

## Table des matières

Introduction générale	7
<b>Chapitre 1: Etude et analyse</b>	9
I.Introduction	10
II.Présentation du projet	10
III.Cahier de charge	11
III.1 Cahier de charge fonctionnel	11
III.1.1 Besoins non fonctionnels	11
III.1.2 Besoins fonctionnels	12
III.1.3 Architecture fonctionnelle de l'application	12
III.2 Cahier de charge technique	15
III.2.1 Architecture technique de l'application	15
IV.Conclusion	19
<b>Chapitre 2: Conception</b>	20
I.Introduction	21
II.Cas d'utilisation	21
III.Diagrammes de séquence	23
III.1 Cas "Mesure IMC"	23
III.2 Cas "Création d'un dossier médical"	24
III.3 Cas "Enregistrement du taux de glycémie"	25
III.4 Cas "Rappel des prise médicamenteuses "	27
III.5 Cas "Enregistrement des bilans sanguins"	29
III.6 Cas "Recherche"	30
IV. Diagrammes de classes	31
IV.1 Cas : "Mesure de l'IMC"	31
IV.2 Cas : "Création d'un dossier médical"	34
IV.3 Cas : "Enregistrement de taux de glycémie"	35
IV.4 Cas : "Rappel des prises médicamenteuse"	36
IV.5 Cas : "Enregistrement des bilans sanguins"	39
IV.6 Cas : "Recherche"	40
IV.7 Diagramme de classe général	40
V. OCL	42
IV. Conclusion	44
<b>Chapitre 3: Réalisation</b>	45
I. Introduction	46
II. L'application	46
II.1 La Fenêtre principale	46
II.2 La Création d'un dossier médical	47
II.3 Calculer l'indice de la masse corporelle	48
II.4 Taux de glycémie	49
II.5 Bilans sanguins	51
II.6 Rappel des prises médicamenteuses	51
II.7 Recherche	53
III. Conclusion	55

<b>Chapitre 4: Conclusion et perspectives</b>	<b>58</b>
Annexe A	60
Annexe B	60
Annexe C	66
Annexe D	66
Annexe E	67
Annexe F	68
Annexe G	69
Annexe H	69
Annexe I	70

## Liste des tableaux

Tab .1.1. Entité Dossier médicale .....	17
Tab .1.2. Entité IMC.....	17
Tab .1.3. Entité Taux de glycémie.....	17
Tab .1.4. Entité Alarme.....	17
Tab .1.5. Entité Rendez vous.....	18
Tab .1.6. Entité Bilan hématologique.....	18
Tab .1.7. Entité Bilan biochimique.....	18
Tab.2.1 Classe IMC.....	31
Tab.2.2 Classe « BaseSql » .....	31
Tab.2.3 Classe Bdd .....	32
Tab.2.4 Classe DossierMed .....	33
Tab.2.5 Classe CreeDossier.....	34
Tab.2.6 Classe TauxSucre.....	35
Tab.2.7 Classe Alarme.....	36
Tab.2.8 Classe Ajout_Alarme.....	37
Tab.2.9 Classes Bilan_Hématologique et Bilan_Biochimique.....	39
Tab.2.10 Classe Recherche.....	40

# Liste des figures

Fig. 1.1 Schéma général de l'application AISanté .....	12
Fig. 1.2. Schéma général du MCD d'ISanté.....	16
Fig. 1.3. Schéma général de l'enchaînement des différents écrans d'AISanté.....	19
Fig. 2.1 diagramme cas d'utilisation.....	22
Fig. . 2.2 Diagramme de séquence "mesure de l'IMC".....	24
Fig. . 2.3 Diagramme de séquence "Création d'un dossier médical".....	25
Fig. 2.4 Diagramme de séquence " Enregistrement du taux de glycémie " .....	26
Fig. 2.5 Diagramme de séquence " Ajout d'une alarme des prises médicamenteuses".	28
Fig.2.6 Diagramme de séquence " Enregistrement des bilans sanguins ".....	29
Fig. 2.7 Diagramme de séquence " Recherche".....	30
Fig. . 2.8 Diagramme de classe " mesure de l'IMC ".....	33
Fig. . 2.9 Diagramme de classe " CreeDossier ".....	34
Fig. 2.10 Diagramme de classe " Enregistrement du taux de gycémie".....	36
Fig. . 2.11Diagramme de Classe " rappel des prises médicamenteuses".....	38
Fig. . 2.12 Diagramme de Classe " Enregistrement des bilans sanguins".....	39
Fig. . 2.13 Diagramme de Classe "recherche ".....	40
Fig. 2.14Diagramme de Classe de AISanté.....	41
Fig. 3.1 Fenêtre principale.....	46
Fig. 3.2 Création d'un dossier médical.....	47
Fig. 3.3 Formulaire de création d'un dossier médical.....	47
Fig. 3.4 Validation du formulaire crée.....	48
Fig. 3.5 Fenêtre de calcul de l'IMC.....	48
Fig. 3.6 Résultat de l'IMC.....	49

