder au système

2_ Le système demande a l'utilisateur taper le nom et mot de passe.

3_ l'utilisateur saisit le nom et mot de passe et valider.

4-Le system vérifier la validité du nom et mot de passe et ouvrir
l'application.

➤ **Scénario d'erreur**

-le nom et mot de passe erroné e: affiche un message d'erreur.

- **Cas d'utilisation gérer les patients**
- **Cas d'utilisation gérer les patients <<Ajouter patient>>**

- **Cas d'utilisations : Ajouter un patient**

- Acteur** : l'agent de gestion des patients.

- But** : Permet d'ajoute un patient

- Pré condition** : l'agent de gestion des patients est authentifier, le patient n'existe pas

- Post condition** : le patient est enregistre /annule l'ajouté.

- **Scénario nominal**

- 1-l'agent de gestion des patients demande l'ajout d'un nouveau patient.

- 2-le système afficher l'interface de saisie.

- 3- l'agent de gestion des patients saisit les informations du nouveau patient

- 4-l'agent de gestion des patients demande la sauvegarde des informations.

- 5-le système vérifie les informations saisie puis sauvegarder.

- **Scénario d'erreur**

- **Cas d'utilisation gérer les patients <<Consulter les données d'un patient>>**

- **Cas d'utilisations : Consulter les données d'un patient.**

- Acteur** : l'agent de gestion des patients.

- But** : permet la Consultation des données d'un patient.

- Pré condition** : l'agent de gestion des patients est authentifié.

- Post condition** : affichage les données du patient / annulé l'opération.

➤ **Scénario nominal**

- 1 - l'agent de gestion des patients demande la recherche d'un patient.
- 2- Le system affiche la fenêtre de recherche.
- 3 -l'agent de gestion des patients saisit les informations de recherche
- 4 -Le system affiche les données du patient cherché.

➤ **Scénario d'erreur**

- le Numéro du patient saisit n'existe pas : affiche un message d'erreur.

➤ **Cas d'utilisation gérer les patients << modifier les données d'un patient>>**

➤ **Cas d'utilisations : modifier un patient.**

-**Acteur** : l'agent de gestion des patients.

-**But** : permet la modification des informations d'un patient.

-**Pré condition** : l'agent de gestion des patients est authentifié.

-**Post condition** : la mis a jour des données du patient

➤ **Scénario nominal**

- 1-l'agent de gestion des patients demande la modification d'un patient.
- 2 -le système afficher l'interface de recherche d'un patient.
- 3- l'agent de gestion des patients saisit les informations de recherche
- 4-le système affiche les informations du patient.
- 5- l'agent de gestion des patients modifier les donnée du patient puis valider.
- 6-le système vérifie les données.
- 7-Si pas d'erreur alors sauvegardé.

➤ **Scénario d'erreur**

- Si' il y a des données mal saisies : affiche un message d'erreur.

➤ **Cas d'utilisation gérer les patients << Supprimer les données d'un patient>>**

➤ **Cas d'utilisations : Supprimer un patient.**

-**Acteur** : l'agent de gestion des patients.

-**But** : permet la suppression d'un patient.

-**Pré condition** : l'agent de gestion des patients est authentifié.

-**Post condition** : le système Supprimer les données du patient

➤ **Scénario nominal**

1- l'agent de gestion des patients demande la suppression des données d'un patient.

2-le système afficher l'interface de recherche d'un patient.

3-l'agent de gestion des patients saisit les informations de recherche

4- le système afficher les informations du patient.

5-l'agent de gestion des patients demande la validation de la suppression.

6-Le système demande la confirmation de suppression.

7- l'agent de gestion des patients confirme la suppression.

➤ **Scénario d'erreur**

1-l'agent de gestion des patients saisit le Numéro du patient n'existe pas : affiche un Message d'erreur.

➤ **Cas d'utilisation gérer les véhicule**

➤ **Cas d'utilisation gérer les véhicules << Ajouter véhicule >>**

➤ **Cas d'utilisations : Ajouter un véhicule.**

-**Acteur** : personnel du parc.

-**But** : faire l'ajoute d'un véhicule.

-**Pré condition** : personnel du parc est authentifier, le véhicule n'existe pas.

-**Post condition** : les données d'un nouveau véhicule est enregistrer /annule l'ajoute.

➤ **Scénario nominal**

- 1_ personnel du parc demande l'ajout d'un nouveau véhicule.
- 2_ le système affiche le formulaire d'ajout.
- 3_ personnel du parc saisit les données du nouveau véhicule.
- 4_ personnel du parc demande la sauvegarde des informations.
- 5_ le système effectuer une vérification des informations saisies
- 6_ si pas d'erreur sauvegarder.

➤ **Scénario d'erreur**

- Si' il y a des données mal saisies : affiche un message d'erreur.

➤ **Cas d'utilisation gérer les véhicules <<Consulter les données d'un véhicule >>**

➤ **Cas d'utilisations : Consulter les données d'un véhicule.**

- Acteur** : personnel du parc.
- **But** : permet la Consultation des données d'un véhicule.
- **Pré condition** : personnel du parc est authentifié.
- Post condition** : affichage les données du véhicule / annulé

L'opération.

➤ **Scénario nominal**

- 1_ personnel du parc demande la consultation d'un véhicule.
- 2_ Le system affiche l'interface de recherche.
- 3_ personnel du parc saisie Numéro du véhicule.
- 4-Le system affiche les informations du véhicule.

➤ **Scénario d'erreur**

- le Numéro du véhicule saisit n'existe pas : affiche un message d'erreur.

➤ **Cas d'utilisation gérer les véhicules << Modifier les données d'un véhicule>>**

➤ **Cas d'utilisations : modifier un véhicule.**

- **Acteur** : personnel du parc.
- **But** : faire la modification d'un véhicule.
- **Pré condition** : personnel du parc est authentifié.
- **Post condition** : la mise à jour des données d'un véhicule.

➤ **Scénario nominal**

- 1-personnel du parc demande la modification d'un véhicule.
- 2- le système affiche l'interface de recherche d'un véhicule.
- 3- personnel du parc saisit le Numéro de véhicule.
- 4-Le système affiche les informations du véhicule.
- 5-personnel du parc fait la modification.
- 6- si pas d'erreur alors sauvegardé.

➤ **Scénario d'erreur**

- Si' il y a des erreurs de saisies : affiche un message d'erreur.

➤ **Cas d'utilisation gérer les véhicules <<Supprimer les données d'un véhicule >>**

➤ **Cas d'utilisations : Supprimer un véhicule.**

- **Acteur** : personnel du parc.
- But** : faire la suppression d'un véhicule.
- **Pré condition** : personnel du parc est authentifié.
- **Post condition** : le système Supprime le véhicule choisit / annulé la
Suppression

Scénario nominal

- 1- personnel du parc demande la suppression d'un véhicule.
- 2- Afficher le formulaire de recherche.
- 3-personnel du parc saisit le Numéro du véhicule.
- 4-le système affiche les informations du véhicule
- 5-personnel du parc fait la suppression.
- 6-Le système demande la validation de la suppression.
- 7-l'agent de gestion des patients confirme la suppression.

➤ Scénario d'erreur

- 1-personnel du parc saisit le Numéro du véhicule n'existe pas : affiche un message d'erreur.
- 2-le système affiche l'interface de recherche d'un véhicule.

➤ Cas d'utilisation assuré planning**➤ Cas d'utilisation << créer un planning >>****➤ cas d'utilisation : créer un planning.**

-Acteur : régulateur.

-But : permet de créer un nouveau planning.

-Pré condition : le régulateur est authentifié.

-Post condition : le planning crée par la système.

➤ Scénario nominal :

- 1-régulateur demande la création d'un nouveau planning.
- 2-le système affiche le formulaire de création d'un nouveau planning.
- 3-régulateur saisie les données du nouveau planning puis valider.
- 4-le système vérifie les données si pas d'erreur alors sauvegarder.

➤ Scénario d'erreur :

- Si une donnée est mal saisie afficher un message d'erreur.

➤ **Cas d'utilisation assurer planning << consulter le planning>>**

➤ **Cas d'utilisation : consulter le planning.**

-**Acteur** : régulateur.

-**But** : permet de consulter un planning.

- **Pré condition** : le régulateur est authentifié.

- **Post condition** : le planning est affiché /annuler.

➤ **Scénario nominal :**

1- régulateur demande la consultation d'un planning.

2-le système afficher la fenêtre de consultation des plannings.

3- le régulateur saisie le numéro de la semaine.

4- le système affiche le planning concerné.

➤ **Scénario d'erreur :**

- s'il ya erreur dans le numéro de la semaine afficher un message d'erreur.

➤ **Cas d'utilisation << modifier planning >>.**

➤ **cas d'utilisation : modifier planning.**

-**Acteur** : régulateur.

-**But** : permet la modification d'un planning.

-**Pré condition** : le régulateur est authentifié.

-**Post condition** : la mis à jour du planning.

➤ **Scénario nominal :**

1- le régulateur demande la modification d'un planning.

2-le système afficher l'interface de modification.

3-le régulateur saisie le numéro de la semaine.

4-le système afficher le planning.

5-le régulateur modifier les données des plannings puis valider.

6- le système vérifier les données puis sauvegarder.

➤ **Scénario d'erreur :** si une donnée est mal saisie afficher un message d'erreur.

- **Cas d'utilisation établir rapport**
- **Cas d'utilisation établir rapport << créer rapport >>**

- **Cas d'utilisation : créer un rapport.**
 - Acteur** : personnel du parc.
 - But** : permet d'établir un rapport.
 - Pré condition** : personnel du parc est authentifié.
 - Post condition** : le rapport est établir ou annuler.
- **Scénario nominal :**
 - 1-le personnel du parc demande la création d'un rapport.
 - 2-le système afficher le formulaire de création d'un nouveau rapport.
 - 3-le personnel du parc saisie les données du nouveau rapport.
 - 4-le système vérifier les données puis enregistrer le rapport.
- **Scénario d'erreur :**
 - si une donnée est mal saisie afficher un message d'erreur.

- **Cas d'utilisation établir rapport << consulter rapport >>**

- **Cas d'utilisation : consulter rapport.**
 - Acteur** : personnel du parc
 - But** : permet la consultation d'un rapport.
 - Pré condition** : le personnel du parc est authentifié.
 - Post condition** : mis à jour du rapport.
- **Scénario nominal :**
 - 1-le personnel du parc demande la consultation d'un rapport.
 - 2-le personnel du parc afficher l'interface de consultation.
 - 3-le personnel du parc saisie le numéro d'un rapport.
 - 4-le système afficher les données du rapport.
 - 5-le personnel du parc fait la consultation et valider.
 - 6-le système vérifier l'information et sauvegarder.
- **Scénario d'erreur :**
 - Si une donnée est mal saisie afficher un message d'erreur.

➤ **Cas d'utilisation établir rapport<< Modifier rapport>>**

➤ **Cas d'utilisation : modifier rapport.**

-**Acteur** : personnel du parc.

-**But** : permet la modification d'un rapport.

-**Pré condition** : personnel du parc est authentifié.

-**Post condition** : mis à jour du rapport.

➤ **Scénario nominal :**

1-le personnel du parc demande la modification d'un rapport.

2-le système afficher l'interface de recherche.

3-le personnel du parc saisie le numéro de rapport.

4-le système afficher les données du rapport.

5-Le personnel du parc fait la modification et valider.

6- le système vérifier les modifications et sauvegarder.

➤ **Scénario d'erreur :**

-Si des donnée est mal saisie afficher un message d'erreur.

➤ **Cas d'utilisation établir le dossier du patient.**

➤ **Cas d'utilisations : établir le dossier du patient**

-**Acteur** : l'agent de gestion des patients.

-**But** : faire l'établissement de dossier.

- **Pré condition** : l'agent de gestion est authentifié, le dossier du patient

N'existe pas.

-**Post condition** : le dossier du patient est ajouté.

➤ **Scénario nominal**

1- l'agent de gestion des patients demande l'établissement de dossier.

2-le système afficher le formulaire de saisie

4- le système vérifie les informations.

4.1-le système enregistre les données.

4.2-si erreur : le système affiche un message.

➤ **Scénario d'erreur**

-Si' il y a des données mal saisies : affiche un message d'erreur.

➤ **Cas d'utilisation << accorder l'admission du patient >>**

➤ **Cas d'utilisation : accorder l'admission.**

- Acteur : agent de gestion des patients.
- But : saisie sur l'admission du patient (les informations concernant l'identification du patient, l'assurance, etc....).
- Pré condition : l'agent de gestion du patient est authentifié.
- Post condition :

➤ **Scénario nominal :**

- 1- L'agent demande la saisie d'une admission.
- 2- Le système affiche l'interface de saisie.
- 3- L'agent de gestion des patients saisie les informations.
- 4- Si la situation vis-à-vis à l'assurance est assuré alors :

➤ **Cas d'utilisation << consulter statistique>>**

➤ **Cas d'utilisation : consulté statistique.**

- Acteur : tous les acteurs.
- But : voir les statistiques.
- Pré condition : /
- Post condition : les statistique sont affichées ou annulée.

➤ **Scénario nominal :**

- 1- l'utilisateur demande de voir les statistiques.
- 2- le système affiche l'interface des critères de choix.
- 3- l'utilisateur sélectionne un critère du choix.
- 4- Le système affiche les statistiques.

➤ **Cas d'utilisation << Etablit la Facture >>**

➤ **Cas d'utilisations : établit la facture. .**

-**Acteur** : l'agent de facturation

-**But** : permet d'établir la facture d'un patient.

-**Pré condition** : l'agent de facturation est authentifié.

-**Post condition** : la facture est établie / annulé l'opération.

➤ **Scénario nominal :**

1-l'agent de facturation demande l'établissement d'une
Facture du patient.

2-Le system affiche la formulaire de la facture.

3-l'agent de facturation saisit la facture puis demande la
sauvgarde

4-Le system vérifier les données de la facture et faire
l'enregistrement.

➤ **Scénario d'erreur**

-Si' il y a des données mal saisies : affiche un message d'erreur

➤ **Cas d'utilisation << Editer la Facture >>**

➤ **Cas d'utilisation : éditer la facture.**

-**Acteur** : l'agent de facturation.

-**But** : permet l'édition facture d'un patient

-**Pré condition** : l'agent de facturation est authentifié

-**Post condition** : imprimer la facture du patient / annulé

L'opération.

➤ **Scénario nominal :**

1-l'agent de facturation demande l'édition d'une facture.

2-le système affiche l'interface d'édition.

3-L'agent de facturation saisit l'identification du patient.

4-Le système affiche la liste des missions (kilométrage) du patient.

5-L'agent de facturation lance le processus d'impression de la Facture.

6-Le système imprime la facture.

2.3.1.4 les diagrammes de séquence système.

Le diagramme de séquence représente la succession chronologique des opérations réalisées par un acteur. Il indique les objets que l'acteur va manipuler et les opérations qui font passer d'un objet à l'autre.

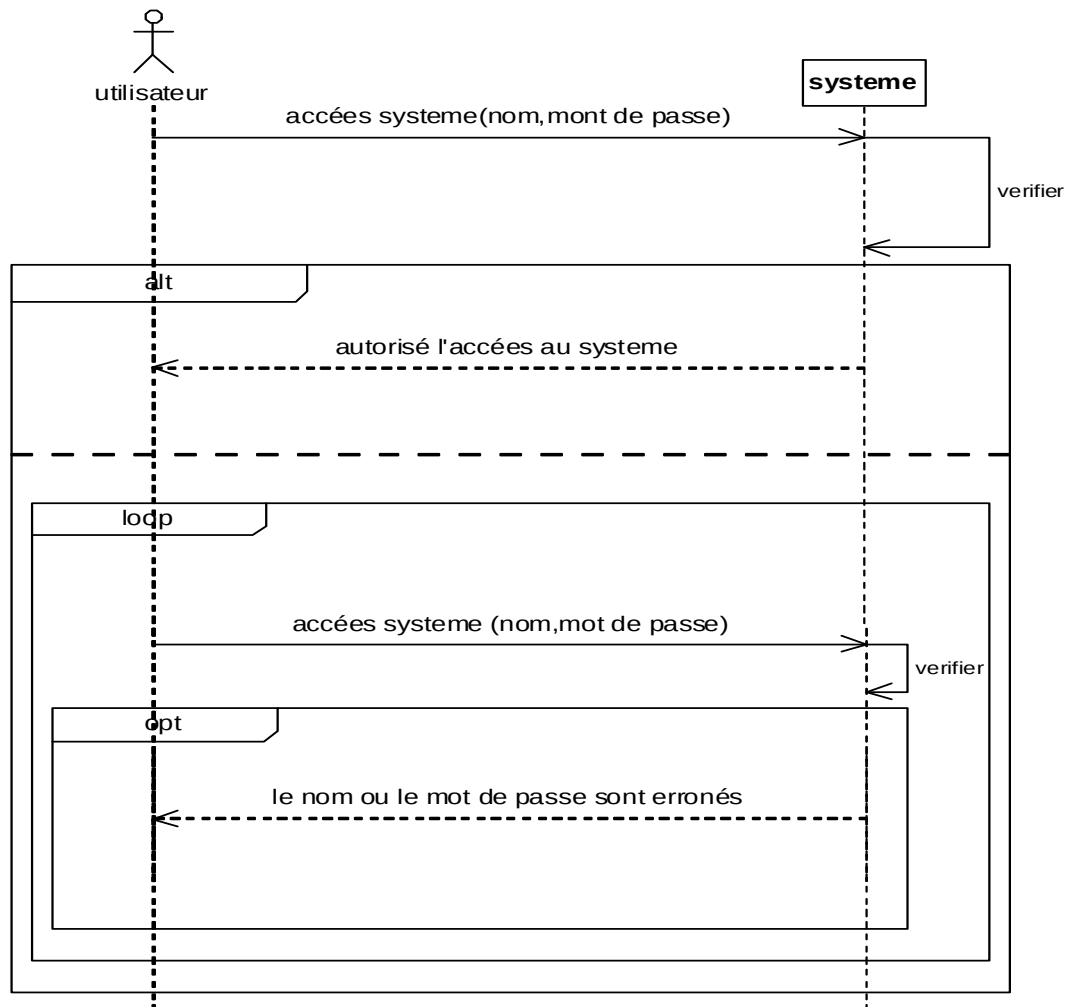


Figure 2.2 Diagramme de séquence système de cas d'utilisation Authentifier.

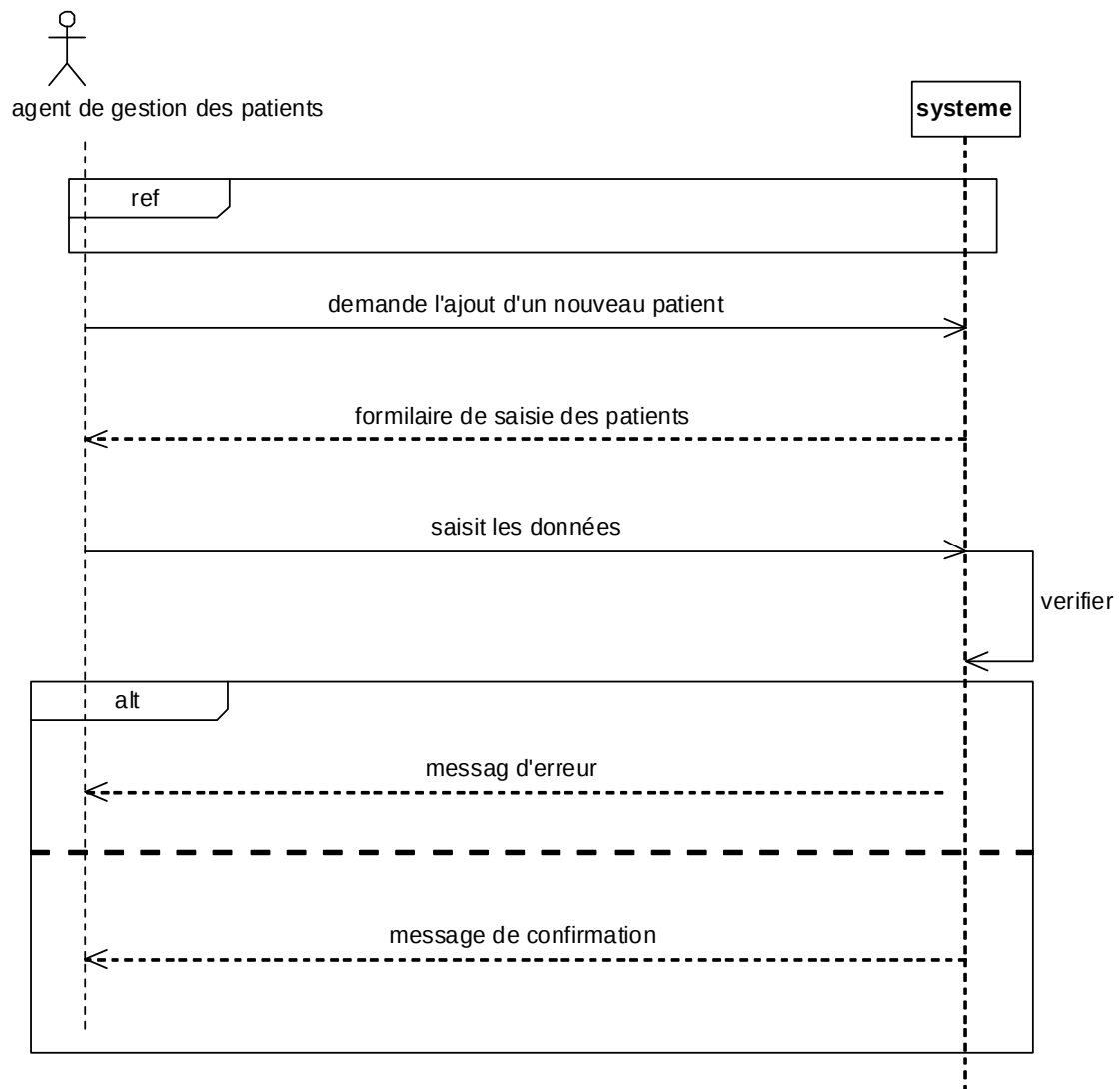


Figure 2.3 Diagramme de séquence système de cas d'utilisation Ajouter patient.

