

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaïa



Faculté des Sciences Exactes
Département d'Informatique

En vue de l'obtention du diplôme de MASTER en Informatique
Option : Administration & Sécurité des Réseaux
Mémoire de Fin d'Etudes

Thème

CONCEPTION ET RÉALISATION D'UNE APPLICATION DE
SUIVIE DE PRESTATION D'ASSISTANCE MALADIE
CAS *SOUMMAM ASSISTANCE*

Réalisé par :

M^r ADALOU Mohand Hamimi.
M^{lle} TAZIBET Fadila.

Soutenu devant le jury composé de :

Président :	<i>M^r</i> Abdelkamel TARI	MCA	UAMB
Encadreur :	<i>M^r</i> Ali LARBI	MAA	UAMB
Examineur :	<i>M^r</i> Foudil MIR	MAA	UAMB
Examineur :	<i>M^r</i> Saadi M.	UAMB	

Junin 2012

TABLE DES MATIÈRES

Table des matières	i
Liste des Figures	iv
Liste des Tableaux	v
Liste Des Acronymes	vi
Introduction Générale	1
1 Généralités	3
Introduction	4
1.1 Généralités sur Les Systèmes d'Information (S.I)	4
1.1.1 Définition	4
1.1.2 Représentation des Systèmes d'Information	4
1.1.3 Historique et Evolution des S.I	5
1.2 Le Language de Modélisation Unifié (UML)	7
1.2.1 Historique de l'UML	7
1.2.2 La notation UML	7
1.2.3 Les Diagrammes UML	7
1.2.4 Les Processus de développement	9
1.2.4.1 Définition du Processus Unifié	9
1.2.4.2 Les activités du processus unifié	9
1.2.4.3 La Méthode 2TUP	10
1.3 Les Bases de Données (BDD)	12
1.3.1 Définition d'une Base de Données	12
1.3.2 Système de Gestion de Bases de Données (SGBD)	12
1.3.2.1 Exemples	13
1.4 Les Réseaux Informatiques	13
1.4.1 Les Concepts Réseau	13
1.4.1.1 Réseau Informatique	13
1.4.1.2 Réseau Intranet	13
1.4.1.3 Réseau Extranet	13
1.4.1.4 Types de Réseaux	13
1.4.2 Présentation de l'Architecture Client/Serveur	14
1.4.3 Fonctionnement d'un système Client/Serveur	14
1.4.4 Les différentes architectures Client/Serveur	14
1.4.4.1 Architecture à 1-tiers	14
1.4.4.2 Architecture à 2-tiers	15

1.4.4.3	Architecture à 3-tiers	17
	Conclusion	19
2	Analyse et Spécification des Besoins	20
	Introduction	21
2.1	Spécification des besoins	21
2.1.1	Présentation de l'organisme d'accueil	21
2.1.2	Historique	21
2.1.3	Organigramme de l'entreprise	22
2.1.4	Effectif de l'entreprise	23
2.1.5	Objectif de l'étude	23
2.1.6	Présentation du Domaine d'Étude	24
2.1.7	Situation Informatique	24
2.1.8	Problématique	24
2.1.9	Solutions	24
2.2	Analyse des besoins	25
2.2.1	Besoins fonctionnels et non fonctionnels	25
2.2.2	Identification des acteurs	26
2.2.3	Identification des messages	26
2.2.4	Modélisation du Système	27
2.2.4.1	Le Diagramme de Cas d'Utilisation	27
2.2.4.2	Paquetage du Cas d'Utilisation Gérer les Factures	28
2.2.4.3	Les Diagrammes d'Activité	29
	Conclusion	31
3	Conception	32
3.1	Introduction	33
3.2	Réalisation du diagramme de classes	33
3.2.1	Les règles de Gestion	33
3.2.2	Dictionnaire de Données	33
3.2.3	Le Diagramme de Classes	35
3.3	Règles de Passage du Diagramme de Classes vers le Modèle Relationnel	36
3.4	Règles de Normalisation	36
3.5	Le Modèle Relationnel	37
	Conclusion	38
4	Réalisation	39
4.1	Environnement et outils de développement	40
4.2	Outils de développement	40
4.2.1	Windev 15	40
4.2.2	WLangage	40
4.2.3	HyperFileSQL	41
4.2.4	Structured Query Language	41
4.2.5	Adobe Photoshop CS5	41
4.2.6	Argo UML	41
4.3	Ergonomie	41
4.3.1	Choix des Couleurs	41
4.3.2	Logo	42
4.3.3	Le Corps	42

4.4	Implémentation de la base de données	43
4.5	Architecture de développement	44
4.5.1	L'Interface	44
4.5.2	Les Procédures	45
4.5.3	Les Requêtes	46
4.6	Arborescence de l'application	47
4.7	Interfaces de l'application	48
4.7.1	Fenêtre d'Authentification	48
4.7.2	Page d'Accueil	49
4.7.3	Page Gestion du Parc Véhicule	50
4.7.4	Page Planification	51
4.7.5	Page Facturation	52
4.8	Test	53
4.8.1	Test du cas d'utilisation "Authentification"	53
4.8.2	Test du cas d'utilisation "Ajout d'un Assuré"	54
4.8.3	Test du cas d'utilisation "Modifier un Assuré"	54
4.8.4	Test du cas d'utilisation "Supprimer un Assuré"	55
4.9	Perspective	55
	Conclusion Générale	57
	BIBLIOGRAPHIE	59

Table des figures

1.1	Représentation d'un Système d'Information	5
1.2	Schéma de l'évolution des Systèmes d'Information	6
1.3	Le Système d'Information soumis à deux types de contraintes	10
1.4	le Cycle de vie en Y	11
1.5	Fonctionnement d'un Système Client/Serveur	14
1.6	Représentation de l'Architecture à 2 niveaux	15
1.7	Processus d'Accès aux Données en mode deux tiers	16
1.8	Représentation de l'architecture à 3 niveaux	17
2.1	Organigramme de Soummam Assistance	22
2.2	Diagramme de Cas d'Utilisation	27
2.3	Paquetage du cas d'utilisation "Gerer des Factures"	28
2.4	Diagramme d'activité du cas d'utilisation "Gérer les Factures"	29
2.5	Diagramme d'activité du cas d'utilisation "Gérer les séances d'Entretiens des Véhicules"	30
3.1	Le Diagramme de Classes	35
4.1	Logo de l'entreprise "Soummam Assistance"	42
4.2	Exemple de fenêtre de l'application	42
4.3	Base de données de l'application "Soummam Assistance Centre de Gestion"	43
4.4	Table Entretien	43
4.5	Architecture de développement de l'application "Soummam Assistance Centre de Gestion"	44
4.6	Procedure "Entrtien"	45
4.7	Requête "Entretien"	46
4.8	Arborescence de l'application	47
4.9	Fenêtre d'Authentification	48
4.10	Fenêtre d'Accueil	49
4.11	Fenêtre "Gestion du parc véhicule"	50
4.12	Fenêtre "Planification des RDV"	51
4.13	Fenêtre "Gestion des Factures"	52
4.14	Test de l'Authentification	53
4.15	Test de l'Ajout d'un Assuré	54
4.16	Test de la modification d'un Assuré	54
4.17	Test de la suppression d'un Assuré	55

Liste des tableaux

2.1	Effectif de Soummam Assistance	23
2.2	Comparaison des deux S.I manuel et automatisé	23
3.1	Dictionnaire de Données	34

Liste Des Acronymes

Acronyme	Signification
1NF	1 Normal F orm.
2NF	2 Normal F orm.
2TUP	2 Tracks U nified P rocess.
3NF	3 Normal F orm.
AGL	A telier de G énie L ogiciel.
API	A pplication P rogramming I nterface.
BDD	B ase D e D onnées.
CLI	C ommand L ine I nterface.
DCE	D ata T erminal E quipment.
DPFTS	D emande de P rise en charge de F rais de T ransport S anitaire.
DRFTS	D emande de R emboursement de F rais de T ransport S anitaire.
GPL	G eneral P ublic L icense.
HFSQL	H yper F ile S QL.
HP	H ewlett P ackage.
HTTP	H yper T ext T ransfer P rotocol.
IBM	I nternational B usiness M achines.
LAN	L ocal A rea N etwork.
MAC	M edia A ccess C ontrol.
MAN	M etropolitan A rea N etwork.
MLD	M odèle L ogique des D onnées.
MySQL	M y S tructured Q uery L anguage.
ODBC	O pen D ata B ase C onnectivity.
OMG	O bject M anagement G roup.
OMT	O bject M odeling T echnique.
OOSE	O bject O riented S oftware E ngineering.
PAN	P ersonal A rea N etwork.
RDV	R endez-vous.
SGBD	S ystème de G estion de B ases de D onnées.
SI	S ystème d' I nformation.
SQL	S tructured Q uery L anguage.
TCP/IP	T ransport C ontrol P rotocol/ I nternet P rotocol.
UML	U nified M odeling L anguage.
UP	U nified P rocess.
WAN	W ide A rea N etwork.

Introduction Générale

*"il n'y a pas de problèmes qu'on se pose,
il y a des problèmes qui se posent, il n'y a pas de problèmes résolus,
il y a des problèmes plus ou moins résolus"*
*** Henri Poincaré**

De nos jours, une des tendances les plus en vue et qui concerne tout les secteurs de développement est l'informatisation.

Depuis l'apparition de l'informatique et son introduction dans le monde économique, les entreprises et entités publiques aspirent à optimiser et à rendre fiable la gestion de leurs structures internes.

Avant l'invention de l'ordinateur, on enregistrerait toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps dans la recherche de ces informations ou la dégradation de ces dernières.

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention à permis d'informatiser les systèmes de données des entreprises, qui est la partie essentielle dans leur développement d'aujourd'hui.

La gestion automatisée peut être un élément capital de rentabilisation d'une entreprise surtout des moyennes et petites entreprises à faibles ressources humaines comme dans notre cas. Ce traitement automatisé pourra permettre essentiellement l'allègement du travail de l'homme et fournir les résultats précis, opportuns et nécessaires à la prise de décisions. L'objectif de notre projet présenté dans ce rapport est la conception et la réalisation d'une application de gestion de prestation d'assistance maladie pour l'entreprise Soummam Assistance de Bejaia.

Pour la réalisation de ce projet, nous avons opté pour le langage de modélisation UML (Unifier Modeling Language), qui propose une architecture satisfaisante de modélisation orientée objet avec 2TUP (2 Track Unified Process) comme processus unifié ; une solution de développement logiciel adaptée à tout type de projet. Ses traits distinctifs tiennent en trois notions : piloté par les cas d'utilisation, centré sur l'architecture, itératif et incremental.

Pour l'implémentation, nous avons utilisé l'atelier de Génie Logiciel WinDev 15.

Après avoir présenté les outils et la méthode adoptée, nous allons maintenant exposer le plan du mémoire qui se subdivise en quatre principaux chapitres.

Dans le premier chapitre, nous présentons un état de l'art sur les systèmes d'informations, l'outil de modélisation UML, ainsi que la démarche du processus de développement logiciel(Processus Unifié 2TUP), et enfin nous énumérons les architectures réseaux.

Le second chapitre présente l'étude de l'existant et expose la problématique et les objectif visés.

Le troisième chapitre présente l'analyse conceptuelle de la solution proposée.

Dans le dernier chapitre, nous allons détailler la réalisation de la solution proposée, il présente la mise en place du nouveau système à l'aide de l'Atelier de Génie Logiciel Windev 15.

Nous finirons enfin par une conclusion générale.

1

Généralités

Introduction

Dans ce chapitre, nous allons définir quelques généralités portant sur le domaine des Systèmes d'informations l'informatique. Nous allons commencer par une présentation des systèmes d'information(S.I), définir le langage de modélisation unifié UML (Unified Modeling Language), la démarche générique du processus de développements logiciel qui l'accompagne, et enfin énumérer les architectures réseaux.

1.1 Généralités sur Les Systèmes d'Information (S.I)

Pour l'entreprise, les systèmes d'information qui permettent de traiter, d'acheminer et de sécuriser des données représentent un enjeu important, à la fois stratégique, économique, technique et humain.

En facilitant la prise de décision, un système d'information bien conçu peut permettre d'obtenir des avantages concurrentiels importants et durables.

1.1.1 Définition

Un système d'information est un système qui utilise des technologies de l'information pour saisir, transmettre, stocker, retrouver, manipuler ou afficher de l'information utilisée dans un ou plusieurs processus de gestion[14].

Selon, Jean PINETA, "un système d'information est un ensemble structuré de règles, de méthodes, d'outils et de ressources, visant à rassembler et à organiser des documents utiles de l'entreprise ainsi que les informations concernant ces documents " ;

" c'est un ensemble construit et articulé de techniques, de procédures, de règles, destiné à réaliser des tâches d'acquisition, de stockage, de traitement et de diffusion des informations, dans l'objectif d'aider les individus et les groupes d'individus de l'entreprise (service, atelier, département, instances de décisions, groupe de travail, etc.) à prendre des décisions de gestion ".[12]

1.1.2 Représentation des Systèmes d'Information

Un système d'information d'une organisation peut être représenté comme deux triangles ou les problèmes et les besoins de l'organisation sont transformés aux données et traitements à l'aide des technologies de l'information. (Figure 1.1).[14]



FIGURE 1.1 – Représentation d'un Système d'Information

1.1.3 Historique et Evolution des S.I

L'histoire des Systèmes d'Information s'éclaircit si on la situe dans la perspective des efforts d'industrialisation du travail de bureau qui débutèrent en 1880, à Chicago.

D'abord utilisée pour faire croître la productivité par automatisation des tâches répétitives, l'informatique s'enrichit des apports de la bureautique communicante et équipe les processus de production au plus près de leur déroulement. Dès lors les organisations entrent dans l'ère du travail assisté par ordinateur généralisé, et elles doivent chercher à articuler au mieux ces deux ressources : le cerveau de l'être humain et l'informatique décentralisée.

Aujourd'hui le système d'information (SI) est au cœur de la création de valeur au sein des organisations. C'est un ensemble complexe, souvent hétérogène, car constitué d'éléments qui se sont juxtaposés au fil du temps, au gré des choix stratégiques, des évolutions technologiques des systèmes informatiques en place, du développement de l'organisation elle-même. Le besoin d'intégration de ses différents composants est actuellement fortement ressenti par l'ensemble des organisations.

L'amélioration de l'efficacité et de l'efficience des organisations est la préoccupation permanente des décideurs. Dans une économie qui se mondialise, où la concurrence s'accroît, où les organisations cherchent à offrir davantage de services aux clients, l'information est de plus en plus une variable stratégique essentielle. Elle est nécessaire au processus de décision. Mais, si l'information accroît la connaissance et réduit ainsi l'incertitude, c'est le système d'information, grâce à toutes ses composantes et en liaison avec les autres moyens opérationnels, qui permet aux organisations de répondre aux besoins, de fournir en tous

lieux les biens et services personnalisés.

Ainsi, l'information est nécessaire au sein des organisations, pour décider, agir, apprendre (acquérir par l'expérience), comprendre (par exemple en analysant les faits passés), prévoir (par exemple par extrapolation), contrôler (par exemple en confrontant données réelles et prévues)...[14]

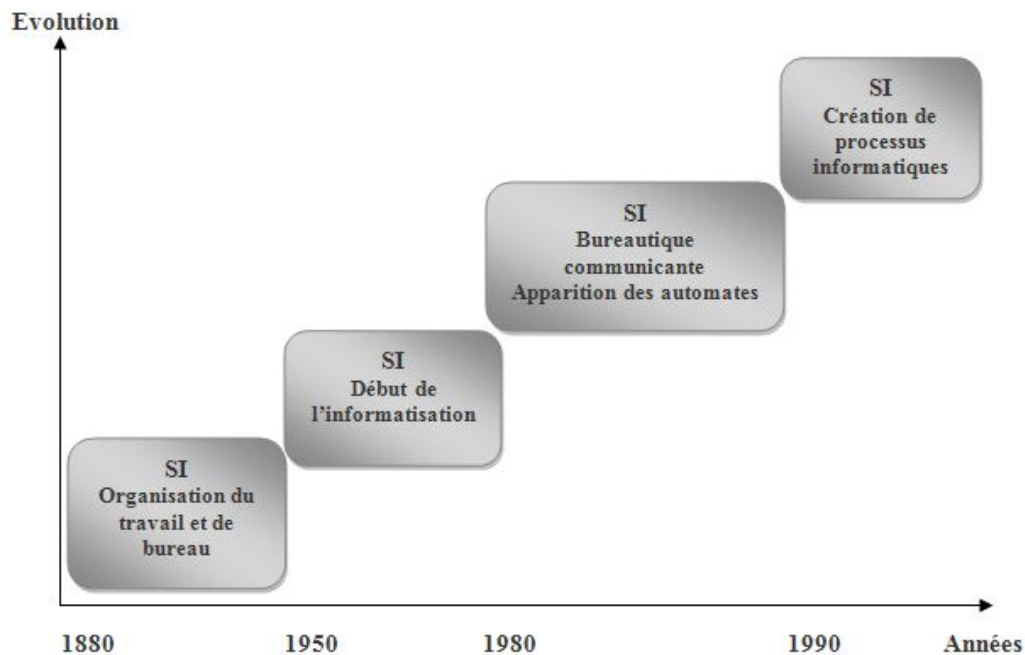


FIGURE 1.2 – Schéma de l'évolution des Systèmes d'Information

Le SI d'une organisation est de fait constitué d'un ensemble de moyens, de ressources, d'éléments organisés afin de :

- recueillir (collecter, saisir).
- traiter (extraire une information pertinente, calculer).
- stocker (mémoriser).
- diffuser de l'information (communiquer).