Fig. 3.7 Taux de glycémie.....	50
Fig. 3.8 Graphe de variation du taux de glycémie.....	50
Fig. 3.9 Moyenne des taux de glycémie.....	50
Fig. 3.10 Bilans biochimiques.....	51
Fig. 3.11 Bilans hématologiques.....	51
Fig. 3.12 création d'une alarme .....	52
Fig. 3.13 formulaire détaillant l'ajout d'un nouveau rappel. ....	52
Fig. 3.14 Supprimer un rappel.....	52
Fig. 3.15 Dossier médical personnel.....	53
Fig. 3.16 Recherche.....	54
Fig. 3.17 Auto remplissage.....	54
Fig. 3.18 Résultats de recherche1.....	54
Fig. 3.19 Résultats de recherche2.....	55

## Liste des abréviations

<b>Acronym</b>	<b>Explication</b>	<b>Acronyme</b>	<b>Explication</b>
<b>e</b>			
<b>AISanté</b>	<b>Android Internet Santé.</b>	<b>XML</b>	<b>eXtensible MarkupLanguage.</b>
<b>API</b>	<b>Application Programming Interface.</b>	<b>URL</b>	<b>Uniform Resource Locator.</b>
<b>UML</b>	<b>Unified Modeling Language.</b>	<b>HTML</b>	<b>HyperText Markup Language.</b>
<b>OCL</b>	<b>Object Constraint Language.</b>	<b>HDL</b>	<b>High Density Lipoprotein (Anglais)</b>
<b>IDE</b>	<b>Integrated Development Environment.</b>	<b>LDL</b>	<b>Low Density Lipoprotein (Anglais)</b>
<b>IMC</b>	<b>Indice de Masse de Corporelle</b>	<b>VLDL</b>	<b>Very Low Density</b>
<b>TG</b>	<b>Taux de Glycémie.</b>	<b>VGM</b>	<b>Volume Globulaire Moyen.</b>
<b>SDK</b>	<b>Software Development Kit.</b>	<b>TGMH</b>	<b>Teneur Globulaire Moyenne en Hémoglobine.</b>
<b>MCD</b>	<b>Modèle Conceptuel de Données</b>	<b>VS</b>	<b>Vitesse de Sédimentation (fr).</b>
<b>BDD</b>	<b>Base De Données.</b>	<b>2TUP</b>	<b>2 Track Unified Process.</b>
<b>ADT</b>	<b>Android Development Tools.</b>	<b>UI</b>	<b>User Interface.</b>
<b>OS</b>	<b>Open Source.</b>	<b>SQL</b>	<b>Structured Query Language</b>

# **Introduction générale**

Les technologies de l'information et de la communication ont été la révolution la plus importante et innovante qui a marquée ces dernières décennies. En effet, loin d'être un phénomène éphémère ces technologies nous ont apporté du confort dans notre vie quotidienne par leurs capacités à traiter l'information dans des délais raisonnables.

Cette révolution a permis l'émergence de la notion de la portabilité et de la mobilité qui permet un accès distant, instantané et un flux sans interruption d'informations. En effet, cela est symbolisé par l'apparition des différents appareils de haute technologie tels que les Smartphones et les tablettes qui sont dotés de plusieurs applications pratiques quelles soient gratuites ou payantes.

Dans ce cadre, et afin de réaliser ce projet de fin d'études, un outil proposé permet de pallier les problèmes concernant la question du maintien à domicile des personnes âgées ou handicapées. Cet outil propose dès maintenant une solution simple, innovante autour du concept d'AI Santé. Cela a pour vocation de gérer et de faciliter l'accès au dossier médical et le suivi individuel à la fois pour le patient et pour le corps médical.