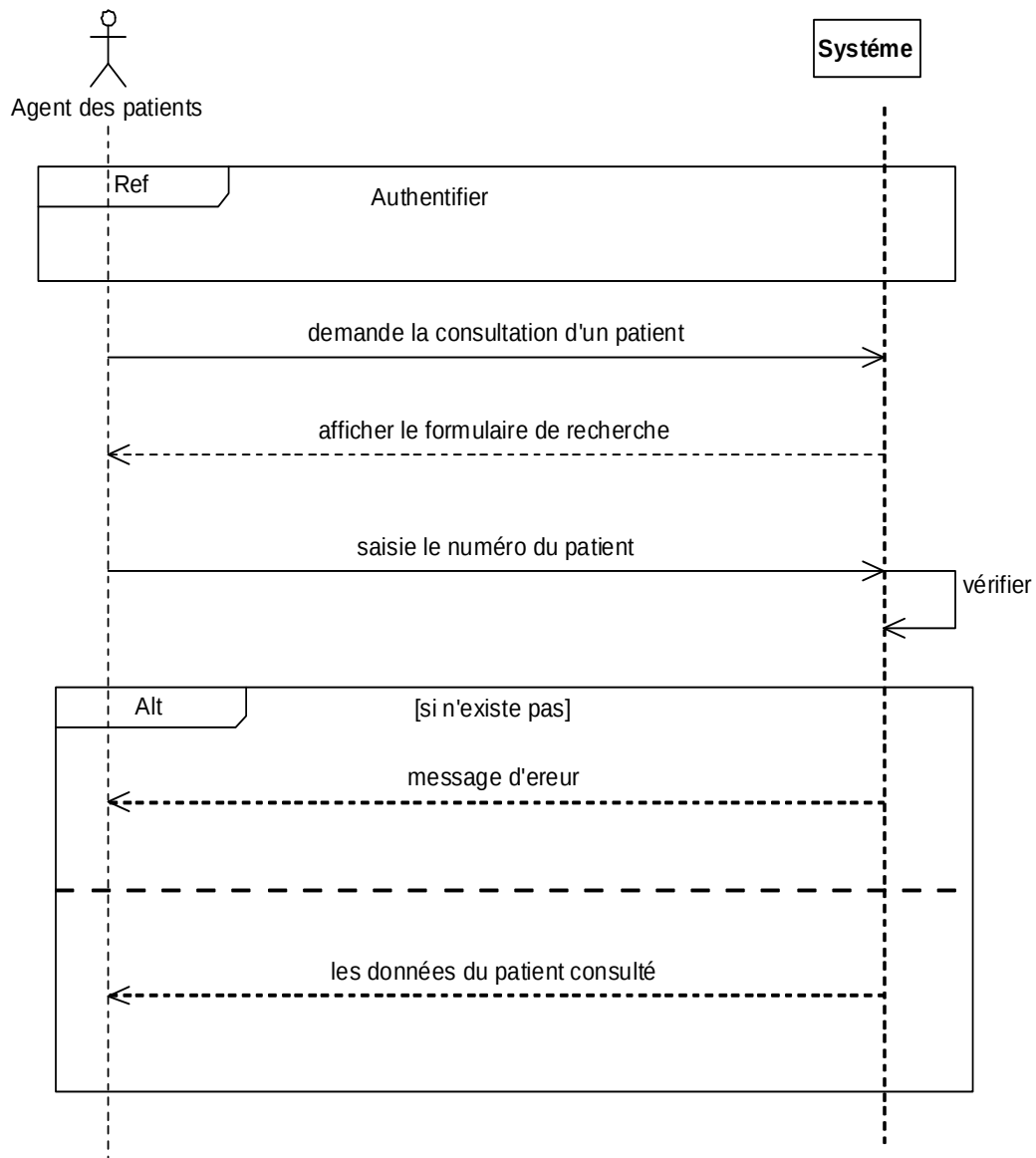


Figure 2.4 Diagramme de séquence système de cas d'utilisation Consulter patient.

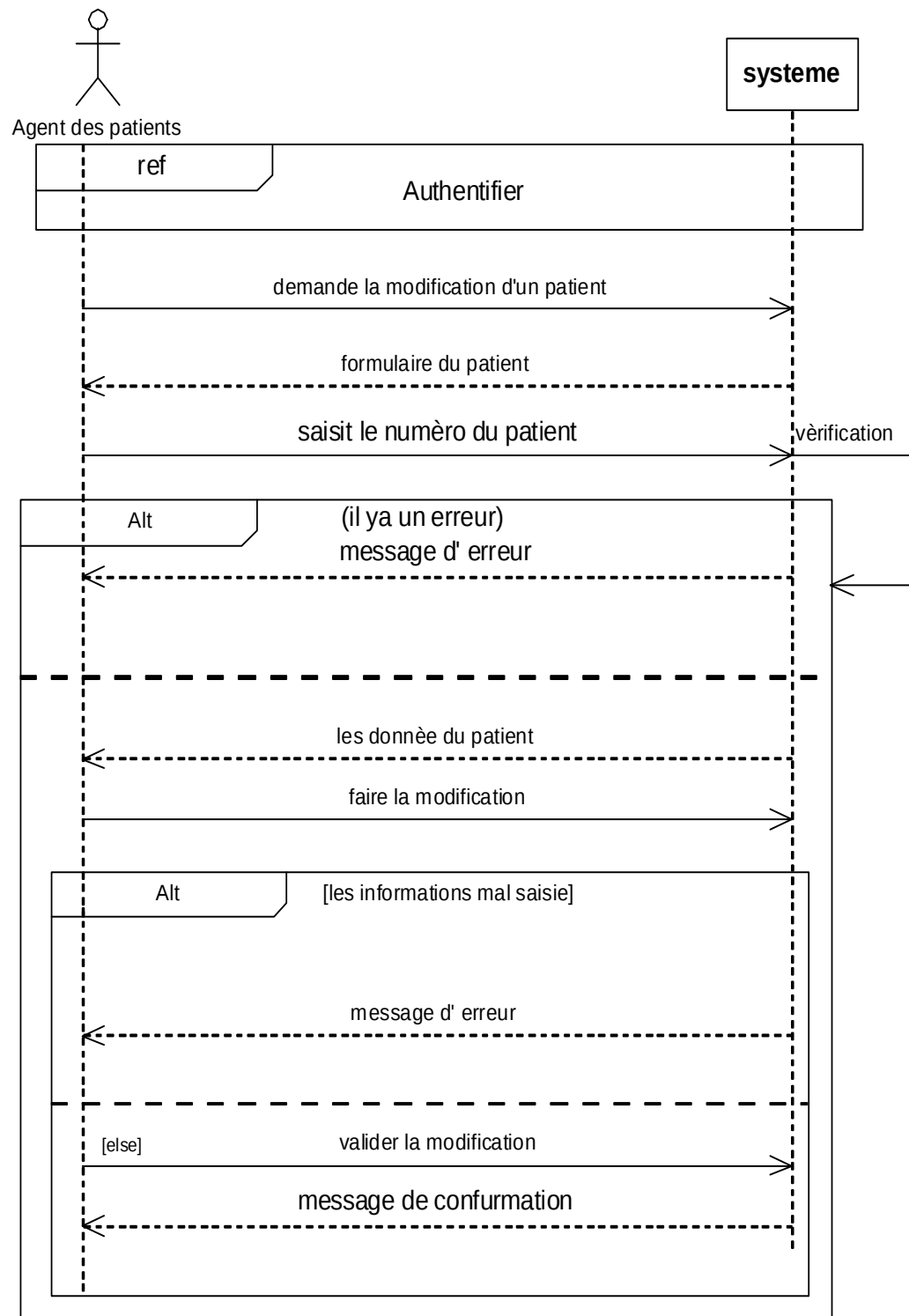


Figure 2.5 Diagramme de séquence système de cas d'utilisation Modifier patient.

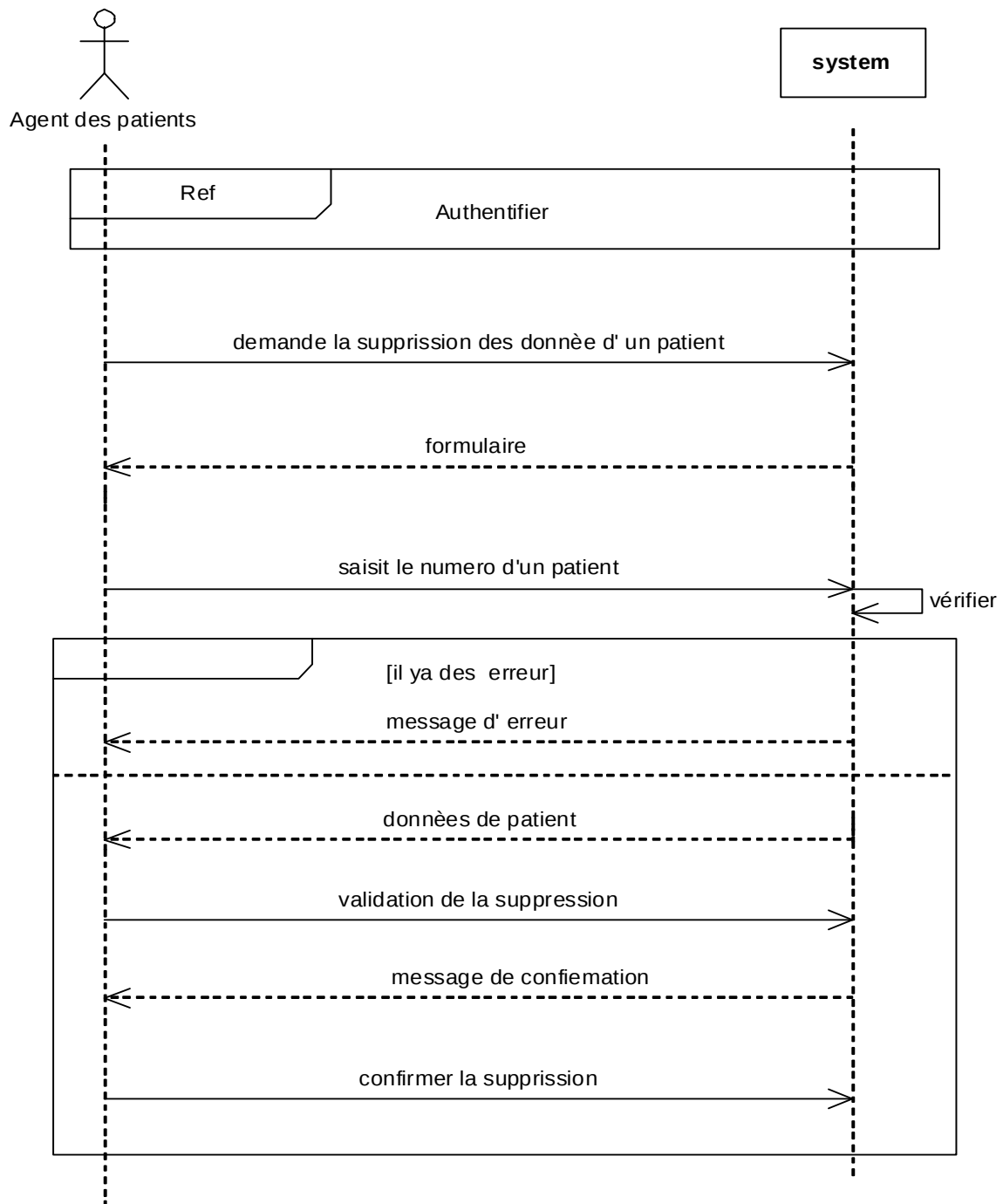


Figure 2.6 Diagramme de séquence système de cas d'utilisation Supprimer patient.

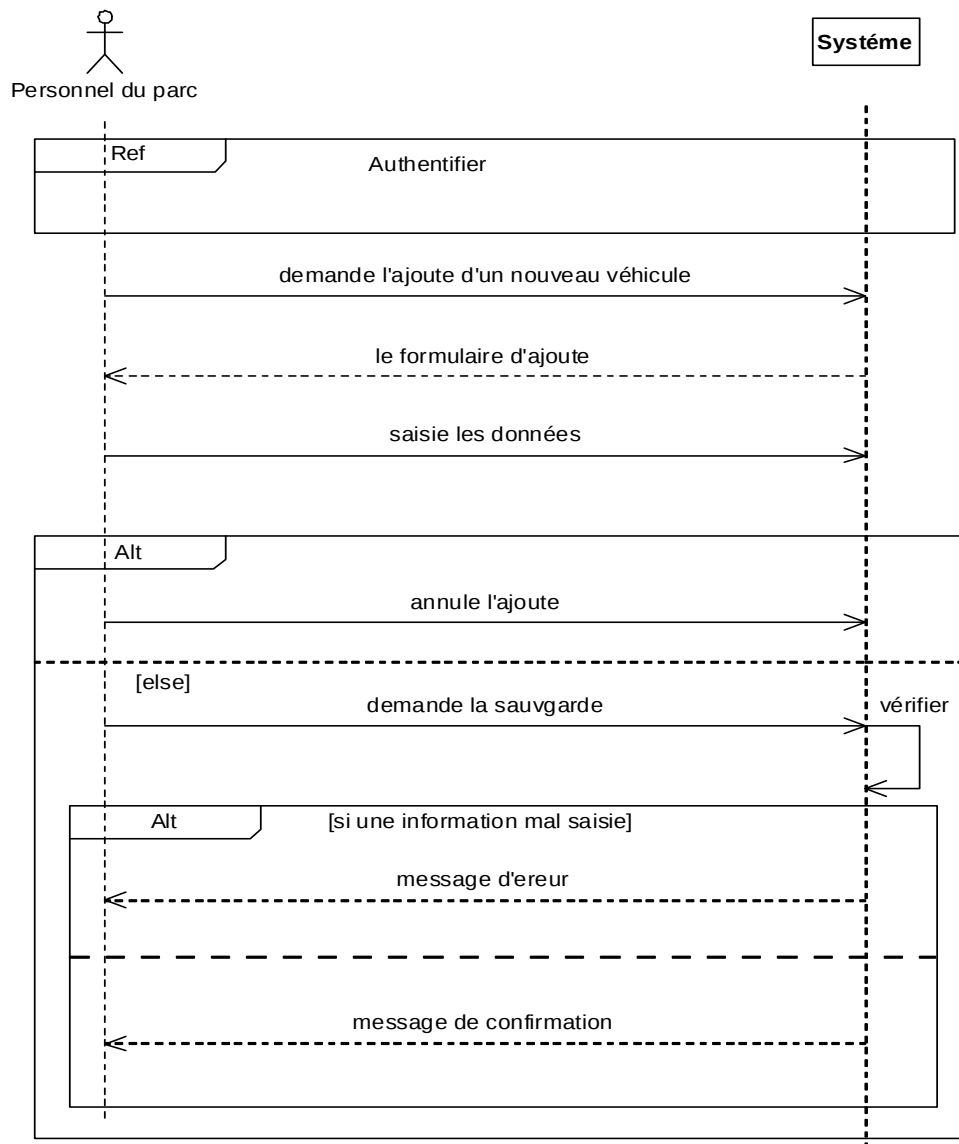


Figure 2.7 Diagramme de séquence système de cas d'utilisation Ajouter véhicule

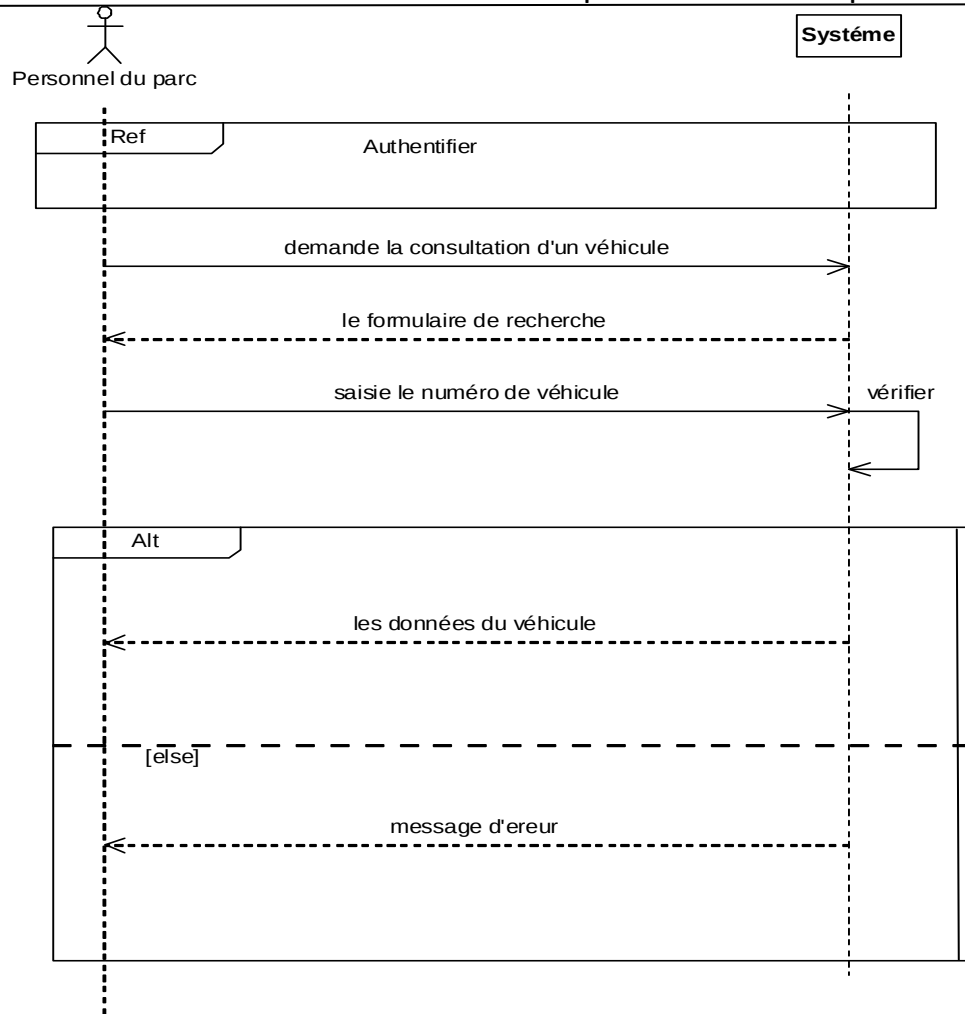


Figure 2.8 Diagramme de séquence système de cas d'utilisation Consulter véhicule.

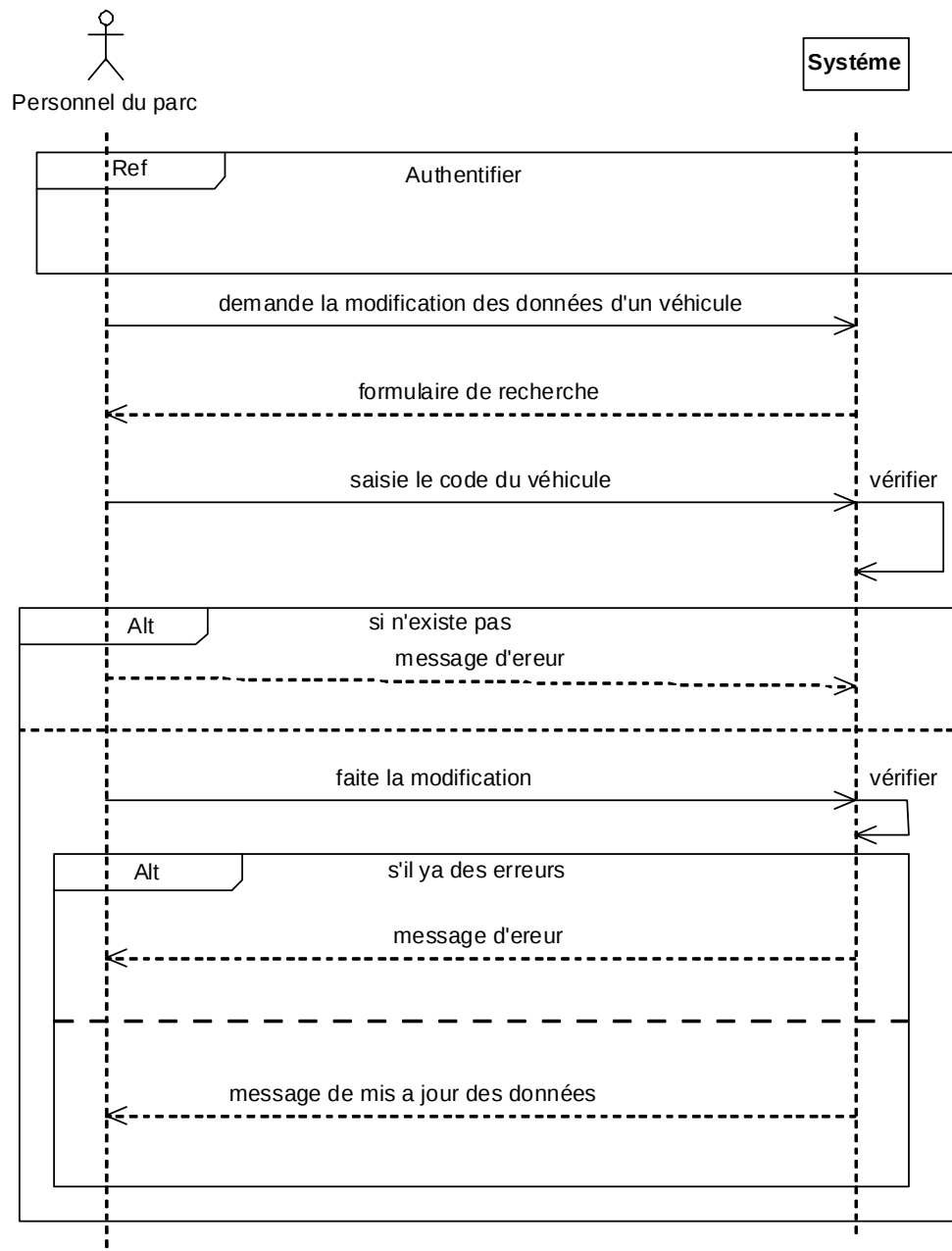


Figure 2.9 Diagramme de séquence système de cas d'utilisation Modifier véhicule.

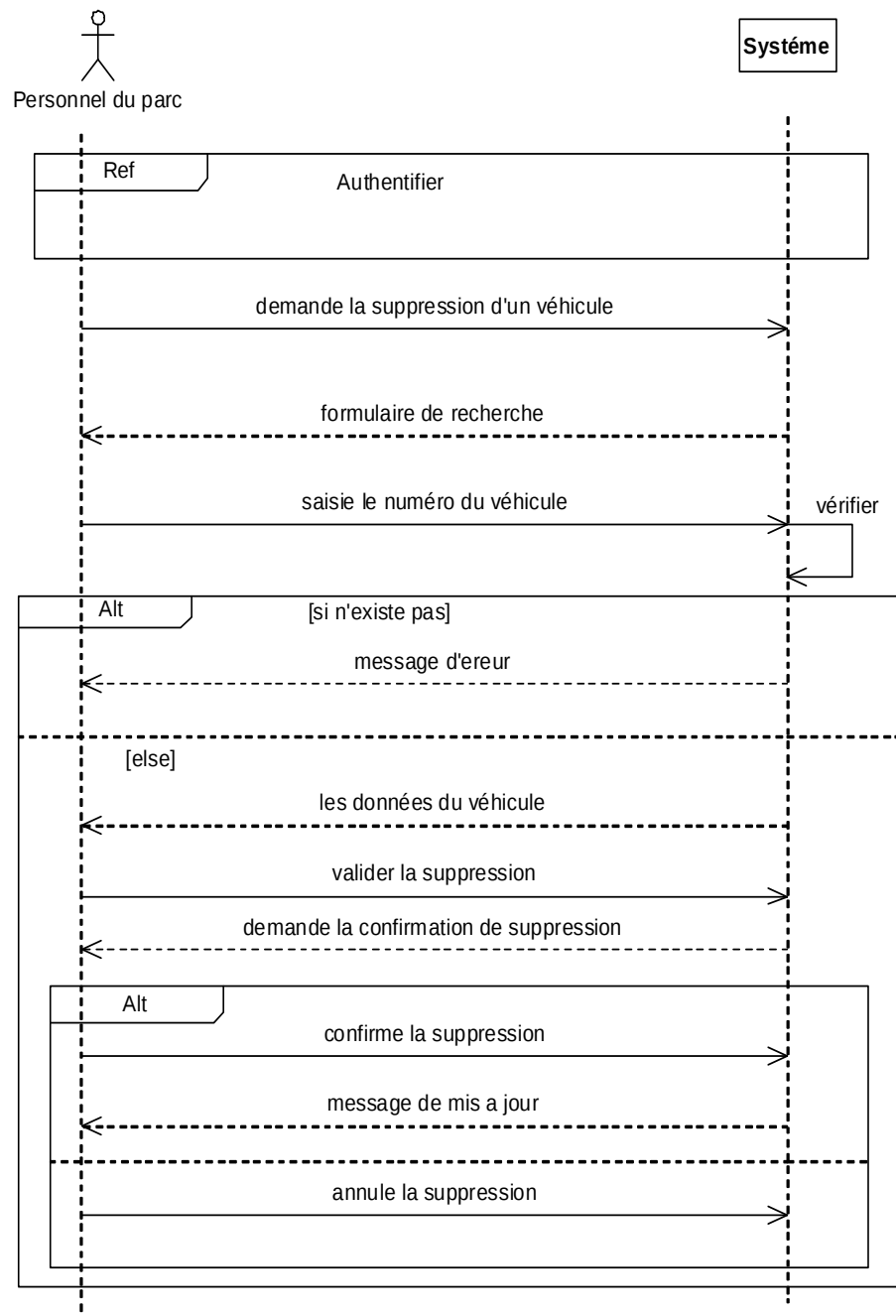


Figure 2.10 Diagramme de séquence système de cas d'utilisation Supprimer véhicule.