Au sein des organisations, les systèmes d'information sont souvent différenciés selon la nature de l'information et des traitements :

- les systèmes d'aide à la décision : Ce sont des systèmes qui exploitent des modèles d'analyse de problèmes.
- les systèmes opérationnels : Ce sont des systèmes qui exploitent des ensembles de données.
- Les systèmes de gestion de la connaissance : Ce sont des systèmes qui fournissent en particulier des outils pour le travail collaboratif.

1.2 Le Language de Modélisation Unifié (UML)

Par le passé, le modèle Entité-Relation représentait une grande partie des approches les plus utilisées.

Actuellement, les nouvelles technologies s'appuient sur le modèle objet.

En termes d'analyse et de modélisation objet, UML est devenu un standard incontournable, stabilisé et industriel.

1.2.1 Historique de l'UML

Au début des années 90, une cinquantaine de méthodes d'analyse et de conception objet qui existaient, seulement trois d'entre se sont nettement détachées. En effet, la volonté de converger vers une méthode unifiée était déjà bien réelle et c'est pour cette raison que les méthodes OMT, BOOCH et OOSE se sont démarquées des autres.

OMT (Object Modeling Technique) de James Rumbaugh et BOOCH de Grady Booch ont été les deux méthodes les plus diffusées en France durant les années 90. Par ailleurs, OOSE de Ivar Jacobson s'est aussi imposée dans le monde objet pour la partie formalisation des besoins.

Pour aller plus loin dans le rapprochement, James Rumbaugh et Grady Booch se sont retrouvés au sein de la société Rational Software et ont été ensuite rejoints par Ivar Jacobson en se donnant comme objectif de fusionner leur méthode et créer UML (Unified Methode Language).

Il est important de noter que contrairement à ce qui avait été envisagé au départ, le processus de développement a été sorti du champ couvert par le projet de norme. UML est donc une norme du langage de modélisation objet qui a été publiée, dans sa première version, en novembre 1997 par l'OMG (Object Management Group), instance de normalisation internationale du domaine de l'objet.

En quelques années, UML s'est imposée comme standard à utiliser en tant que langage de modélisation objet.

1.2.2 La notation UML

UML (Unified Modeling Language), se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. UML modélise l'ensemble des données et des traitements en élaborant des différents diagrammes. En clair, il ne faut pas designer UML en tant que méthode (Il y manque la démarche) mais plutôt comme une boîte d'outils qui sert à améliorer les méthodes de travail[6].

1.2.3 Les Diagrammes UML

UML dans sa version 2 propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles[5] :

1. **Les diagrammes structurels** : Ces diagrammes, représentent l'aspect statique d'un système (classes, objets, composants...) :
 - Diagramme de classe : Ce diagramme représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. C'est le diagramme pivot de l'ensemble de la modélisation d'un système.
 - Diagramme d'objet : Le diagramme d'objet permet la représentation d'instances des classes et des liens entre instances.
 - Diagramme de composant : Ce diagramme représente les différents constituants du logiciel au niveau de l'implémentation d'un système.
 - Diagramme de déploiement : Ce diagramme décrit l'architecture technique d'un système avec une vue centrée sur la répartition des composants dans la configuration d'exploitation.
 - Diagramme de paquetage : Ce diagramme donne une vue d'ensemble du système structuré en paquetage. Chaque paquetage représente un ensemble homogène d'éléments du système (classes, composants...).
 - Diagramme de structure composite : Ce diagramme permet de décrire la structure interne d'un ensemble complexe composé par exemple de classes ou d'objets et de composants techniques. Ce diagramme met aussi l'accent sur les liens entre les sous-ensembles qui collaborent.

2. **Les diagrammes de comportement** : Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés par UML :
 - Diagramme des cas d'utilisation : Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Il constitue un des diagrammes les plus structurants dans l'analyse d'un système.
 - Diagramme d'état-transition : Ce diagramme montre les différents états des objets en réaction aux événements.
 - Diagramme d'activités : Ce diagramme donne une vision des enchaînements des activités propres à une opération ou à un cas d'utilisation. Il permet aussi de représenter les flots de contrôle et les flots de données.
 - Diagramme de séquence : Ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets.
 - Diagramme de communication : Ce diagramme est une autre représentation des scénarios des cas d'utilisation qui met plus l'accent sur les objets et les messages échangés.
 - Diagramme global d'interaction : Ce diagramme fournit une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activités.
 - Diagramme de temps : Ce diagramme permet de représenter les états et les interactions d'objets dans un contexte où le temps a une forte influence sur le comportement du système à gérer.

1.2.4 Les Processus de développement

Un processus de développement définit une séquence d'étapes, en partie ordonnée, qui concourt à l'obtention d'un système logiciel ou à l'évolution d'un système existant ; pour produire des logiciels de qualité, qui répondent aux besoins des utilisateurs dans des temps et des coûts prévisibles[18].

1.2.4.1 Définition du Processus Unifié

Un processus unifié est un processus construit sur UML (Unified Modeling Language). Les processus unifiés sont le résultat de l'unification, non pas des processus, mais plus exactement les meilleures pratiques du développement objet [18].

Un processus unifié se distingue par les caractéristiques suivantes :

Itératif : Le logiciel nécessite une compréhension progressive du problème à travers des raffinements successifs et développer une solution effective de façon incremental par des itérations multiples.

Piloté par les risques : Les causes majeures d'échec d'un projet logiciel doivent être écartées en priorité.

Centré sur l'architecture : Le choix de l'architecture logicielle est effectué lors des premières phases de développement du logiciel. La conception des composants des produits est basée sur ce choix.

Conduit par les cas d'utilisation : Le processus est orienté par les besoins utilisateurs présentés par des cas d'utilisation.

1.2.4.2 Les activités du processus unifié

Expression des besoins : UP propose d'appréhender l'expression des besoins en se fondant sur une bonne compréhension du domaine concerné pour le système à développer et une modélisation des procédures du système existant.

Ainsi, UP distingue deux types de besoins :

- les besoins fonctionnels qui conduisent à l'élaboration des cas d'utilisation.
- les besoins non fonctionnels (techniques) qui aboutissent à la rédaction d'une matrice des exigences.

Analyse : L'analyse permet une formalisation du système à développer en réponse à l'expression des besoins formulée par les utilisateurs. L'analyse se concrétise par l'élaboration de tous les diagrammes donnant une représentation du système tant statique (diagramme de classe principalement), que dynamique (diagramme des cas d'utilisation, de séquence, d'activité, d'état-transition...)[18].

Conception : La conception prend en compte les choix d'architecture technique retenus pour le développement et l'exploitation du système. La conception permet d'étendre la représentation des diagrammes effectuée au niveau de l'analyse en y intégrant les aspects techniques plus proches des préoccupations physiques[18].

Implémentation L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutables et d'autres éléments du même type.

Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes

sous formes de codes sources.

Cette phase reste, comme dans toutes les autres méthodes, la plus lourde en charge par rapport à l'ensemble des autres phases (au moins 40 %)[18].

Test Les tests permettent de vérifier :

- la bonne implémentation de toutes les exigences (fonctionnelles et techniques).
- le fonctionnement correct des interactions entre les objets.
- la bonne intégration de tous les composants dans le logiciel.

Classiquement, différents niveaux de tests sont réalisés dans cette activité : test unitaire, test d'intégration, test de réception, test de performance et test de non régression[18].

1.2.4.3 La Méthode 2TUP

2TUP (2 Tracks Unified Process) est un processus UP, dédié 'a la modélisation des systèmes d'information. Il apporte une réponse aux contraintes de changement continu imposé aux systèmes d'information. En ce sens, il renforce le contrôle sur les capacités d' 'evolution et de correction de tels systèmes. Il intègre également la notion de composant selon différentes granularités : composants métier, composants logiciels...[18]

L'axiome fondateur de ce processus est de diviser la démarche en deux branches : fonctionnelle (approche par les fonctionnalités) et technique (étude de leur mise en œuvre).

Le bénéfice attendu est de pouvoir réutiliser l'aspect fonctionnel en cas de changement d'architecture technique, et de pouvoir réutiliser l'aspect technique en cas d'évolution du fonctionnel (ou tout simplement pour développer une autre application présentant les mêmes contraintes d'architecture).

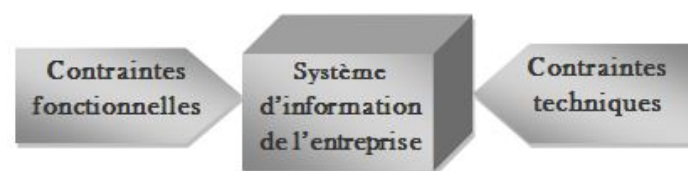


FIGURE 1.3 – Le Système d'Information soumis à deux types de contraintes

1. La branche gauche (fonctionnelle) :

capitalise la connaissance du métier de l'entreprise. Elle constitue généralement un investissement pour le moyen et le long terme.

Les fonctions du système d'information sont en effet indépendantes des technologies utilisées.

Cette branche comporte les étapes suivantes :

- La capture des besoins fonctionnels, qui produit un modèle des besoins focalisé sur le métier des utilisateurs.
- L'analyse.

2. La branche droite (architecture technique) :

capitalise un savoir-faire technique. Elle constitue un investissement pour le court et moyen terme. Les techniques développées pour le système peuvent l'être en effet indépendamment des fonctions à réaliser. Cette branche comporte les étapes suivantes :

- La capture des besoins techniques.
- La conception générique.

3. La branche du milieu :

à l'issue des évolutions du modèle fonctionnel et de l'architecture technique, la réalisation du système consiste à fusionner les résultats des 2 branches. Cette fusion conduit à l'obtention d'un processus en forme de Y. (Figure 1.4)

Cette branche comporte les étapes suivantes :

- La conception préliminaire.
- La conception détaillée.
- Le codage.
- L'intégration.

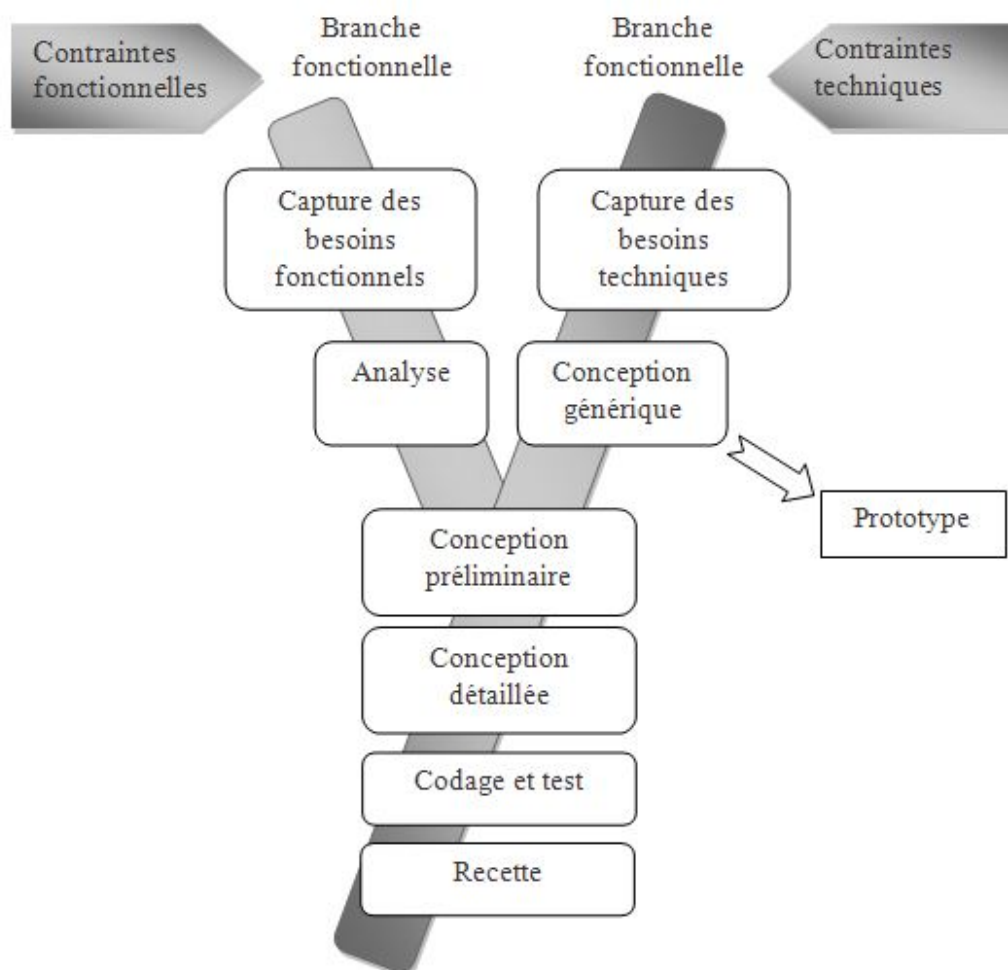


FIGURE 1.4 – le Cycle de vie en Y

1.3 Les Bases de Données (BDD)

1.3.1 Définition d'une Base de Données

Une BDD est un ensemble volumineux, structuré et minimalement redondant de données, reliées entre elles, stockées sur supports numériques centralisés ou distribués, servant pour les besoins d'une ou plusieurs applications, interrogeables et modifiables par un ou plusieurs utilisateurs travaillant potentiellement en parallèle.[4]

On distingue trois niveaux de modélisation pour les bases de données :

Le Modèle Conceptuel :

Il permet de décrire le réel selon une approche ontologique, sans prendre en compte les contraintes techniques.

Le Modèle Logique :

Il permet de décrire une solution, en prenant une orientation informatique générale (type de SGBD typiquement), mais indépendamment de choix d'implémentation précis.

Le Modèle Physique :

Il correspond aux choix techniques, en termes de SGBD choisi et de sa mise en œuvre (programmation, optimisation, etc.).

Exemples :

Exemple de formalisme de modélisation conceptuelle :

- Le modèle Entité-Association a été le plus répandu dans le cadre de la conception de bases de données.
- Le modèle UML, qui se généralise pour la conception en informatique, se fonde sur une approche objet.

Exemple de formalisme de modélisation logique :

- Le modèle relationnel est le modèle dominant.
- Le modèle relationnel-objet (adaptation des modèles relationnels et objet au cadre des SGBD) est actuellement en pleine croissance.
- Le modèle objet "pur" reste majoritairement au stade expérimental et de la recherche.

1.3.2 Système de Gestion de Bases de Données (SGBD)

Un SGBD est un logiciel qui prend en charge la structuration, le stockage, la mise à jour et la maintenance d'une base de données. Il est l'unique interface entre les informaticiens et les données (définition des schémas, programmation des applications), ainsi qu'entre les utilisateurs et les données (consultation et mise à jour).

1.3.2.1 Exemples

- Oracle est un SGBD relationnel (et Relationnel-Objet dans ses dernières versions) très reconnu pour les applications professionnelles.
- MySQL est un SGBD relationnel libre (licence GPL et commerciale), simple d'accès et très utilisé pour la réalisation de sites Web dynamiques. Depuis la version 4 MySQL implémentent la plupart des fonctions attendues d'un SGBD relationnel.
- PostgreSQL est un SGBD relationnel et relationnel-objet très puissant qui offre une alternative open-source aux solutions commerciales comme Oracle ou IBM.
- Access est un SGBD relationnel Microsoft, qui offre une interface conviviale permettant de concevoir rapidement des applications de petite envergure ou de réaliser des prototypes à moindre frais.

1.4 Les Réseaux Informatiques

1.4.1 Les Concepts Réseau

1.4.1.1 Réseau Informatique

Un Réseau Informatique est un ensemble d'ordinateurs reliés entre eux grâce à des lignes physiques et échangeant des informations sous forme de données numériques (des valeurs binaires, c'est-à-dire codées sous forme de signaux pouvant prendre deux valeurs : 0 et 1)[11].