Le manuscrit se compose de plusieurs chapitres. Le premier chapitre détaille l'analyse des besoins fonctionnels et techniques pour mener à aboutir un cahier de charge qui constitue le point de départ de ce travail. Le deuxième chapitre explique la phase de conception de ce projet avec ces différents diagrammes. La réalisation de l'application est détaillée dans le dernier chapitre. Enfin ce manuscrit clôture par une conclusion et des perspectives de ce travail.

**Chapitre 1 :**

**Etude et analyse**

## I. Introduction :

Pour mener à bien ce projet, le cycle de vie adopté est en mode Y [1], identifier par le processus unifié 2TUP, ce mode qui permet une gestion complète et détaillée à la fois des projets informatiques. Cela consiste à analyser les besoins fonctionnels et techniques en parallèle (deux branches parallèle), suivie d'une phase de conception et une phase de réalisation et test. La phase d'analyse consiste à comprendre les besoins fonctionnels et techniques qui interviennent dans la construction de la solution finale. Cette phase se conclue par la rédaction d'un cahier de charge qui constitue la feuille de route pour le reste du projet.

## II. Présentation du projet :

L'application baptisé *AISanté* est une application Smartphone (\*) dont l'objectif principal est la gestion de la santé des patients, cela permet un suivie individuel régulier, un accès rapide au dossier du patient en cas urgence ou en cas de consultation.

L'application devra notamment :

- Permettre de créer un dossier médical individuel et de fournir un rapport médical détaillé en se basant sur les différentes données fournies par le patient lui même.
- Permettre d'enregistrer le taux de glycémie pour les patients diabétiques.
- Permettre de mémoriser et d'analyser le dernier bilan sanguin effectué par le patient.
- Permettre de planifier le suivi des prises des médicaments prescrites par les différents médecins.
- Permettre d'effectuer des recherches dans un dictionnaire médical en occurrence 'Doctissimo' (Annexe A)

Les objectifs de second niveau sont :

- Permettre d'établir des statistiques relatives aux informations enregistrées.

## **III. Cahier de charge :**

### **III.1 Cahier de charge fonctionnel :**

Le cahier de charge fonctionnel décrit l'aspect pratique de l'application telle qu'un fonctionnement concret et un résultat qui converge vers un vrai pronostic.

Il existe deux types de besoins que l'application se doit de satisfaire : Les besoins non fonctionnels qui correspondent à la manipulation de l'application et précisent l'environnement de celle-ci, tandis que les besoins fonctionnels listent les opérations réalisables par cette dernière.

#### **III.1.1 Besoins non fonctionnels :**

Interface utilisateur :

L'application AISanté se doit d'être cohérente d'un point de vue ergonomique, ce point sera un facteur essentiel étant donnée l'utilisation intensive dont sera sujette cette application. Un menu d'aide à l'utilisateur présentant l'interface et les fonctionnalités seront disponibles dans un contexte attractif voire conviviale.

Configuration requise par l'application :

La configuration minimale adoptée est celle d'un Smartphone basé sur un système d'exploitation ANDROID (Annexes B)

Développement de l'application :

SDK Android (Annexes B)

Eclipse IDE for Java Developers (Annexes I)

Environnement d'exécution :

-Dalvik Virtual machine (DVM)

Environnement réseau :

Dans un premier lieu l'application fonctionnera en mode monoposte, puis en mode réseau (dans un développement futur), cela permettra de centraliser les données des différents patients et de faciliter l'accès aux corps médical dans un premier temps ; corps médical qui est distant de base et cela sans avoir recours à utiliser le support mobile pour venir en aide rapidement aux malade puis établir aussi des statistiques médicales relatives aux informations

enregistrées dans un souci de cohérence tablant sur différents paramètres tels que : le nombre de diabétiques, le nombre d'obèses ou d'anorexiques, les problèmes cardiaques, etc ...

### III.1.2 Besoins fonctionnels :

AISanté doit permettre ce qui suit :

- Accéder rapidement au dossier médical du patient.
- Fournir les fonctions décrites ci-dessus.
- Générer une page résumant l'état général du patient
- Sauvegarder des données cruciales : *IMC, Taux de glycémie, Bilan sanguin ...*

### III.1.3 Architecture fonctionnelle de l'application :

AISanté comprendra principalement deux grands modules divisés en sous-modules accessibles à l'aide d'un menu principal :

- Un module recherche qui permet d'effectuer des recherches dans l'encyclopédie médicale référée auparavant '*Doctissimo*'
- Un module d'application qui contient les sous-modules suivants :

-Création du dossier médical.