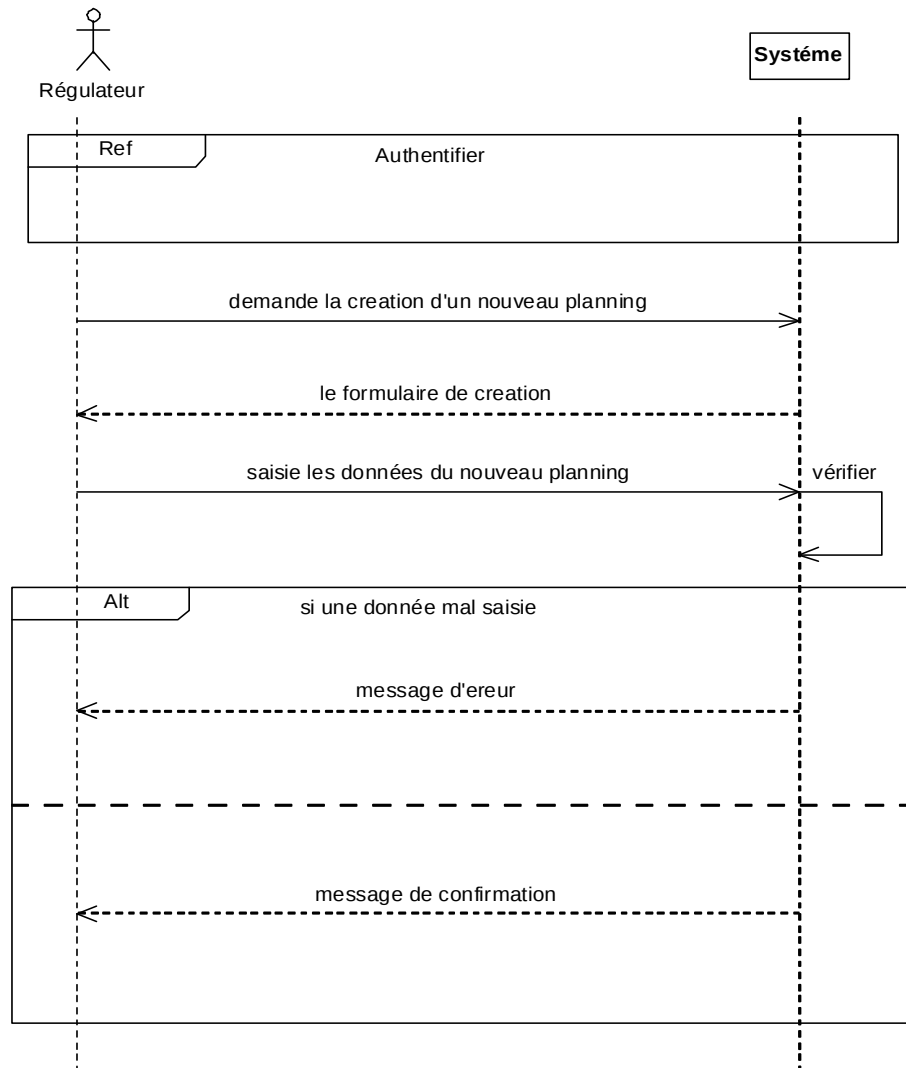


Figure 2.11 Diagramme de séquence système de cas d'utilisation Créer planning.

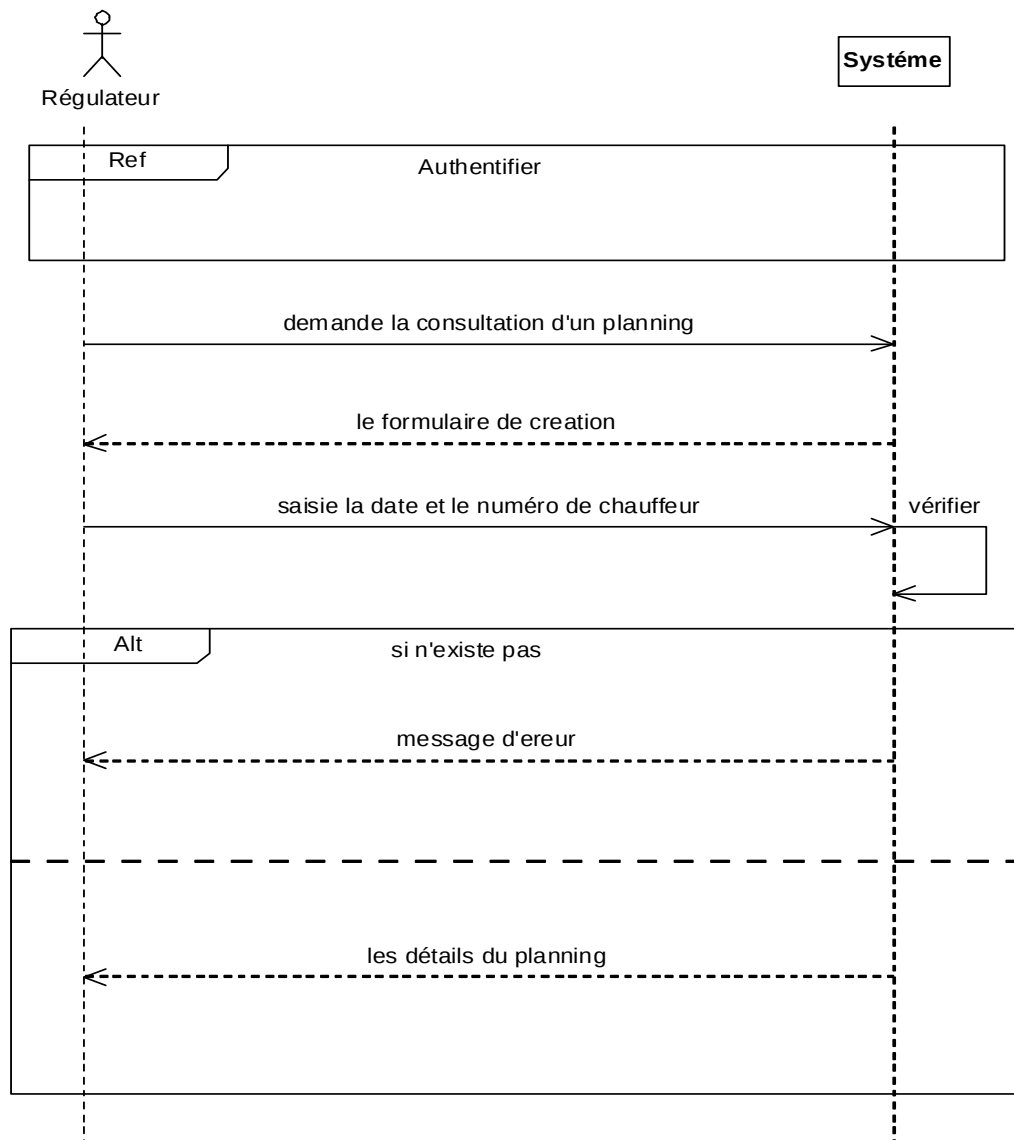


Figure 2.12 Diagramme de séquence système de cas d'utilisation Consulter planning.

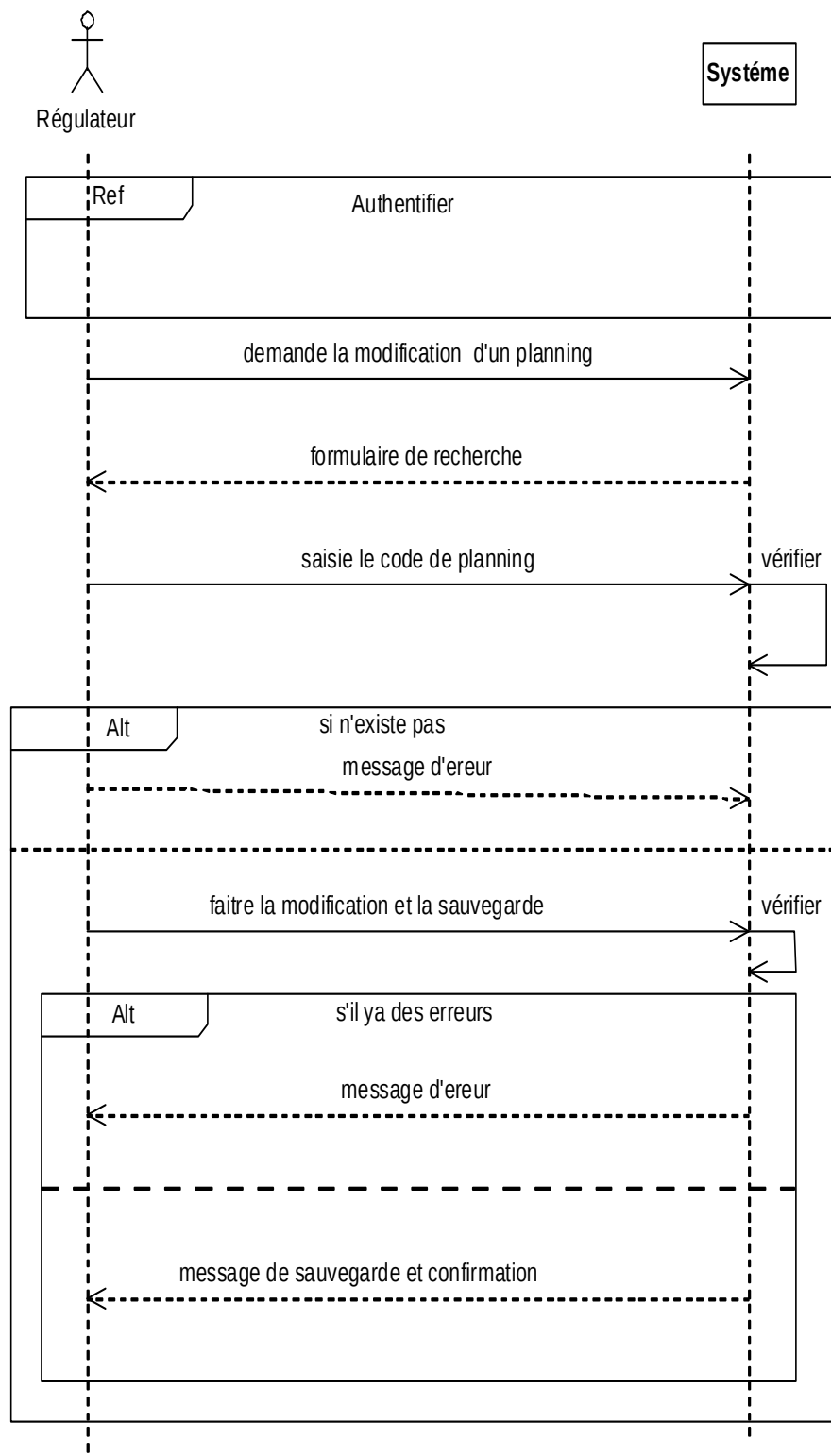


Figure 2.13 Diagramme de s séquence système de cas d'utilisation Modifié planning.

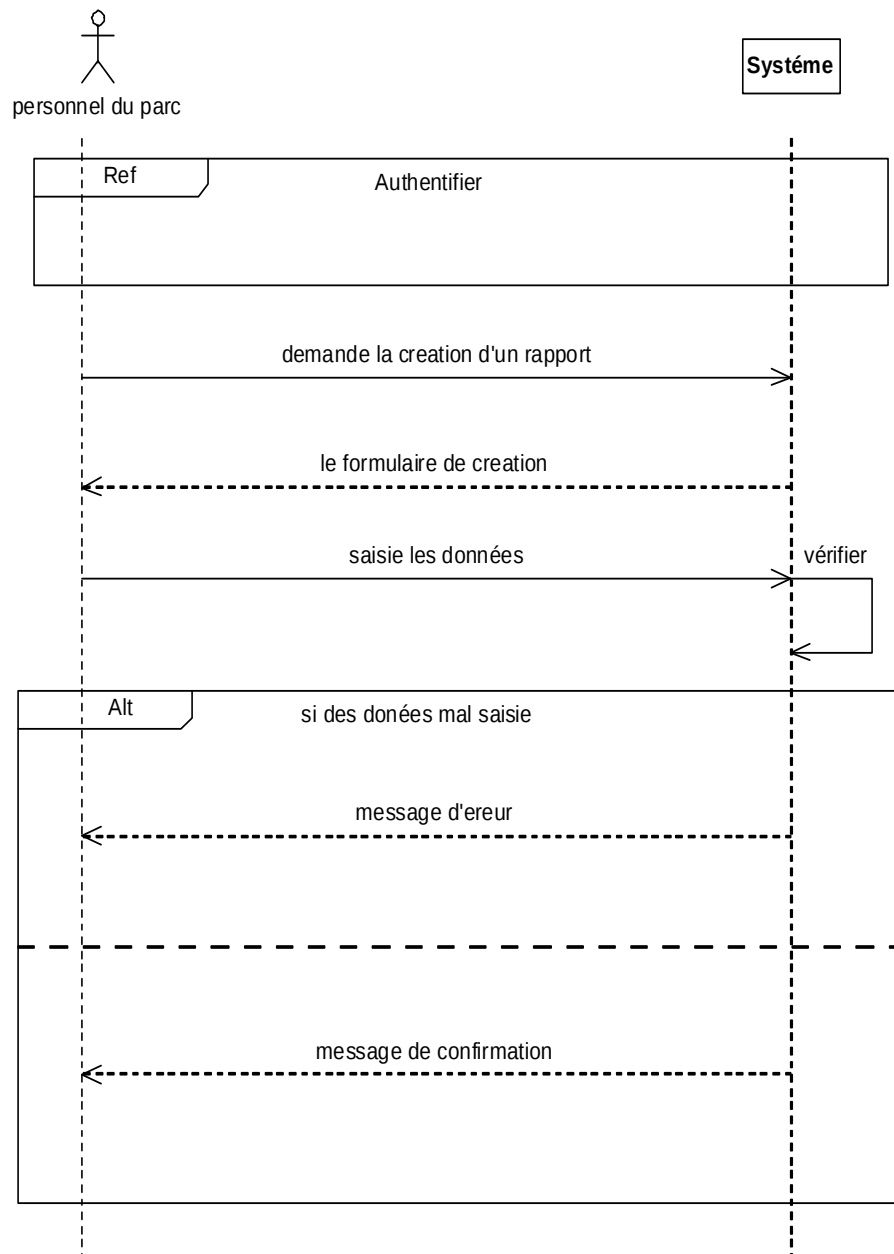


Figure 2.14 Diagramme de séquence système de cas d'utilisation Créer rapport.

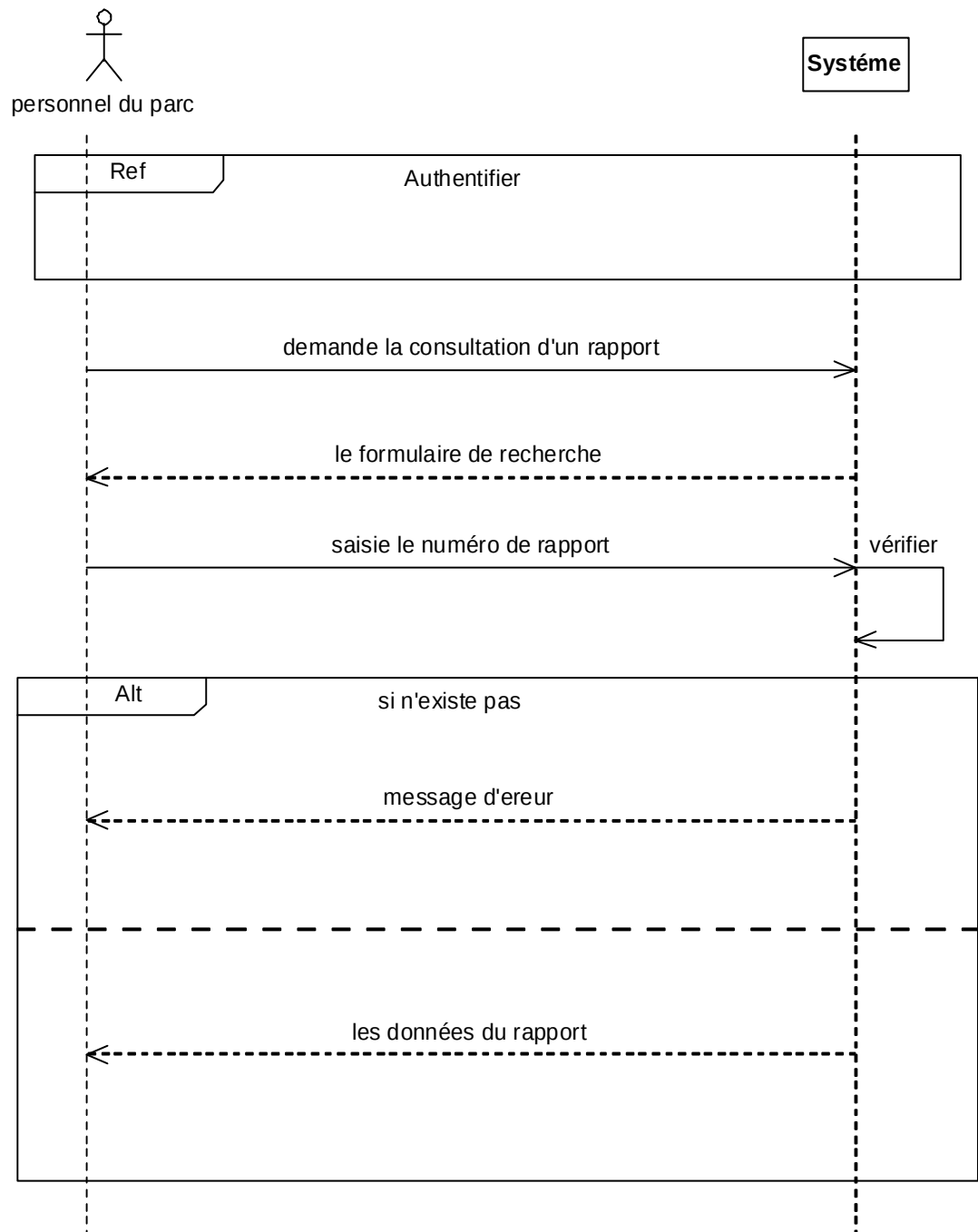


Figure 2.15 Diagramme de séquence système de cas d'utilisation Consulter rapport.

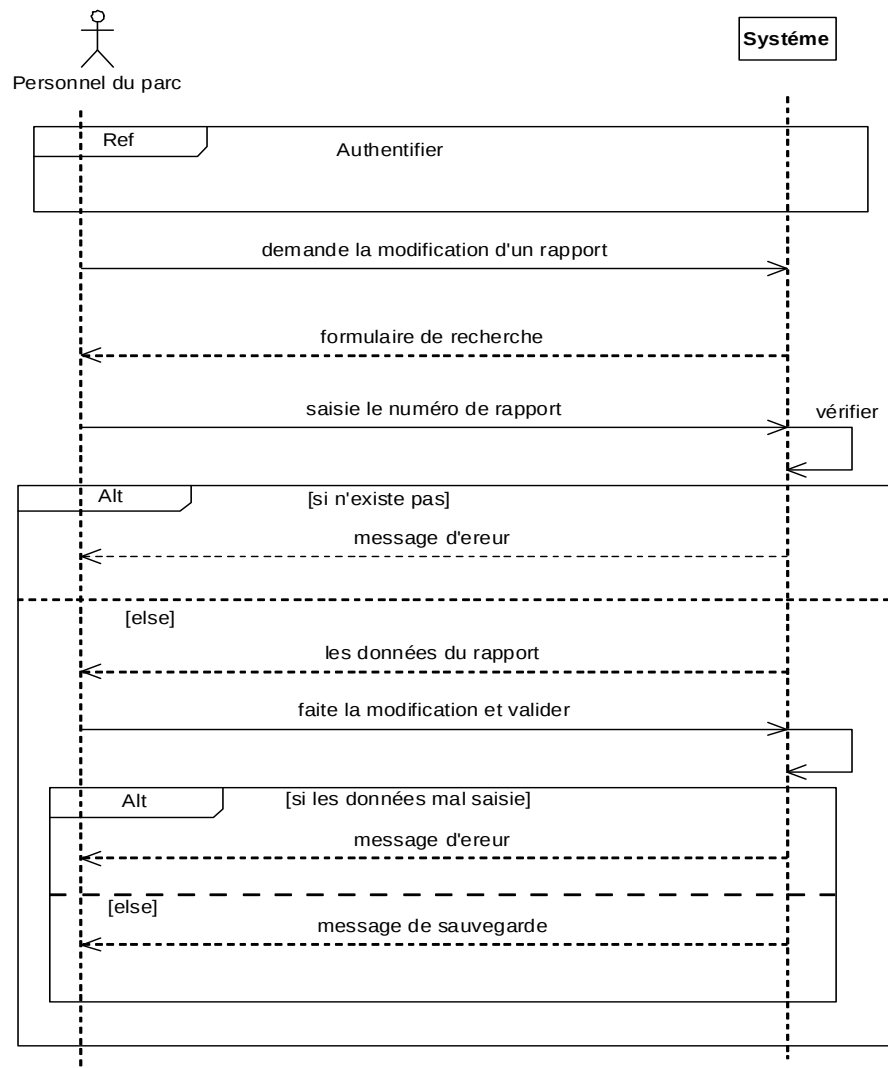


Figure 2.16 Diagramme de séquence système de cas d'utilisation Modifier rapport.

