

Conception et réalisation d'une application de suivi de patients dans un établissement hospitalier

Table des matières	i
Liste des figures	iii
Liste des tableaux	y
Introduction générale	1
1 Présentation de l'organisme d'accueil	3
1.1 Historique	3
1.2 Activité	3
1.3 Plan hospitalier	4
1.4 Effectif de l'hôpital	4
1.5 Mission de l'organisme d'accueil	4
1.6 Organigramme général	6
1.7 Présentation du champ d'étude	7
1.8 Organigramme fonctionnel du champ d'étude	7
1.9 Situation informatique	7
1.10 Problématiques et objectifs	8
2 Analyse et Conception	10
2.1 Introduction	10
2.2 Spécification des besoins	10
2.3 Présentation de l'UML	11
2.3.1 Définition	11
2.3.2 Diagramme de cas d'utilisation	11
2.3.3 Diagramme de collaboration	14

2.3.4	Diagramme de séquence	17
2.3.5	Diagramme d'activité	21
2.3.6	Diagramme de classes	26
2.3.7	Dictionnaire de données	29
2.3.8	Le modèle relationnel	33
2.4	Conclusion	34
3	Réalisation	35
3.1	Introduction	35
3.2	Outils de développement	35
3.2.1	Implémentation de la base de données	35
3.2.2	Environnement de développement	36
3.3	Description de l'application	37
3.3.1	Menu de l'application	38
3.3.2	Les interfaces de l'application	40
3.3.3	Les imprimés	47
3.4	Description des requêtes	50
3.4.1	La liste des patients admis	50
3.4.2	La liste des gardes patients	50
3.4.3	La liste des services	50
3.4.4	La liste des nouveaux-nés	50
3.4.5	La liste des patients traités	50
3.4.6	La sortie par évacuation d'un patient	51
3.4.7	La sortie normale (guérison) d'un patient	51
3.4.8	Recherche d'un patient par numéro	51
3.4.9	Compteur de taille des Bases De Données	51

3.5 Conclusion 52

Conclusion générale et perspectives 53

Bibliographie 54

LISTE DES FIGURES

1.1 Organigramme général de l'hôpital d'Akbou 6

1.2 Organigramme fonctionnel du champ d'étude 7

2.1 Diagramme de cas d'utilisation 13

2.2 Diagramme de collaboration de l'authentification 14

2.3 Diagramme de collaboration de l'ajout 15

2.4 Diagramme de collaboration de la suppression 15

2.5 Diagramme de collaboration de la modification 16

2.6 Diagramme de collaboration de la recherche 16

2.7 Diagramme de collaboration de l'impression 17

2.8 Diagramme de séquence du cas d'utilisation "Authentification" . . . 18

2.9 Diagramme de séquence du cas d'utilisation "Ajout" 18

2.10 Diagramme de séquence du cas d'utilistaion "Suppression" 19

2.11 Diagramme de séquence du cas d'utilistaion "Modification" 20

2.12 Diagramme de séquence du cas d'utilistaion "Recherche" 20

2.13 Diagramme de séquence du cas d'utilistaion "Impression" 21

2.14 Diagramme d'activité de l'authentification 22

2.15 Diagramme d'activité d'ajout 23

2.16 Diagramme d'activité de modification 24

2.17 Diagramme d'activité de suppression 25

2.18 Diagramme d'activité de recherche 26

2.19 Diagramme de classes 32

3.1	Interface de Borland Delphi 7	37
3.2	Menu de l'application (partie 1)	38
3.3	Menu de l'application (partie 2)	39
3.4	Formulaire d'authentification	40
3.5	La fenêtre du menu principal	41
3.6	Le bouton "Nouveau > Admission> Patient"	41
3.7	Le bouton "Nouveau> Sortie"	42
3.8	Le bouton "Modifier"	42
3.9	Le bouton "Supprimer"	42
3.10	Le bouton "Affichage"	43
3.11	Le bouton "Recherche"	43
3.12	Le bouton "Statistiques"	43
3.13	Le bouton "Imprimer"	44
3.14	Le bouton "Outils"	44
3.15	Le bouton "Aide"	44
3.16	Le formulaire d'admission	45
3.17	Le formulaire de sortie	46

Liste des figures

LISTE DES TABLEAUX

1.1	Tableau des moyens du bureau des entrées	8
2.1	Méthodes et attributs des classes	28
2.2	Dictionnaire de données (partie 1)	29
2.3	Dictionnaire de données (partie 2)	30
2.4	Dictionnaire de données (partie 3)	31

Introduction générale

A

ACTUELLEMENT, le monde connaît une avance technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement de l'entreprise et d'autres établissements.

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

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

Les hôpitaux font partie intégrante des établissements que l'informatique pourra beaucoup aider. En effet, la croissance de la population hospitalière nécessite la mise en place d'une gestion rationnelle prise et rapide, or et jusqu'à ce jour, la manière de gérer manuellement est encore dominante d'où la nécessité d'introduire l'informatique dans les administrations hospitalières.

L'objectif de notre projet présenté dans ce rapport est la conception et la réalisation d'une application monoposte simple de gestion des entrées/sorties des patients dans un établissement hospitalier. Pour ce faire, nous avons été affectés au sein du bureau des entrées de l'établissement public hospitalier AKLOUL Ali d'Akbou.

Introduction générale

Nous avons organisé ce mémoire de la façon suivante :

Le premier chapitre présente l'établissement d'accueil à savoir l'Établissement Public Hospitalier AKLOUL Ali d'Akbou et notre champ d'étude.

Le deuxième chapitre présente la conception de notre système d'information que nous allons modéliser avec le langage UML.

La réalisation et l'implémentation de notre application fera l'objet du troisième chapitre dans lequel nous illustrerons les différentes parties de l'application à savoir la base de données et les différentes requêtes qui permettent l'accès à celle-ci.

Enfin, nous terminerons ce document par une conclusion générale.

1.10 Problématiques et objectifs

Problématiques

Pour détecter les problèmes existants, nous avons interrogé le personnel du bureau des entrées de l'hôpital d'Akbou et il nous a cité quelques anomalies, mais pour localiser leur source, nous

nous sommes mis en pratique avec lui et après une observation continuelle, nous avons pu recenser les insuffisances suivantes :

- Volume important des informations traitées manuellement, ce qui provoque parfois des erreurs dans l'établissement des documents.
- Recherche difficile sur les registres qui engendre une perte de temps.
- Insécurité des informations.
- Possibilité d'erreur dans le remplissage des différents documents et registres.
- Possibilité d'erreur dans les calculs des statistiques.
- Nombre important des archives qui engendre une difficulté de stockage.
- Détérioration des archives à force de leur utilisation trop fréquente.
- Mauvaise codification sur quelques objets dans la gestion d'information.

Objectifs

Afin d'y remédier à tous ses problèmes, nous avons assigné à notre étude les objectifs suivants :

- Rapidité dans l'établissement des différents documents.
- Facilité de la recherche et l'accès aux informations.
- Stockage des informations sur des supports informatiques ce qui assurera leur sécurité.
- Gain de temps dans les calculs des statistiques.
- Automatiser les tâches qui se traitent manuellement.
- Proposer une bonne codification.