1.4.1.2 Réseau Intranet

Un Intranet est un ensemble de services internet (par exemple un serveur web) interne à un réseau local, c'est-à-dire accessible uniquement à partir des postes d'un réseau local, ou bien d'un ensemble de réseaux bien définis, et invisible de l'extérieur. Il consiste à utiliser les standards client-serveur de l'internet (en utilisant les protocoles TCP/IP), comme par exemple l'utilisation de navigateurs internet (client basé sur le protocole HTTP) et des serveurs web (protocole HTTP), pour réaliser un système d'information interne à une organisation ou une entreprise[11].

1.4.1.3 Réseau Extranet

Un Extranet est un réseau du type internet, dont la liste de sécurité est externalisée, c'est-à-dire gérée par un organisme ou une entité externe aux utilisateurs. Par opposition, pour un réseau intranet, la liste de sécurité est gérée en interne.

La liste de sécurité est l'ensemble des données regroupant les identifiants (nom d'utilisateur (login), adresse IP, adresses MAC, clefs logiques ou physiques) autorisés à se connecter[11].

1.4.1.4 Types de Réseaux

On distingue différents types de réseaux selon leur taille (en termes de nombre de machines), leur vitesse de transfert des données ainsi que leur étendue. On définit les catégories de réseaux suivantes :

- Réseaux personnels ou PAN (Personal Area Network)
- Réseaux locaux ou LAN (Local Area Network)
- Réseaux métropolitains ou MAN (Métropolitain Area Network)

- Réseaux étendus ou WAN (Wide Area Network)

1.4.2 Présentation de l'Architecture Client/Serveur

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacités d'entrée leur fournit des services. Ces services sont des programmes fournissant des données telles que des données, des fichiers, une connexion, etc.

Le concept du client-serveur est apparu dans les années 1990 pour pouvoir allier les avantages de l'informatique centralisée (puissance et sécurité) et de l'informatique individuelle (convivialité).

Les architectures client/serveur cherchent à répondre au problème suivant :

- où héberger les données ?
- où héberger les applications qui gèrent les données ?
- où héberger la logique applicative, c'est à dire les applications chargées de la gestion du dialogue avec l'utilisateur ? (exemple : les formulaires de saisie, les états, etc.)

1.4.3 Fonctionnement d'un système Client/Serveur

Un système client/serveur fonctionne selon le schéma suivant :

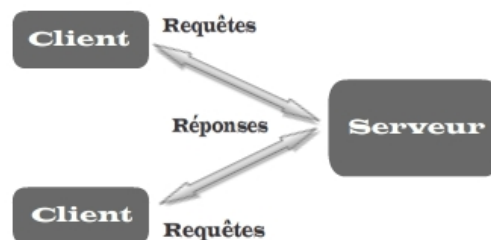


FIGURE 1.5 – Fonctionnement d'un Système Client/Serveur

- Le client émet une requête vers le serveur grâce à son adresse IP et le port qui désigne un service particulier du serveur.
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port.

1.4.4 Les différentes architectures Client/Serveur

1.4.4.1 Architecture à 1-tiers

Dans une approche d'application de type 1-tiers, les trois couches sont fortement et intimement liées, et s'exécutent sur la même machine. Dans ce cas, on ne peut pas parler d'architecture client-serveur mais d'informatique centralisée[2].

Dans un contexte simple utilisateur, la question ne se pose pas, mais dans un contexte multi-utilisateurs, on peut voir apparaître deux types d'architectures mettant en œuvre

des applications 1-tiers : des applications sur site central ; des applications réparties sur des machines indépendantes communiquant par partage de fichiers.

1.4.4.2 Architecture à 2-tiers

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tiers signifiant étage en anglais) caractérise les systèmes clients/serveurs dans lesquels le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service.

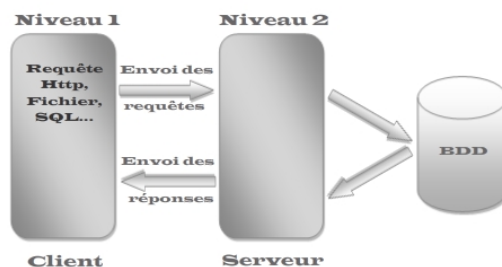


FIGURE 1.6 – Représentation de l'Architecture à 2 niveaux

Dans une architecture deux tiers, encore appelée client/serveur de première génération ou client/serveur de données, le poste client se contente de déléguer la gestion des données à un service spécialisé. Le cas typique de cette architecture est l'application de gestion fonctionnant sous Microsoft Windows et exploitant un SGBD centralisé.

Ce type d'application permet de tirer partie de la puissance des ordinateurs déployés en réseau pour fournir à l'utilisateur une interface riche, tout en garantissant la cohérence des données, qui restent gérées de façon centralisée.

La gestion des données est prise en charge par un SGBD centralisé, s'exécutant le plus souvent sur un serveur dédié. Ce dernier est interrogé en utilisant un langage de requête qui, le plus souvent, est SQL[2].

Le dialogue entre client et serveur se résume donc à l'envoi de requêtes et au retour des données correspondant aux requêtes. Ce dialogue nécessite l'instauration d'une communication entre client et serveur.

Le client provoque l'établissement d'une conversation afin d'obtenir des données ou un résultat de la part du serveur.

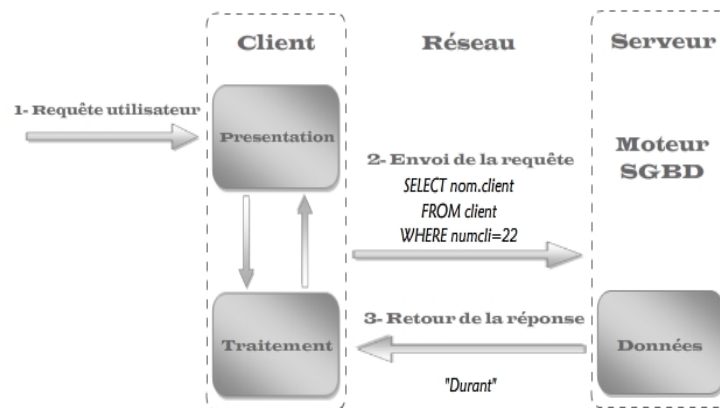


FIGURE 1.7 – Processus d'Accès aux Données en mode deux tiers

Cet échange de messages transite à travers le réseau reliant les deux machines. Il met en œuvre des mécanismes relativement complexes qui sont, en général, pris en charge par un middleware¹.

Exemples

SQL*Net : Interface propriétaire permettant de faire dialoguer une application cliente avec une base de données Oracle. Ce dialogue peut aussi bien être le passage de requêtes SQL que l'appel de procédures stockées.

ODBC : Interface standardisée isolant le client du serveur de données. C'est l'implémentation par Microsoft du standard CLI défini par le SQL Access Group. Elle se compose d'un gestionnaire de driver standardisé, d'une API s'interfaçant avec l'application cliente (sous Ms Windows) et d'un driver correspondant au SGBD utilisé.

DCE : Permet l'appel à des procédures distantes depuis une application.

Le choix d'un middleware est déterminant en matière d'architecture, il joue un grand rôle dans la structuration du système d'information.[9]

Limites de l'architecture à 2-tiers

L'expérience a démontré qu'il était coûteux et contraignant de vouloir faire porter l'ensemble des traitements applicatifs par le poste client. On en arrive aujourd'hui à ce que l'on appelle le client lourd, avec un certain nombre d'inconvénients.

- on ne peut pas soulager la charge du poste client, qui supporte la grande majorité des traitements applicatifs.
- le poste client est fortement sollicité, il devient de plus en plus complexe et doit être mis à jour régulièrement pour répondre aux besoins des utilisateurs.

1. programme permettant au poste client et au poste serveur de communiquer.

- la conversation entre client et serveur est assez bruyante et s'adapte mal à des bandes passantes étroites. De ce fait, ce type d'application est souvent cantonné au réseau local de l'entreprise.
- les applications se prêtent assez mal aux fortes montées en charge car il est difficile de modifier l'architecture initiale.
- la relation étroite qui existe entre le programme client et l'organisation de la partie serveur complique les évolutions de cette dernière.
- ce type d'architecture est grandement rigidifié par les coûts et la complexité de sa maintenance.

Malgré tout, l'architecture deux tiers présente de nombreux avantages qui lui permettent de présenter un bilan globalement positif :

- elle permet l'utilisation d'une interface utilisateur riche.
- elle a permis l'appropriation des applications par l'utilisateur
- elle a introduit la notion d'interopérabilité.[2]

1.4.4.3 Architecture à 3-tiers

Dans l'architecture à 3 niveaux (appelées architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

1. Le client : le demandeur de ressources (Niveau 1)
2. Le serveur d'application (appelé aussi middleware) : le serveur chargé de fournir la ressource mais faisant appel à un autre serveur pour certaine demande de ressources (Niveau 2).
3. Le serveur secondaire (généralement un serveur de base de données), fournissant un service (ressources) au premier serveur (Niveau 3).[2]

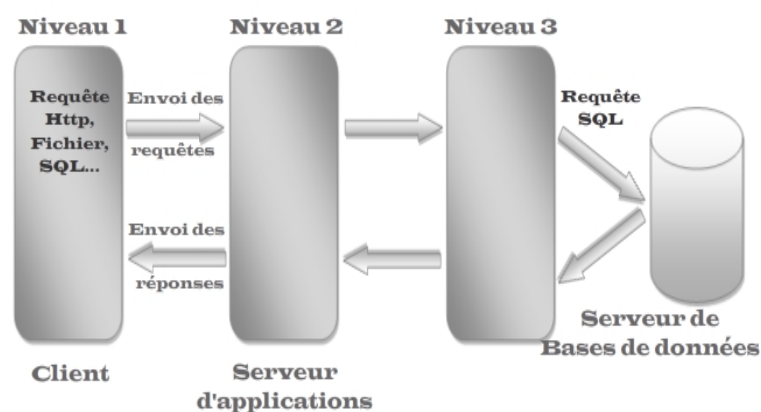


FIGURE 1.8 – Représentation de l'architecture à 3 niveaux

Limites de l'architecture à 3-tiers

L'architecture trois-tiers a corrigé les excès du client lourd en centralisant une grande partie de la logique applicative sur un serveur HTTP. Le poste client, qui ne prend à sa

charge que la présentation et les contrôles de saisie, s'est trouvé soulagé et plus simple à gérer.

En revanche, le serveur HTTP constitue la pierre angulaire de l'architecture et se trouve souvent fortement sollicité : il est difficile de répartir la charge entre client et serveur.

On se retrouve confronté aux épineux problèmes de dimensionnement serveur et de gestion de la montée en charge rappelant l'époque des mainframes. De plus, les solutions mises en œuvre sont relativement complexes à maintenir et la gestion des sessions est compliquée, mais reste possible.

Les contraintes semblent inversées par rapport à celles rencontrées avec les architectures deux-tiers : le client est soulagé, mais le serveur est fortement sollicité.

Le juste équilibre de la charge entre client et serveur semble atteint avec la génération suivante : les architectures n-tiers.^[2]

Comparaison entre l'architecture 2-tiers et 3-tiers

L'architecture à deux niveaux est donc une architecture client/serveur dans laquelle le serveur est polyvalent, c'est-à-dire qu'il est capable de fournir directement l'ensemble des ressources demandées par le client.

Dans l'architecture à trois niveaux par contre, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur web/serveur de base de données par exemple). L'architecture à trois niveaux permet :

- Une plus grande flexibilité/souplesse.
- Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau.
- De meilleures performances, étant donné le partage des tâches entre les différents serveurs.

Conclusion

A l'issu de ce chapitre, nous avons réussi à définir les différent concepts informatiques qui nous ont semblé essentiels à la réalisation de notre projet. Nous nous sommes ainsi familiarisé avec l'outil de modélisation UML d'une part et d'autre part, avec la démarche du processus de développement logiciel (Processus Unifié 2TUP),ainsi que l'identification de l'architecture reseau approprié à notre application. Ce qui va nous guider dans la réalisation des prochaines étapes de notre projet.

2

Analyse et Spécification des Besoins

Introduction

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, et d'identifier les cas d'utilisation initiaux.

Notre objectif dans cette étape est donc d'exprimer les besoins attendus du futur système à développer.

2.1 Spécification des besoins

2.1.1 Présentation de l'organisme d'accueil

La SARL Soummam Assistance est une entreprise lucrative privée, offrant un service de prestation qui consiste à couvrir le transport sanitaire des malades, de leurs domiciles jusqu'à leurs centres de soins.

2.1.2 Historique

L'unité de transport sanitaire Soummam Assistance a été fondée en juin 2000, par Docteur Nouredine BELMOURI, agréé par le ministère de la santé et de la population, implantée au début au niveau de la wilaya de Béjaia.

Ce n'est qu'en juillet 2001 que cette unité est devenue la SARL Soummam Assistance couvrant aujourd'hui quatre wilayas agréées à l'échelle nationale (Tizi-Ouzou, Bejaia, Constantine et Sétif) et ce, à la demande des malades dialysés.

Son premier objectif est d'améliorer les conditions du malade en lui offrant le confort nécessaire de sorte à remplir toutes les conditions de sa prise en charge tout au long de ses déplacements.

2.1.3 Organigramme de l'entreprise

L'organigramme suivant illustre l'ensemble des services de l'entreprise Soummam Assistance.

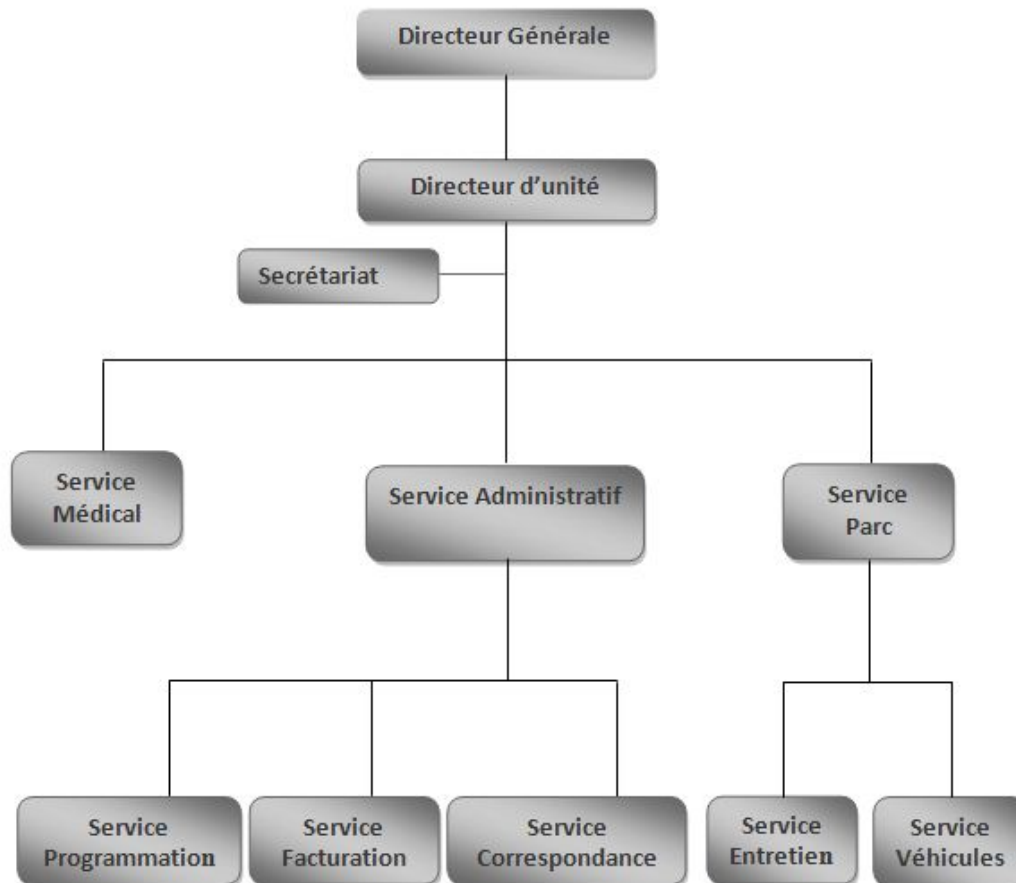


FIGURE 2.1 – Organigramme de Soummam Assistance

Présentation des services

L'entreprise se compose de trois principaux services qui sont :

- **Service Médical** : Ce service s'occupe de la gestion médicale et paramédicale de l'entreprise. La loi stipule que chaque malade transporté doit être accompagné par un infirmier.
- **Service Parc** : Ce service s'occupe de la gestion de l'ensemble des véhicules et des chauffeurs de l'entreprise, ainsi que l'entretien des véhicules et ambulances.
- **Service Administratif** : Ce service est structuré en trois sous service
 - Le service programmation : s'occupe de l'établissement du planning de rendez-vous, ainsi que la gestion de ces derniers.
 - Le service facturation : s'occupe de la gestion des factures de l'ensemble des malades transportés.
 - Le service correspondance : s'occupe de la gestion des interactions entre le service administratif de l'entreprise et l'organisme d'assurance maladie et autres.