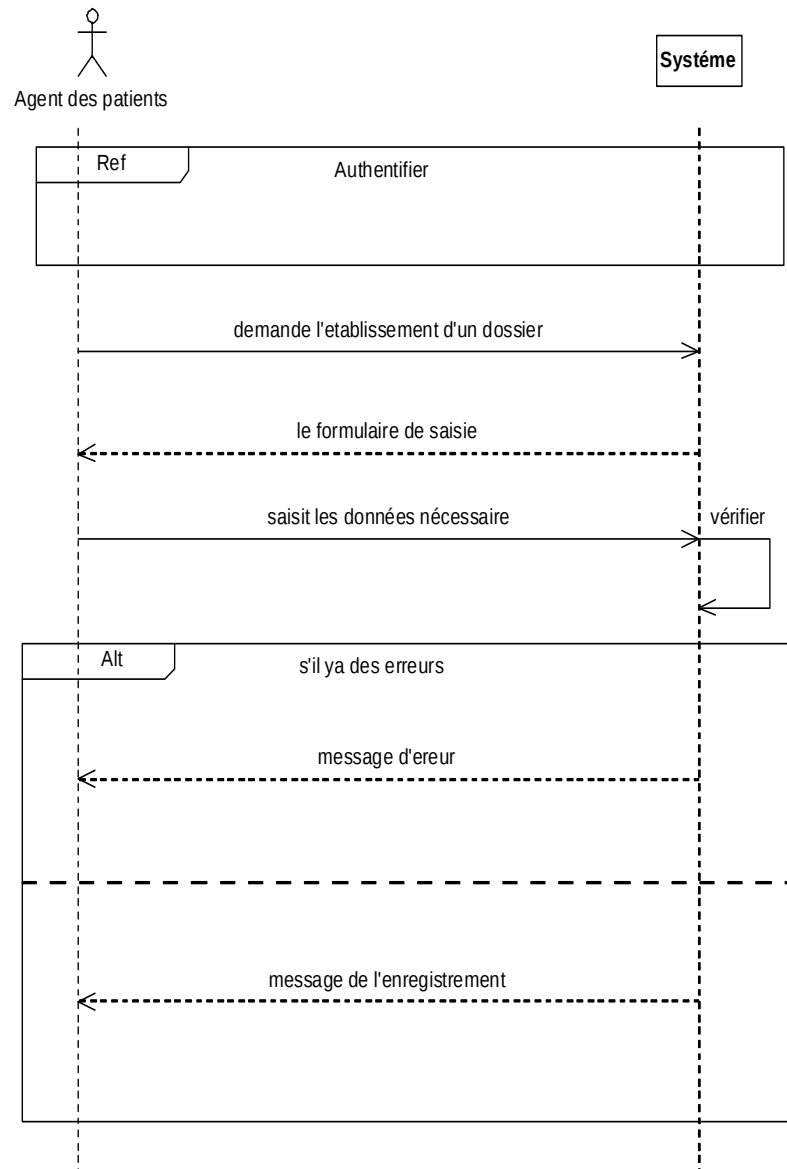


Figure 2. 17 Diagramme de séquence système de cas d'utilisation Etablir le dossier d'un patient.

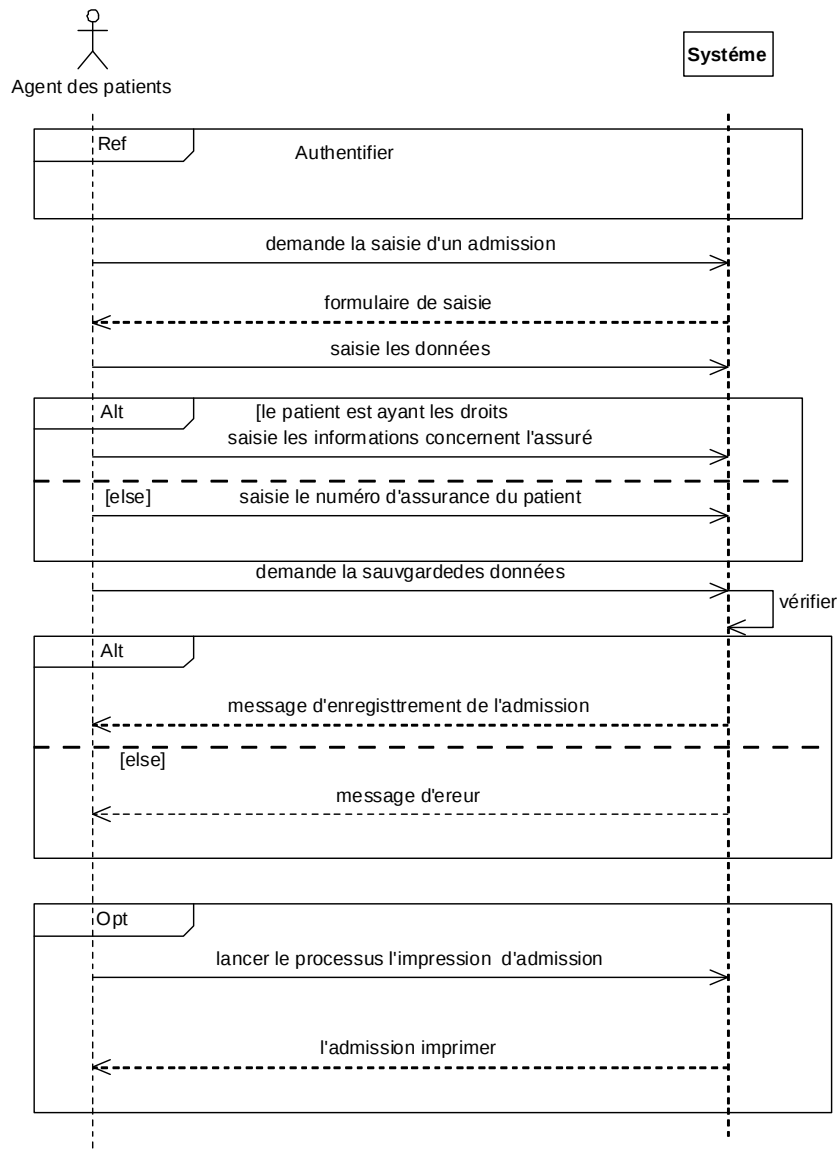


Figure 2.18 Diagramme de séquence système de cas d'utilisation Accorder l'admission.

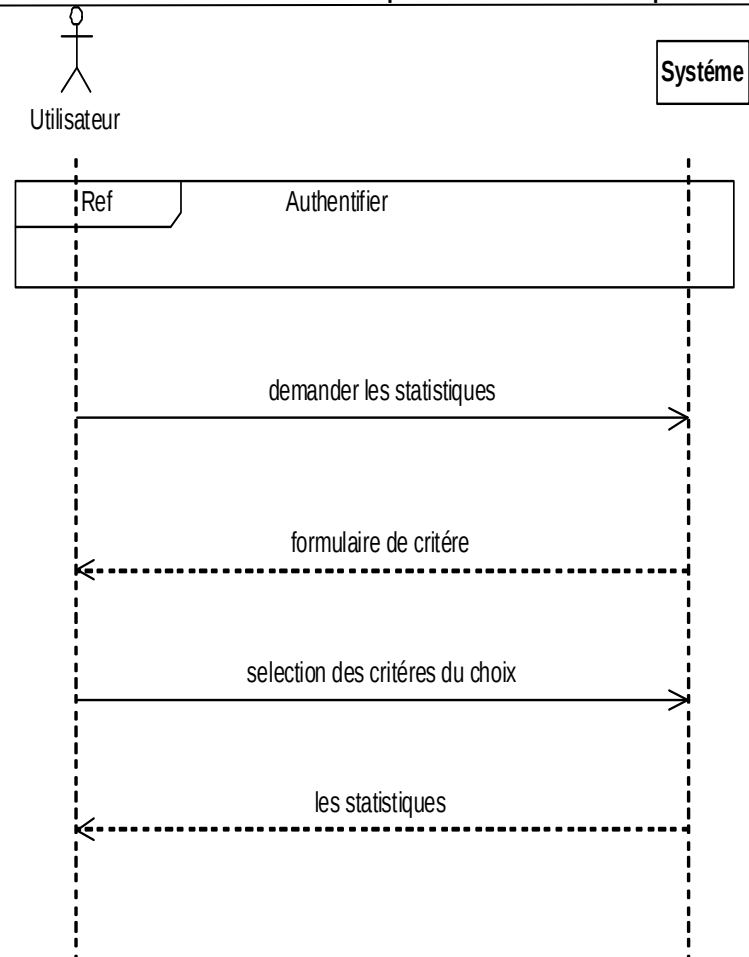


Figure 2.19 Diagramme de séquence de cas d'utilisation Consulter les statistiques.

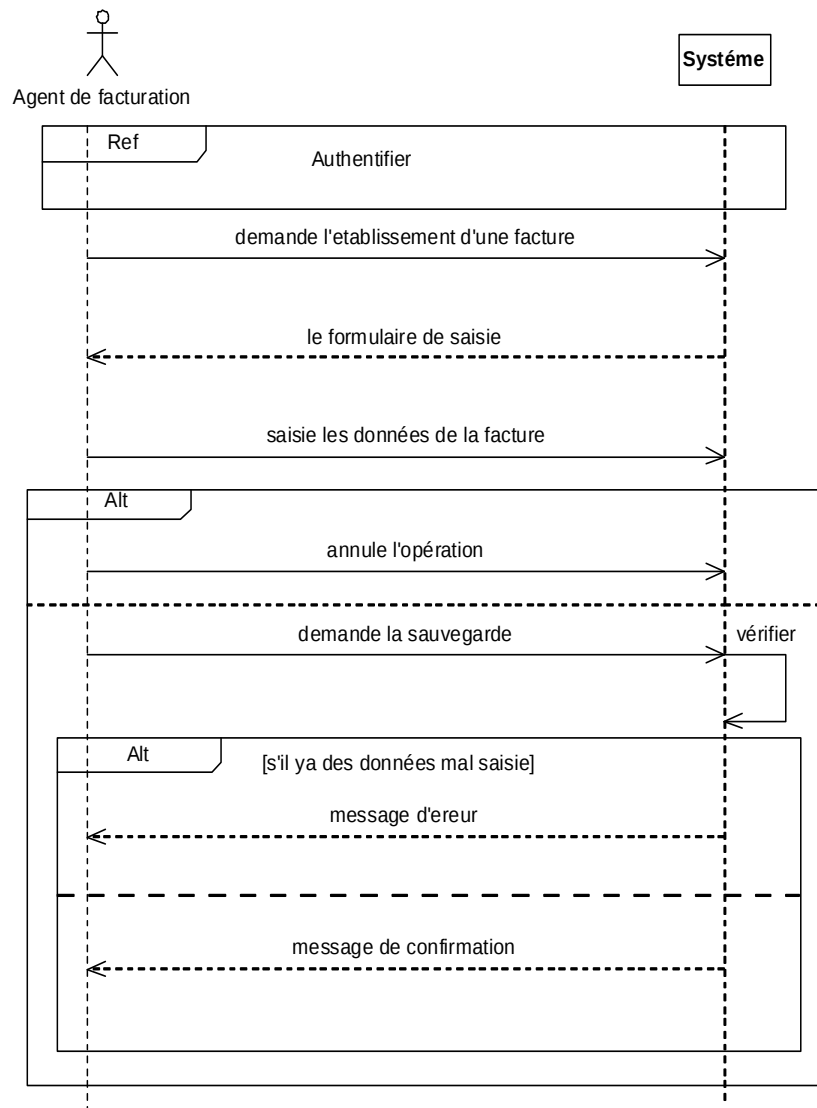


Figure 2.20 Diagramme de séquence système de cas d'utilisation Etablir la facture.

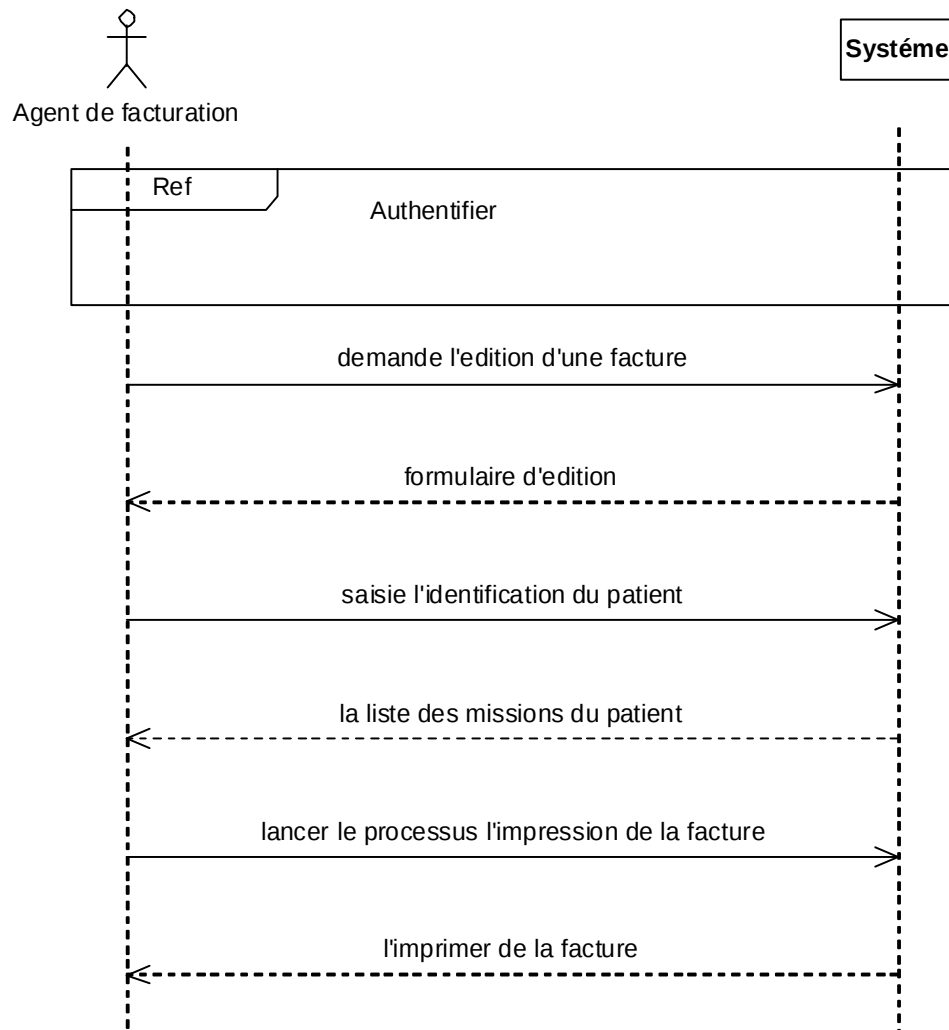


Figure 2.21 Diagramme de séquence système de cas d'utilisation Editer la facture.

2.3.1.5 Les diagrammes de classes participantes.

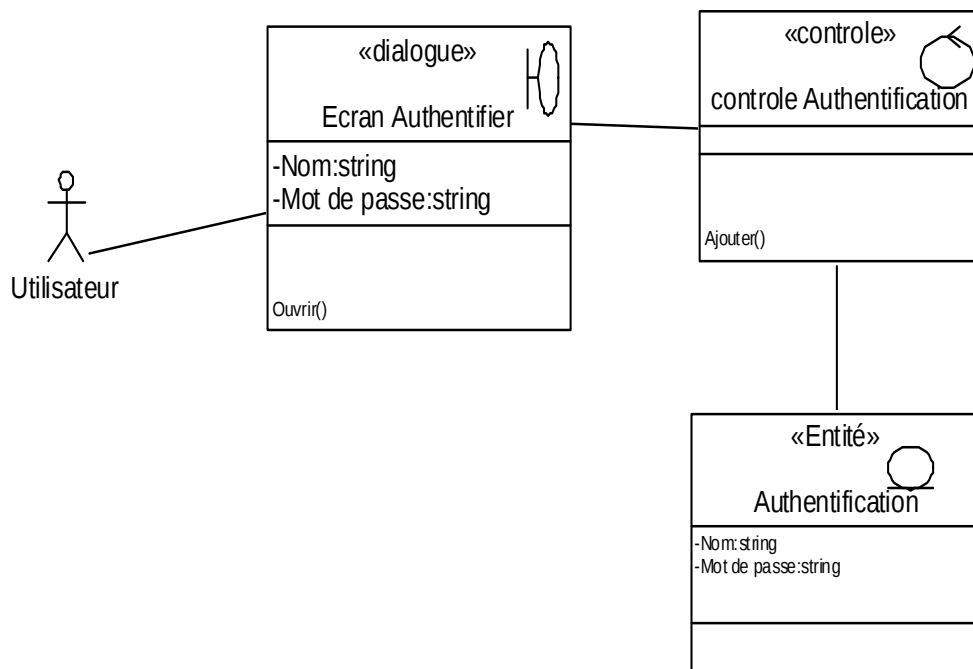


Figure 2.22 Diagramme de classe participant du cas d'utilisation Authentifier.

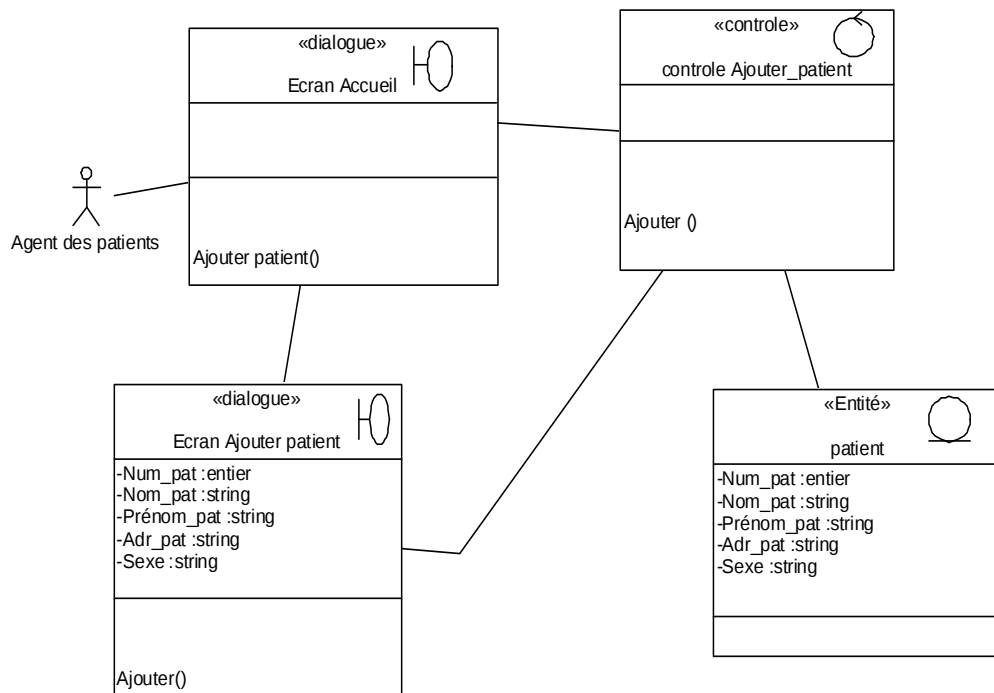


Figure2.23 Diagramme de classe participant du cas d'utilisation Ajouter patient.

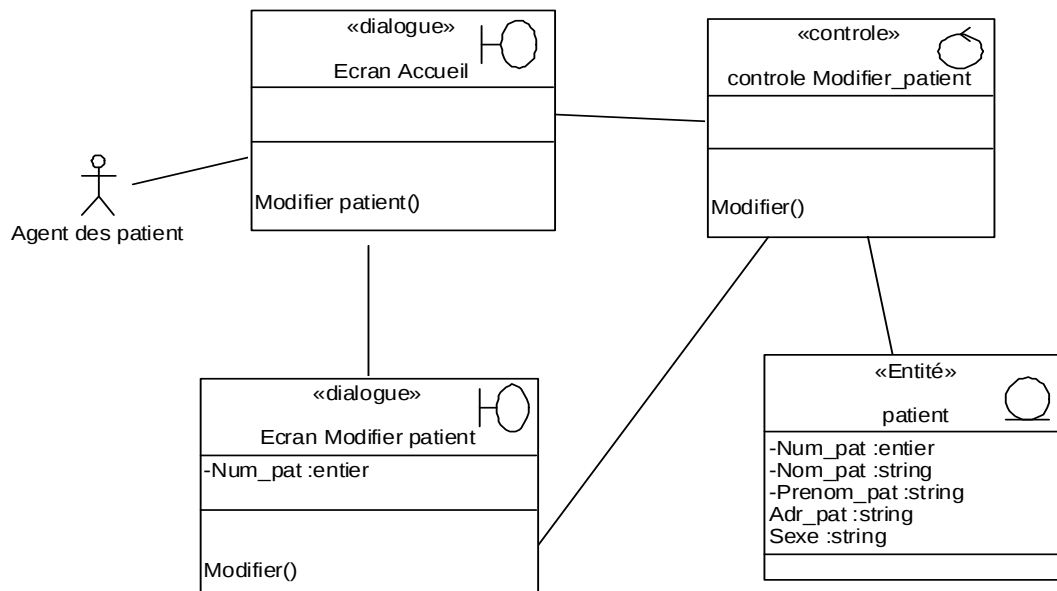


Figure 2.24 Diagramme de classe participant du cas d'utilisation Modifier patient.

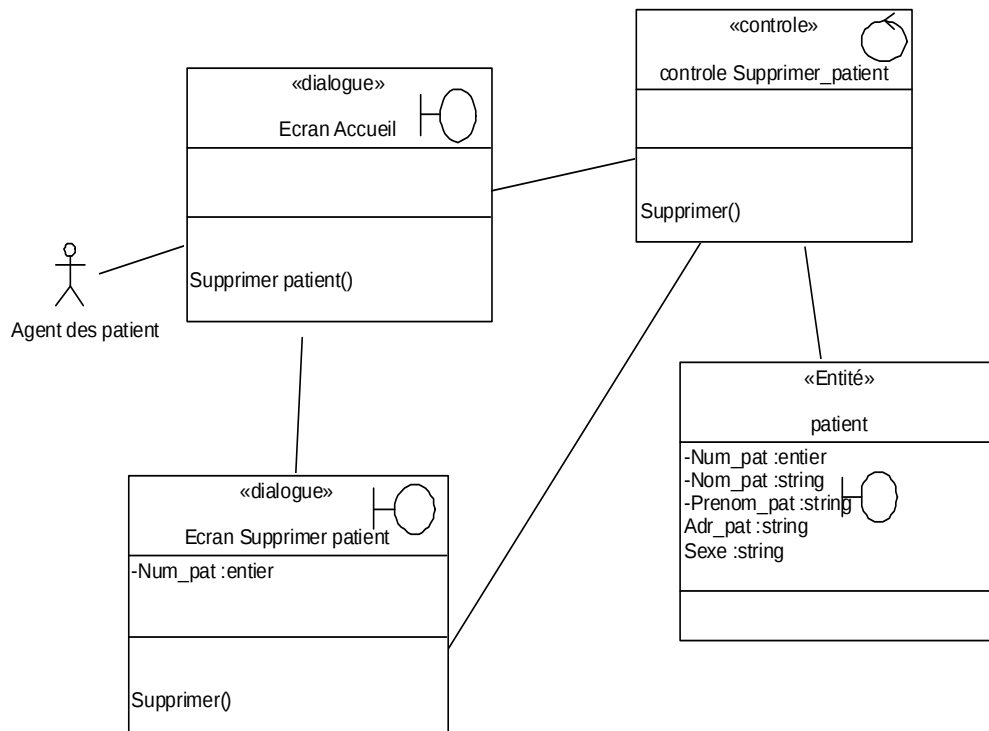


Figure 2.25 Diagramme de classe participant du cas d'utilisation Supprimer patient.