Chapitre : 2

Analyse et Conception

2.1 Introduction

Cette partie est consacrée aux étapes fondamentales pour le développement de notre système de gestion d'un patient hospitalisé. Pour la conception et la réalisation de notre application, nous avons choisis de modéliser avec le formalisme UML (Unified Modeling Language) qui offre une flexibilité marquante qui s'exprime par l'utilisation des diagrammes.

2.2 Spécification des besoins

C'est une étape primordiale au début de chaque démarche de développement. Son but est de veiller à développer un logiciel adéquat, sa finalité est la description générale des fonctionnalités du système, en répondant à la question : Quelles sont les fonctions du système?

Notre système doit répondre aux exigences suivantes :

- Le système doit pouvoir récupérer des informations de chaque entité à partir de son matricule pour mettre à jour la base des données de l'application.
- L'insertion des patients et d'autres entités et les orienter vers une salle d'un service quelconque.
- Modification des informations à propos du patient et des autres entités.
- La suppression.
- L'impression des documents comme (bulletin d'admission, billet de salle, certificat de séjour, déclaration de décès ...etc.).
- Calcul de statistiques : le nombre de nouveau-nés, le nombre de décès, le nombre d'accidentés, nombre de lits libres, . . .etc.

2.3 Présentation de l'UML

2.3.1 Définition

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. Il véhicule en particulier :

- Les concepts des approches par objets : classe, instance, classification, etc.
- Intégrant d'autres aspects : associations, fonctionnalités, événements, états, séquences, etc.

UML définit neuf types de diagrammes devisés en deux catégories:

1. Diagrammes statiques (structurels) : diagramme de classe, d'objet, de composant, de déploiement et de diagramme de cas d'utilisation.
2. Diagrammes dynamique (comportementaux) : diagramme d'activité, de séquence, d'état-transition et de diagramme de collaboration. [PAM05]

Pour la modélisation des besoins, nous utilisons les diagrammes UML suivant : Diagramme de cas d'utilisation, diagramme de séquence, diagramme de collaboration et diagramme d'activité.

2.3.2 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un graphe d'acteurs, un ensemble de cas d'utilisation englobés par la limite du système, des associations de communication entre les acteurs et les cas d'utilisation, et des généralisations entre cas d'utilisation. [NK01]

Il est destiné à représenter les besoins des utilisateurs par rapport au système. [GAB04]

Identification des acteurs

Les acteurs d'un système sont les entités externes à ce système qui interagissent avec lui. Dans notre application, le seul acteur qui interagit avec le système est l'agent de saisie du bureau des entrées.

Identification des cas d'utilisations

Un cas d'utilisation est utilisé pour définir le comportement d'un système ou la sémantique de toute autre entité sans révéler sa structure interne. Chaque cas d'utilisation spécifie une séquence d'action, y compris des variantes, que l'entité réalise, en interagissant avec les acteurs de l'entité. La responsabilité d'un cas d'utilisation est de spécifier un ensemble d'instances, où une instance de cas d'utilisation représente

une séquence d'actions que le système réalise et qui fournit un résultat observable par l'acteur. [NK01]

Voici les cas d'utilisation de notre application :

- Authentification : l'application vérifie que l'utilisateur est bien ce qu'il prétend être et lui donne ensuite l'autorisation d'accès .
- Ajout : pouvoir ajouter des nouveaux patients, nouveau-nés, . . .etc.
- Modification : sert à modifier l'information dans la base de données
- Recherche : rechercher des informations sur un patient, un nouveau-né . . .etc. pour pouvoir se renseigner ou renseigner les visiteurs.
- Imprimer :

Bulletins propre aux patients: (bulletin d'admission, billet de salle, certificat de séjour, certificat de présence et déclaration de décès).

Pour le garde-patient : (billet de salle).

Pour la naissance : (billet de salle, déclaration de naissance).

- Calcul des statistiques : nombre de nouveau-nés, nombre des accidentés, moyenne des décès et des naissances par mois .. etc.

D'où la présentation de notre diagramme de cas d'utilisation (FIG. 2.1)

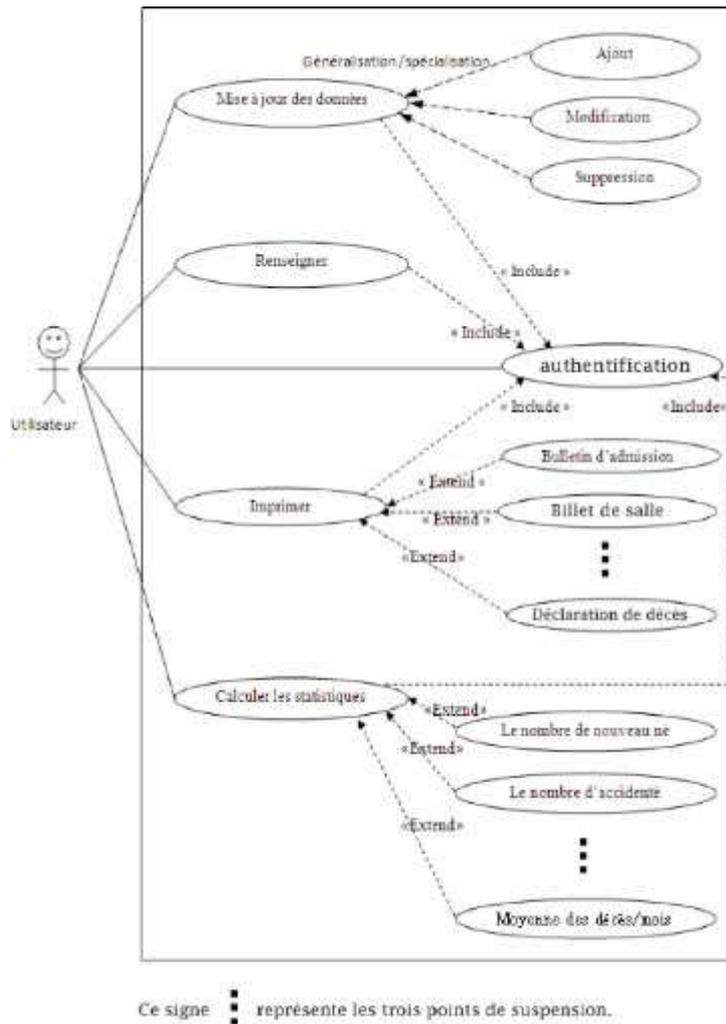


FIG. 2.1 Diagramme de cas d'utilisation

2.3.3 Diagramme de collaboration

Un diagramme de collaboration montre une interaction organisée autour d'un ensemble d'objets et de leurs liens. En revanche, un diagramme de collaboration ne montre pas le temps dans une dimension séparée; ainsi la séquence des messages et les fils concurrents doivent être déterminés en utilisant les numéros de séquence. [NK01]

C'est une autre représentation des scénarios des cas d'utilisation qui met plus l'accent sur les objets et les messages échangés. [GAB04]

Les figures 2.2 à 2.7 représentent les diagrammes de collaboration du cas d'utilisation.