2.1.4 Effectif de l'entreprise

service	Moyens Humains	Moyens matériels
Direction générale	1 Directeur Général 1 Secrétaire Général	1 Tel/fax 1 Ordinateur
Administration	1 Réceptionniste 1 Responsable Administratif 1 Responsable Facturation 1 Responsable Correspondance	1 Tel/fax 1 Ordinateurs 1 Imprimante/Photocopieuse 2 imprimantes simples
Parc	2 Mécaniciens 1 Responsable des Achats 17 chauffeurs	15 véhicules 4 ambulances

TABLE 2.1 – Effectif de Soummam Assistance

2.1.5 Objectif de l'étude

Le développement de technologie de l'information et en particulier de la micro-informatique permet actuellement à toute entreprise, quelle que soit sa taille, d'informatiser ses opérations et sa gestion. Ceci a l'avantage d'effectuer des tâches répétitives, le système manuel est encombrant, moins rapide et présente souvent des erreurs de manipulation.

Notre objectif dans cette étude est de développer et implémenter une application de gestion pour l'entreprise Soummam Assistance qui permettra essentiellement l'amélioration et la simplification du travail administratif.

Le tableau 2 représente une comparaison entre les deux systèmes d'information : manuel et automatisé.

Tâche	S.I. Manuel	S.I. Automatisé
Stocker une grande masse d'information (facture, informations sur les assurés, malades, chauffeurs et véhicules)	Sur des fiches dans un carton <ul style="list-style-type: none"> • Encombrant • Risque de détérioration • Difficile de faire plusieurs copies 	Sur un disque <ul style="list-style-type: none"> • Espace réduit. • Facilité de faire plusieurs copies • Sécurité de l'information
Accès à l'information (rechercher la facture d'une ancienne date).	Recherche manuelle. Facture par facture (lente).	La machine effectue une recherche rapide et instantanée pour l'utilisateur.
Traitement (établir une facture).	<ul style="list-style-type: none"> • Traitement lent. • Risque d'erreur. 	Calcul automatique sans risque d'erreur

TABLE 2.2 – Comparaison des deux S.I manuel et automatisé

2.1.6 Présentation du Domaine d'Étude

Dans notre étude on s'intéressera au service parc et administratif de l'entreprise Soummam Assistance de Bejaia là où s'est effectué Notre stage. Dans le service administration nous allons nous intéresser à l'automatisation du service facturation et dans le service parc nous allons automatiser la gestion des séances d'entretien effectuées sur les véhicules.

2.1.7 Situation Informatique

Le service administratif de l'entreprise Soummam Assistance dispose de ce matériel informatique suivant :

- Deux (02) ordinateurs HP ayant les caractéristiques suivants :
 - Processeur Intel(R) pentium(R) Dual, vitesse d'horloge : 2.0 GHz.
 - Capacité mémoire 512 Mo de RAM.
 - Capacité de disque dur est de 160 Go.
 - Un système d'exploitation Vista à 32 bits.
 - Écrans de 15".
- Trois (03) imprimantes :
 - Deux (02) imprimante simple de marque HP LaserJet 1018 : 15 pages/minutes.
 - Imprimante photocopieuse d'une marque Canon IR 2018.

Le service parc ne dispose d'aucun matériel informatique.

2.1.8 Problématique

La non automatisation de la gestion des différentes tâches effectuées au sein de Soummam Assistance est le problème principal à l'origine des inconvénients suivants :

- Le risque de perte et de détérioration de documents.
- Le coût élevé de la reproduction des documents entraîne l'achat de grandes quantités de fournitures de bureau.
- Le manque de ressources humaines alourdi considérablement les tâches de l'effectif restreint des employés et rend le traitement de ces dernières difficile et lourd.
- La gestion manuelle de l'information rend difficile la circulation et le repérage de cette dernière au niveau du service administratif.
- Le manque de souplesse dans le travail.

Ces problèmes énoncés ne facilitent pas la gestion de l'information dans l'entreprise. Notre objectif est donc de partir de ces constats afin de pallier ces goulots d'étranglement.

2.1.9 Solutions

- Implémenter une base de données qui gardera la trace de toutes les informations concernant les malades transportés par l'entreprise, les chauffeurs, véhicules et toute information nécessaire dont l'entreprise a besoin.
- Implémenter une base de données qui gardera la trace de toutes les informations concernant les malades transportés par l'entreprise, les chauffeurs, véhicules et toute information nécessaire dont l'entreprise a besoin.
- Établir une application efficace répondant aux besoins immédiats de l'entreprise, ergonomique et facile à utiliser.

2.2 Analyse des besoins

2.2.1 Besoins fonctionnels et non fonctionnels

Le système dont l'entreprise Soummam Assistance veut se doter doit être opérationnel, évolutif, convivial et offrant les informations nécessaires à temps réel.

Pour ceci, le système à réaliser doit satisfaire les besoins de la totalité des utilisateurs. Nous présenterons dans ce qui suit tous les besoins fonctionnels ainsi que les besoins non fonctionnels du système.

Besoins fonctionnels :

- Gestion des assurés.
- Gestion des malades.
- Gestion des Rendez-vous.
- Gestion des factures.
- Gestion des chauffeurs.
- Gestion des véhicules.
- Gestion des Séances d'Entretiens des véhicules.

Besoins non fonctionnels :

A part les besoins fondamentaux, notre futur système doit répondre aux critères suivants :

La rapidité de traitement : En effet, vu le nombre important d'opérations quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.

La performance : Un logiciel doit être avant tout performant c'est à-dire à travers ses fonctionnalités, répond à toutes les exigences des utilisateurs d'une manière optimale.

La convivialité : Le futur logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur.

2.2.2 Identification des acteurs ¹

le seul acteur susceptible d'interagir avec le système dans notre cas present est l'Administrateur de l'application, nous l'appellerons ici : Administrateur.

2.2.3 Identification des messages ²

Le système émet les messages suivants

- Les formulaires de demande de prise en charge de frais de transport sanitaire.
- Les formulaires de demande de remboursement de transport sanitaire.
- Les factures.
- Les informations concernant les assurés, malades, chauffeurs et véhicules.

Le système reçoit les messages suivants

- Les créations, suppressions, modifications des assurés.
- Les créations, suppressions, modifications des malades.
- Les créations, ajouts, modifications des véhicules.
- Les créations, suppressions, modifications des chauffeurs.
- Les créations, ajouts, modifications des Rendez-vous.
- Les créations, ajouts, modifications des factures.

1. un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié.[13]

2. un message représente la spécification d'une communication unidirectionnelle entre les objets qui transporte de l'information avec l'intention de déclencher une activité chez le récepteur.[13]

2.2.4 Modélisation du Système

2.2.4.1 Le Diagramme de Cas d'Utilisation

Les principales fonctionnalités de l'application à concevoir sont régies autour des besoins de l'Administrateur. Elles sont illustrées dans le diagramme de cas d'utilisation suivant :

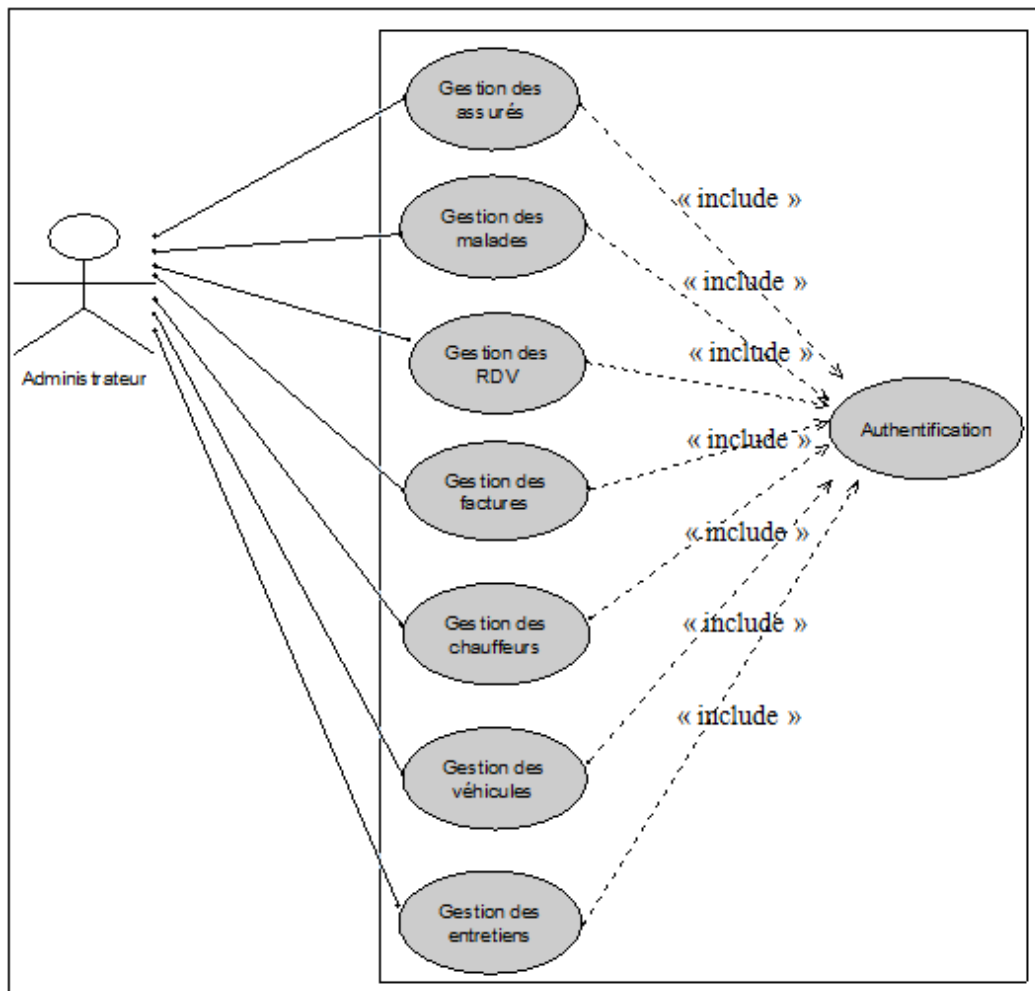


FIGURE 2.2 – Diagramme de Cas d'Utilisation

Description des différents Cas d'Utilisation

Gestion des Assurés : permet à l'administrateur de gérer les personnes qui prennent en charge les malades couverts par leur assurance sociale.

Gestion des Malades : permet à l'administrateur de gérer les malades transportés par l'entreprise.

Gestion des Chauffeurs : permet à l'administrateur de gérer les chauffeurs de l'entreprise.

Gestion des Rendez-vous : permet à l'administrateur de gérer les rendez-vous des malades transportés par l'entreprise.

Gestion des Factures : permet à l'administrateur de gérer les factures des malades transportés par l'entreprise.

Gestion des Véhicules : permet à l'administrateur de gérer les véhicules de l'entreprise.

Gestion des Entretien : permet à l'administrateur de gérer les séances d'entretiens des véhicules.

Authentification : Permet à un acteur de s'authentifier avant d'accéder à l'application.

2.2.4.2 Paquetage du Cas d'Utilisation Gérer les Factures

En raison de similarités entre les diagrammes, nous avons préféré donner un seul diagramme de cas d'utilisations, qui est la Gestion de la Facturation illustré dans la Figure 2.3.

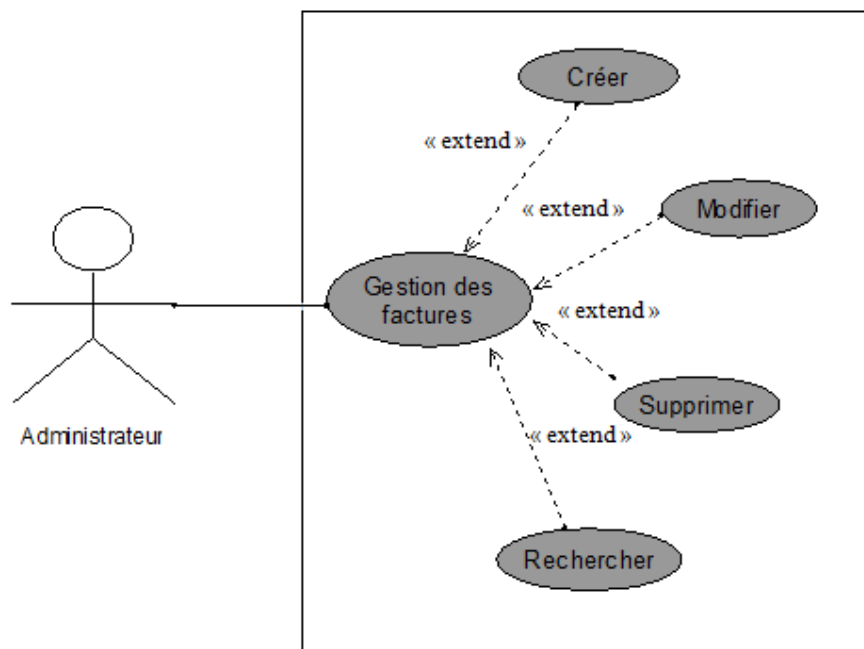


FIGURE 2.3 – Paquetage du cas d'utilisation "Gerer des Factures"

Description textuelle du cas d'utilisation

Créer une Facture : Lors de l'établissement d'une Facture, l'administrateur doit pouvoir enregistrer toutes les informations nécessaires relatives au malade transporté par la Soummam Assistance afin d'établir sa facture.

Modifier une Facture : L'administrateur peut se tromper en remplissant les informations communiquées, la récupération et correction de l'erreur est envisagée.

Supprimer une Facture : L'administrateur peut être amené à supprimer une facture relative à un malade si cette dernière n'a plus besoin d'être sauvegardée dans le système d'information.

Rechercher une Facture : L'administrateur peut être amené à rechercher une facture quelconque d'un malade donné en fonction de n'importe quelle information concernant soit le malade soit la facture (par date d'établissement, par nom, prénom du malade, trajet effectué etc.).