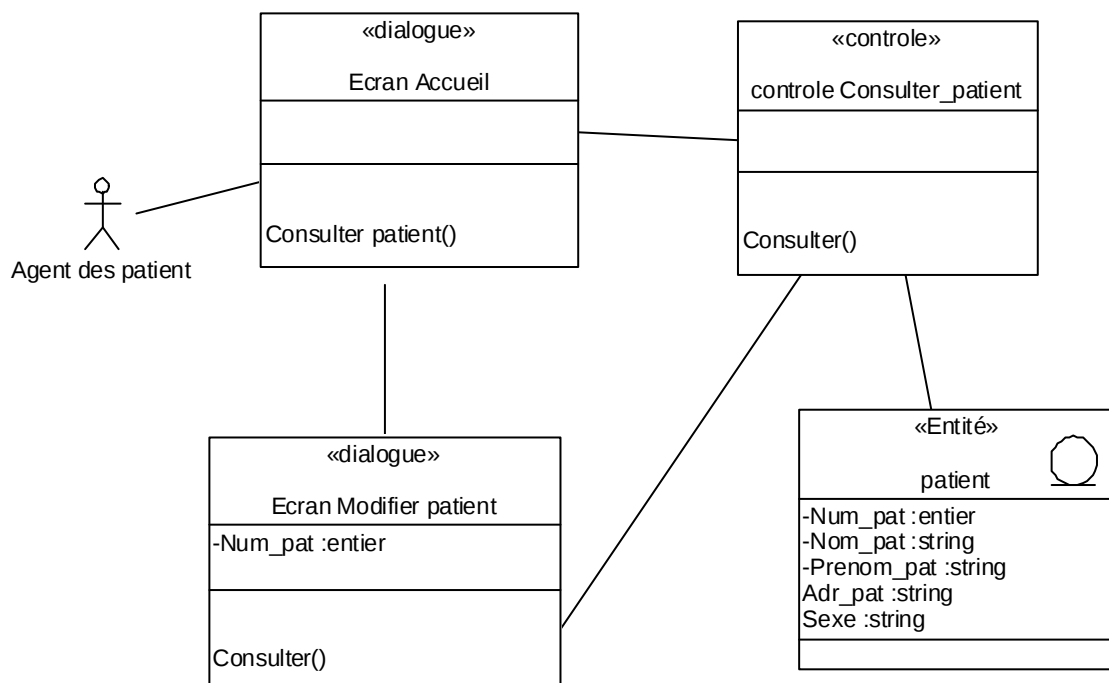


Figure 2.26 Diagramme de classe participant du cas d'utilisation Consulter patient.

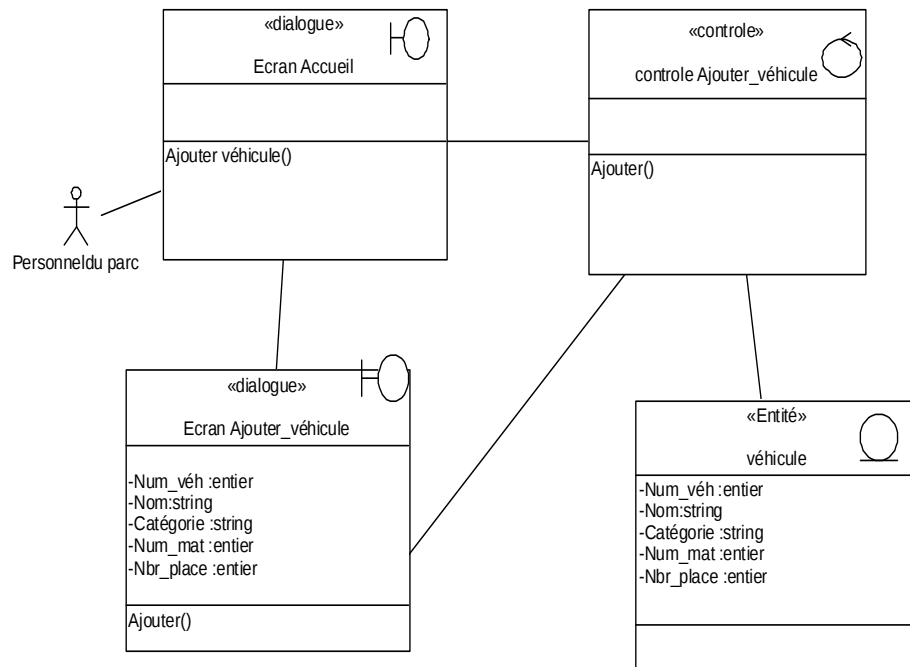


Figure 2.27 Diagramme de classe participant de cas d'utilisation Ajouter véhicule.

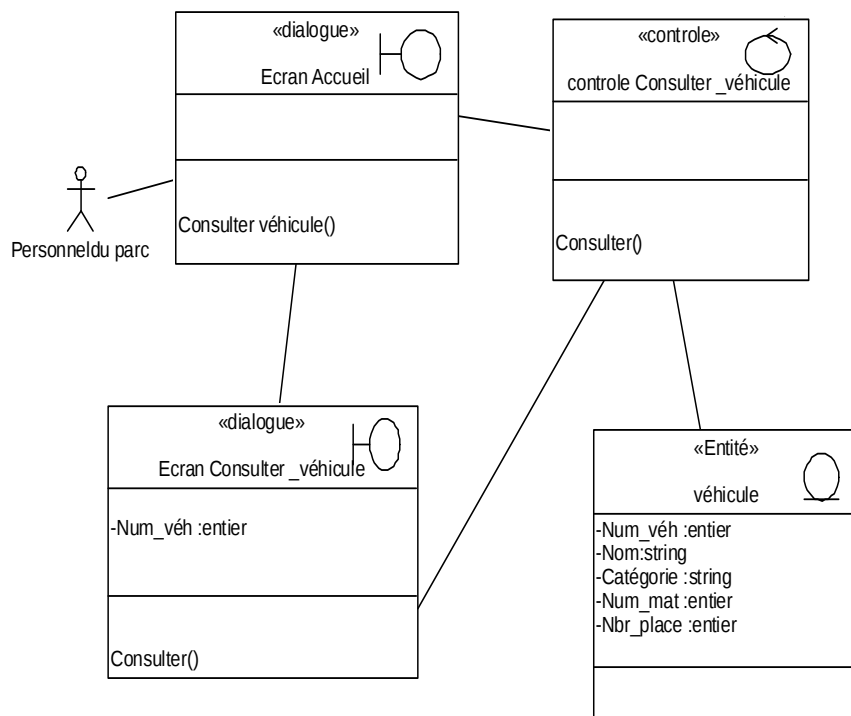


Figure 2.28 Diagramme de classe participant de cas d'utilisation Consulter véhicule.

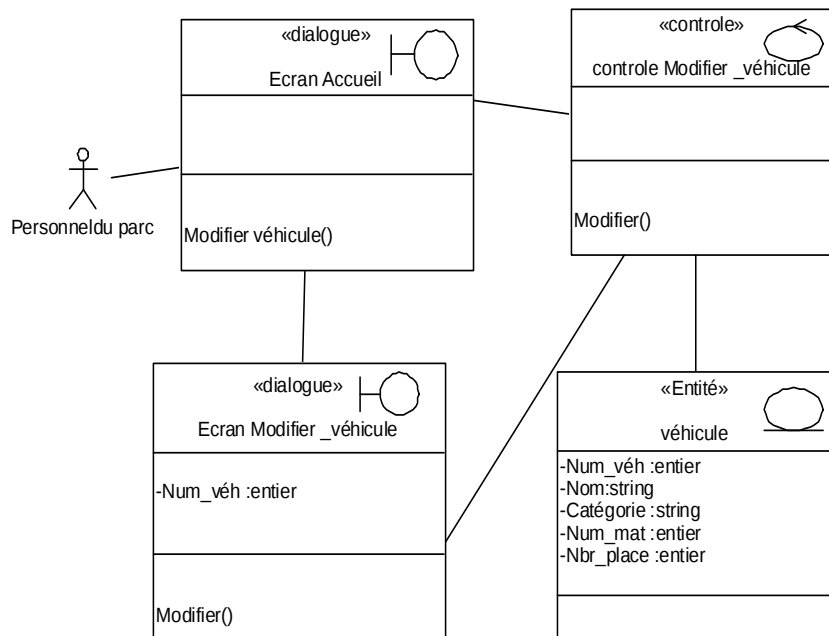


Figure 2.29 Diagramme de classe participant du cas d'utilisation Modifier véhicule.

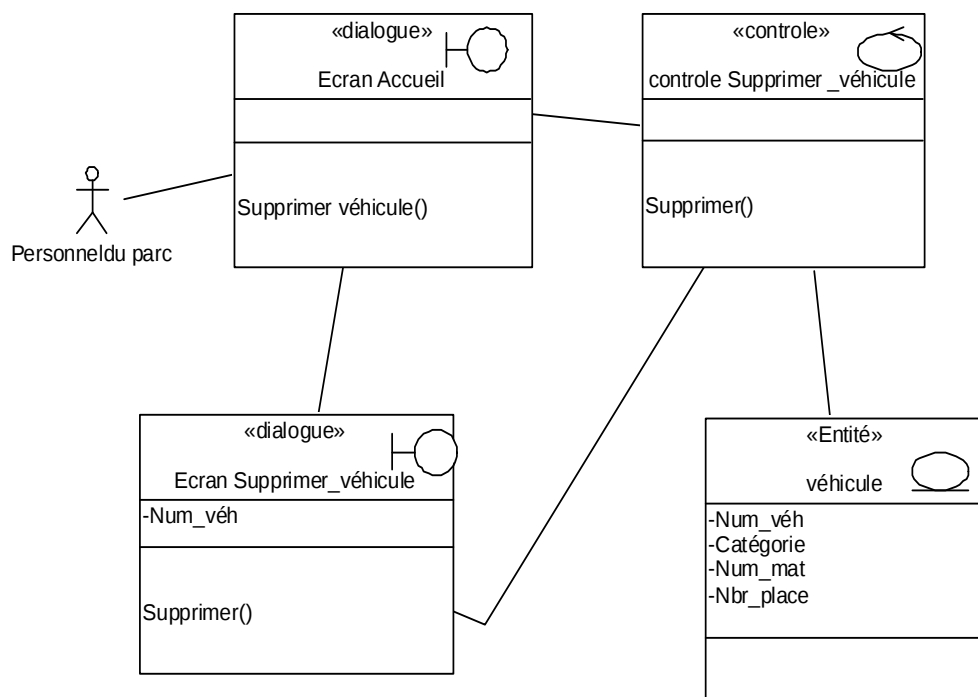


Figure 2.30 Diagramme de classe participant du cas d'utilisation Supprimer véhicule.

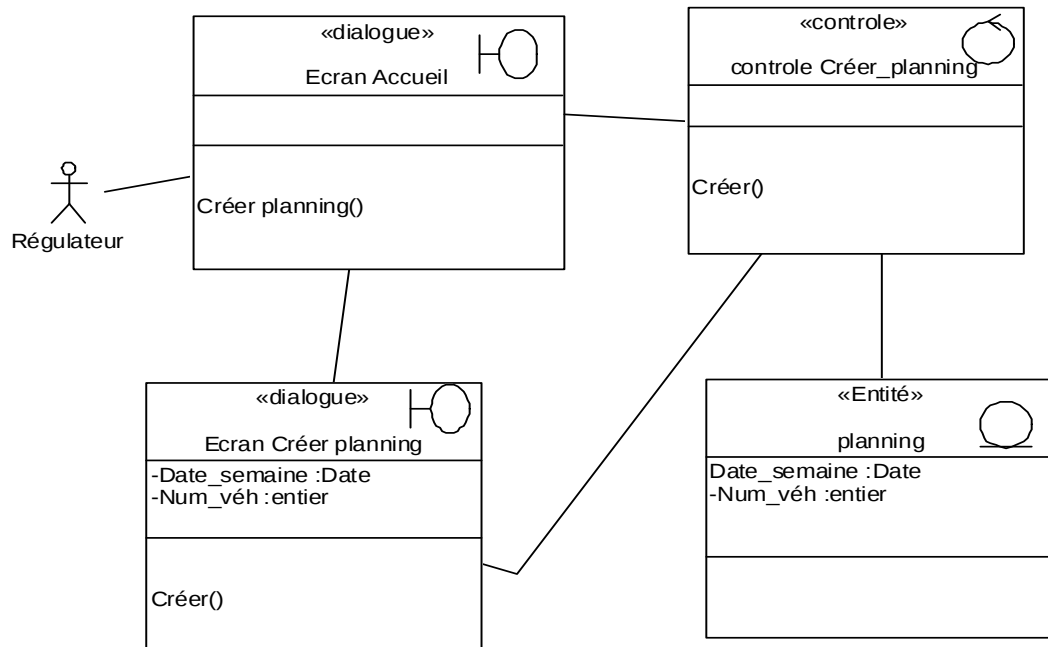


Figure 2.31 Diagramme de classe participant du cas d'utilisation Créer planning.

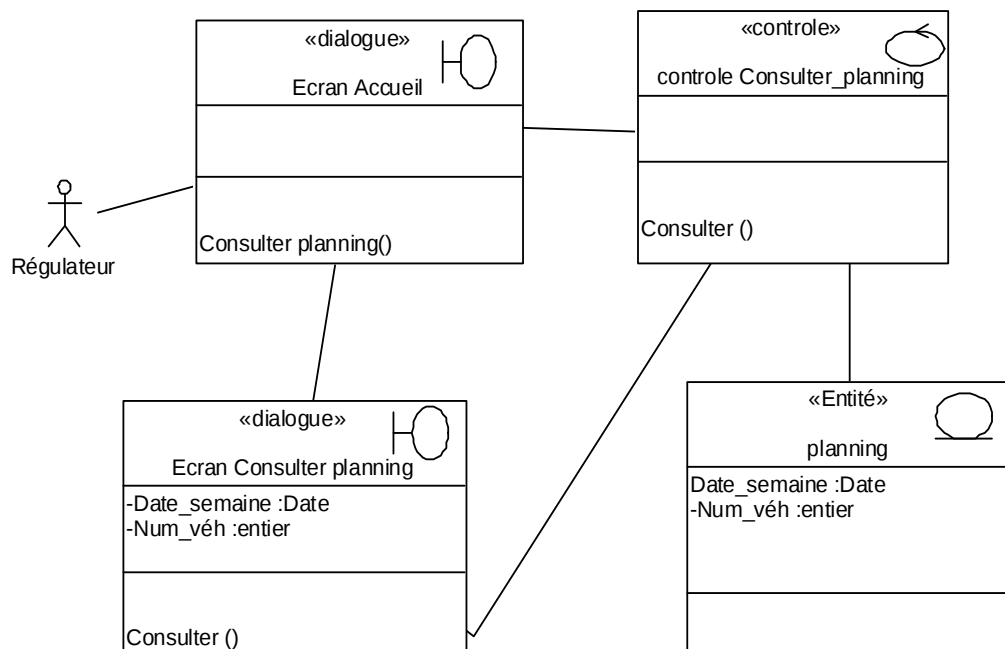


Figure 2.32 Diagramme de classe participant du cas d'utilisation Consulter planning.

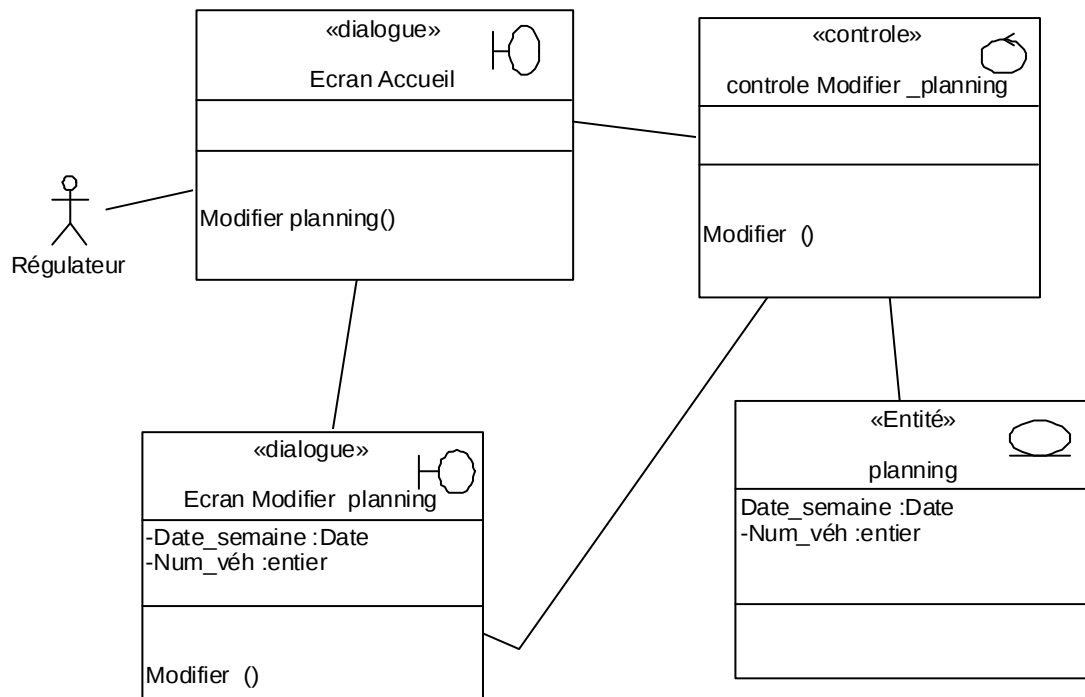


Figure 2.33 Diagramme de classe participant du cas d'utilisation Modifier planning.

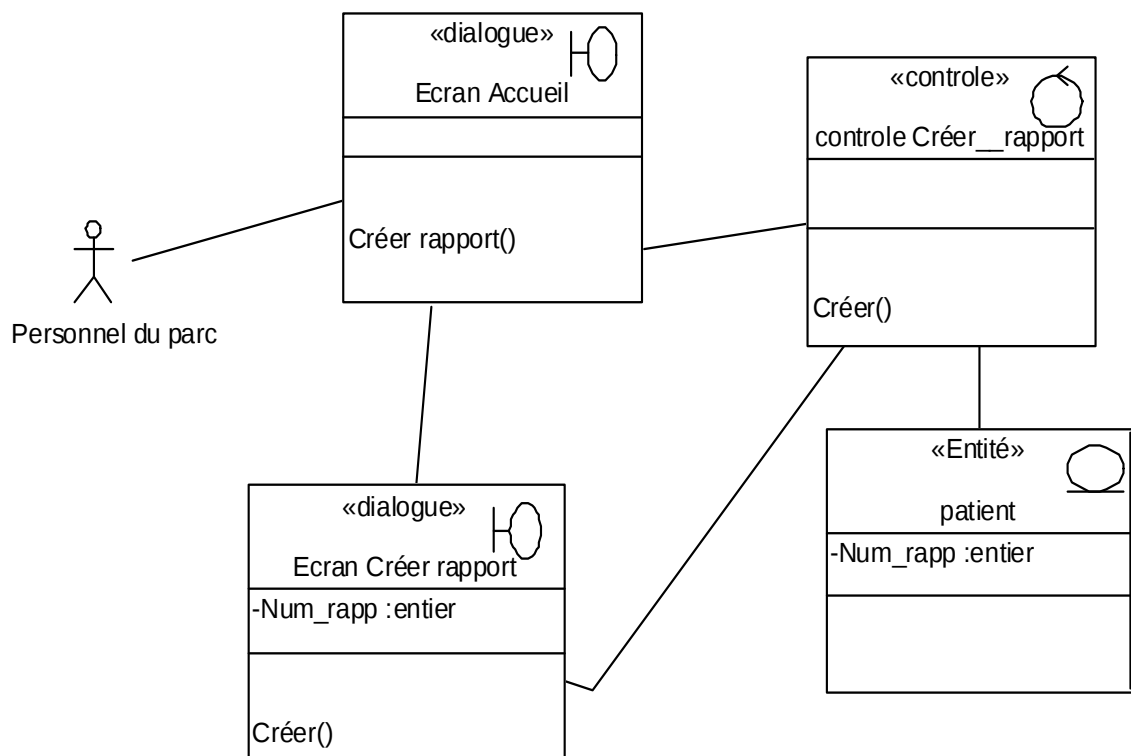


Figure 2.34 Diagramme de classe participant du cas d'utilisation Créer rapport.

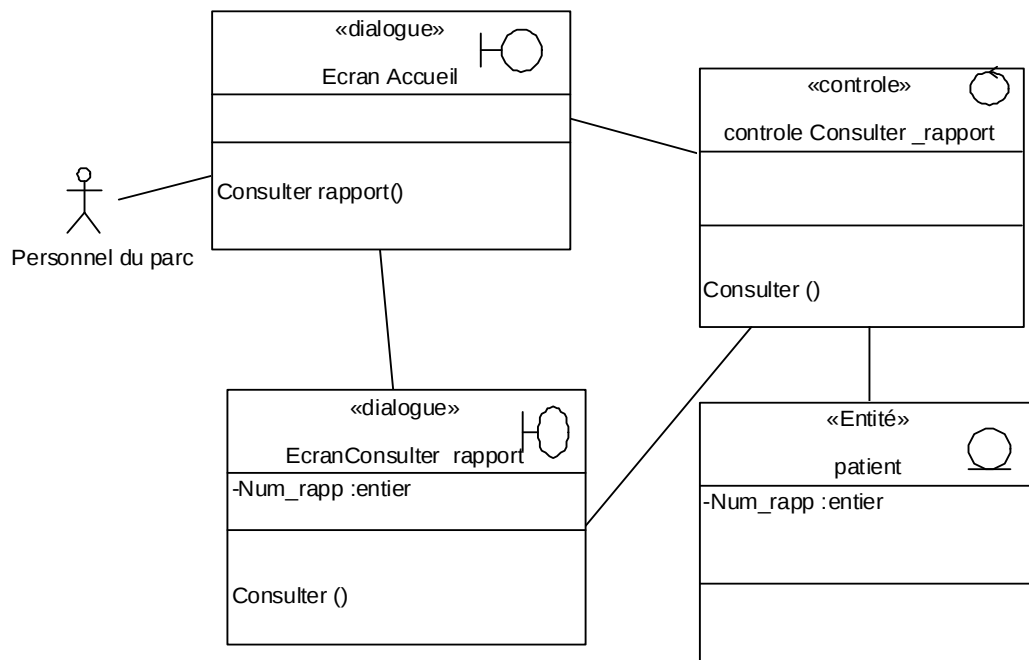


Figure 2.35 Diagramme de classe participant du cas d'utilisation Consulter rapport.