Diagramme de collaboration d'authentification

Ce diagramme décrit les messages échangés entre les différents objets pour montrer le fonctionnement de l'opération d'authentification : l'utilisateur saisit le mot de passe puis le système vérifie sa validité, ensuite c'est le système qui retourne la page d'accueil de l'application à l'utilisateur.

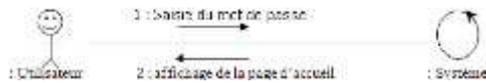


FIG. 2.2 Diagramme de collaboration de l'authentification

Diagramme de collaboration d'ajout

Ce diagramme illustre la façon avec laquelle l'opération d'ajout d'une information (nouveau patient, nouveau garde-patient, naissance.. etc.) s'effectue.

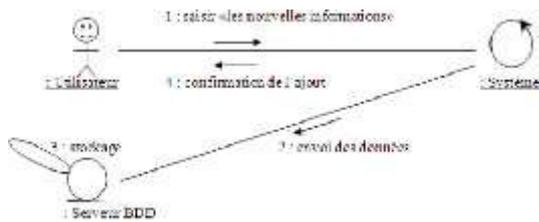


FIG. 2.3 Diagramme de collaboration de l'ajout

Diagramme de collaboration de Suppression

Ce diagramme nous montre les déferents messages entre les objets intervenant dans la suppression d'une donnée : l'utilisateur choisit l'information à supprimer puis le confirme la suppression et enfin la donnée sera supprimée au niveau de la base de données.

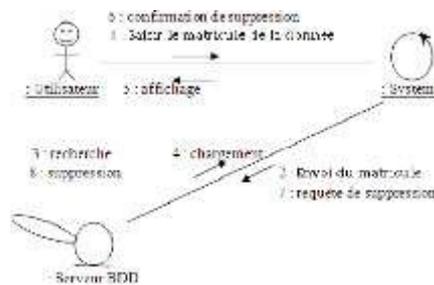


FIG. 2.4 Diagramme de collaboration de la suppression

Diagramme de collaboration de Modification

Ce diagramme montre comment modifier une donnée : l'utilisateur demande la modification en saisissant le matricule de la donnée et le système recherche cette dernière dans la base de données et l'affiche à l'utilisateur qui va la modifier et l'enregistrer, le système la stocke ensuite dans la base de données.

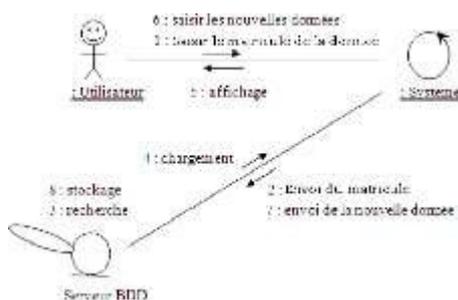


FIG. 2.5 Diagramme de collaboration de la modification

Diagramme de collaboration de Recherche

Pour la recherche (renseignement), il suffit que l'utilisateur saisis l'une de ces informations : numéro du patient, nom du patient ou date d'admission, dans le formulaire de recherche et le système effectue une recherche au niveau de la base de données pour lui afficher le résultat.

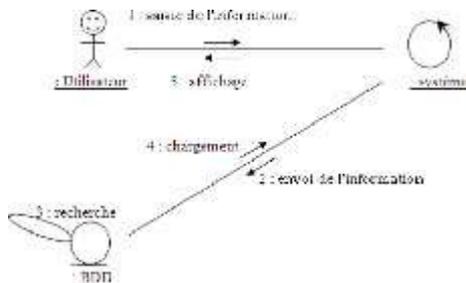


FIG. 2.6 - Diagramme de collaboration de la recherche

Diagramme de collaboration d'Impression

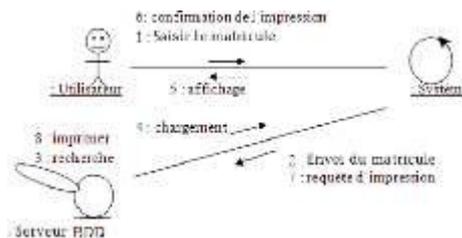


FIG. 2.7 Diagramme de collaboration de l'impression

2.3.4 Diagramme de séquence

Il permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets. [GAB04] Un **diagramme de séquence** montre une interaction présentée en séquence dans le temps. En particulier, il montre aussi les objets qui participent à l'interaction par leur "ligne de vie" et les messages qu'ils échangent présentés en séquence dans le temps. [NK01]

Voici quelques notions de base du diagramme : [DUM08]

- Scénario : une liste d'actions qui décrivent une interaction entre un acteur et le système.
- Interaction : un comportement qui comprend un ensemble de messages échangés par un ensemble d'objets dans un certain contexte pour accomplir une certaine tâche.
- Message : Un message représente une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une réaction chez le récepteur.

Les figures 2.8 à 2.13 représentent les diagrammes de séquence des cas d'utilisation.

Diagramme de séquence du cas d'utilisation "authentification"

1. l'utilisateur demande le formulaire d'authentification.
2. L'application affiche le formulaire d'authentification.
3. L'utilisateur saisit le mot de passe.
4. Le système vérifie la validité du mot de passe.
5. L'application affiche la page d'accueil.

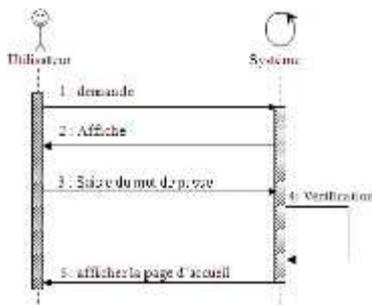


FIG. 2.8 Diagramme de séquence du cas d'utilisation "Authentification"

Diagramme de séquence du cas d'utilisation "Ajout"

1. L'utilisateur demande le formulaire d'ajout.
2. L'application affiche le formulaire d'ajout.
3. L'utilisateur saisit les nouvelles données.
4. L'application envoie la requête.
5. L'application stocke les données au niveau de la base de données.
6. L'application confirme l'enregistrement.

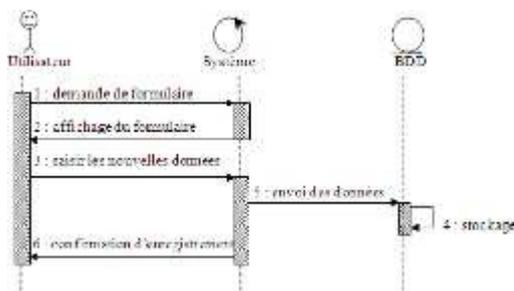


FIG. 2.9 - Diagramme de séquence du cas d'utilisation "Ajout"

Diagramme de séquence du cas d'utilisation "Suppression"

1. L'utilisateur demande le formulaire de suppression.
2. L'application affiche le formulaire de suppression.
3. L'utilisateur saisit le matricule de l'information à supprimer.
4. L'application demande la recherche à la base de données.
5. Une procédure de recherche se fera au niveau de la base de données.
6. Chargement de la donnée à partir de la BDD (Base De Données) vers l'application.
7. L'application affiche la donnée.
8. L'utilisateur confirme la suppression.
9. Au niveau de la BDD la donnée sera supprimée.