2.2.4.3 Les Diagrammes d'Activité

Diagramme d'activité du cas d'utilisation gérer les factures :

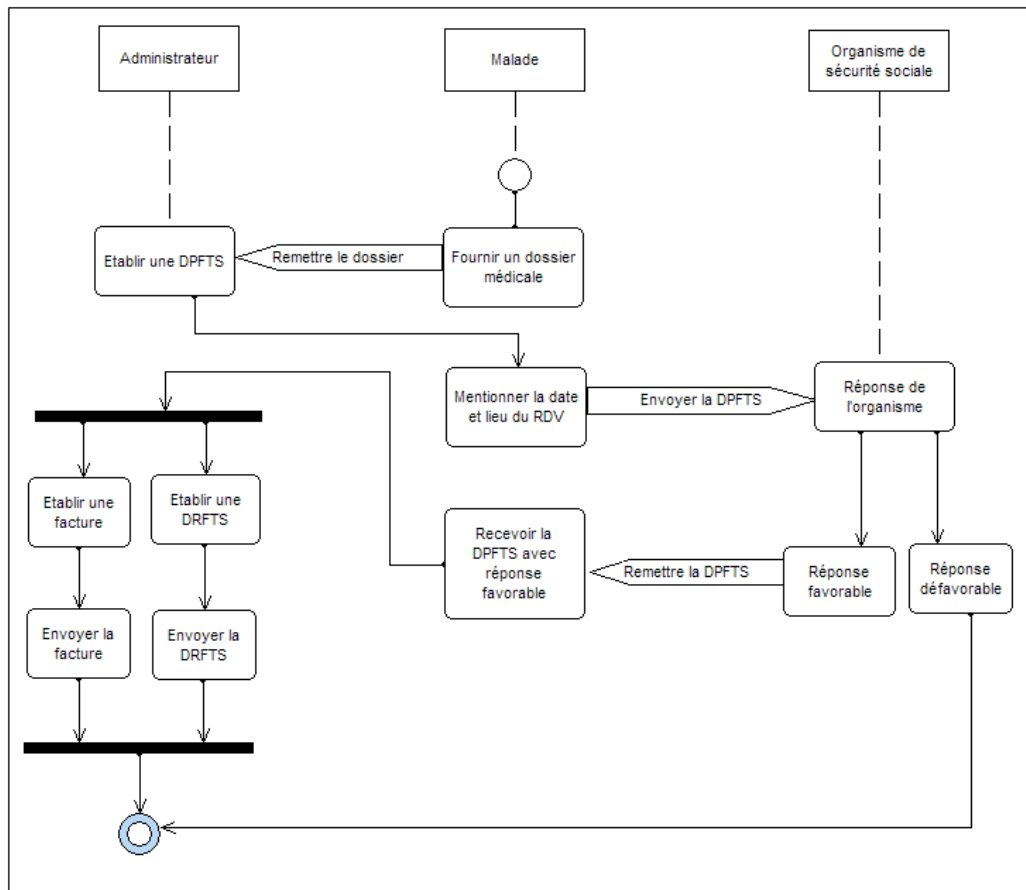


FIGURE 2.4 – Diagramme d'activité du cas d'utilisation "Gérer les Factures"

Diagramme d'activité du cas d'utilisation "Gérer les séances d'Entretiens des Véhicules" :

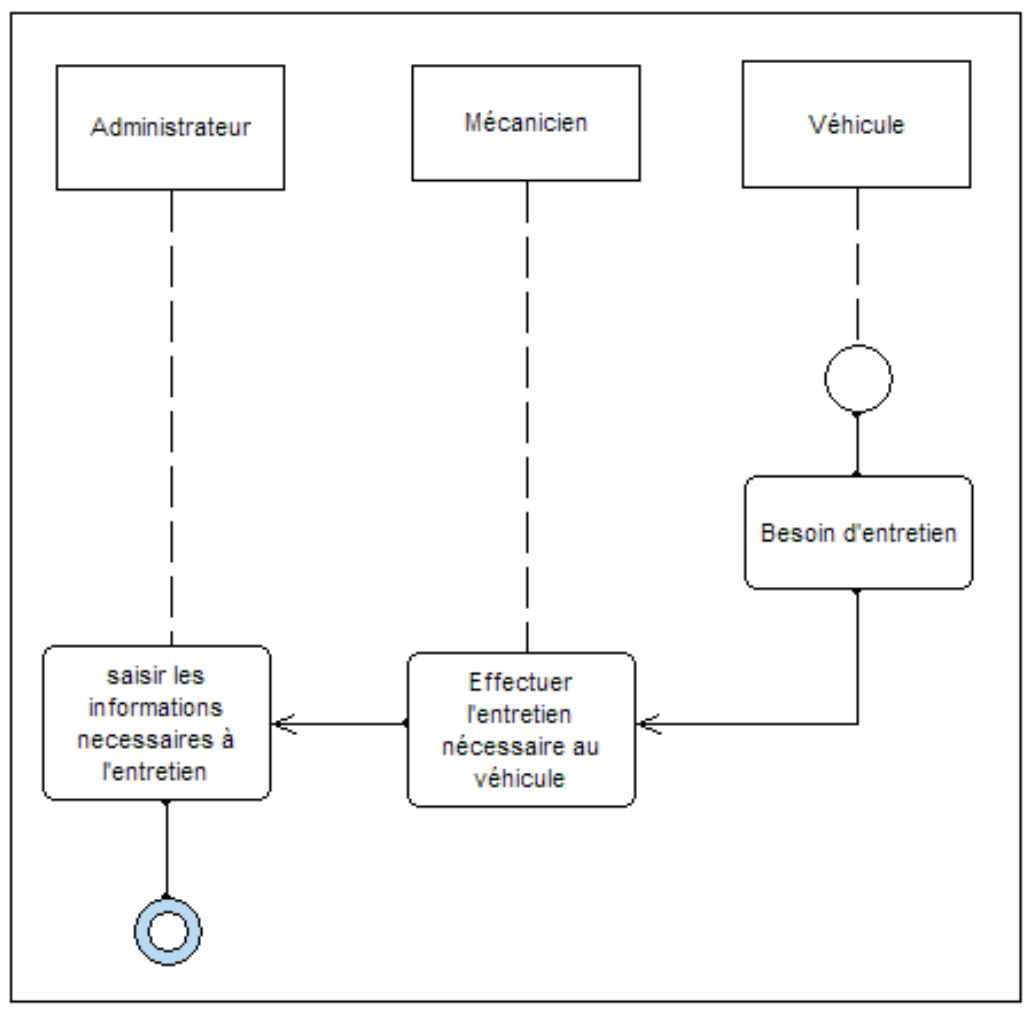


FIGURE 2.5 – Diagramme d'activité du cas d'utilisation "Gérer les séances d'Entretiens des Véhicules"

Conclusion

À l'issue de cette étape nous avons pu exprimer les objectifs attendus du futur système à concevoir, ainsi que l'analyse associée aux cas d'utilisations. Ce qui va nous permettre de passer à la prochaine étape intitulée "conception" que nous allons détailler dans le chapitre suivant.

3

Conception

3.1 Introduction

La phase de conception de logiciel consiste à traduire les besoins du système en spécifiant comment l'application pourra les satisfaire avant de procéder à sa réalisation. En effet, dans ce chapitre nous essayons d'étendre la représentation des diagrammes effectués au niveau du chapitre précédent en y intégrant les aspects techniques les plus proches des préoccupations physiques.

3.2 Réalisation du diagramme de classes

3.2.1 Les règles de Gestion

Le diagramme de classe du système étudié est basé sur les règles de gestion suivantes :

1. Un Malade est Assuré par un et un seul Assuré.
2. Un Assuré peut Assurer un ou plusieurs Malades.
3. Une Intervention peut concerner plusieurs Malades.
4. Une Intervention peut concerner plusieurs Rendez-vous.
5. Une Intervention peut concerner plusieurs Véhicules.
6. Une Intervention peut concerner plusieurs Chauffeurs.
7. Une Facture peut concerner un ou plusieurs Rendez-vous.
8. Un Rendez-vous est établi dans une et une seule Facture.
9. Un Rendez-vous peut être de type Dialyse, Rééducation ou Evacuation.
10. Un Véhicule possède une et une seule Catégorie.
11. Une Catégorie peut concerner un ou plusieurs Véhicules.
12. Un Entretien est subit par un et un seul Véhicule.
13. Un Véhicule subit un ou plusieurs Entretiens.

3.2.2 Dictionnaire de Données

Nom de La Classe	Codification	Désignation	Type	Longueur
Assuré	Id_A	Identifiant de l'Assuré	N	4
	Nom_A	Nom de l'assuré	A	20
	Pré_A	Prénom de l'assuré	A	20
	Date_nais_A	Date de naissance de l'assuré	D	8
	NSS	Numéro de sécurité sociale	N	25
	Agence	Agence de l'assuré	A	25
	Centre_P	Centre de paiement	A	25
	Num_tel_A	Numéro de téléphone de l'assuré	N	10

Chauffeur	Id_C	Identifiant du chauffeur	N	4
	Nom_C	Nom du chauffeur	A	20
	Pré_C	Prénom du chauffeur	A	20
	Adr_C	Adresse du chauffeur	A	30
	Num_tel_C	Numéro de téléphone du chauffeur	N	10
	Num_PC	Numéro du Permis de Conduire	N	10
Catégorie	Id_Cat	Identifiant de la catégorie	N	4
	Des_Cat	Désignation de la catégorie	A	25
Dialyse	Id_Dial	Identifiant de la dialyse	N	4
	Nbr_se	Nombre de séances de dialyse	N	3
Évacuation	Id_ev	Identifiant de l'évacuation	N	4
Entretien	Id_Entretien	Identifiant de l'entretien	N	4
	Type	Type de l'entretien	A	20
	Cout	Cout de l'entretien	N	8
	Date_E	Date de l'entretien	D	8
	Motif_E	Motif de l'entretien	A	30
	Km	kilométrage	N	6
	Observ	Observations sur l'entretien	A	200
Facture	Id_Facture	Identifiant de la facture	N	4
	Prix_unit	Prix unitaire	N	3
	Qte	Quantité	N	8
	Montant	Montant de la facture	N	10
Malade	Id_M	Identifiant du Malade	N	4
	Nom_M	Nom du Malade	A	20
	Pré_M	Prénom du Malade	A	20
	Adr_M	Adresse du Malade	A	30
	Qualite	Qualité du Malade	A	20
Rééducation	Id_reeduc	Identifiant de la rééducation	N	4
	Nbr_se	Nombre de séances de rééducation	N	2
RDV	Id_RDV	Identifiant du rendez-vous	N	4
	Centre_S	Adresse de la structure de soins	A	30
	Motif_T	Motif de Transport	A	15
	Trajet	Trajet à effectuer	A	30
	Distance	Distance Parcourue	N	4
	Date_RDV	Date du rendez-vous	D	8
Véhicule	Id_V	Identifiant du véhicule	N	4
	Mat	Matricule du véhicule	A	10
	Date_ach	Date d'achat du véhicule	D	8
Nom de la Classe Associative	Codification	Désignation	Type	Longueur
Intervention	Hr_Dep	L'heure de départ du chauffeur	N	4
	Hr_Ret	L'heure de retour du chauffeur	N	4

TABLE 3.1 – Dictionnaire de Données

3.2.3 Le Diagramme de Classes

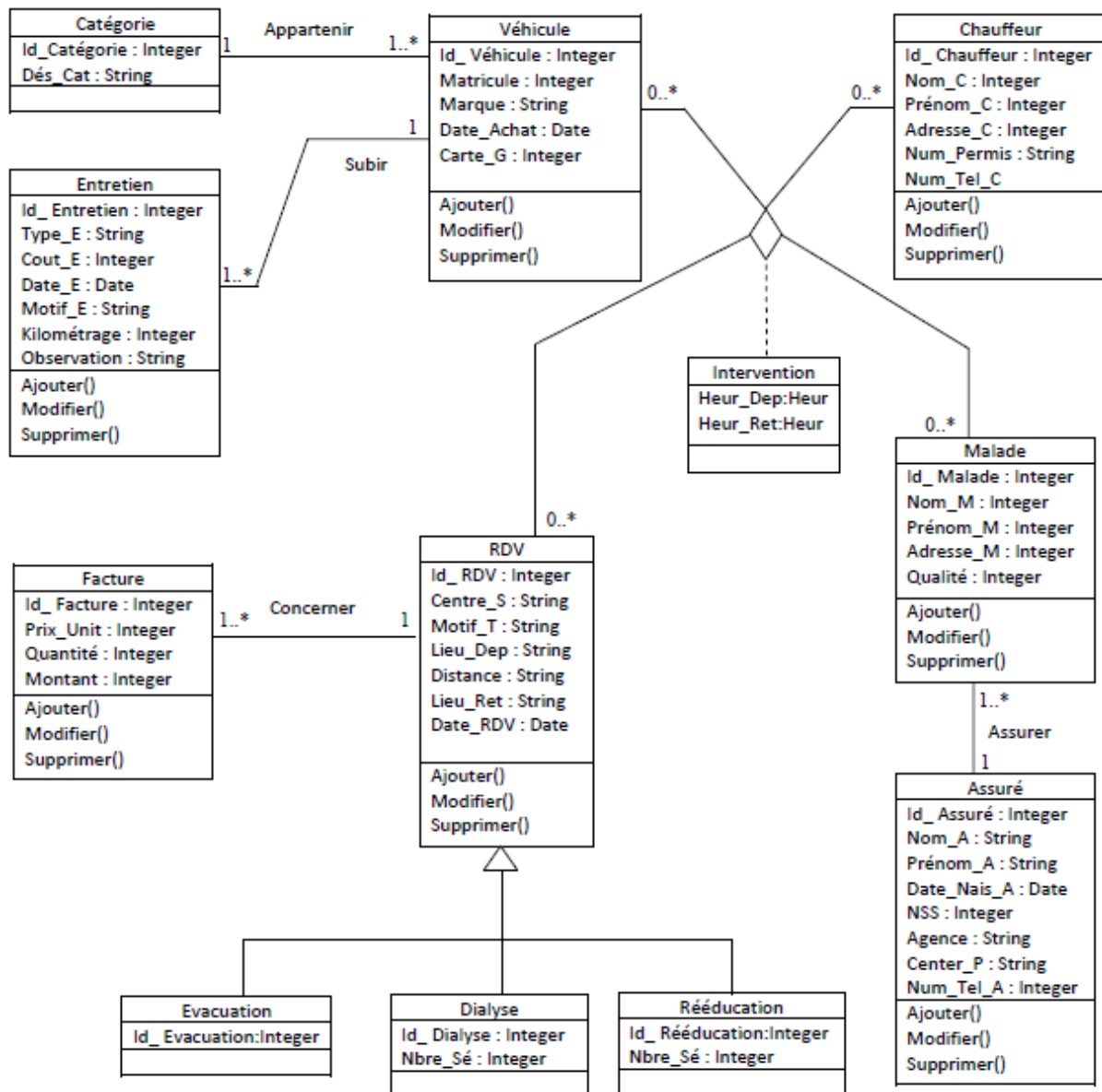


FIGURE 3.1 – Le Diagramme de Classes

3.3 Règles de Passage du Diagramme de Classes vers le Modèle Relationnel

Dans ce qui suit, nous allons présenter les différentes règles de passages, qui, à partir du Diagramme de Classes vont nous servir à l'élaboration du Modèle Relationnel.[3]

- Affecter une table à chaque classe.
- Pour toute classes C1 et C2 liées par une association A de type $*..1$, $*..n$ respectivement, dans le modèle relationnel on crée deux Tables T1, T2 dont les attributs sont respectivement les attributs des Classes C1, C2, en insérant la clé de C1 comme clé étrangère dans la Tables T2.
- Pour toute classes C1 et C2 liées par une association A de type $*..n$, $*..m$, dans le modèle relationnel on crée deux Tables T1, T2 dont les attributs sont respectivement les attributs des Classes C1, C2, ainsi qu'une troisième table T3 ayant la concatenation des deux clés des Tables T1 et T2 comme clé primaire.
- Pour toute classes C1 et C2 liées par une association A de type $*..1$, $*..1$, dans le modèle relationnel on crée deux Tables T1, T2 dont les attributs sont respectivement les attributs des Classes C1, C2, en insérant les clés primaires de C1 et/ou C2 comme clé étrangère dans les Tables T2 et/ou T1 respectivement selon le choix adéquat.

3.4 Règles de Normalisation

La normalisation correspond au processus d'organisation des données dans une base de données. Ce processus comprend la création de tables et l'établissement de relations entre celles-ci conformément à des règles conçues à la fois pour protéger les données et pour rendre la base de données plus flexible grâce à l'élimination de la redondance et des dépendances incohérentes.

Bien que plusieurs niveaux de normalisation soient possibles, la troisième forme normale est considérée comme étant le niveau le plus élevé nécessaire à la plupart des applications.[15]

1NF (Première forme normale) : Une relation est en première forme normale si et seulement si :

- tout attribut contient une valeur atomique.
- Chaque entité doit disposer d'un identifiant qui la caractérise de manière unique).[15]

2NF (Deuxième forme normale) : Une relation est en deuxième forme normale si et seulement si :

- tout attribut n'appartenant pas à une clé ne dépend pas que d'une partie de cette clé.
- Les propriétés d'une entité ne doivent dépendre que de l'identifiant de l'entité et non d'une partie de cet identifiant).[15]

3NF (troisième forme normale) : Une relation est en troisième forme normale si et seulement si :

- elle est en 2NF.
- tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non clé.
- Les propriétés d'une entité doivent dépendre de l'identifiant de la table de manière directe).[15]

3.5 Le Modèle Relationnel

Assuré (*Id_A*, Nom_A, Pre_A, Date_nais_A, NSS, Agence, Centre_P, Num_tel_A).

Chauffeur (*Id_C*, Nom_C, Pre_C,Adr_C, Num_tel_C,Num_PC).

Catégorie (*Id_Cat*, Des_Cat).

Dialyse (*Id_dial*, Nbr_Se, # Id_RDV).

Entretien (*Id_E*, Type, Cout, Date_E, Motif_E, Km, Observ, #Id_V).

Evacuation (*Id_Ev*, # Id_RDV).

Facture (*Id_F*, Prix_Unit, Qte, Montant, # Id_RDV).

Malade (*Id_M*, Nom_M, pre_M,Adr_M, Qualité, # Id_A).

Réducation (*Id_Reeduc*, Nbr_Se, # Id_RDV).

RDV (*Id_RDV*, Motif_T, Centre_S, Trajet,Distance,Date_RDV).

Véhicule (*Id_V*, Matricule, Date_Achat, # Id_Catégorie).