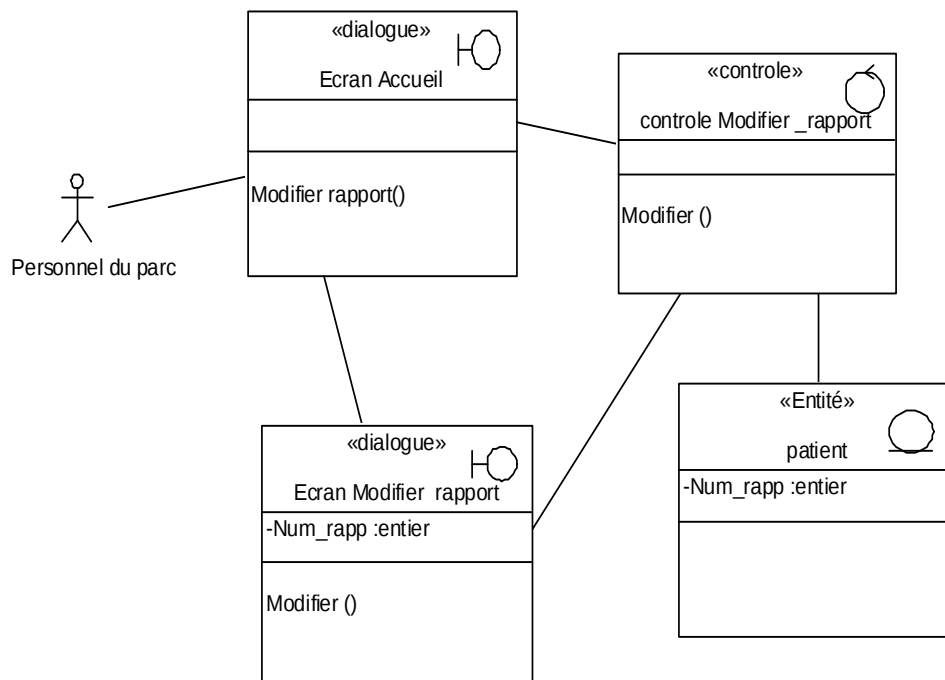


Figure 2.36 Diagramme de classe participant du cas d'utilisation Modifier rapport.

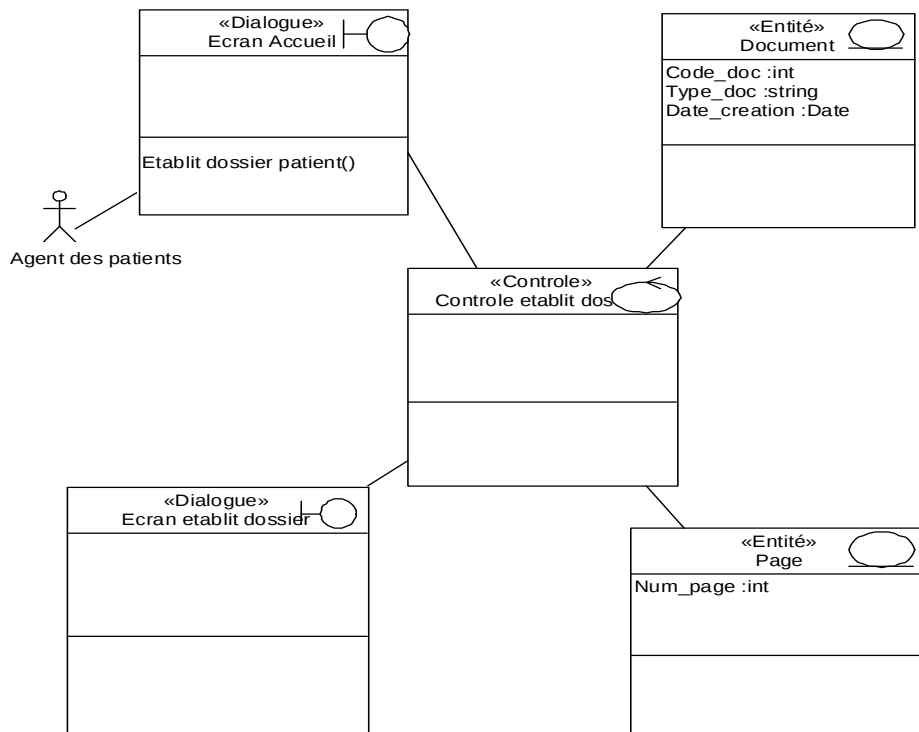


Figure 2.37 Diagramme de classe participant du cas d'utilisation Etablit le dossier.

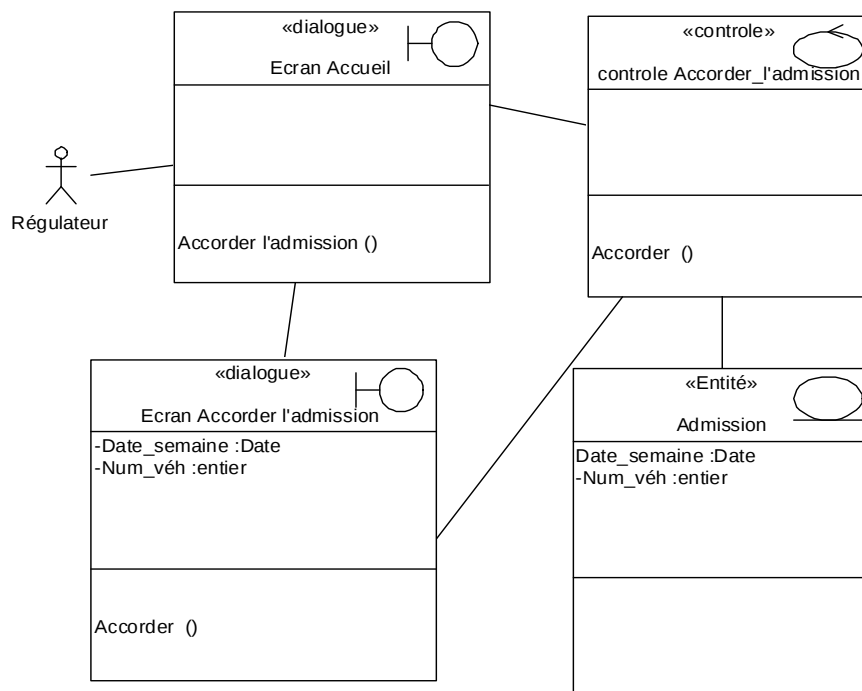


Figure 2.38 Diagramme de classe participant du cas d'utilisation Accorder l'admission.

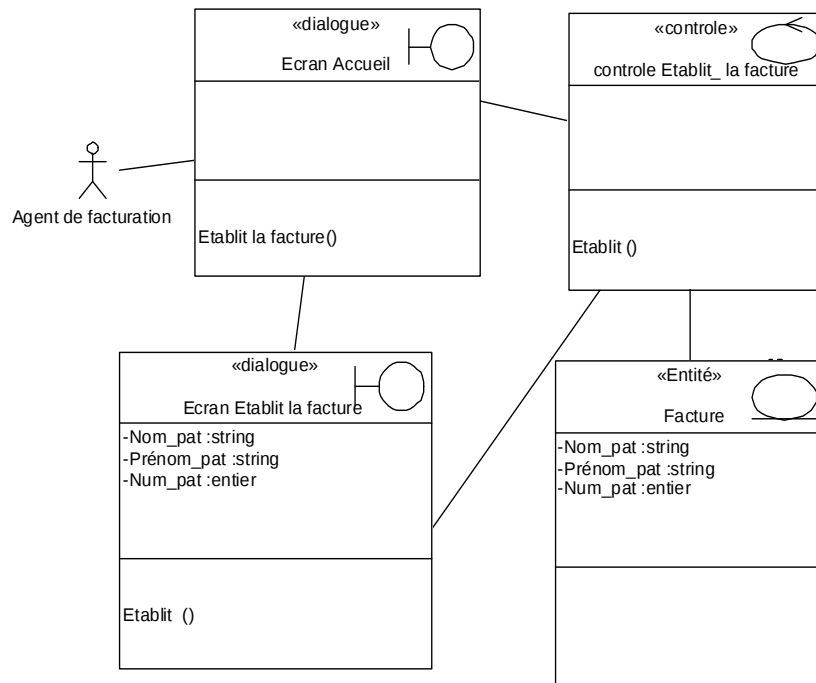


Figure 2.39 Diagramme de classe participant du cas d'utilisation Etablir la facture.

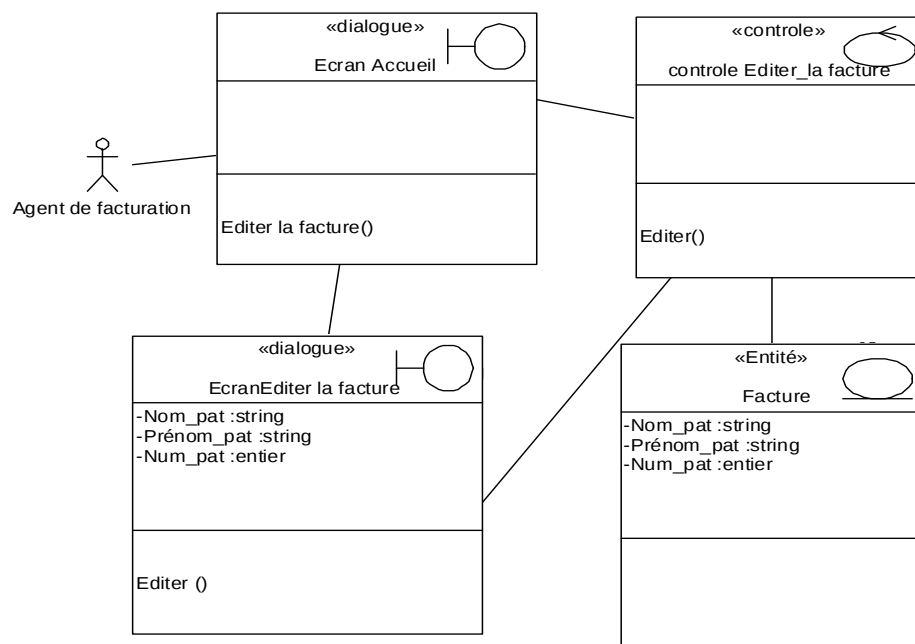


Figure 2.40 Diagramme de classe participant du cas d'utilisation Editer la facture.

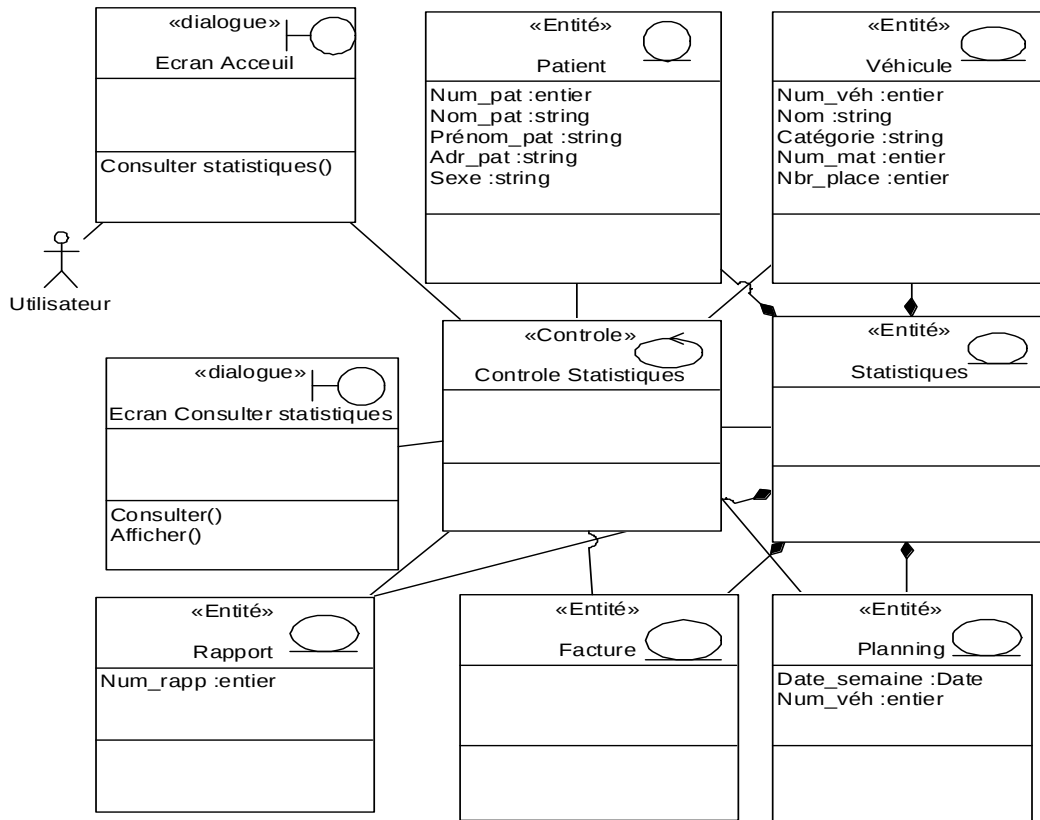


Figure2.41 Diagramme de classe participant du cas d'utilisation Consulter les statistiques.

2.3.2 Capture des besoins techniques.

2.3.2.1 Style d'architecture

Le style d'architecture en niveaux spécifie le nombre de niveaux géographiques et organisationnels où vont se situer les environnements d'exécution du système.

Puisque l'environnement d'exécution de notre système est départemental et local, on utilise l'architecture deux tiers (à deux niveaux) qui est la plus simple d'un système client/serveur.

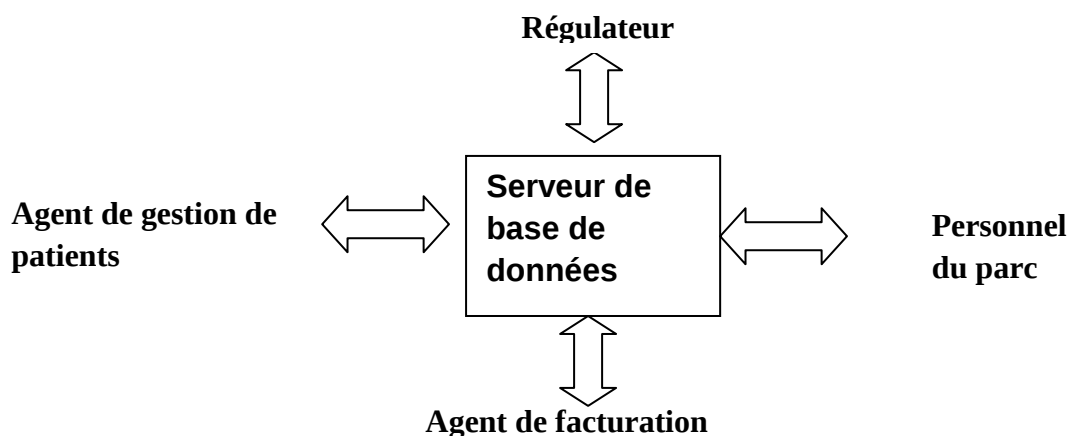


Figure 2.42 architecture client serveur du système

2.3.2.2 La configuration matérielle du système

Elle est schématisée par le diagramme de déploiement suivant :

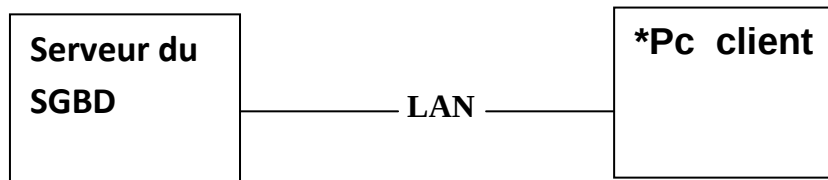


Figure 2.43 Configuration matérielle du système

2.3.2.3. Spécification de l'architecture en Tiers

Le choix de style d'architecture conditionne la façon d'organiser et de déployer les composants d'exploitation du système.

Dans l'architecture 2-tiers, les clients gèrent l'interface utilisateur et les processus d'exploitation. Les serveurs ont pour responsabilité de traiter le stockage de données.

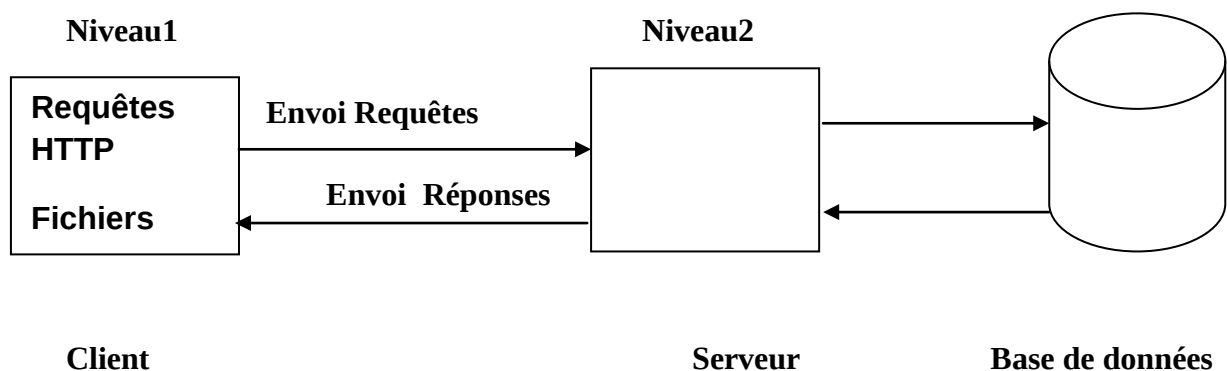


Figure 2.44 Architecture à deux niveaux (Client/serveur)

2.4 Conclusion

La capture des besoins fonctionnels et techniques est une étape qui travail d'un coté pour compléter les recueils des besoins opérés pendant la phase d'étude préliminaire, est d'un autre coté, elle donne une première vue pour la prochaine étape concernant l'analyse.

3.1. Introduction

Dans ce chapitre, nous passons à l'analyse de notre application, tout en présentant les Différentes étapes d'analyse suivies, selon la méthode adoptée, de l'identification de catégorie jusqu'à l'élaboration du modèle de classe optimisé du système.

3.2. Découpage en catégorie.

3.2.1. Définition d'une catégorie

Une catégorie consiste en regroupement logique de classes à forte cohérence interne et faible couplage externe.

On a découpé le problème en trois catégories :

- Identification du patient
- Transport des patients
- Parc

3.3. Répartition des classes candidates en catégorie

- **La catégorie « Identification du patient »**

- + Patient.
- + Admission.
- + Type patient.
- + Ville.
- + Assuré.

- **La catégorie « Transport des patients »**

- + Facture.
- + Moment.
- + Tranche.
- + Hôpital.
- + Centre de dialyse.

- **La catégorie «Parc »**

- + Véhicule.
- + Catégorie.
- + Chauffeur.
- + Permis.

3.4. Dépendances entre les catégories

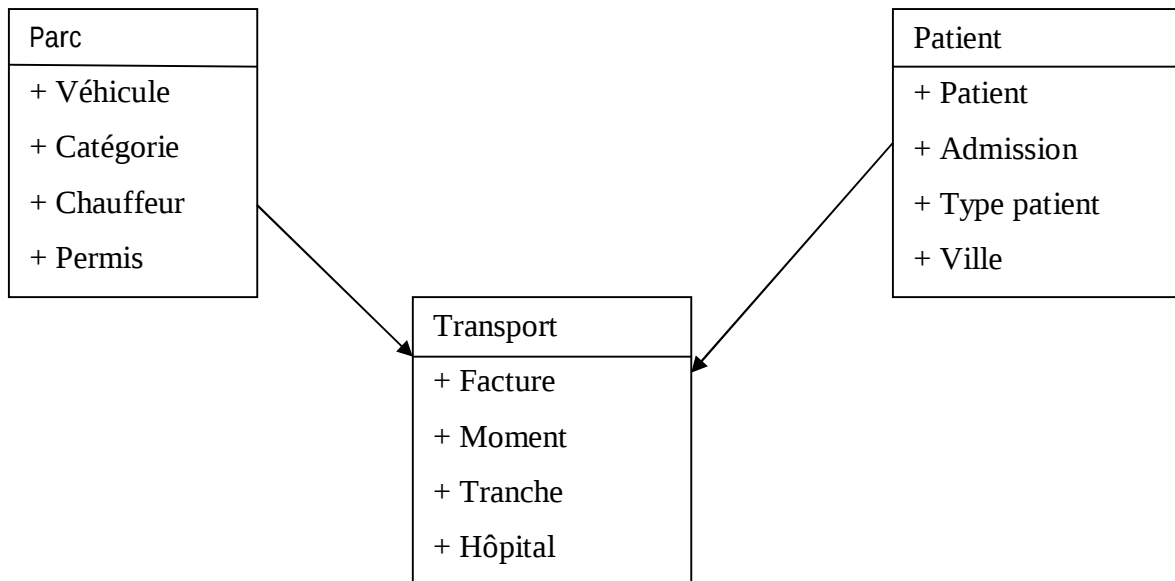


Figure. 3 .1. Modèle structurel d'analyse.

3.5. Les diagrammes de classe préliminaire :

3.5.1. Le diagramme de classe préliminaire de la catégorie « identification du patient » :

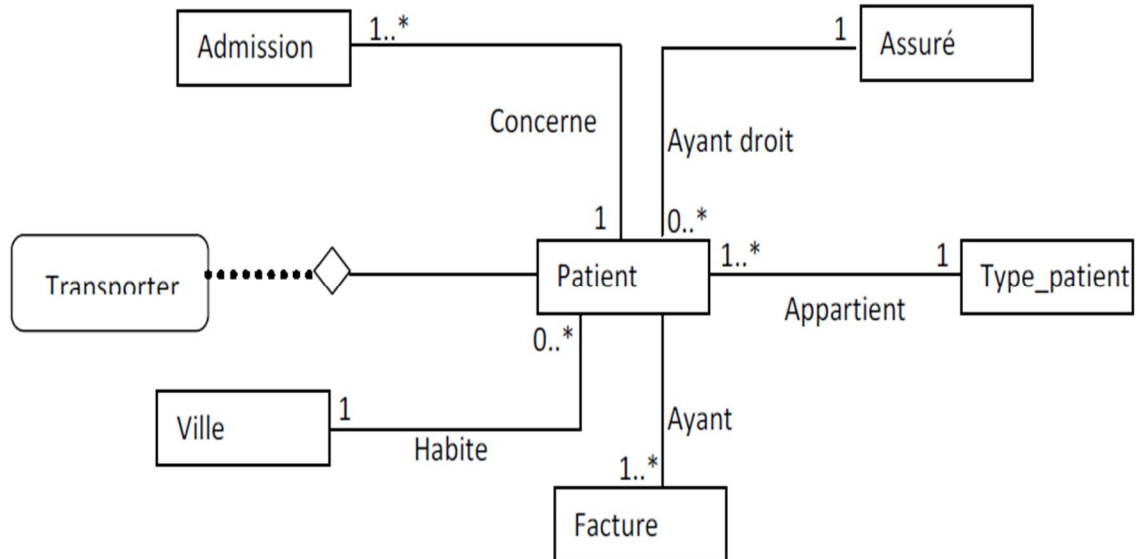


Figure .3.2 Diagramme préliminaire de la catégorie patient.