FIG. 2.10 - Diagramme de séquence du cas d'utilisation "Suppression"

Diagramme de séquence du cas d'utilisation "Modification"

1. L'utilisateur demande la modification.
2. L'application affiche le formulaire de modification.
3. L'utilisateur saisit le matricule de la donnée à modifier.
4. L'application envoie le matricule de la donnée à la BDD.
5. Une fonction de recherche se fait au niveau de la BDD.
6. La BDD charge la donnée demandée vers l'application.
7. L'application affiche la donnée demandée à l'utilisateur.
8. L'utilisateur saisit les nouvelles données.
9. L'application envoie les nouvelles données à la BDD.
10. Au niveau de la BDD se fait le stockage.
11. Confirmation de la modification.

FIG. 2.11 - Diagramme de séquence du cas d'utilistaion "Modification"

Diagramme de séquence du cas d'utilisateur "Recherche"

1. L'utilisateur demande le formulaire de renseignement.
2. L'application affiche le formulaire.
3. L'utilisateur saisit le matricule de la donnée.
4. L'application envoi le matricule de la donnée à la BDD.
5. Une fonction de recherche se fait au niveau de la BDD.
6. La BDD charge la donnée envers l'application.
7. L'application Affiche la donnée.

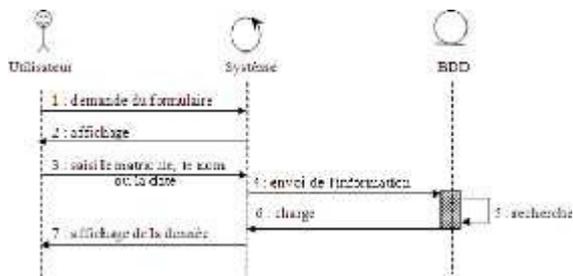


FIG. 2.12 - Diagramme de séquence du cas d'utilistaion "Recherche"

Diagramme de séquence du cas d'utilisation "Impression"

1. L'utilisateur demande l'impression.
2. L'application affiche le formulaire d'impression.
3. L'utilisateur saisit le matricule de la donnée et soumet la requête.
4. L'application consulte la BDD.
5. Une recherche se fera au niveau de la BDD.
6. Le formulaire se fera charger à partir de la BDD.
7. L'application affiche le formulaire à l'utilisateur.
8. L'utilisateur confirme l'impression.
9. L'application envoi la requête à la BDD.
10. La fonction d'impression se fera à partir de la base de données.

11. Le formulaire ou le bulletin est imprimé.

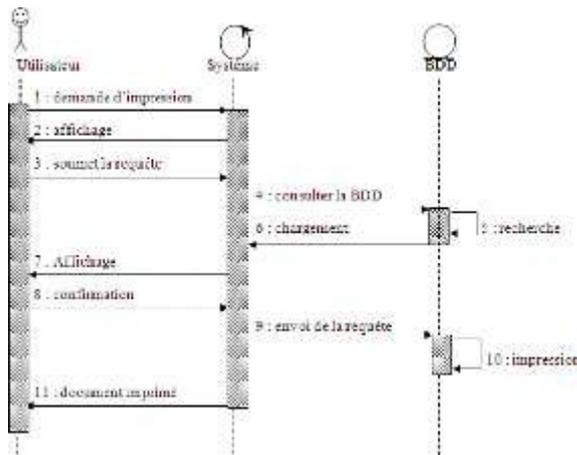


FIG. 2.13 Diagramme de séquence du cas d'utilisation "Impression"

2.3.5 Diagramme d'activité

Il donne une vision des enchaînements des activités propre à une opération ou à un cas d'utilisation. [GAB04]

Le diagramme d'activité est attaché à une catégorie de classes et décrit le **déroulement des** activités de cette catégorie. Le déroulement s'appelle "flot de contrôle". Il indique la part prise par chaque objet dans l'exécution d'un travail. Il sera enrichi par les conditions de séquence. [J.S03]

Les figures 2.14 à 2.18 présentent les diagrammes d'activités du cas d'utilisation. Diagramme d'activité de l'authentification

Le diagramme d'activité d'authentification nous permet de voir les comportements internes du système, lors du démarrage de l'application par l'utilisateur, le système lui affiche le formulaire d'authentification, après que le mot de passe soit saisi le système vérifie sa validité et affiche la page d'accueil sinon il affiche un message d'erreur.

FIG. 2.14 Diagramme d'activité de l'authentification

Diagramme d'activité d'ajout

Après une demande d'ajout d'une donnée par l'utilisateur (patient, garde- **patient**, **naissance**), le système lui affiche le formulaire d'ajout pour qu'il puisse saisir ces données et confirmer leur enregistrement au niveau de la base de données.