Intervention (#*Id_M*, #*Id_RDV*, #*Id_C*, #*Id_V*, H_Dep, H_Ret).¹

1. Les attributs en *Italique* représentent les Clés Primaires des différentes tables.

Conclusion

Après avoir identifié les règles de gestion du système d'information et les classes associées, nous avons procédé à l'établissement du dictionnaire des données afin d'élaborer un diagramme de classe détaillé. À base de ce dernier, nous avons abouti au modèle logique des données en faisant appel aux règles de passages.

4

Réalisation

Introduction

Après avoir terminé la conception de notre système d'information, nous passons à la dernière phase de notre étude, qui est l'implémentation, dans ce chapitre nous allons présenter les outils utilisés pour le développement de notre application "Soummam Assistance Centre de Gestion" et les interfaces les plus importantes de l'application.

4.1 Environnement et outils de développement

L'environnement de développement de notre application est basée sur : Serveur de Gestion de Base de Données HyperFileSQL Client/Serveur, l'Atelier de Génie Logiciel Windev15, avec son langage de programmation WLangage, l'outil de traitement d'images Adobe Photoshop CS5 ainsi que l'outil de modélisation UML ARGO UML. Dans ce qui suit nous donnerons une petite description de ces derniers.

4.2 Outils de développement

4.2.1 Windev 15

Windev est un AGL (Atelier de Génie Logiciel) complet édité par PCSOFT permettant de concevoir, développer et déployer des applications Windows dans de nombreux domaines : gestion, industrie, médical...

Les applications développées peuvent inclure des Bases de Données.

La technologie utilisée assure un fonctionnement intégral des applications et exploitables sous plusieurs plates-formes Windows, MAC ou UNIX.[10]

4.2.2 WLangage

Le WLangage est un langage de programmation de 5e génération inclus dans les outils de développement WinDev, WebDev et WinDev Mobile. Il est propriétaire et ne peut être manipulé qu'avec les outils PC SOFT. Le WLangage est né en 1992 avec la première version de WinDev.

Le WLangage est un langage de programmation procédurale qui permet la programmation impérative et la programmation orientée objet. C'est en fait un langage de programmation multi-paradigme.[10]

Le WLangage permet l'utilisation de classes et inclut entre autres :

- l'encapsulation (public, protégé, privé).
- la composition de classes.
- l'association de classes.
- l'héritage multiple.
- l'abstraction et le polymorphisme.

4.2.3 HyperFileSQL

C'est un SGBD exploité par les AGL WinDev, WebDev et WinDev Mobile. Une application HyperFileSQL Client/Serveur consiste à exécuter l'application sur différents postes utilisateur (appelés machines clientes) et à déporter la ou les bases de données et les traitements sur un poste serveur. Ce mode de fonctionnement permet des temps de réponses plus rapides et plus fiables, ainsi qu'une maintenance de la base de données facilitée.[10]

4.2.4 Structured Query Language

SQL est un Langage de requêtes structuré, est un langage informatique destiné à interroger ou piloter une base de données.

La partie langage de manipulation de données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données.

La partie langage de définition de données permet de créer, et de modifier l'organisation des données dans la base de données.

La partie langage de contrôle de transaction permet de commencer et de terminer des transactions, et la partie langage de contrôle de données permet d'autoriser ou d'interdire l'accès à certaines données à certaines personnes.[7]

Ce langage nous a servi aux différentes requêtes que nous avons utilisées pour interroger la base de données.

4.2.5 Adobe Photoshop CS5

Photoshop est un logiciel, de traitement et d'images assisté par ordinateur édité par Adobe. Il est principalement utilisé pour le traitement de photographies numériques, mais sert également à la création d'images.

Photoshop est un logiciel travaillant sur images matricielles (également appelées 'bitmap') car les images sont constituées d'une grille de points appelés pixels. L'intérêt de ces images est de reproduire des graduations subtiles de couleurs.[16]

Ce logiciel nous a servi à réaliser les différentes images qui se trouvent dans les différentes maquettes.

4.2.6 Argo UML

Argo UML est un outil de modélisation UML. Il permet d'analyser, de dessiner, de coder, de tester et de déployer. L'application permet de dessiner tous les types de diagrammes UML, d'inverser le code source pour le modèle UML, générer le code source à partir de diagrammes et d'élaborer la documentation.

4.3 Ergonomie

4.3.1 Choix des Couleurs

Le personnel de Soummam Assistance nous a laissé la liberté de choisir les couleurs à utiliser, mais nous avons opté principalement pour le blanc, le bleu et le rouge car ce sont les couleurs principales de l'entreprise, correspondant au domaine médical.

4.3.2 Logo

Étant Donné l'entreprise possédant déjà un logo personnalisé Significatif, nous l'avons utilisé avec quelques modifications et nous avons pris le soin de le mettre en évidence sur la partie gauche de chaque interface de l'application.



FIGURE 4.1 – Logo de l'entreprise "Soummam Assistance"

4.3.3 Le Corps

Le corps des fenêtres de l'application contient principalement des formulaires de remplissage ou des tableaux d'affichage. nous donnons ici un exemple de fenêtre vierge sur lequel nous nous sommes basés pour concevoir la plupart des fenêtres de l'application.

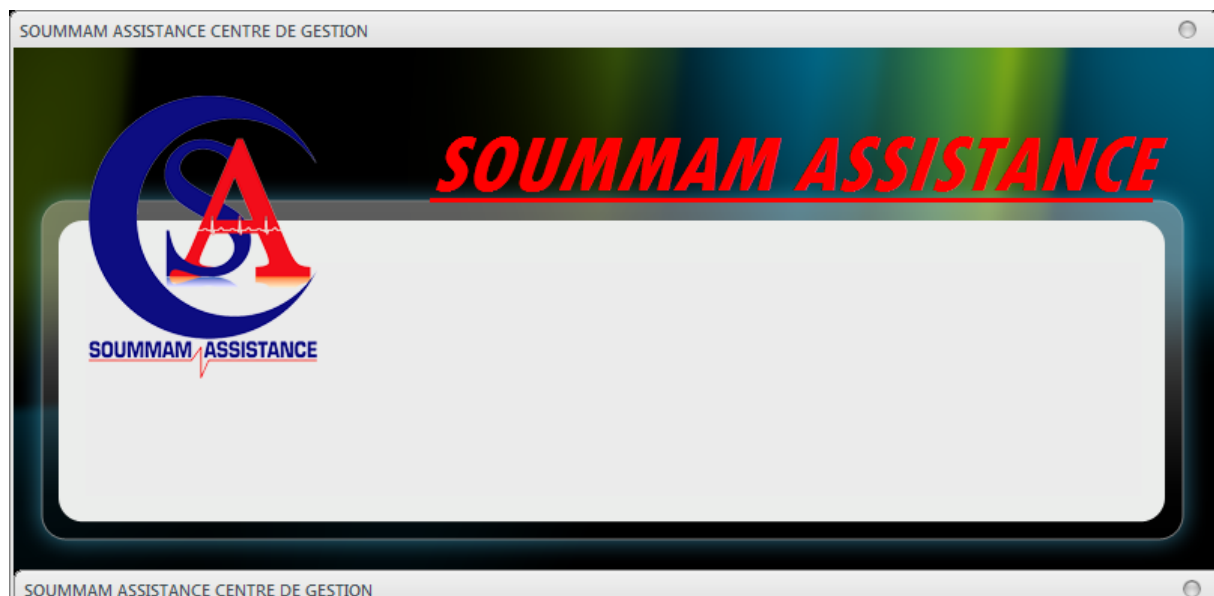


FIGURE 4.2 – Exemple de fenêtre de l'application

4.4 Implémentation de la base de données

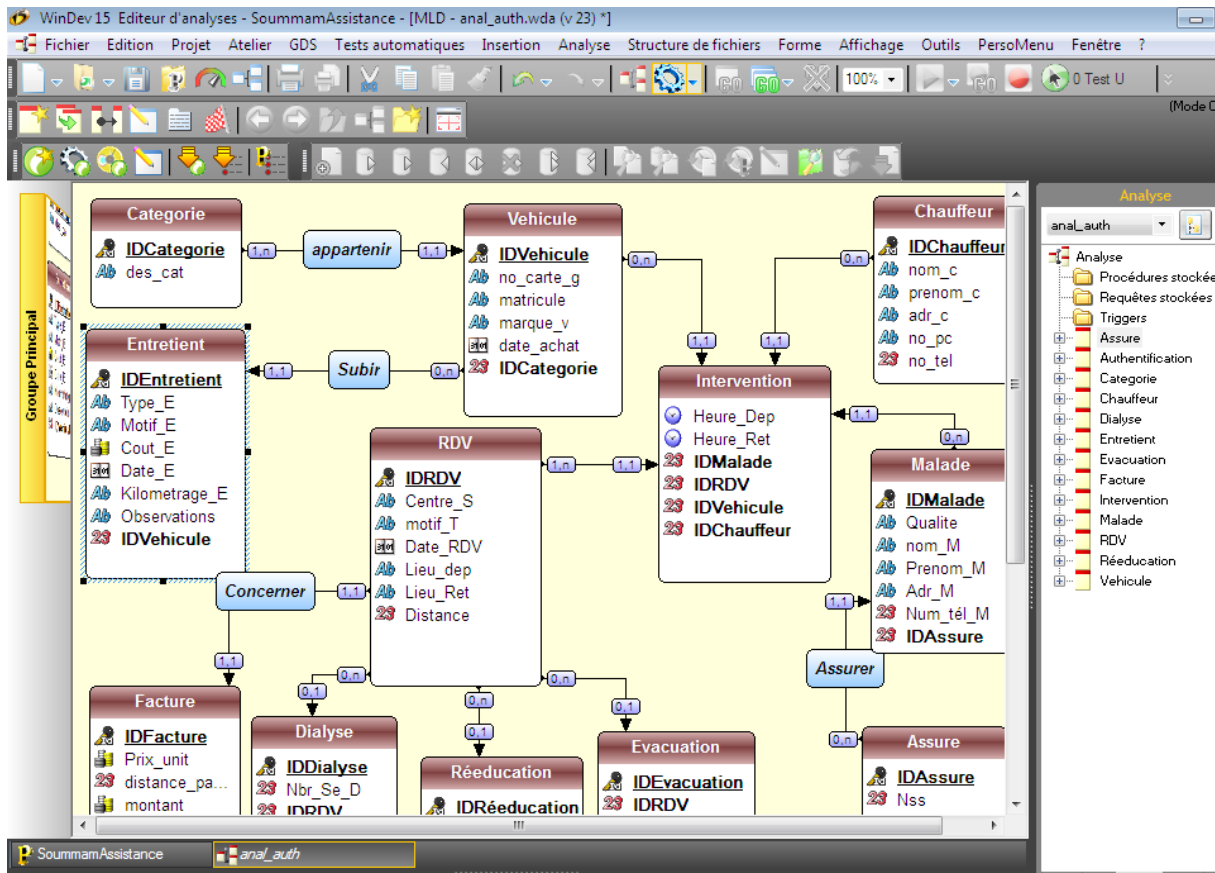


FIGURE 4.3 – Base de données de l'application "Soummam Assistance Centre de Gestion"

Exemple de Table : Table_Entretien

Cle	Nom	Libellé	Type	Taille
IDEntretien	Identifiant de Entretien		Id. automatique	4
Type_E	Type Entretien		Texte	20
Motif_E	Motif Entretien		Texte	50
Cout_E	Cout Entretien		Monétaire	10
Date_E	Date Entretien		Date	8
Kilometrage_E	Kilometrage		Texte	50
Observations	Observations		Texte	100
IDVehicule	IDVehicule		Numérique	4

FIGURE 4.4 – Table Entretien

4.5 Architecture de développement

L'architecture de développement de notre application est basée sur les trois concepts suivants : interface, procédure, requête, comme illustrés dans la figure 4.5.

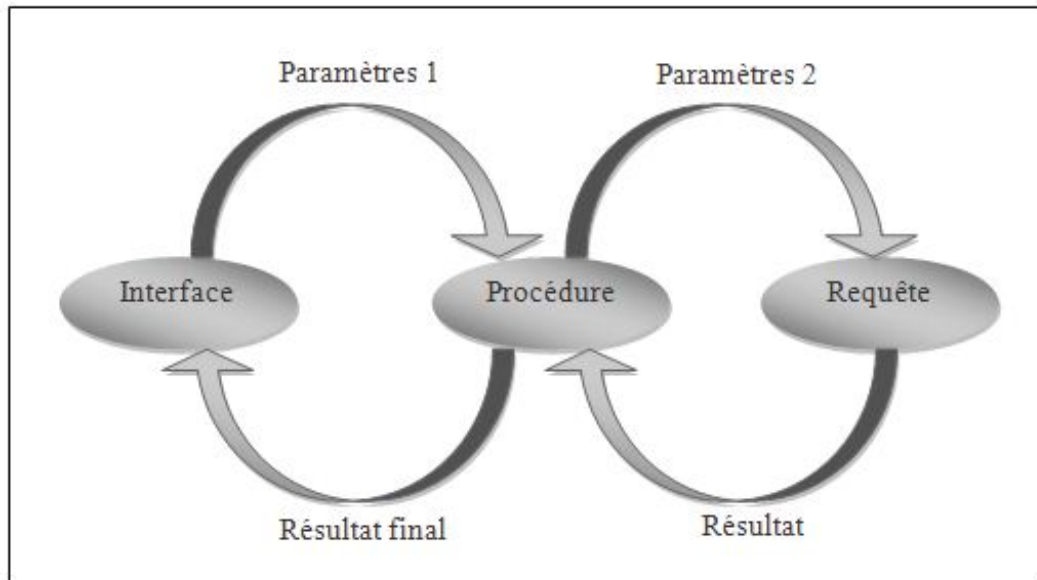


FIGURE 4.5 – Architecture de développement de l'application "Soummam Assistance Centre de Gestion"

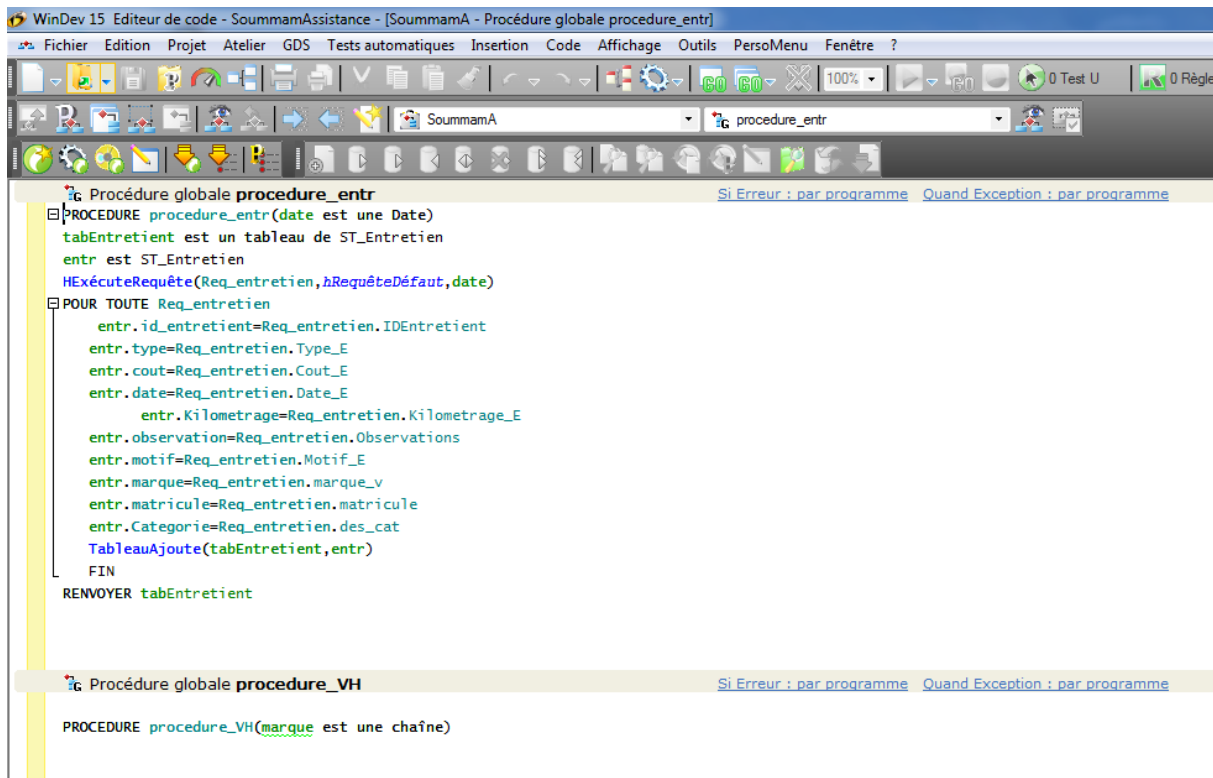
4.5.1 L'Interface

Une interface est un point de contact entre au moins deux objets de nature différente. Les objets peuvent être des systèmes informatiques et/ou des êtres humains. L'interface appartient donc aux deux objets, elle leur est commune, un langage partagé, mais ce qu'elle est pour l'un peut être différent de ce qu'elle est pour l'autre. Elle est une limite qui permet l'échange et la communication.[1]