3.5.2. Le diagramme de classes préliminaire de la catégorie « Parc » :

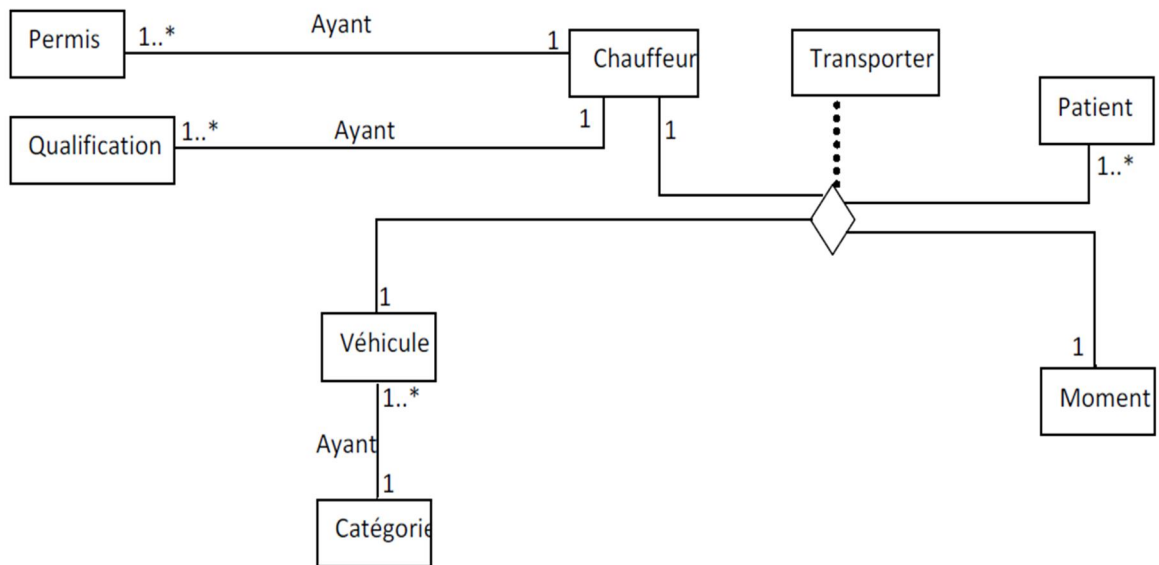


Figure .3.3 Diagramme préliminaire de catégorie parc.

3.5.3. Diagramme de classes préliminaire de la catégorie « Transport du patient » :

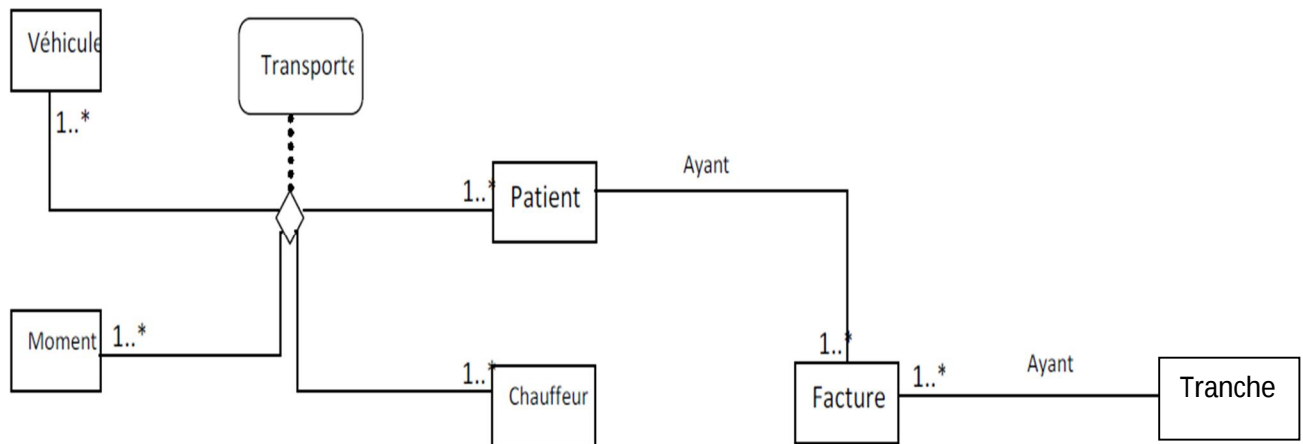


Figure .3.4. Diagramme préliminaire de catégorie transport.

3.6. Développement du modèle statique

Contient les opérations suivantes :

- Affiner les classes.
- Affiner les associations.
- Ajouter les attributs.
- Ajouter les opérations.

3.6.1. Les diagrammes détaillés pour chaque catégorie

3.6.1.1 Diagramme détaillés de catégorie patient

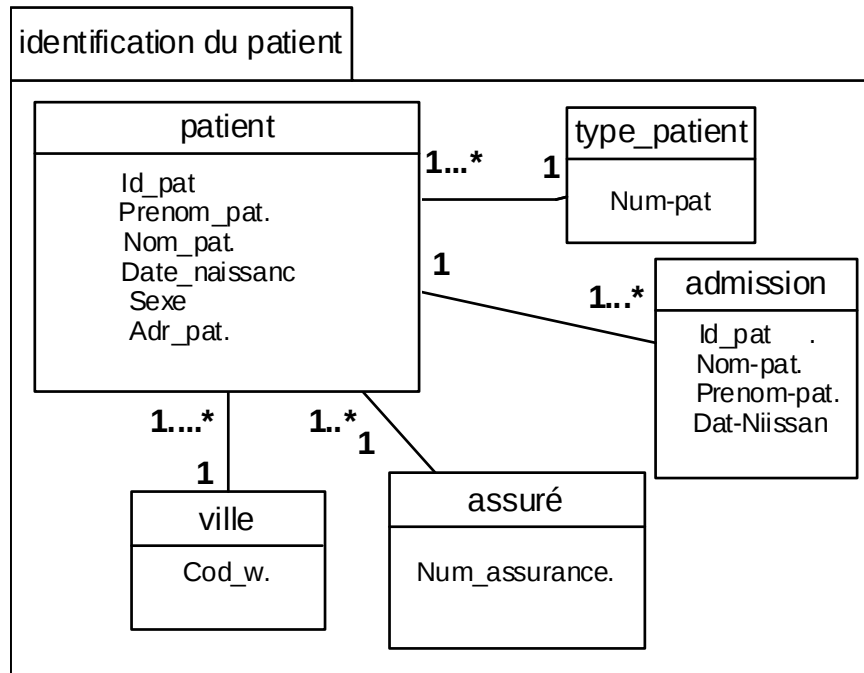


Figure .3.5. Diagramme de classes détaillé de catégorie patient.

3.6.1.2 Diagramme détaillés de catégorie transport

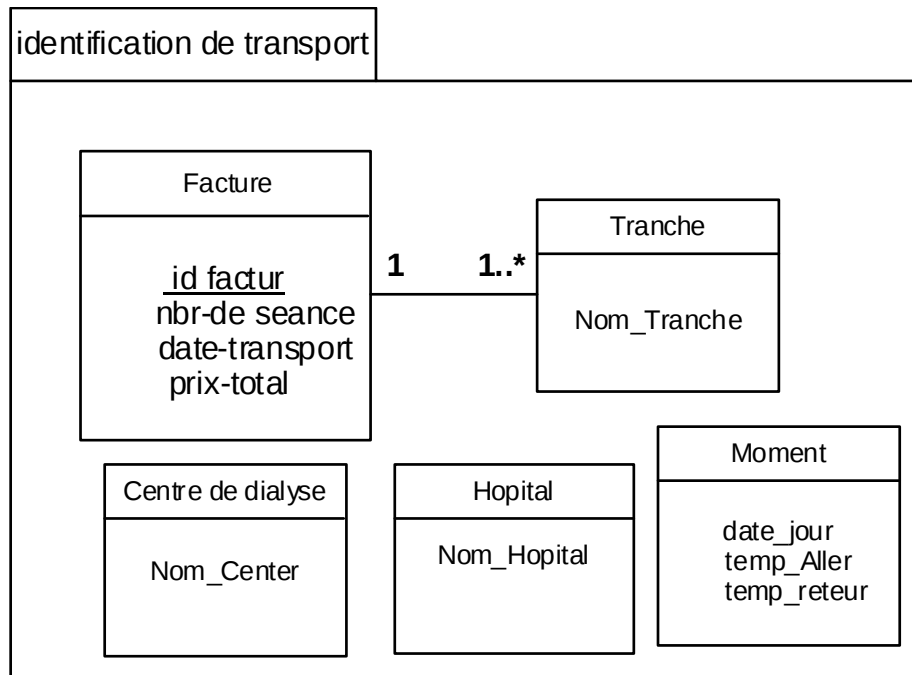


Figure .3.6. Diagramme de classes détaillé de catégorie transport.

3.6.1.3 Diagramme détaillés de catégorie parc

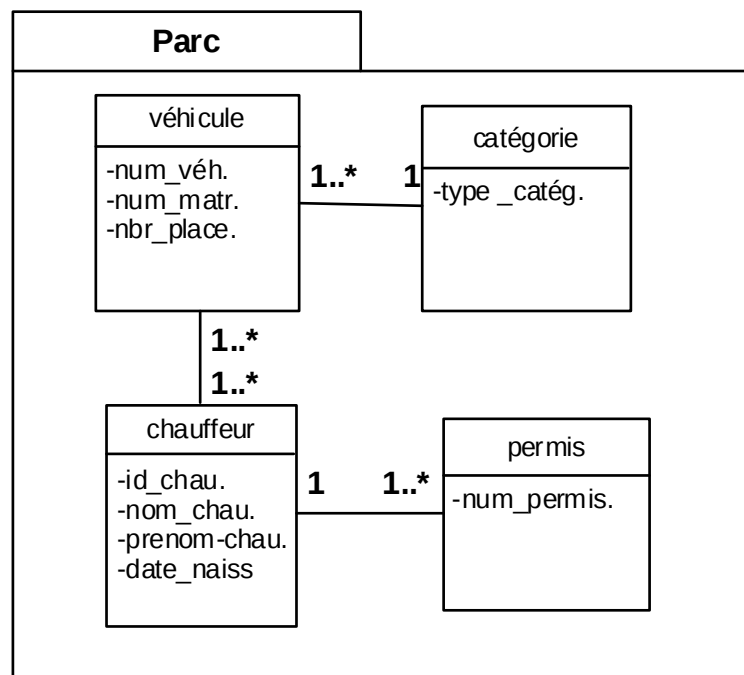


Figure .3.7. Diagramme de classes détaillé de catégorie parc.

3.6.2. Diagramme de classe optimisé du système

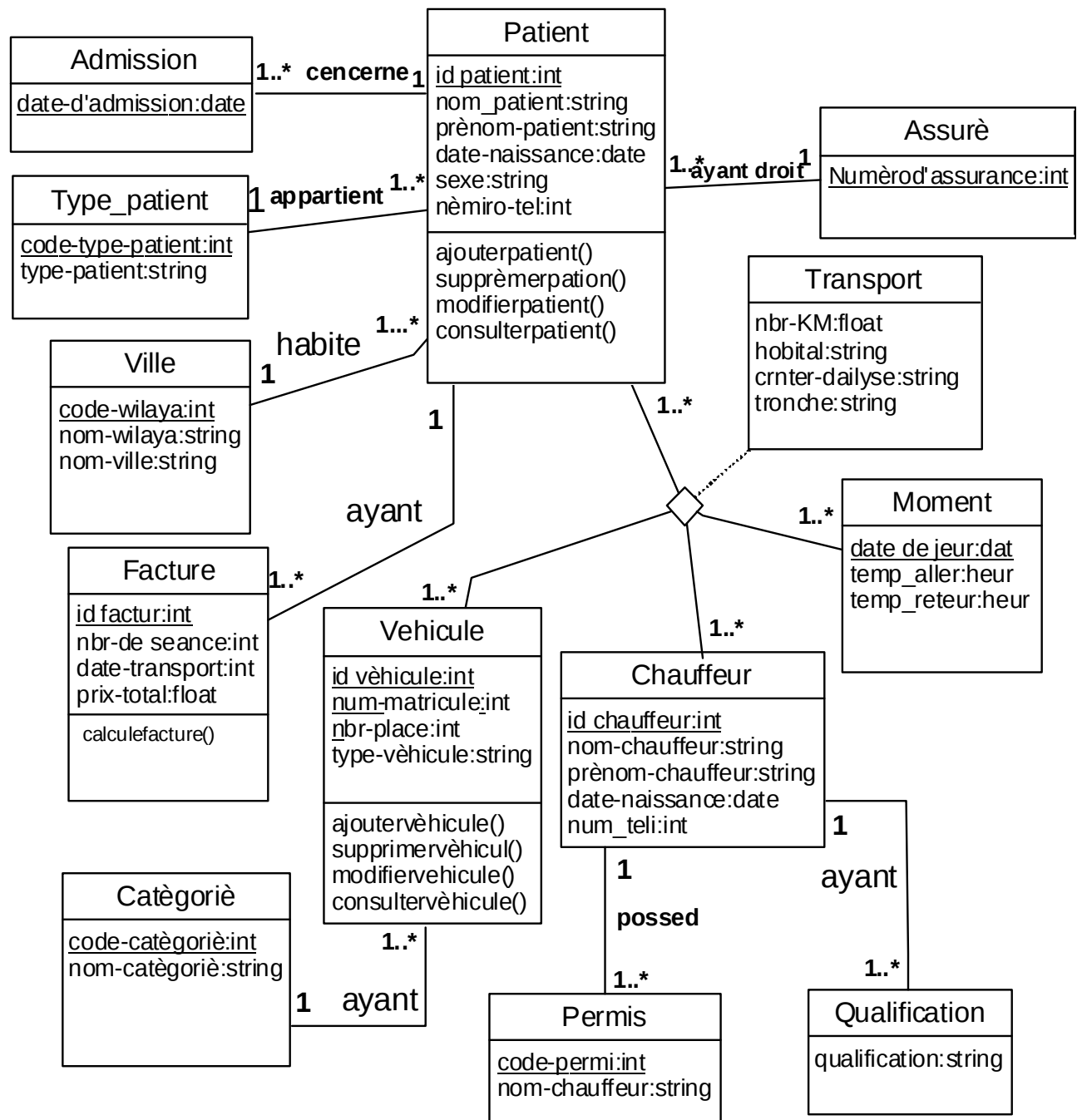


Figure .3.8. Diagramme de classes optimisé du système.

3.7. Conclusion

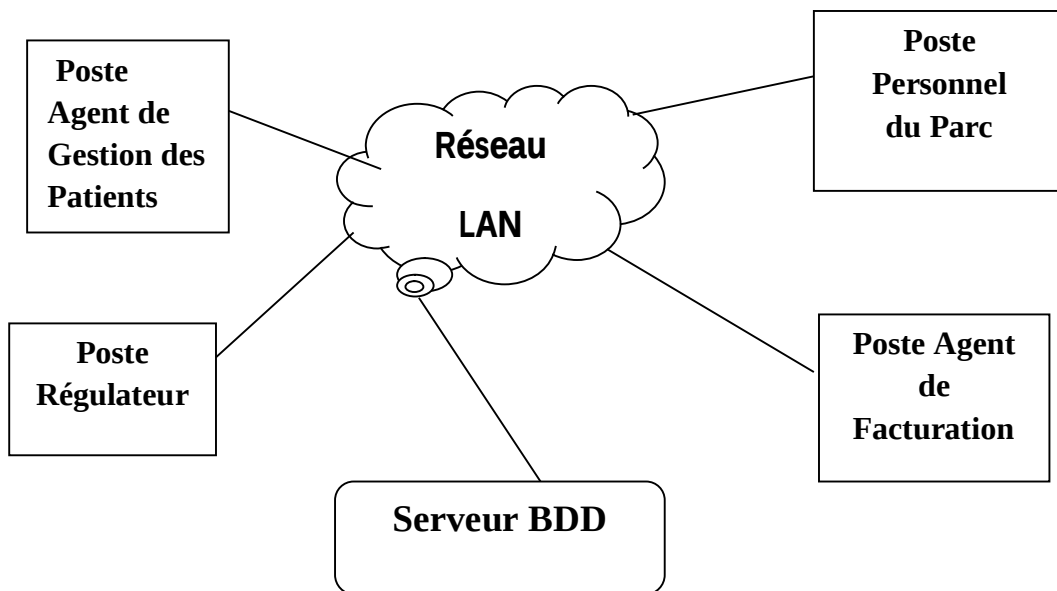
On a décrit dans ce chapitre la phase d'analyse de notre application, cette phase nous a permis d'affiner l'analyse détaillée des fonctionnalités du travail demandé et élaborer les diagrammes de classe préliminaire et les classes optimisées du système pour passer à l'étape de conception du prochain chapitre.

4.1 Introduction

Dans ce chapitre, nous passons à la phase de conception de notre application, tout en présentant les différentes étapes conceptuelles suivies, selon la méthode adoptée 2TUP, En premier lieu nous allons présenter la conception préliminaire de Project et la conception détaillée. Par la suite, nous présenterons un certain ensemble de règles de passage utilisé, nous détaillant ci-après, ensuite présenter la modèle relationnel.

4. 2 Conception préliminaire :

4. 2.1 Concevoir le modèle de déploiement :



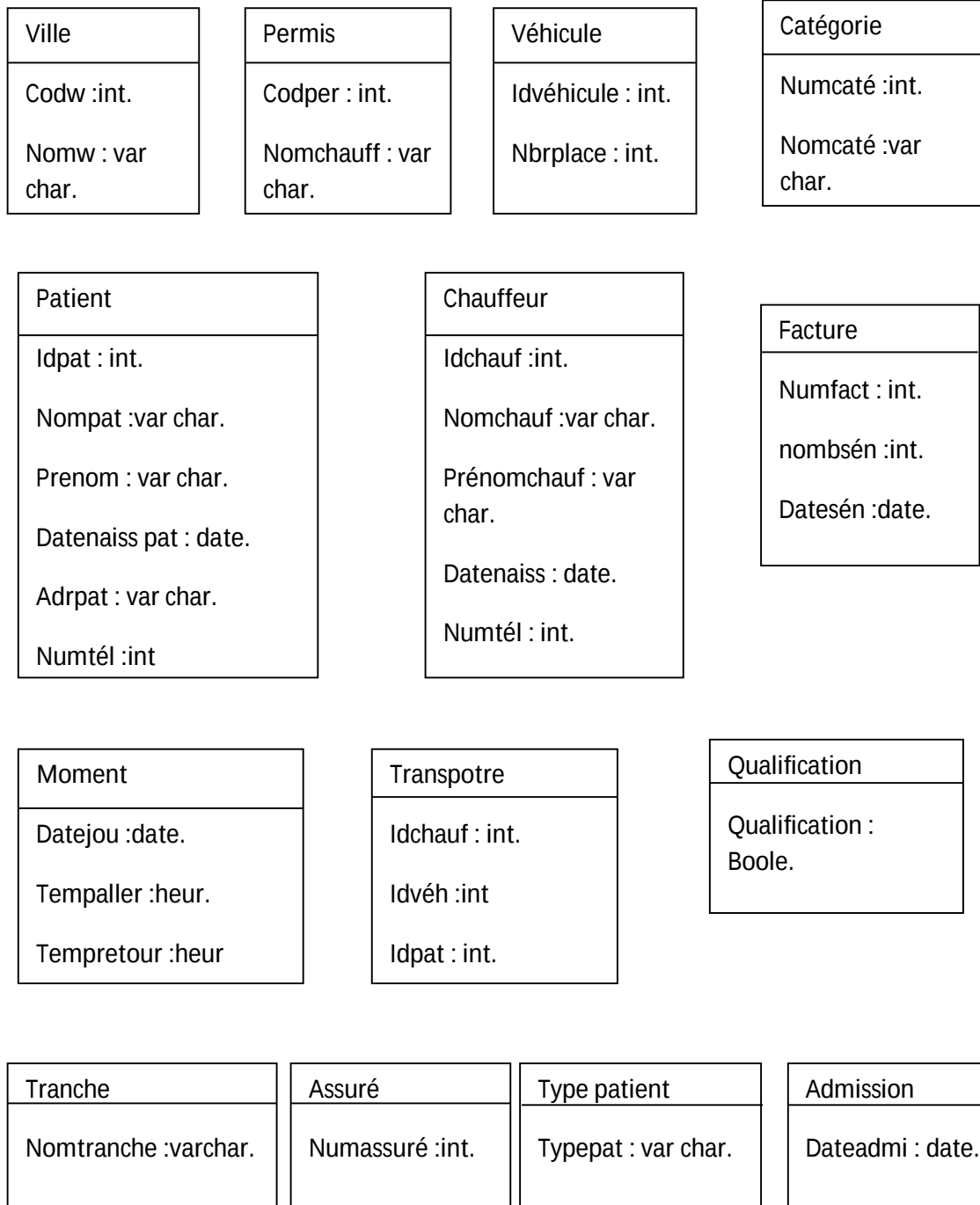
Fuguer.4.1 Concevoir le modèle de déploiement

4.3 Conception détaillée

Pour la conception détaillée de notre système, on procède comme suit :

- Typage des attributs.
- Le passage du modèle objet au modèle relationnel.
- Elaboration du modèle relationnel du système.

4.3.1 Typage des attributs :



4. 3.2 Le passage du modèle objet au modèle relationnel

4. 3.2.1 Règles de passage :

Les classes entités et leurs associations seront convertit en une base de données relationnelle, qui sera sollicitée par l'application pour consultation et mise à jour. Pour ce faire nous nous sommes basées sur les quatre règles :

1 Transformation des classes (R1):

chaque classe devient une relation. Les attributs de la classe deviennent des attributs de la relation. Si la classe possède un identifiant, il devient la clé primaire de la relation, sinon, il faut ajouter une clé primaire arbitraire.

2 Transformation des associations :**- Association 1..* :**

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

- Association *...* :

association/classe – association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des identités connecté à l'association. Chaque attribut devient clé étrangère si entité/classe connectée dont il devient une relation en vertu de la règle R1. Les attributs d'association/classe – association doivent être ajoutés a la nouvelle relation. ces attributs ne sont ni clé primaire, ni clé étrangère.