FIG. 2.15 Diagramme d'activité d'ajout

Diagramme d'activité de modification

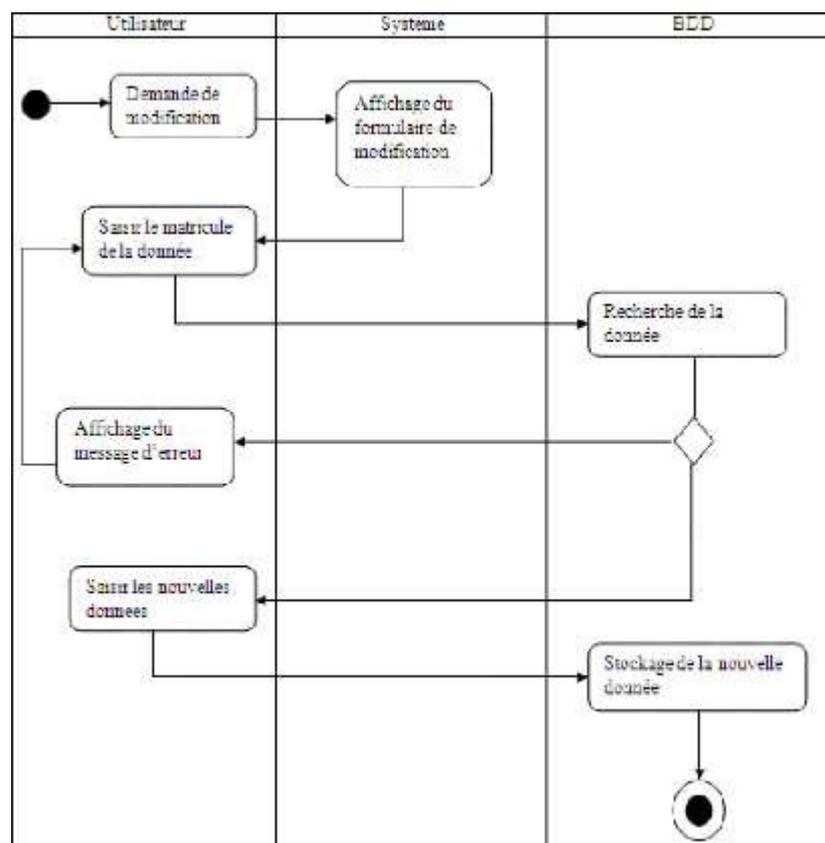


FIG. 2.16 Diagramme d'activité de modification

Diagramme d'activité de suppression

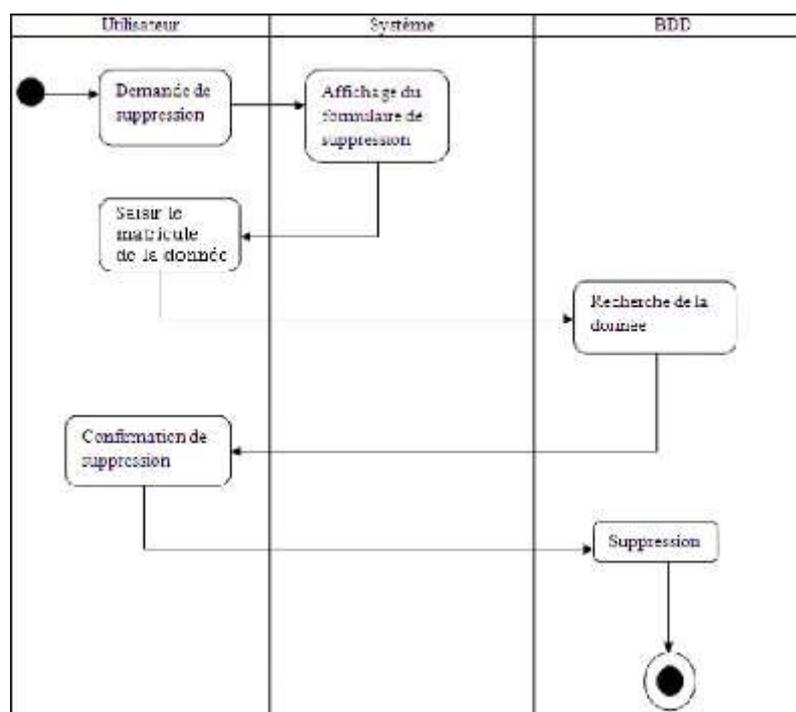


FIG. 2.17 Diagramme d'activité de suppression

Diagramme d'activité de recherche

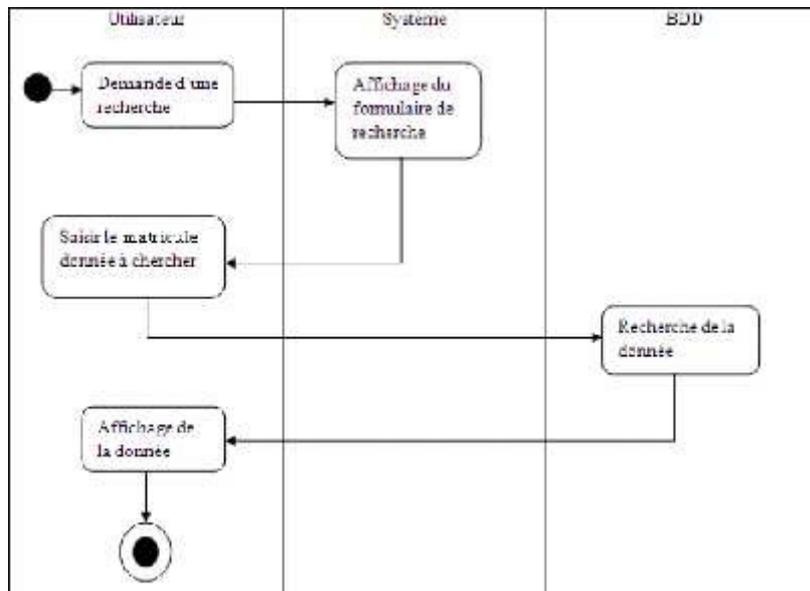


FIG. 2.18 Diagramme d'activité de recherche

2.3.6 Diagramme de classes

C'est une collection d'éléments de modèle statique, tels que des classes, des interfaces et leurs relations, connectés entre eux comme un graphe. [NK01]

Il 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. [GAB04]

Identification des classes

Une classe est une description d'un groupe d'objets partageant un ensemble commun de propriétés (les attributs), de comportements (les opérations) et de relations avec d'autres objets (les associations et les agrégations). [LAT01]

Une classe contient : [Sca05]

Des attributs (ou champs, ou variables d'instances) : Les attributs d'une classe décrivent la structure de ses instances (les objets).

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

Une agrégation est une association correspondant à une relation qui lorsqu'elle est lue dans un sens signifie "est une partie de" et lorsqu'elle est lue dans l'autre sens elle signifie "est composé de".

Les classes sur lesquelles se porte notre application sont les suivantes :

Patient :c'est la classe la plus essentielle de notre application; toute personne entrant dans l'hôpital ayant droit à l'hospitalisation.

Mod_adm: Une instance de cette classe est la façon dont le patient est entré à l'hôpital (entré normale, naissance, accident).

Salle/Unité/Service : Une instance de cette classe est l'emplacement où le patient séjournera.

Personne : Une instance de cette classe représente la personne qui accompagnera le patient lors de son entré à l'hôpital ou un garde malade qui pourra garder le patient tout au long de son séjour à l'hôpital.

- **Personnel _m** : Cette classe représente le personnel médical dont les médecins, les chirurgiens.. .etc. C'est-à-dire toutes les personnes qui agissent ou suivent le patient tout au long de son séjour à l'hôpital.

Traitement : Une instance de cette classe représente les traitements et les remèdes que le patient peut prendre durant son hospitalisation.

- **Mod _sortie** : Une instance de cette classe représente la manière dont le patient va sortir de l'hôpital (Sortie par guérison, par évacuation, ou bien par décès).

Naissance : Cette classe est la classe des nouveaux-nés.