-Gestion de taux de glycémie.

-IMC.

-Rappel des prises des médicaments.

-Gestion des différents bilans sanguins.

Ci-dessous, une représentation des différents modules qui composent cette application :

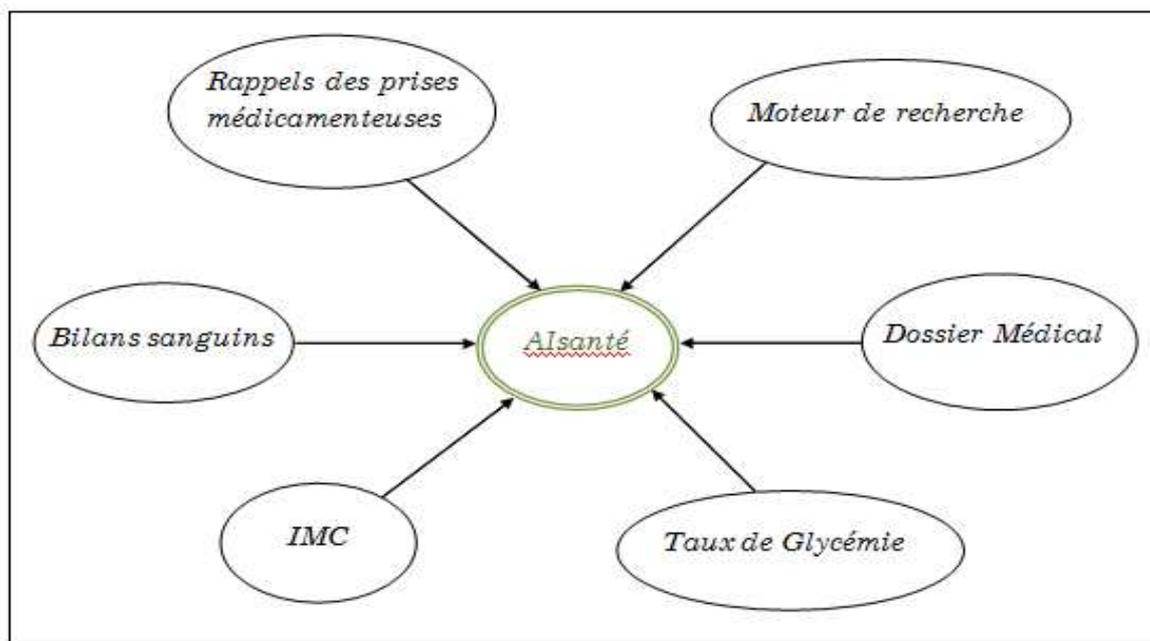


Fig. 1.1. Schéma général de l'application *ISanté*

Le module « Dossier médical personnel » :

En tant que descriptif, Le « dossier médical personnel » permet d'accéder rapidement aux données qui sont propres aux patients ainsi que quelques points médicaux de base.

Le module «dossier médical personnel» est unique pour chaque patient et est composé des champs suivants:

1. Nom
2. Prénom
3. Sexe
4. Age
5. Adresse
6. Groupe sanguin
7. Liste des pathologies connues du patient (allergie, diabète, problèmes gastrique, etc...)
8. Numéro du téléphone de la personne à contacter en cas d'urgence.

Ensuite, l'application doit pouvoir aussi compléter automatiquement le dossier médical (fournir un rapport médical) en se basant sur les différentes données fournis par le patient (IMC, taux de glycémie,...).

9. Dernier taux de glycémie mesuré par le patient,
10. Dernière mesure de l'IMC,
11. Dernier Bilan sanguin détaillé,
12. es différentes prises médicamenteuse en cours.

Le module « IMC » :

*AISanté* permet de d'enregistrer les différentes mesures d'*IMC* (Annexe C). Ainsi, elle permettra un suivie de l'évolution de la masse corporelle de chaque patient au fil du temps. Le module « IMC » est composé des champs suivants:

1. Taille
2. Poids,
3. Résultat.

Le module « Gestion du taux de glycémie » :

*AISanté* doit permettre aussi d'enregistrer les différents taux de glycémie (Annexe D) pendant la journée. De plus, ce taux permettra de faire la distinction entre les mesures faites : à jeun, avant le repas et après le repas ainsi que fournir leurs différentes variations au fil du temps et établir différents ratios tels que : la moyenne des taux à jeun, avant et après le repas.