- Association 1..1 :

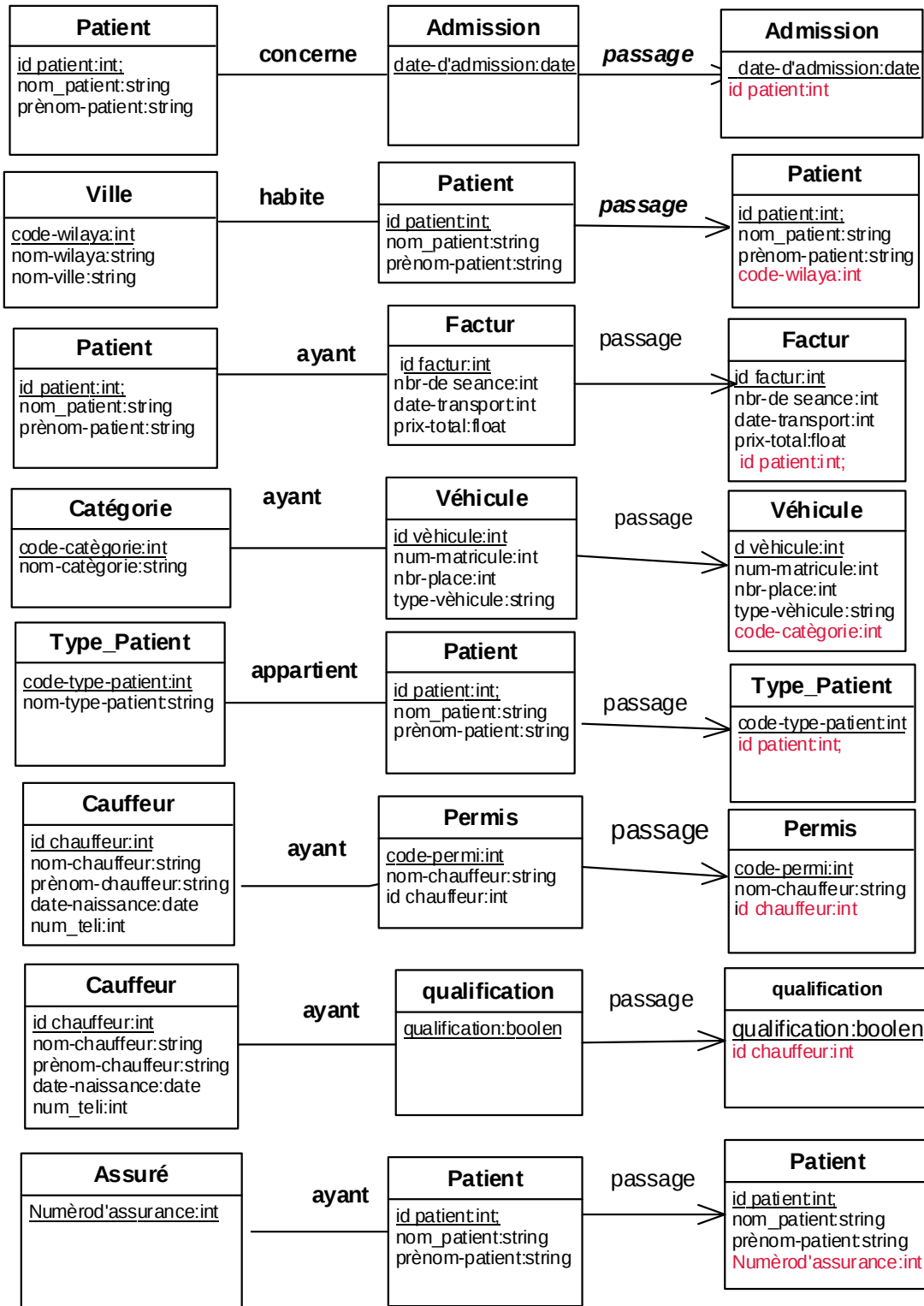
Il faut ajouter un attribut de type clé étrangère dans la relation dérivée de l'entité ayant la cardinalité minimale égale à zéro. Dans le cas de diagramme UML il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un.

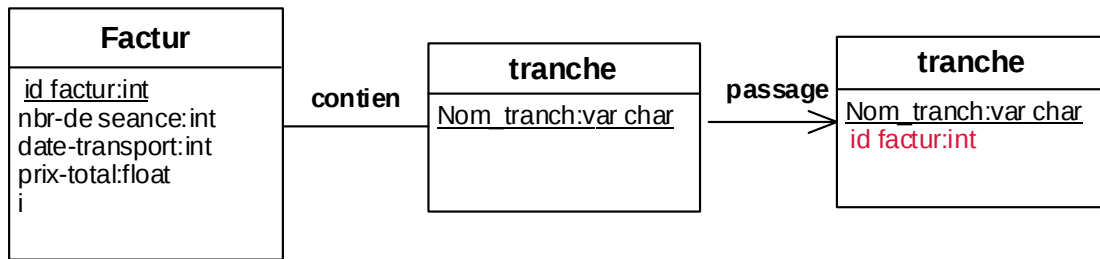
L'attribut porte le nom de la clé primaire de la relation dérivée d'entité classe connectée à l'association. Si les deux cardinalités minimales égales à zéro, le choix est donné entre les deux relations dérivées de la R1.

Si les deux cardinalités minimales égales à un, il est préférable de fusionner les deux entités/classe en une seule.

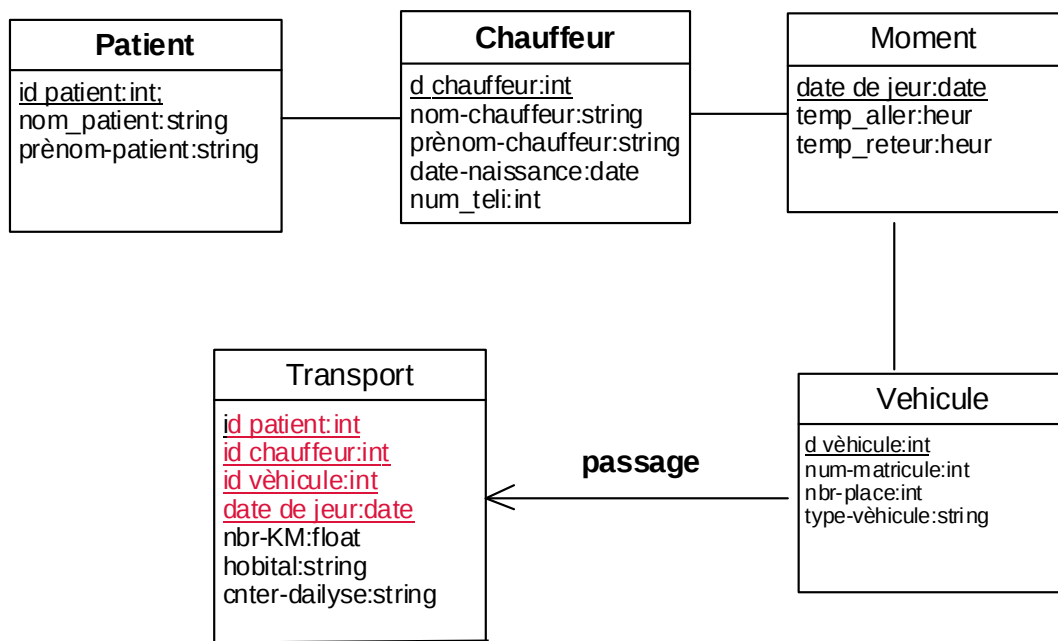
4.3.2.2 Traduction des associations :

„Appliquer la règle de l'association 1.....*:





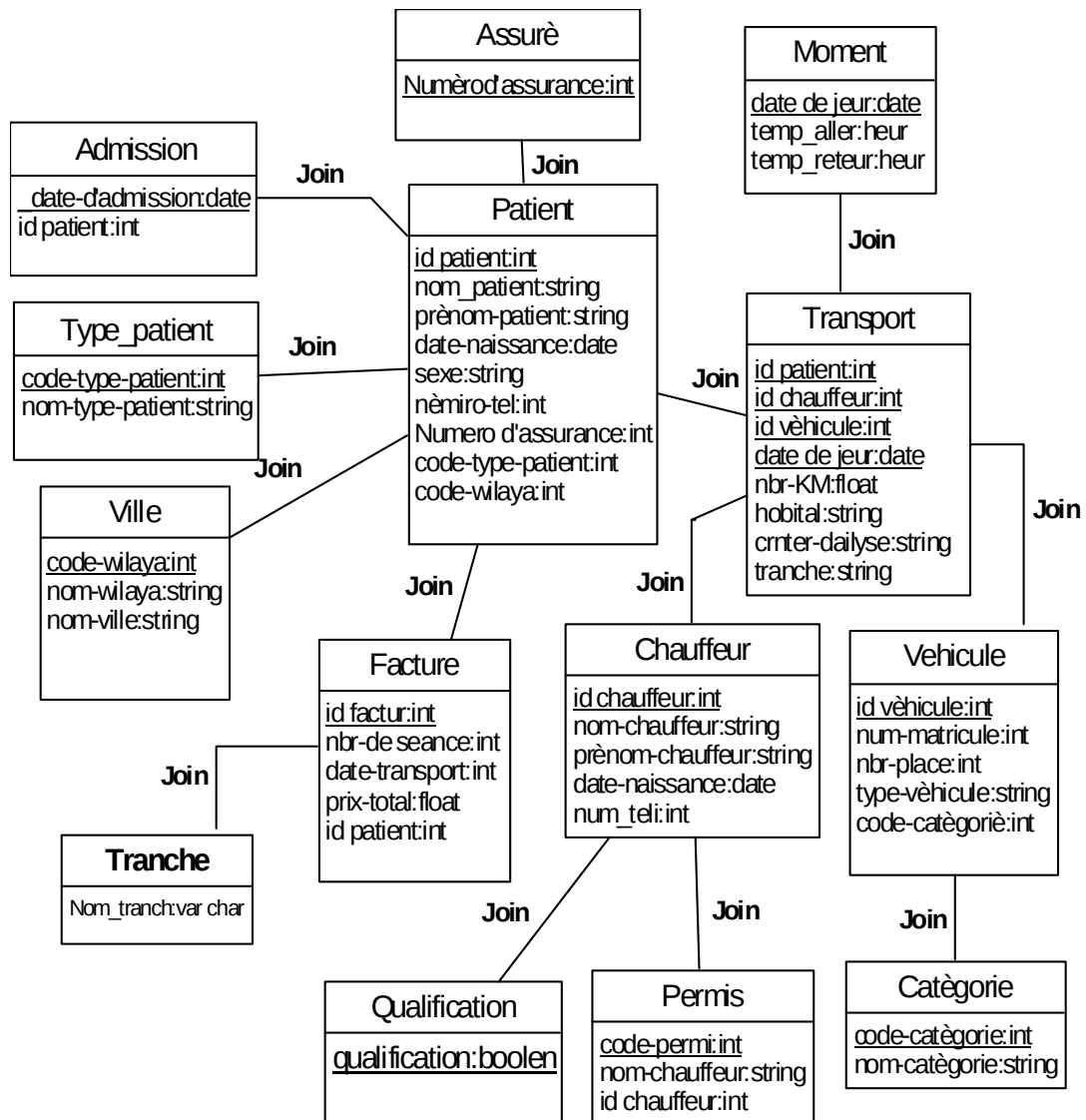
Appliquer la regle d'association *.....* :



4. 3.2.3 Les classes modifiées :

Patient, permis, qualification, véhicule, transporter, facture.

4.3.3 Le modèle relationnel du système :



Fuguer .4.2 Le modèle relationnel du système

4.5 Conclusion :

On a décrit dans ce chapitre la phase de conception de notre application, cette phase nous a permis de spécifier la configuration matérielle de notre application et d'élaborer un modèle relationnel représentant les différentes tables de la base de données du système. Donc nous pouvons déclencher la phase de réalisation de notre système.

5.1 Introduction :

Après avoir terminé l'étude conceptuelle, nous pouvons alors passer à l'étape finale du processus à savoir l'implémentation. Elle a comme objectif d'aboutir à un produit final exploitable par les utilisateurs. Dans cette phase nous allons préparer l'environnement de programmation que nous avons utilisés pour le développement de notre application, puis nous décrivons quelques interfaces de l'application que nous avons extraites via des prises d'écrans.

5. 2 Environnement de développement de l'application

5. 2.1. Pourquoi Delphi ?

Delphi est un outil de développement visuel et rapide sous Windows (Rapid Application Development) qui permet de créer des applications fenêtrées, directement exécutables (.EXE) et redistribuables librement sous Windows ou DOS.

Delphi utilise le langage Pascal Orienté Objet (il est toutefois possible d'utiliser d'anciennes sources en Pascal standard grâce au compilateur en ligne de commande). Ce langage est facile à apprendre et beaucoup plus simple que le C++ traditionnel. Les objets utilisés ont des propriétés et des méthodes. Les propriétés sont les caractéristiques de l'objet (couleur, taille, ...) tandis que les méthodes sont les procédures (classiques ou événementielles) et fonctions qui y sont rattachées.

5. 3 Interfaces graphique.

Dans cette section nous allons présenter quelques interfaces de notre application :

5.3.1 Interface d'authentification : Permet d'accéder au système via le nom d'utilisateur et le mot de passe.

Authentifier

Gestion de Transport Sanitaire

Utilisateur:

Mot de passe:

Nouveau utilisateur

Changer le mot de passe

X Fermer

OK

Annuler

Figure.5.1.Interface <<authentification>>

5.3.2 Interface de menu principale <<assurer planning>>.

Form2

Assurer planning | Consulter les statistiques | Créer une séance | Gérer les chauffeurs

mai 2014

dim.	mar.	mer.	jeu.	ven.	sam.	dim.
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
2	3	4	5	6	7	8

Aujourd'hui : 07/11/2013

X Fermer

medic-assistance.

Figure.5.2.Interface de menu principale <<assurer planning>>.

5.3.3 Interface de menu principale <<gérer les véhicules>>.



Figure.5.3. Interface de menu principale <<gérer les véhicules>>.

5.3.4 Interface de menu principale <<gérer des patients>>.



Figure.5.4. Interface de menu principale <<gérer des patients>>.

5.3.5 Interface de menu principale <<Etablir les factures>>.



Figure.5.5. Interface de menu principale <<Etablir les factures>>.

5.3.6 Interface <<Ajouter patient>>.

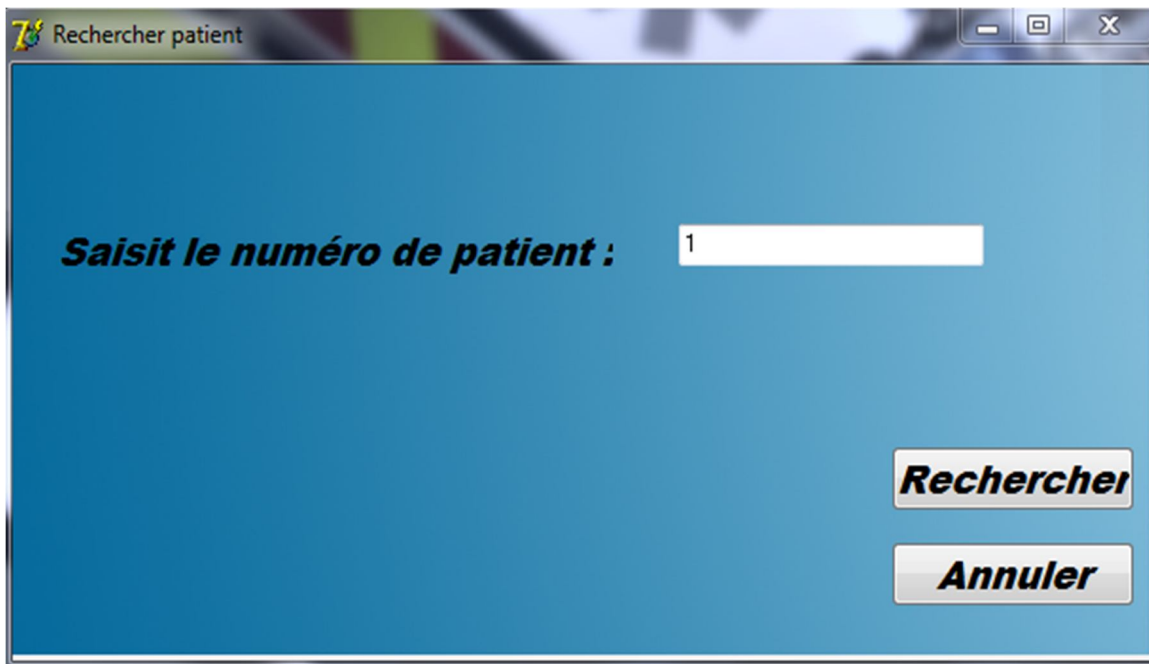
 The screenshot shows a software window titled 'AjouterP'. The main heading is 'saisie les données :'. Below this, there are several labeled input fields:

- Numéro de patient**: A text box containing the number '1'.
- Nom de patient**: A text box containing the name 'IKRAM'.
- Prenom de patient**: A text box containing the name 'BOUL'.
- Date de naissance de patient**: A text box containing the date '12/02/14'.
- Sexe**: A dropdown menu with 'F' selected.
- Numéro de téléphone**: A text box containing the number '05555652'.
- Numéro d'assurance**: A text box containing the number '12345'.
- Code de type de patient**: A text box containing the number '23'.
- Numéro de wilaya**: A text box containing the number '43'.

 On the right side of the form, there are two buttons: 'Ajouter' and 'Exit'.

Figure.5.6 Interface <<Ajouter patient>>.

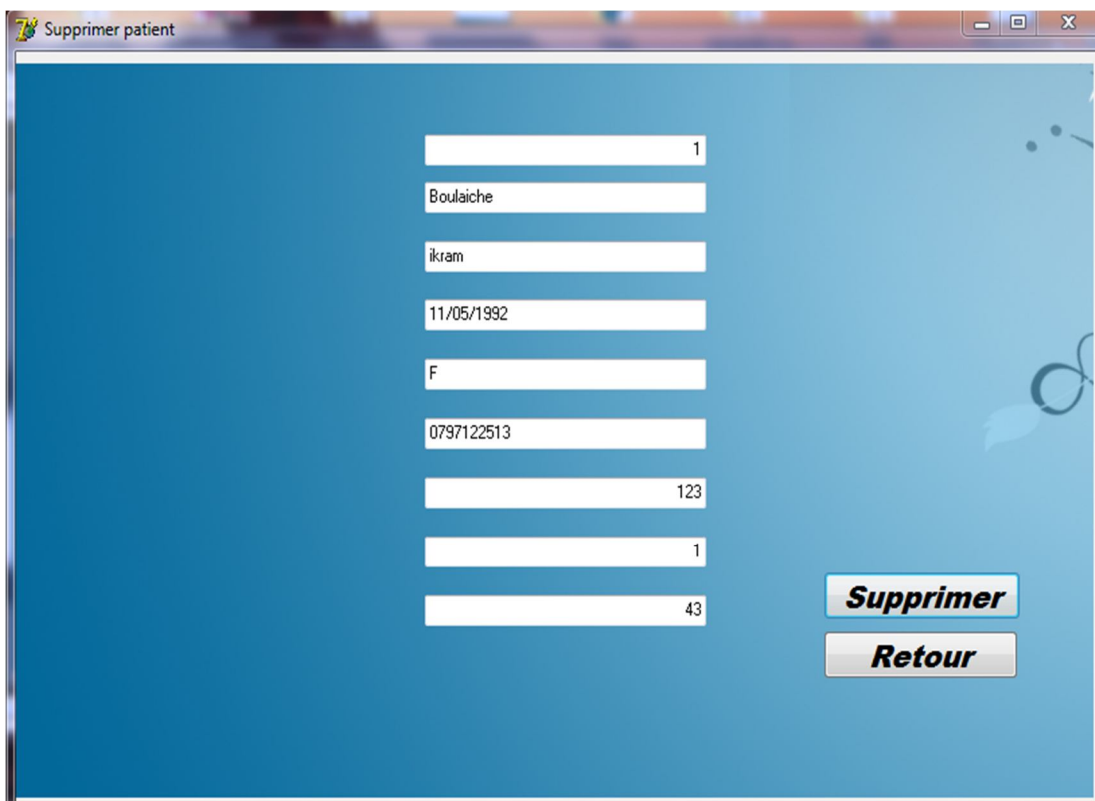
5.3.7 Interface <<Rechercher pour supprimer un patient >>.



The screenshot shows a window titled "Rechercher patient". The main area has a blue background with the text "Saisit le numéro de patient :" in bold. To the right of this text is a text input field containing the number "1". Below the input field, there are two buttons: "Rechercher" and "Annuler", both with a 3D effect.

Figure.5.7 Interface <<Rechercher pour supprimer un patient >>.

5.3.8 Interface << supprimer un patient >>.



The screenshot shows a window titled "Supprimer patient". The main area has a blue background. On the left side, there is a vertical stack of text input fields containing the following information: "1", "Boulaiche", "ikram", "11/05/1992", "F", "0797122513", "123", "1", and "43". To the right of these fields, there are two buttons: "Supprimer" and "Retour", both with a 3D effect.

Figure.5.8 Interface << supprimer un patient >>.

5.3.9 Interface << Modifier un patient >>.

Num_pat	Nom_pat	Prenom_pat	Datenaiss_pat	Sexe
1	Boulaiche	ikram	11/05/1992	F
2	zemouri	amina	13/08/1992	F
3	amimmour	souad	16/02/1992	F
5	bouchloukh	ibtissem	21/07/1992	F
6	bouzeraa	selma	15/09/1992	F

Modifier

Annuler

Figure.5.9 Interface << Modifier un patient >>.

5.3.10 Interface << Consulter un patient >>.

Numéro patient	Nom patient	Prénom patient	Date naissance	Sexe
1	Boulaiche	ikram	11/05/1992	F
2	zemouri	amina	13/08/1992	F
3	amimmour	souad	16/02/1992	F
5	bouchloukh	ibtissem	21/07/1992	F
6	bouzeraa	selma	15/09/1992	F

Retour

Figure.5.10 Interface << Consulter un patient >>.

5.4 Conclusion

Dans cette dernière partie de notre travail, nous avons décrit brièvement la phase d'implémentation tout en donnant une vue globale sur les outils utilisés ainsi que l'application que nous avons réalisée.

Nous avons présenté l'interface de l'application avec toutes les options qu'elle contient afin de faciliter son utilisation.

CONCLUSION GÉNÉRALE

Ce projet qui s'inscrit dans le cadre de la gestion de transport sanitaire, était à l'objectif de répondre aux deux principaux processus de la gestion de transport, à savoir :

Faciliter l'organisation des transports de patients de son domicile vers le centre de dialyse et vice-versa, ainsi que l'informatisation de processus de gestion de dossiers des patients et de conducteurs de véhicules. Pour ce faire, nous sommes passés par plusieurs phases, allant de la collecte d'informations acquises de l'étude de l'existant, passant par la suite par la phase d'analyse et de conception.

Nous avons utilisé le langage de modélisation UML et la méthode 2TUP pour le développement de notre application, et l'environnement de programmation DELPHI pour l'implémentation.

En fait, à la fin de la réalisation de ce mémoire, il est important de dire que ce projet nous a permis d'exploiter nos connaissances théoriques acquises pendant le cycle de notre formation. En outre, ce travail nous a donné un avant-goût du métier de développeur et il nous a permis de concevoir pour la première fois une vraie application pour un vrai client du fait que nous nous sommes mieux rapprochés et familiarisés avec un environnement dynamique.

Par conséquent, nous avons pu avoir une idée plus approfondie sur l'importance des systèmes d'informations dans les entreprises.

Enfin, nous pouvons dire que ce travail constitue une grande satisfaction personnelle et professionnelle qui signe le début d'aboutissement de notre formation.

Les références

Ouvrage :

1. Franck Vallée. Roques, pascal.UML en action: de l'analyse des besoins à la conception (4eme Edition).
2. Benhammada sadek. Cours UML.