Nous définissons les méthodes et les attributs de chaque classe comme suit :

Classe	Attribut	Méthode
Patient	N_pat, Nom_pat, Prn_pat, Sexe_pat, Dn_pat, Ln_pat, Adr_pat, Prn_per_pat, Nom_mer_pat, Prn_mer_pat, Nat_pat, Sit_fam_pat, Cont_pat, Tel_cont, Adr_cont, Nom_prn_acc, Sexe_acc, Age_acc, Lien_fam_acc, Tel_acc, Nom_epou_pat	Ajouter(), modifier(), supprimer()
Mod_adm	Cod_adm, Nom_adm,	
Salle	N_salle, N_lit	
Unité	Cod_unite	
Service	Cod_service, Nom_service	
Garde_patient	Mat_gp, Nom_gp, Prn_gp, Sexe_gp, Age_gp, Lien_fam_gp, Typ_pid, N_pid, Dat_pid, Lieu_pid	

Personnel_m	Mat_per_m, Statut_per_m, Nom_per_m, Prn_per_m	Ajouter(), modifier(), supprimer()
Traitement	Cod_tr, Nom_tr	
Mod_sortie	Cod_sortie, Nom_sortie,	
Naissance	N_naiss, Dat_naiss, Heure_naiss, Etat_naiss, Sexe_naiss, Poid_naiss, Prn_naiss	Ajouter(), modifier(), supprimer()

Tab. 2.1 Méthodes et attributs des classes

2.3.7 Dictionnaire de données

Codification	Désignation	Type	Taille	Observation
N_pat	Numéro du patient	Entier	6	xxxxxx
Nom_pat	Nom du patient	Caractère	20	
Prn_pat	Prénom du patient	Caractère	20	
Sexe_pat	Sexe du patient	Caractère	10	
Dn_pat	Date de naissance	date	8	jj/mm/aaaa
Ln_pat	Lieu de naissance	Caractère	20	
Prn_père_pat	Prénom du père du patient	Caractère	20	
Nom_mer_pat	Nom de la mère du patient	Caractère	20	
Prn_mer_pat	Prénom de la mère du patient	Caractère	20	
Nat_pat	Nationalité du patient	Caractère	10	
Adr_pat	Adresse du patient	Caractère	50	
Sit_fam_pat	Situation familiale	Caractère	10	
Nom_epou_pat	Nom de l'époux	Caractère	20	
	du patient			
Nom_prn_acc	Nom de l'accompagnateur	Caractère	20	
Sexe_acc	Sexe de l'accompagnateur	Caractère	10	
Lien_fam_acc	Lien familiale de l'accompagnateur avec le patient	Caractère	10	
Cont_pat	Contact du patient	Caractère	20	
Tel_cont	Téléphone du contact	entier	10	
Adr_cont	Adresse du contact	Caractère	50	
Mat_gp	Matricule de la personne	Alpha-numérique	5	xxxxx
Nom_gp	Nom du garde-patient	Caractère	20	

Prn_gp	Prénom du garde-patient	Caractère	20	
Sexe_gp	Sexe du garde-patient	Caractère	10	
Age_gp	Age du garde-patient	entier	2	

Tab. 2.2 Dictionnaire de données (partie 1)

Codification	Désignation	Type	Taille	Observation
Lien_fam_gp	Lien familiale du garde- patient avec le patient	Caractère	10	
Typ_pid	Type de pièce d'identité du garde-patient	Caractère	10	
N_pid	Numéro de la pièce d'identité du garde-patient	Entier	8	
Dat_pid	Date de délivrance de la pièce d'identité du garde-patient	Date	8	jj/mm/aaaa
Lieu_pid	Lieu de délivrance de la pièce d'identité du garde-patient	Caractère	20	
Cod_adm	Code d'admission	Caractère	10	
Nom_adm	Mode d'admission	Caractère	10	
Dat_adm	Date d'admission	Date	8	jj/mm/aaaa
Heure_adm	Heure d'admission	Time	4	hh :mm
Cod_sortie	Code de sortie	Alpha-numérique	3	xxx
Nom_sortie	Mode de sortie	Caractère	10	
Dat_sortie	Date de sortie	Date		jj/mm/aaaa
Heure_sortie	Heure de sortie entier	Time	4	hh :mm
Lieu_evac	Nom de l'époux	Caractère	20	
Nom_tr	Diagnostic de sortie	Caractère	20	
Cod_service	Code du service	Alpha-numérique	2	xx
Nom_service	Nom du service	Caractère	20	
Cod_unite	Code de l'unite	Entier	2	xx
Nom_unite	Nom de l'unite	Caractère	20	
N_salle	Numéro de la salle	Entier	2	xx
N_lit	Numéro du lit	Entier	2	xx
Cod_tr	Code du médicament	Alpha-	6	xxxxxx

		numérique		
--	--	-----------	--	--

Tab. 2.3 Dictionnaire de données (partie 2)

Codification	Désignation	Type	Taille	Observation
Mat_per_m	Matricule du personnel médical	Alpha- numérique	5	xxxxx
Statut_per_m	Statut du personnel médical	Caractère	10	
Nom_per_m	Nom du personnel médical	Caractère	20	
Prn_per_m	Prénom du personnel médical	Caractère	20	
N_naiss	Numéro de naissance	Entier	5	xxxxx
Dat_naiss	Date de naissance	Date		jj/mm/aaaa
Heure_naiss	Heure de naissance	Time	xx :xx	
Etat_naiss	État de naissance	Caractère	10	
Sexe_naiss	Sexe du nouveau-né	Caractère	10	
Poid_naiss	Poids de naissance	Entier	4	En grammes
Prn_naiss	Prénom du nouveau-né	Caractère	10	

Tab. 2.4 - Dictionnaire de données (partie 3) Identification des relations

- Chaque patient est admis dans un seul mode d'admission et chaque mode d'admission contient un ou plusieurs patients.
- Un patient est affecté à une seule salle qui se trouve dans une unité d'un service bien déterminé et une seule salle contient un ou plusieurs patients limités.
- Chaque patient peut être accompagné par une personne et cette personne peut être un garde patient.
- Le personnel médical suit un ou plusieurs patients et un patient est suivi par un ou plusieurs agents du personnel médical.
- Le personnel médical est obligatoirement affecté dans un seul service et un service contient un ou plusieurs agents du personnel médical.
- Un patient prend un ou plusieurs médicaments et un médicament est pris par un ou plusieurs patients.
- Un patient doit être sorti par un mode de sortie et un mode de sortie est pris par un ou plusieurs patients.

A partir du dictionnaire de données et des règles de gestion, nous avons pu construire le diagramme de classes :