Le module « Gestion du taux de glycémie » est composé des champs suivants:

1. Valeur mesuré.
2. Liste déroulante (à jeun, avant le repas, après le repas).

Le module « Rappel des prises médicamenteuses » :

Ce module propose aux personnes âgées ou atteintes de maladies cérébrales chroniques (exemple : l'Alzheimer), un service qui permet de rappeler les différentes prises de médicaments prescrits dans son traitement.

Le module « Rappel des prises médicamenteuses » est composé des champs suivants :

1. Nom de médicament,
2. Durée de traitement,
3. Semainier (Fréquence d'administration d'un médicament).
4. Posologie d'un médicament (fréquence de consommation et son moment d'attribution),

Le module « gestion des différents bilans sanguins » :

Il est ainsi parfois difficile d'interpréter des résultats d'examens biologiques avec ces noms barbares (Jargon médical), ces initiales (incompréhensibles aux premiers abords) et ces chiffres. Pour aider à analyser lesdits résultats de la prise sanguine, il est proposé l'ajout d'un module qui puisse faciliter cela. Il existe entre autre plusieurs types de bilans, la pertinence des résultats pousse vers les choix suivants:

1. L'analyse de résultats sur l'hématologie (voir annexe E).
2. L'analyse de résultats de biochimie (voir annexe F).

Le module «gestion des différents bilans sanguins» couvrira les deux choix ci-dessus :

○ le bilan hématologique » est composé des champs suivants:

-Globules rouges      -Globules blancs      -TGMH      -Plaquettes  
-Hémoglobine      -VGM      -VS

○ le bilan biochimique » est composé des champs suivants:

- Taux de glycémie      -Cholestérol totale      -HDL      -LDL  
-VLDL      -Acide urique      -Créatine

Le module «Recherche» :

Ce module permet de se renseigner sur les propriétés d'un médicament en interrogeant la ressource informationnelle sollicitée dans ce projet qui est le dictionnaire médical

'*Doctissimo*', du fait qu'il représente une référence incontestée dans le domaine, conséquence, cette approche aboutit à générer une « Notice virtuelle ».

Ces propriétés sont classées de la manière suivante :

- **Indications** : Elles regroupent l'ensemble des pathologies qui peuvent solliciter le patient à une utilisation potentielle du médicament.
- **Contre indications** : Elles recensent les cas dans lesquels le médicament ne doit jamais être utilisé sous peine ou le patient subit une série des complications.
- **Effets indésirables** : Décrivent l'ensemble des effets secondaires qui se révéleront dans leurs ensembles comme étant hypothétiques à conséquences néfastes éventuels déclarés à ce jour.
- **Précautions d'emploi** : Englobent les cas dans lesquels la prescription d'un médicament doit être sujette à une surveillance particulière.
- **Compositions** : Précisent les ingrédients chimiques propres au médicament : principes actifs, excipients...
- **Classes thérapeutiques** : Classification du médicament.
- **Posologies** : Schémas thérapeutiques en fonctions du type d'utilisation.

## III.2 Cahier de charge technique :

Ce cahier de charge décrit l'aspect technique de l'application tels que : les données, les écrans, les messages ... etc.

### III.2.1 Architecture technique de l'application :

Les données :

Une base de données entités-relations est mise en œuvre pour pouvoir sauvegarder les différentes données de notre application. Le modèle conceptuelle de données (MCD) [2] de ce dernier e st décrit dans la Fig.1.2 :