4.5.2 Les Procédures

Les procédures permettent de décomposer un programme complexe en une série de sous-programmes plus simples, lesquels peuvent à leur tour être décomposés eux-mêmes en fragments plus petits, et ainsi de suite. Une procédure est une suite ordonnée d'instructions qui ne retourne pas de valeur (bloc d'instructions nommé et Paramétré).[17]

Exemple de Procédure



```
WinDev 15 Editeur de code - SoummamAssistance - [SoummamA - Procédure globale procedure_entr]
Fichier Edition Projet Atelier GDS Tests automatiques Insertion Code Affichage Outils PersoMenu Fenêtre ?
SoummamA procedure_entr
Procédure globale procedure_entr Si Erreur : par programme Quand Exception : par programme
PROCEDURE procedure_entr(date est une Date)
tabEntretien est un tableau de ST_Entretien
entr est ST_Entretien
HExécuteRequête(Req_entretien,hRequêteDéfaut,date)
POUR TOUTE Req_entretien
    entr.id_entretien=Req_entretien.IDEntretien
    entr.type=Req_entretien.Type_E
    entr.cout=Req_entretien.Cout_E
    entr.date=Req_entretien.Date_E
    entr.Kilometrage=Req_entretien.Kilometrage_E
    entr.observation=Req_entretien.Observations
    entr.motif=Req_entretien.Motif_E
    entr.marque=Req_entretien.marque_v
    entr.matricule=Req_entretien.matricule
    entr.Categorie=Req_entretien.des_cat
    TableauAjoute(tabEntretien,entr)
FIN
RENVOYER tabEntretien

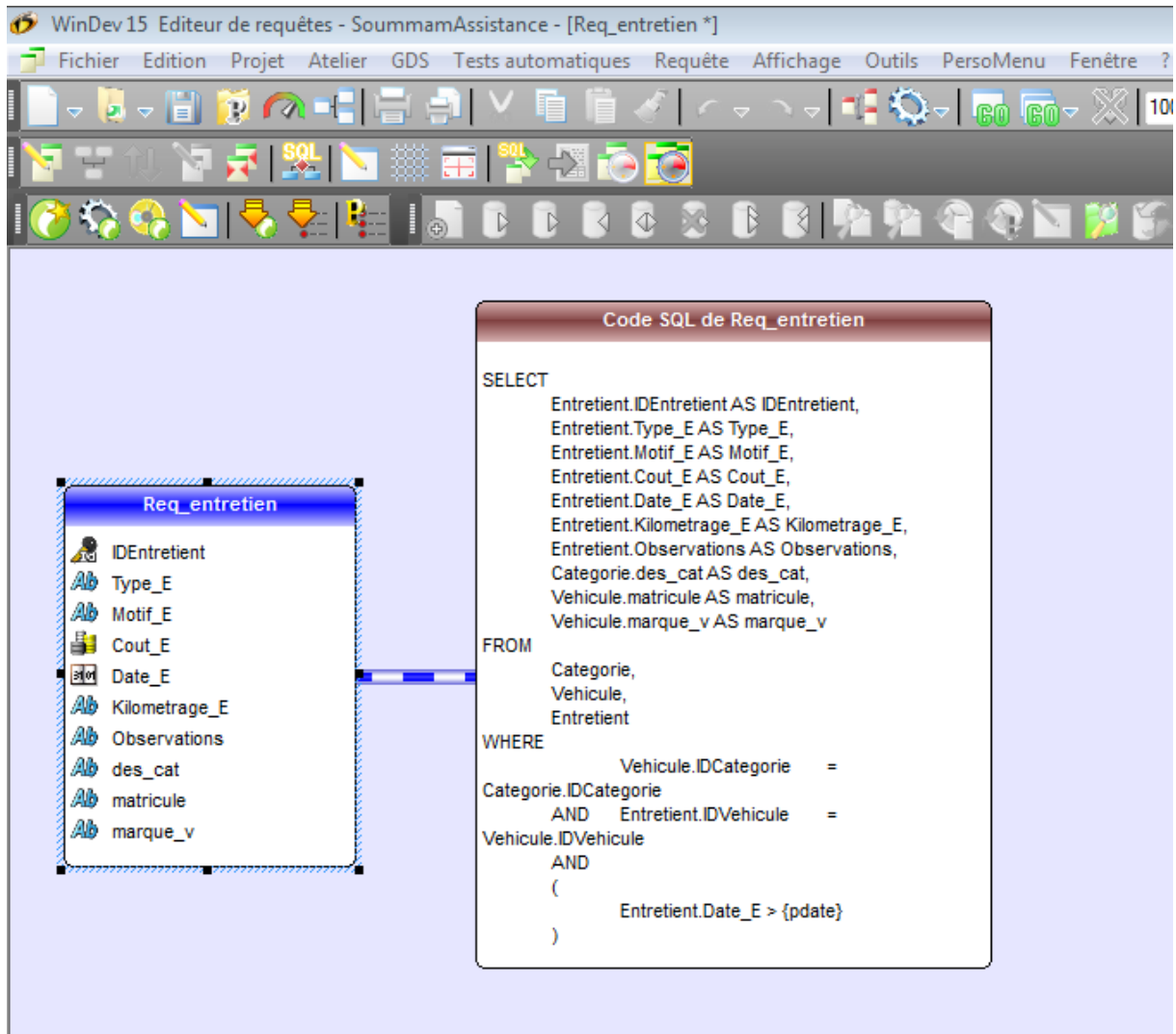
Procédure globale procedure_VH Si Erreur : par programme Quand Exception : par programme
PROCEDURE procedure_VH(marque est une chaîne)
```

FIGURE 4.6 – Procédure "Entrretien"

4.5.3 Les Requêtes

Une Requête est un ordre envoyé à un SGBD pour récupérer des données dans une base de données.[8]

Exemple de Requête



The screenshot shows a SQL editor window titled "WinDev 15 Editeur de requêtes - SoummamAssistance - [Req_entretien *]". The window contains a menu bar with options like "Fichier", "Edition", "Projet", "Atelier", "GDS", "Tests automatiques", "Requête", "Affichage", "Outils", "PersoMenu", and "Fenêtre ?". Below the menu bar is a toolbar with various icons, including a "SQL" icon. The main area of the editor is divided into two panes. On the left, there is a table structure pane titled "Req_entretien" showing the following fields: IDEntretien, Type_E, Motif_E, Cout_E, Date_E, Kilometrage_E, Observations, des_cat, matricule, and marque_v. On the right, there is a code editor pane titled "Code SQL de Req_entretien" containing the following SQL query:

```
SELECT
  Entretien.IDEntretien AS IDEntretien,
  Entretien.Type_E AS Type_E,
  Entretien.Motif_E AS Motif_E,
  Entretien.Cout_E AS Cout_E,
  Entretien.Date_E AS Date_E,
  Entretien.Kilometrage_E AS Kilometrage_E,
  Entretien.Observations AS Observations,
  Categorie.des_cat AS des_cat,
  Vehicule.matricule AS matricule,
  Vehicule.marque_v AS marque_v
FROM
  Categorie,
  Vehicule,
  Entretien
WHERE
  Vehicule.IDCategorie =
  Categorie.IDCategorie
  AND Entretien.IDVehicule =
  Vehicule.IDVehicule
  AND
  (
    Entretien.Date_E > {pdate}
  )
```

FIGURE 4.7 – Requête "Entretien"

4.6 Arborescence de l'application

Notre application se constitue de deux parties, une partie qui concerne le service Administratif sur laquelle on peut gérer les malades et assurés, gérer la planification des rendez-vous et établir des factures, une autre partie qui concerne le service parc qui nous permet de gérer les véhicules, chauffeurs et les séances d'entretiens.

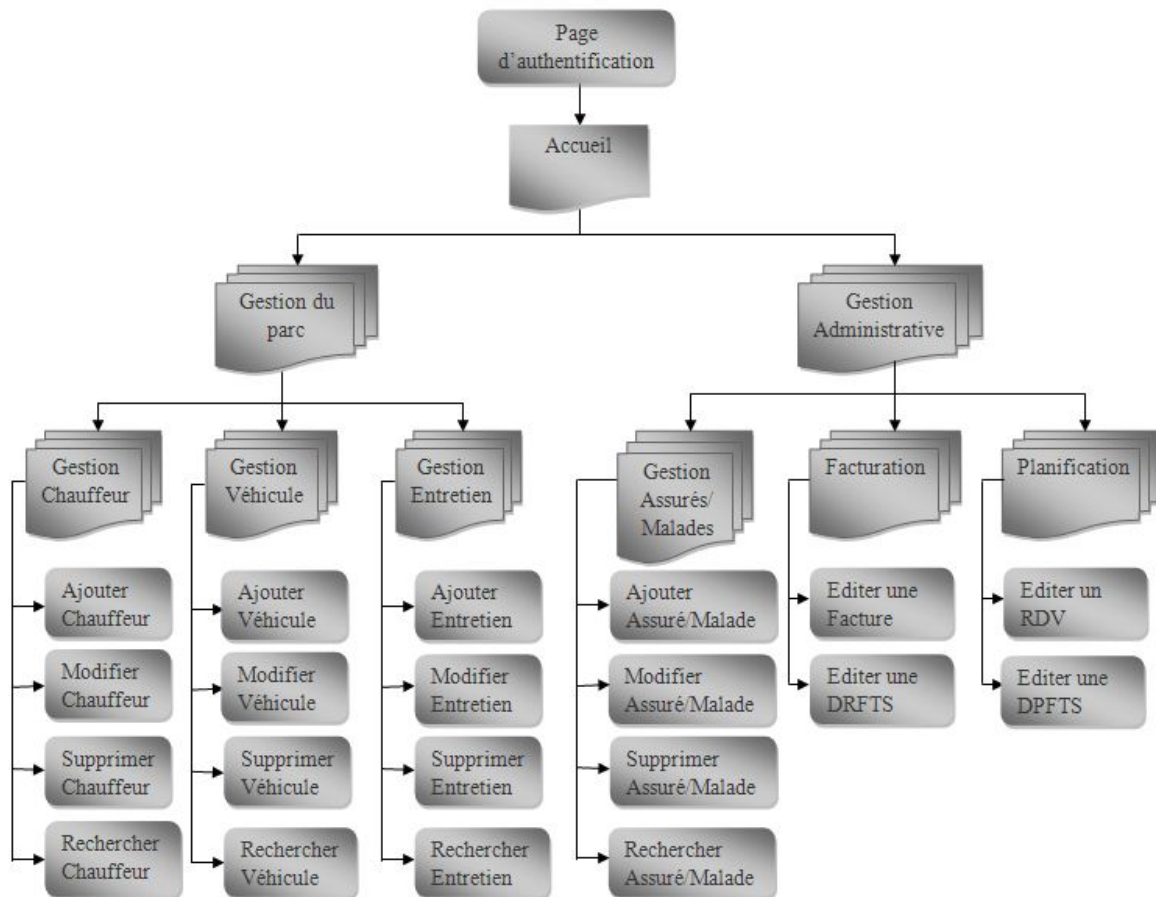


FIGURE 4.8 – Arborescence de l'application

4.7 Interfaces de l'application

Dans ce qui suit, nous allons présenter quelques interfaces de l'application "Soummam Assistance Centre de Gestion".

4.7.1 Fenêtre d'Authentification

Cette page permet à l'administrateur de s'authentifier avec un login et mot de passe, elle renvoie une fenêtre avec l'un des messages suivant :

- Authentification obligatoire : si on n'a rien saisi.
- Vérifiez votre login et/ou mot de passe : si on a saisi un login et/ou mot de passe incorrect.
- Vérifiez votre mot de passe : si on a saisi un mot de passe incorrect.

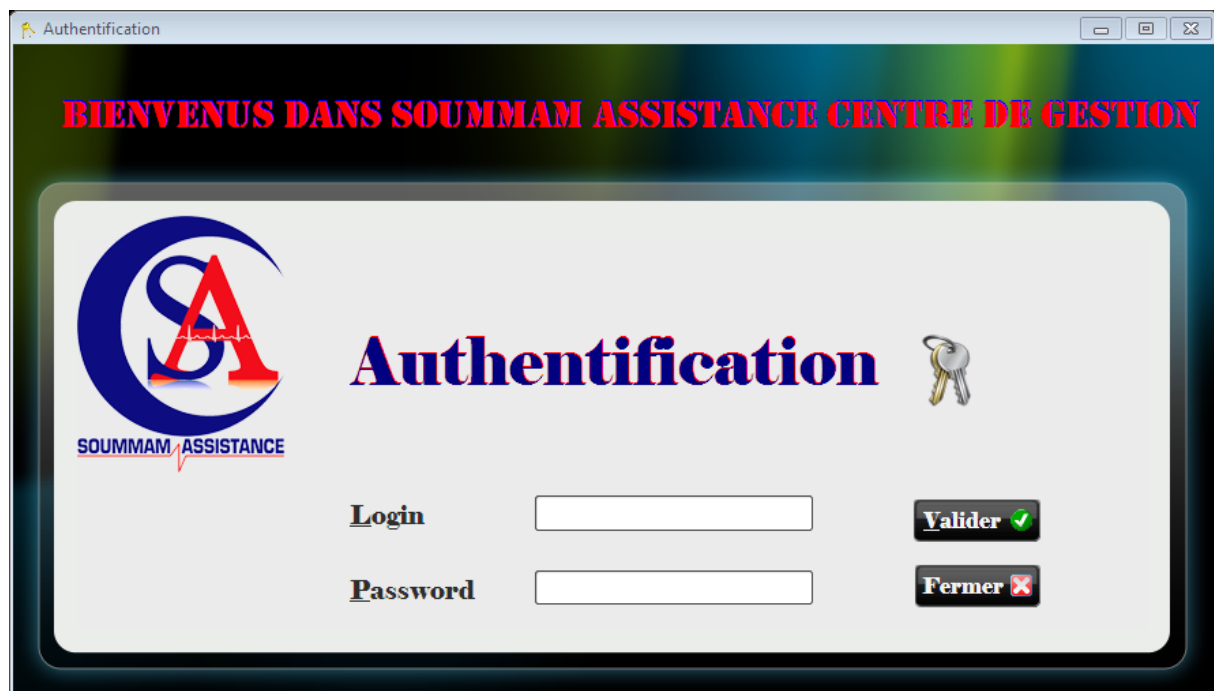


FIGURE 4.9 – Fenêtre d'Authentification

4.7.2 Page d'Accueil

Cette page permet à l'administrateur d'accéder aux différentes fonctionnalités de l'application "Soummam Assistance Centre de Gestion" qui sont principalement la gestion du parc véhicule et la gestion administrative.