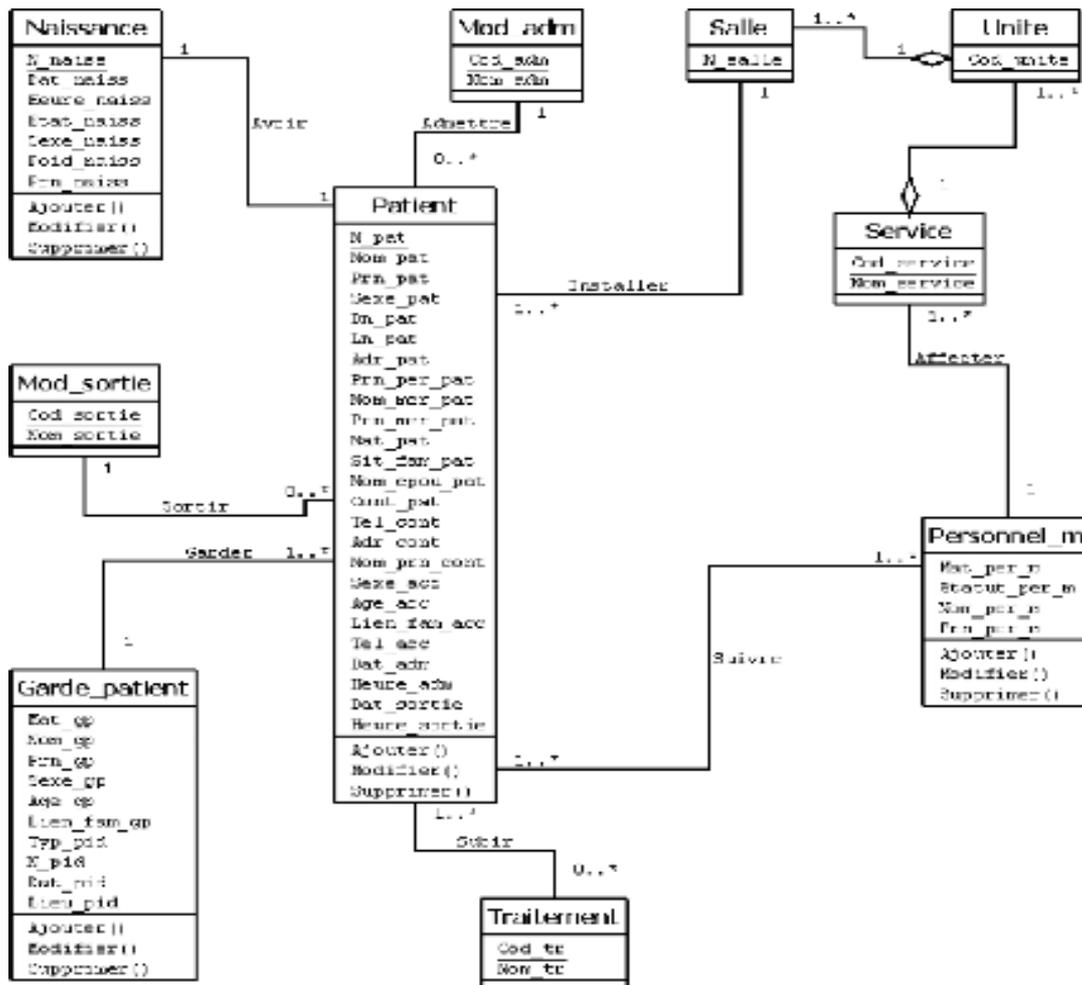


FIG. 2.19 Diagramme de classes

2.3.8 Le modèle relationnel

Du modèle conceptuel au modèle relationnel

A partir de la description conceptuelle que nous avons effectuée, on peut réaliser le modèle relationnel; vu que le système d'information ne peut pas le manipulé directement; et ça en utilisons des règles de passages de l'UML vers le relationnel.

Quelques notions essentielles

- Domaine : c'est l'ensemble des valeurs d'un attribut. [ETI09]
- Relation : c'est un sous ensemble du produit cartésien d'une liste de domaines. C'est en fait un tableau à deux dimensions dont les colonnes correspondent aux Domaines et dont les lignes contiennent des tuples. On associe un nom à Chaque colonne. [ETI09]
- Attribut : c'est une colonne d'une relation, caractérisé par un nom. [ETI09]
- Tuple : c'est la liste des valeurs d'une ligne d'une relation. [ETI09]

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

· L'arité : est le nombre d'attributs d'une relation. [M.C04]

· Clé : On distingue deux types de clés:

Clé primaire : ensemble d'attributs dont les valeurs permettent de distinguer les n-uplets les uns des autres (notion d'identifiant).

Clé étrangère : Attribut qui est clé primaire d'une autre entité.

NB : pour la notation, nous avons choisi de mettre en gras les clés primaires et de mettre * à la fin de chaque clé étrangère.

Les règles de passage

1. Transformation des classes : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé. [C.S02]

Transformation des associations : Nous distinguons trois familles d'associations

2. 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. [C.S02]

3. Association *.* et n-aire et classes-association : la classe-association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association. [C.S02]

4. Association 1.. 1 : 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 de la classe connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule. [C.S02]

En appliquant ces règles de transformation d'un diagramme de classe vers un modèle relationnel, nous avons aboutit au schéma relationnel suivant :

· Patient (N_pat, Nom_pat, Prn_pat, Sexe_pat, Dn_pat, Ln_pat, Prn_per_pat, Nom_mer_pat, Prn_mer_pat, Nat_pat, Adr_pat, Sit_fam_pat, Cont_pat, Tel_cont, Adr_cont, Nom_prn_acc, Sexe_acc, Age_acc, Lien_fam_acc, Tel_acc, Nom_epou_pat, Dat_adm, Heure_adm, Dat_sortie, Heure_sortie, Cod_adm*, N_salle*, Nom_sortie*)

· Personnel_m (Mat_per_m, Statut_per_m, Nom_per_m, Prn_per_m, Cod_service*)

· Garde_patient (Mat_gp, Nom_gp, Prn_gp, Sexe_gp, Age_gp, Lien_fam_gp, Typ_pid_gp, N_pid, Dat_pid, Lieu_pid, Dat_adm_gp, Heure_adm_gp, Dat_sortie_gp, Heure_sortie_gp, N_pat*)

· Mod_adm (Cod_adm, Nom_adm)

- Mod_sartie (Cod _sortie, Nom_sortie)
- Traitement (Cod _tr, Nom_tr)
- Service (Cod_service, Nom_service)
- Unite (Cod_unite, Cod_service*)
- Salle (N_salle, N_lit, Cod_unité*)
- Naissance (N _naiss, Dat_naiss, Heure_naiss, Etat_naiss, Sexe_naiss, Poid_naiss, Prn_naiss, N_pat*)
- Suivre (Mat _per _m, N _pat)
- Subir (Cod _tr, N_pat)

2.4 Conclusion

Dans ce chapitre, nous avons pu concevoir un système d'information pour le suivi du (des) patient(s) dans un établissement hospitalier en se basant sur les diagrammes du langage UML à savoir le diagramme de cas d'utilisation, le diagramme de collaboration, le diagramme de séquence et le diagramme de classe.

Chapitre 3

Réalisation

3.1 Introduction

Dans ce chapitre, consacré à la réalisation et la mise en oeuvre de notre application de suivi des patients dans un établissement hospitalier, nous allons présenter les outils de développement adoptés; soit le système de gestion de base de données Paradox, le langage de manipulation de bases de données SQL ainsi que l'environnement utilisé qui est Borland Delphi 7 et enfin nous montrer les principales interfaces et fenêtres de l'application.