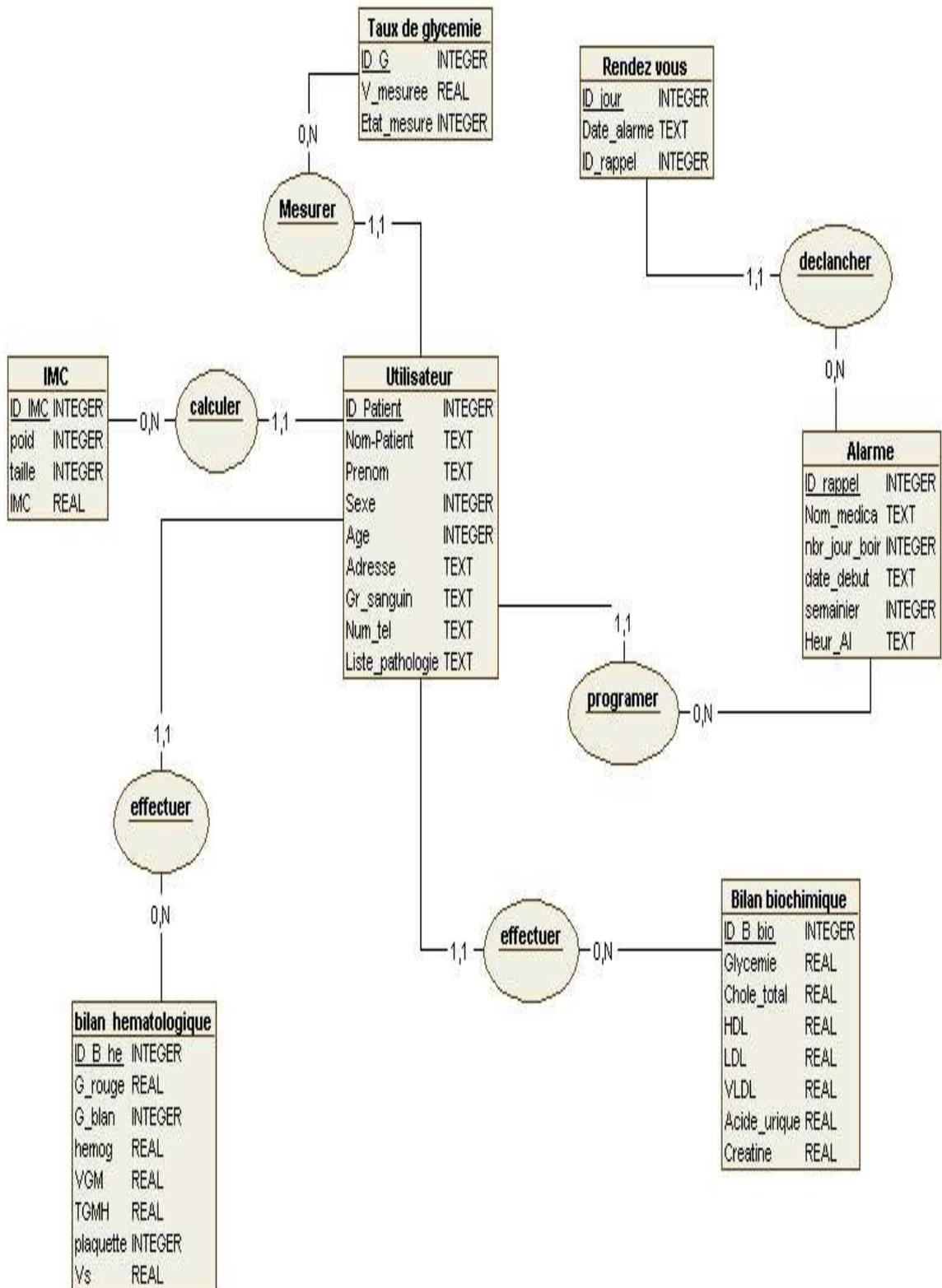


Fig. 1.2. Schéma général du MCD d' AISanté.

Pour implémenter cette base de données, SQL lite (Annexe G) est utilisé à cette fin ; ce dernier est intégré dans Android .

Description des données : Cette section détaille les entités comme il suit :

*Entité Dossier médicale :*

N°	Désignation	Type	Longueur	Valeur	Clef	Description
1	ID_Patient	INT	20	Auto	OUI	Identifiant.
2	Nom-Patient	VARCHAR	25			Nom du patient.
3	Prenom	VARCHAR	25	-		Prénom du patient.
4	Sexe	INT	1	-		Sexe du patient (H ou F).
5	Age	INT	3	-		Age du patient.
6	Adresse	VARCHAR	30			Adresse du patient.
7	Gr_sanguin	VARCHAR	2	-		Groupage sanguin.
8	Num_tel	VARCHAR	10	-		Numéro de téléphone de personne à contacter en cas urgence.
	Liste_pathologies	VARCHAR	20	-		La liste pathologique.

Tab .1.1. Entité Dossier médicale.

*Entité IMC :*

N°	Désignation	Type	Longueur	Valeur	Clef	Description
1	ID_imc	INT	20	Auto	OUI	Identifiant
2	Poids	INT	3	-		Le poids du patient.
3	Taille	INT	3	-		La taille du patient.
4	IMC	REAL	4	-		Le résultat IMC.