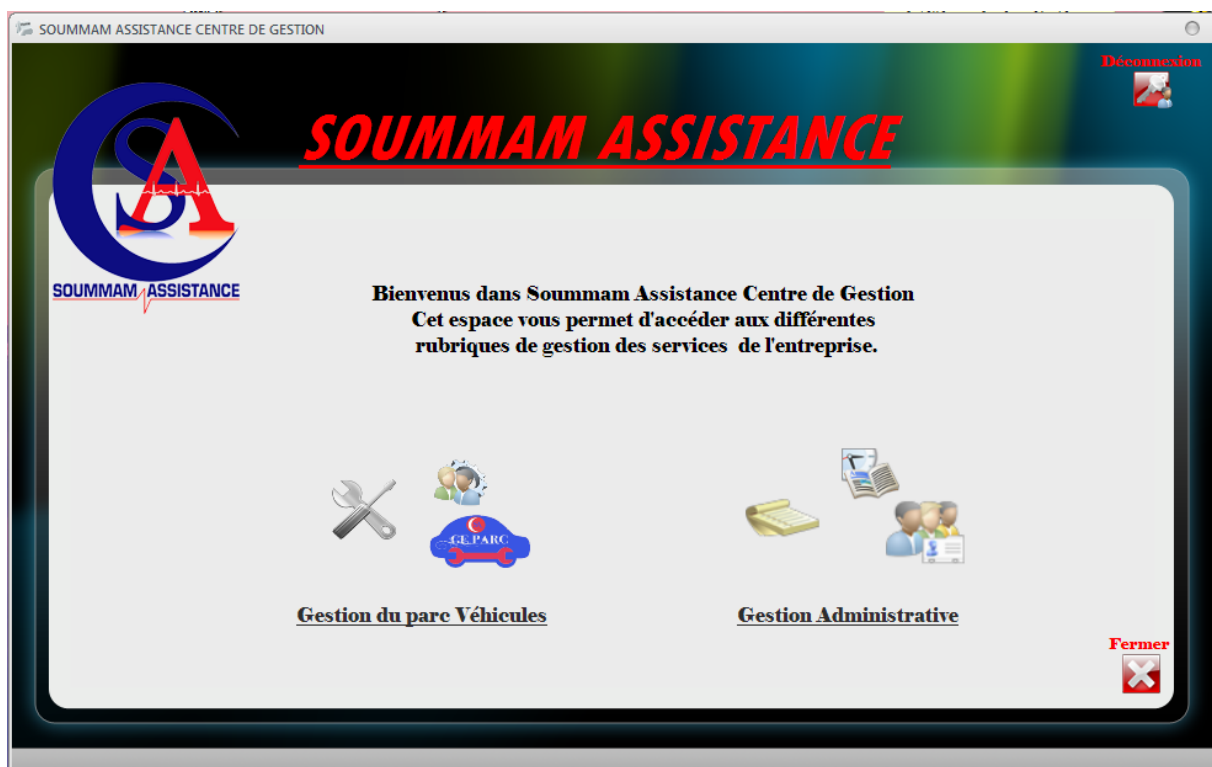


FIGURE 4.10 – Fenêtre d'Accueil

4.7.3 Page Gestion du Parc Véhicule

Cette page permet à l'administrateur de gérer le parc Véhicule, notamment : l'ajout, la modification et suppression des véhicule, chauffeurs ou les séances d'entretiens effectués sur les véhicules.



No_carte_g	Matricule	Marque véhicule	Date achat	Catégorie
158/324/12	10623-110-06	Renault Clio	31/01/2010	###
457/543/45	54321-107-06	Nissan J3	24/12/2007	###
345/876/87	98765-111-10	Accent	28/06/2012	###
AZE2345ERT	1234-709-06	Peugeot	23/05/2009	###
ERTY234567ER	73470-110-16	Accent	14/05/2010	###
NJVGHSFGB	10161016101	SONACOM	07/06/2012	###

FIGURE 4.11 – Fenêtre "Gestion du parc véhicule"

4.7.4 Page Planification

Cette page permet à l'administrateur de planifier un rendez-vous pour un malade.

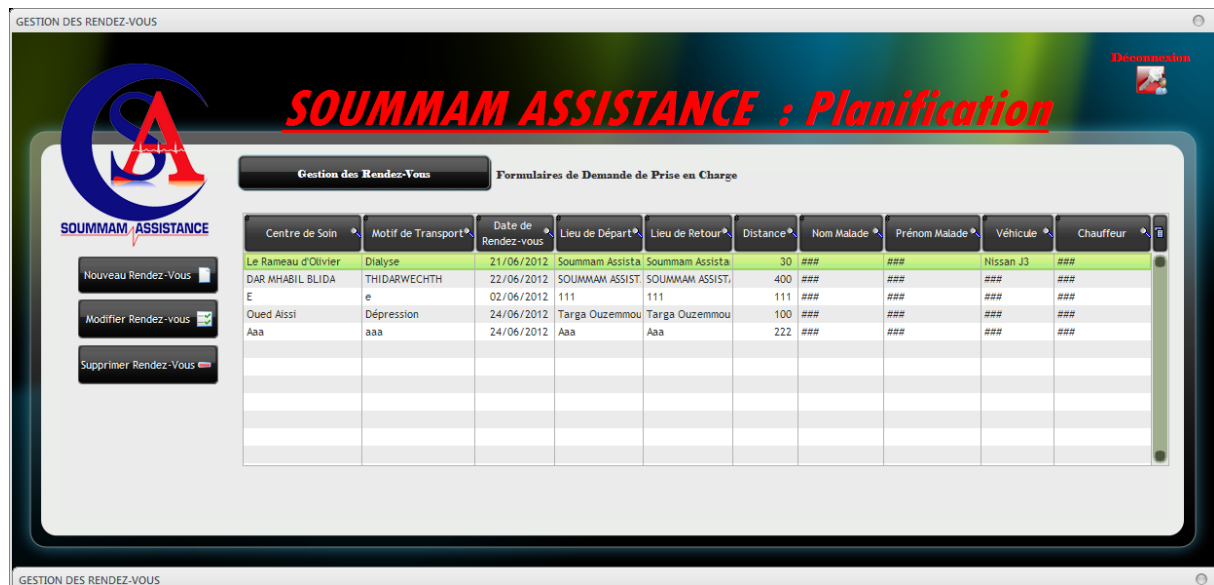


FIGURE 4.12 – Fenêtre "Planification des RDV"

4.7.5 Page Facturation

Cette page permet à l'administrateur d'établir e de gérer les factures des malades transportés.



FIGURE 4.13 – Fenêtre "Gestion des Factures"

4.8 Test

Cette activité consiste à tester les résultats de l'implémentation pour s'assurer du bon déroulement des fonctionnalités du système. Lors de l'évaluation des tests effectués, si nous détectons une anomalie quelconque, nous devrions la corriger.

4.8.1 Test du cas d'utilisation "Authentification"

Pour tester ce cas d'utilisation, nous avons rempli les champs spécifiques pour le login et le mot de passe avec un nom d'utilisateur existant déjà dans la base de données et après la validation, la session est ouverte.

Par la suite, nous avons tenté d'accéder à la même session mais avec un mot de passe différent, l'accès à la session est rejeté. (Idem pour un login inexistant.)

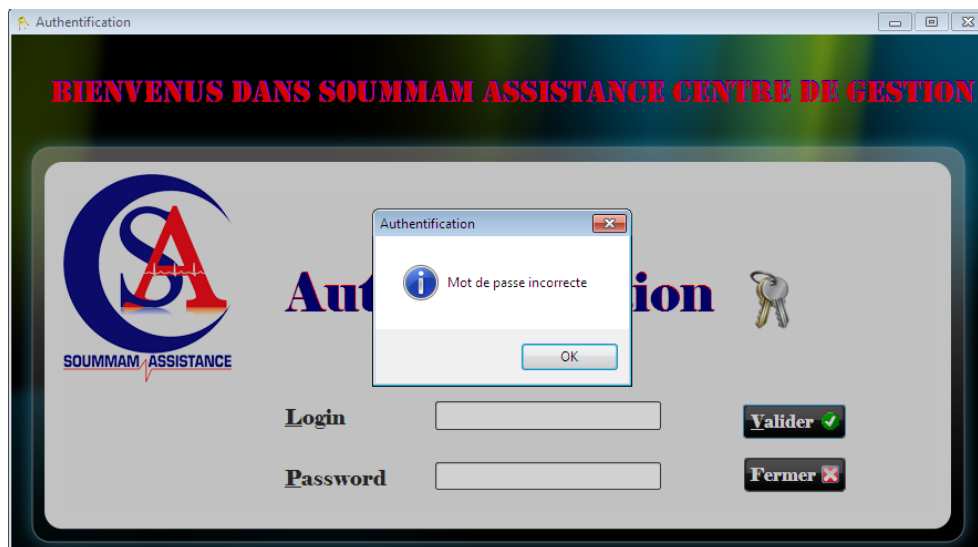


FIGURE 4.14 – Test de l'Authentification

4.8.2 Test du cas d'utilisation "Ajout d'un Assuré"

Nous avons saisi les informations d'un nouvel assuré dans l'interface approprié, ensuite nous avons vérifié leurs enregistrement dans la base de données. La vérification est réussie.



The screenshot shows a web form titled "AJOUTER ASSURÉ" in red. The form contains the following fields and values:


Field	Value
N° Sécurité Sociale	245 648 210
Nom	TAZIBET
Prenom	Fadila
Date de Naissance	06/06/2012
N° Téléphone	776 765 085
Agence Assurance	Béjaia
Centre de Paiement	Ihadadene

At the bottom of the form, there are two buttons: "Annuler" with a red 'X' icon and "Valider" with a green checkmark icon.

FIGURE 4.15 – Test de l’Ajout d’un Assuré

4.8.3 Test du cas d'utilisation "Modifier un Assuré"

Toujours sur le même Assuré, nous avons effectué des modifications sur ses informations. Et Après avoir consulté la base de données, nous avons constaté que les informations modifiées, on été prises en considération. Le test de modification est réussi.



The screenshot shows a web form titled "MODIFIER ASSURÉ" in red. The form contains the following fields and values:

Field	Value
N° Sécurité Sociale	542 254 879
Nom	ADALOU
Prenom	Amrane
Date de Naissance	09/04/1953
N° Téléphone	0
Agence Assurance	Béjaia
Centre de Paiement	Iheddanden

At the bottom of the form, there are two buttons: "Annuler" with a red 'X' icon and "Valider" with a green checkmark icon.

FIGURE 4.16 – Test de la modification d’un Assuré

4.8.4 Test du cas d'utilisation "Supprimer un Assuré"

Nous avons testé ce cas d'utilisation pour l'assuré ayant le numéro 1, qui a été établi et enregistré dans la base de données. Après avoir vérifié la base de données, nous avons trouvé l'inexistence de l'assuré, donc le teste est réussi.

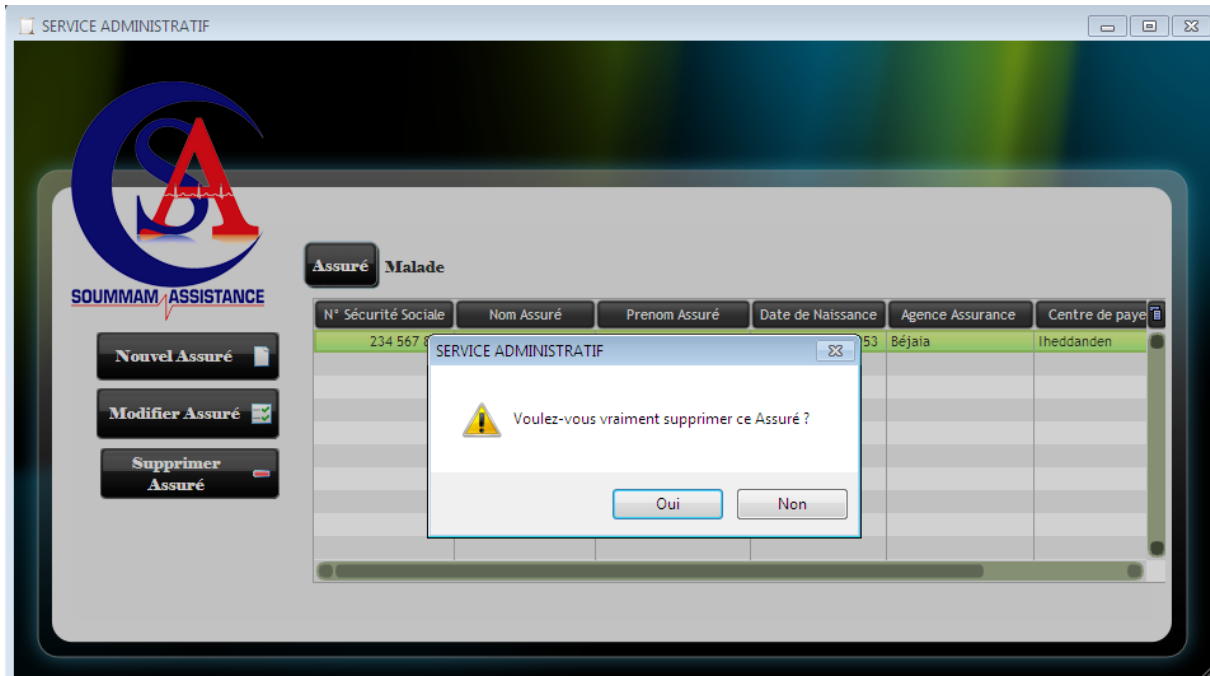


FIGURE 4.17 – Test de la suppression d'un Assuré

4.9 Perspective

Actuellement la maquette informatique développée pour la gestion du parc est limité à la sauvegarde des informations concernant les chauffeurs, véhicules et les séances d'entretiens effectuées sur ces derniers.

L'idée de départ était de garder trace de toute information concernant les véhicules de l'entreprise afin de faciliter leur gestion, mais nous envisageons d'automatiser la planification des séances d'entretiens effectuées sur les véhicules de sorte qu'un entretien quelconque puisse être programmé dans le futur.

Conclusion

Ce dernier chapitre de notre projet décrit la phase de réalisation. Nous y avons précisé en particulier les outils de développement utilisés pour l'application "SOUMMAM ASSISTANCE Centre de Gestion". Puis nous avons présenté les interfaces les plus importantes dans notre application.

Nous pensons que cette application répond à quelques règles de réalisation de projets qui consistent en la conception d'une base de données et la réalisation d'une interface.

Conclusion Générale

*"il ne faut pas dire que la science
est utile parce qu'elle nous permet de construire des machines.*

*il faut dire : les machines sont utiles
parce qu'elles nous permettent de faire de la science"*

**** Henri Poincaré***

L'objectif de notre projet de fin d'étude était de concevoir et développer un système d'information pour l'entreprise SOUMMAM ASSISTANCE de Béjaia.

Pour mener à bien notre mission, nous avons mis en œuvre toutes nos connaissances acquises durant notre cursus universitaire, ajouté à l'expérience bénéfique d'un stage pratique effectué au sein de l'entreprise SOUMMAM ASSISTANCE.

Nous avons fondé une bonne analyse de l'existant suivie d'une conception minutieuse de notre base de données et de nos interfaces, ce qui nous a permis de capturer tous les besoins du service administratif et du parc véhicules de l'entreprise afin de réaliser le travail demandé.

En outre, ce projet nous a offert la possibilité de revoir et de suivre avec beaucoup d'attention les méthodes fondamentales de la modélisation et de la conception des systèmes d'informations grâce auxquelles l'application a été mise au point.

Nous avons la certitude que le nouveau système développé sera bénéfique et apportera des améliorations significatives pour l'entreprise locale ainsi qu'aux autres unités implantées sur le territoire national.

Nous sommes d'autant plus satisfaits que la réalisation de ce projet lié au stage représente une opportunité pour nous de découvrir le monde professionnel, ses réalités, ses difficultés et ses engagements imposés.

Enfin, nous espérons que ce modeste travail servira de référence et de base à tous ceux qui voudront s'engager dans un travail homologue.

BIBLIOGRAPHIE

- [1] A. BUREAUD. *Qu'est-ce qu'une Interface*. [Http://www.olats.org](http://www.olats.org), 2004.
- [2] Télécharger Cours. *Réseaux & Télécoms*. [Http://www.TelechargerCours.com](http://www.TelechargerCours.com), 2012.
- [3] I. JACKOBSON & M. CRISTERSON. *Le Génie Logiciel Orienté Objet*. Addison Wesley, 2000.
- [4] S. CROZAT. *Conception de Bases de Données I*. UTC Formation, 2009.
- [5] D. GABAY & J. GABAY. *UML 2 Analyse et Conception*. DUNOD, 2007.
- [6] F. JULIARD. *UML Unified Method Language*. Journal Universitaire De Bretagne Sud UFR SSI-IUP Vannes, 2001-2002.
- [7] A. AITCHISON & A. MACHANIC. *Expert SQL Server 2008 Development*. Apress, 2008.
- [8] A. MEIER. *Introduction Pratique aux Bases de Données Relationnelles*. Springer, 1998.
- [9] MrProof. *L'architecture Client/Serveur*. [Http://www.Pratiquerlinformatique.com](http://www.Pratiquerlinformatique.com), 2011.
- [10] P. JONSSON & G. OVERGAARD. *Windev15, documentation version 15*. PC SOFT, 2010.
- [11] J.F. PILLIOU. *Tout sur les Réseaux et Internet*. DUNOD, 2006.
- [12] J. PINETA. *Reengineering des systèmes documentaires*. Organisations, 1986.
- [13] P. ROQUES. *UML 2 en pratique*. Eyrolles, 2006.
- [14] S.ALTER. *Management Perspective*. Benjamin Cummings Publishing Company, 1996.
- [15] F. BROUARD & SQLPro. *Normalisation des noms des objets des bases de données*. [Http://sqlpro.developpez.com](http://sqlpro.developpez.com), 2003.
- [16] Adobe Team. *ADOBE PHOTOSHOP CS3 Guide Utilisateur*. Adobe, 2007.
- [17] J. TISSEAU. *Initiation à l'algorithmique*. Presse Universitaire, École Nationale d'Ingénieurs de Brest, 2009.
- [18] P. ROQUES & F. VALLEE. *UML 2 en action*. Eyrolles, 2004.