3.2 Outils de développement

3.2.1 Implémentation de la base de données

La base de données

Une base de données est composée de données stockées dans des mémoires de masse sous une forme structurée, et accessibles par des applications différentes et des utilisateurs différents. Une base de données doit pouvoir être utilisée par plusieurs utilisateurs en même temps. [Sca05]

Système de Gestion de Bases de Données

Un SGBD (Système de Gestion de Bases de Données) est un ensemble de logiciels chargés d'assurer les fonctions minimales suivantes :

- Le maintien de la cohérence des données entre elles
- Le contrôle d'intégrité des données accédées
- Les opérations classiques sur les données (consultation, insertion, modification, suppression)
- Les autorisations d'accès aux données. [Sca05]

Et pour la création des tables de notre base de données on a utilisés Paradox 7 qui est un SGBDR (Système de Gestion de Bases de Données Relationnelles) édités par Corel. Il est compatible avec les requêtes SQL (Structured Query Language) et dispose d'une interface graphique pour saisir les requêtes (QBE - Query By Example). Il permet aussi de configurer, avec des assistants ou librement, des formulaires de saisie incorporant des tables filles sans nécessiter de sous-formulaires, des états imprimables, des pages html liées aux données d'une base et d'incorporer des fiches créées sous Delphi. [tea09]

Langage de Manipulation de Bases de Données

SQL (Structured Query Language) est un langage de manipulation de bases de données mis au point dans les années 70, et il permet trois types de manipulations :

- La maintenance des tables : création, suppression, modification de la structure des tables.
- La manipulation des données : sélection, modification, suppression d'enregistrements.
- La gestion des droits d'accès aux tables : contrôle des données; droits d'accès, validation des modifications. [SB07]

3.2.2 Environnement de développement

Qu'est-ce que Delphi?

Borland Delphi est un environnement de développement de type RAD (Rapid Application Development) basé sur le langage Pascal. Il permet de réaliser rapidement et simplement des applications Microsoft Windows XP, Microsoft Windows 2000 et Microsoft Windows 98, avec un minimum de programmation. [DAR]

Nous avons choisi la version 7 de Delphi car elle fournit tous les outils nécessaires pour développer, tester et déployer des applications, notamment une importante bibliothèque de composants réutilisables, une suite d'outils de conception, de modèles d'applications, de fiches et d'experts de programmation que les versions précédentes du logiciel ne possédaient pas. [Pub02]

Il existe d'autres systèmes de développement rapide sous Windows mais Delphi est particulièrement très bien placé grâce à ces propriétés : [GIN09]

Moins de lignes de code et rapidité de compilation

Possibilité d'utiliser des procédures événementielles partagées

Notion de modèles réutilisables (fiches, menus, objets)

- Richesse des composants fournis

Assembleur intégré, compilateur en ligne de commande

Débogage facile au niveau du code source et du processeur

Possibilité d'allocation dynamique de la mémoire en utilisant les pointeurs Voici un aperçu de l'interface de travail de Borland Delphi 7 :

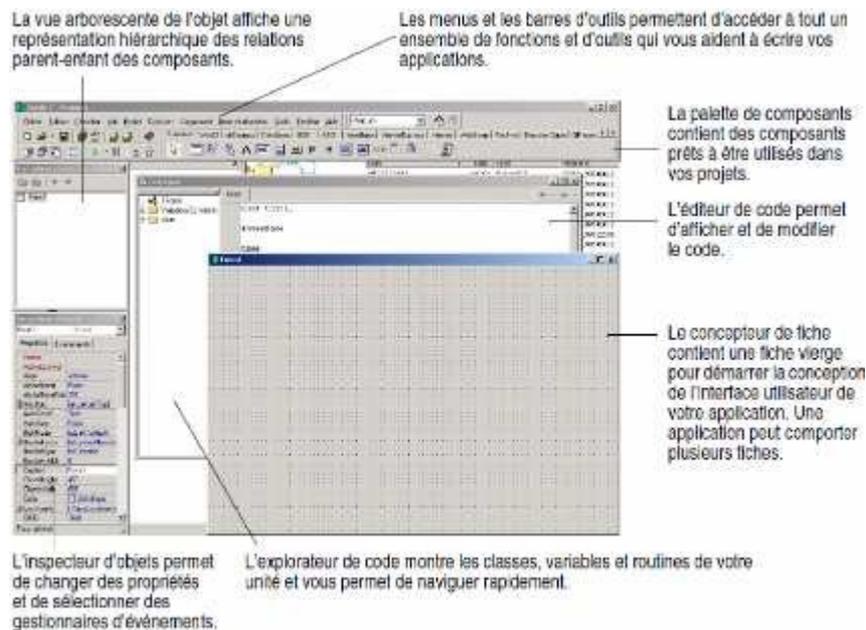


FIG. 3.1 Interface de Borland Delphi 7

3.3 Description de l'application

L'application que nous avons conçue permet d'effectuer les tâches suivantes :

· Les admissions:

- de patients (normale, naissance et accidenté)

de gardes-patient

· Les sorties :

de patients (normale, évacuation et décès)

de gardes-patient

· L'affichage:

des patients hospitalisés des patients décédés des patients évacués des patients guéris

des gardes-patient des naissances

des services

des patients traités

· Les statistiques :

compteur de taille des bases de données

la moyenne des naissances par mois

la moyenne des décès par mois

· Les imprimés :

bulletin d'admission

- billet de salle (patient, garde-patient et naissance)

certificat de séjour certificat de présence déclaration de naissance déclaration de décès

3.3.1 Menu de l'application

FIG. 3.2 Menu de l'application (partie 1)

FIG. 3.3 Menu de l'application (partie 2)

3.3.2 Les interfaces de l'application

Le formulaire d'authentification

Au lancement de notre application, un formulaire s'affiche à l'écran, il nous demandera d'introduire le mot de passe d'authentification pour accéder au menu principal.

FIG. 3.4 Formulaire d'authentification

Si le mot de passe introduit n'est pas valide, alors l'application revoit message d'erreur suivant



Le menu principal

Après avoir saisi le mot de passe valide, la fenêtre ci-après s'affiche, elle comporte le menu principal où l'utilisateur pourra sélectionner la tâche à effectuer.

Le menu de notre application contient sept (07) boutons qui sont : Fichier, Affichage, Recherche, Statistiques, Imprimer, Outils et Aide.

FIG. 3.5 La fenêtre du menu principal

Voici les principaux boutons du menu principal:

FIG. 3.6 Le bouton "Nouveau > Admission > Patient"

FIG. 3.8 Le bouton "Modifier"

FIG. 3.9 Le bouton "Supprimer"

FIG. 3.7 Le bouton "Nouveau > Sortie"

FIG. 3.10 Le bouton "Affichage"

FIG. 3.11 Le bouton "Recherche"

FIG. 3.12 Le bouton "Statistiques"

FIG. 3.13 Le bouton "Imprimer"

FIG. 3.14 Le bouton "Outils"

FIG. 3.15 Le bouton "Aide"

Les formulaires de l'application

Nous prenons ici deux exemples, qui sont l'admission et la suppression d'un patient.

Quand on clique sur le bouton patient du menu "Nouveau > Admission", le formulaire d'admission s'affiche à l'écran pour pouvoir saisir les informations du patient à admettre dans l'hôpital:

FIG. 3.16 Le formulaire d'admission

Après avoir saisi toutes les informations de l'admission, il suffit de cliquer sur le bouton "Enregistrer" pour les sauvegarder dans la base de données.

Lorsque le bouton "Sortie" du menu "Fichier > Nouveau" est activé, le formulaire de sortie d'un patient apparaîtra:

FIG. 3.17 Le formulaire de sortie

Après avoir introduit sélectionné le patient à supprimer, il suffit de cliquer sur le bouton "Supprimer" pour pouvoir l'effacer de la base de données.