Tab .1.2. Entité IMC.

*Entité Taux de glycémie :*

N°	Désignation	Type	Longueur	Valeur	Clef	Description
1	ID_G	INT	20	Auto	OUI	Identifiant
2	V_mesuree	REAL	4	-		La valeur de la mesure
3	Etat_mesure	INT	1	-		Etat du patient l'heur de mesurer

Tab .1.3. Entité Taux de glycémie.

*Entité Alarme:*

N°	Désignation	Type	Longueur	Valeur	Clef	Description
1	ID_rappel	INT	20	Auto	OUI	Identifiant.
2	Nom_medica	VARCHAR	30	-		Nom du médicament.
3	Nbr_jour_b	INT	4	-		nombre de jours du traitement.

4	Date_debut	TEXT	8	-	Date de début du traitement.
6	Semainier	INT	1	-	Nombre de prise (semainier).
7	Heur_AL	TEXT	23	-	Heur de prise de médicament.

Tab .1.4. Entité Alarme.

*Entité Rendez-vous:*

N°	Désignation	Type	Longueur	Valeur	Clef	Description
1	ID_jour	INT	20	-	OUI	Identifiant.
2	Date_alarme	TEXT	8	-		Date de l'alarme.
3	ID_rappel	INT	20	-		Clé étrangère de la table alarme.

Tab .1.5. Entité Rendez-vous.

*Entité bilan hématologique:*

N°	Désignation	Type	Longueur	Valeur	Clef	Description
1	ID_B_he	INT	20	Auto	OUI	Identifiant.
2	G_rouge	REAL	4	-		Valeur des Globules rouges.
3	G_blan	INT	6	-		Valeur des Globules blanches.
4	Hemog	REAL	2	-		Valeur d'Hémoglobine.
6	VGM	REAL	4	-		Valeur du VGA.
7	TGMH	REAL	2	-		Valeur de TGMH.
8	VS	REAL	2	-		Valeur de VS.
9	Plaquette	INT	6	-		Valeur des plaquettes..

Tab .1.6. Entité Bilan hématologique.

*Entité bilan biochimique:*

N°	Désignation	Type	Longueur	Valeur	Clef	Description
1	ID_B_bio	INT	20	Auto	OUI	Identifiant.
2	Glycemie	REAL	4	-		Valeur de glycémie.
3	Chole_total	REAL	4	-		Valeur du cholestérol total.
4	Hdl	REAL	4	-		Valeur du Hdl.
5	LDL	REAL	4	-		Valeur du LDL.
6	VLDL	REAL	4	-		Valeur du VLDL.
7	Acide unique	REAL	2	-		Valeur d'acide urique.
8	Creatine	REAL	2	-		Valeur du créatine.

Tab .1.7. Entité Bilan biochimique.

*Les Ecrans*

L'écran principal de l'application permet de naviguer dans les différents modules proposés par AISanté. Il permet de lancer le module « recherche » et le module « application » (voir Fig.1.3).

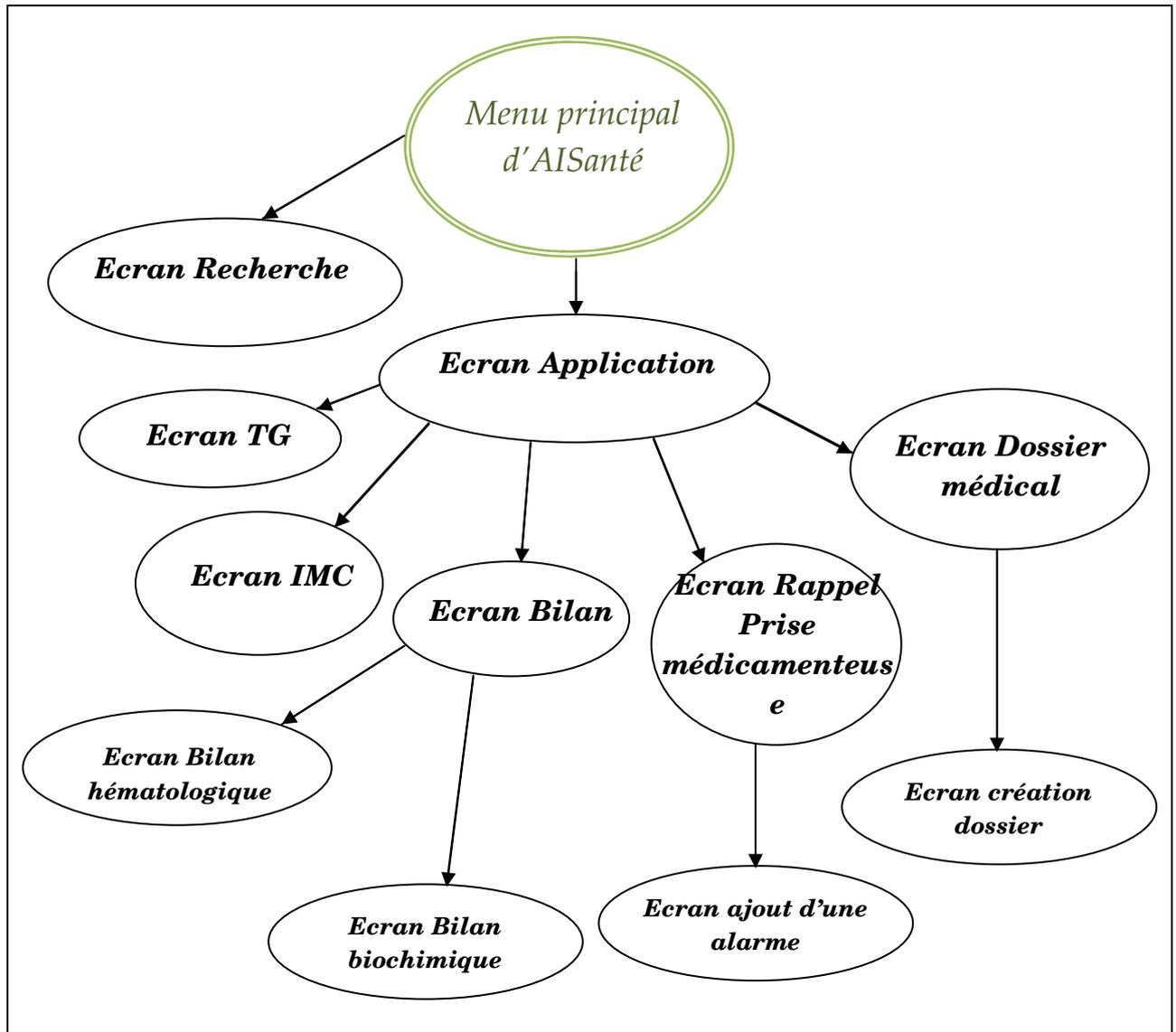


Fig. 1.3. Schéma général de l'enchaînement des différents écrans d'AISanté.

## IV. Conclusion :

La phase théorique s'achève avec une certaine adéquation avec les vœux et buts de ce projet. L'analyse étant réalisée, le cahier de charge concrétisé, la phase pratique sera entamée dans le prochain chapitre qui se structurera autour de sa propre introduction et détaillera la schématisation de la partie conceptuelle.

**Chapitre 2 :**

**Conception**

## I. Introduction :

La phase de conception constitue une étape fondamentale qui précède celle du développement, elle permet de détailler les différents diagrammes et scénarios à implémenter dans la phase suivante.

Pour que les différentes notations soient lisibles et compréhensibles par toute personne intervenante dans le développement de ce projet, l'approche optée est pour le langage unifié de modélisation UML (Unified Modeling Language) [3]. UML définit treize types de diagrammes devisés en deux catégories:

- Diagrammes statiques (structurels)[4] : diagramme de classe, d'objet, de composant, de déploiement et de diagramme de cas d'utilisation.
- Diagrammes dynamique (comportementaux) : diagramme d'activité, de séquence, d'état-transition et de diagramme de collaboration.

Dans ce travail, les diagrammes UML utilisés sont les suivant : Diagramme de cas d'utilisation, diagramme de séquence, et diagramme de classes sont modéliser a l'aide de l'outil de modélisation StarUml[5].

## II. Cas d'utilisation :

Le diagramme de cas d'utilisation est utilisé pour exprimer 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. Le rôle 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.

La définition de cas d'utilisations est détaillée dans un ensemble décrit ci-dessous :

- Créer un Dossier Médical personnel.
- Mesurer l'IMC.
- Enregistrer le taux de glycémie.
- Rappeler des prises de médicamenteuse.
- Enregistrer les bilans sanguins.
- Rechercher des informations sur un médicament.

[4] : un langage de modélisation graphique à base de pictogrammes.

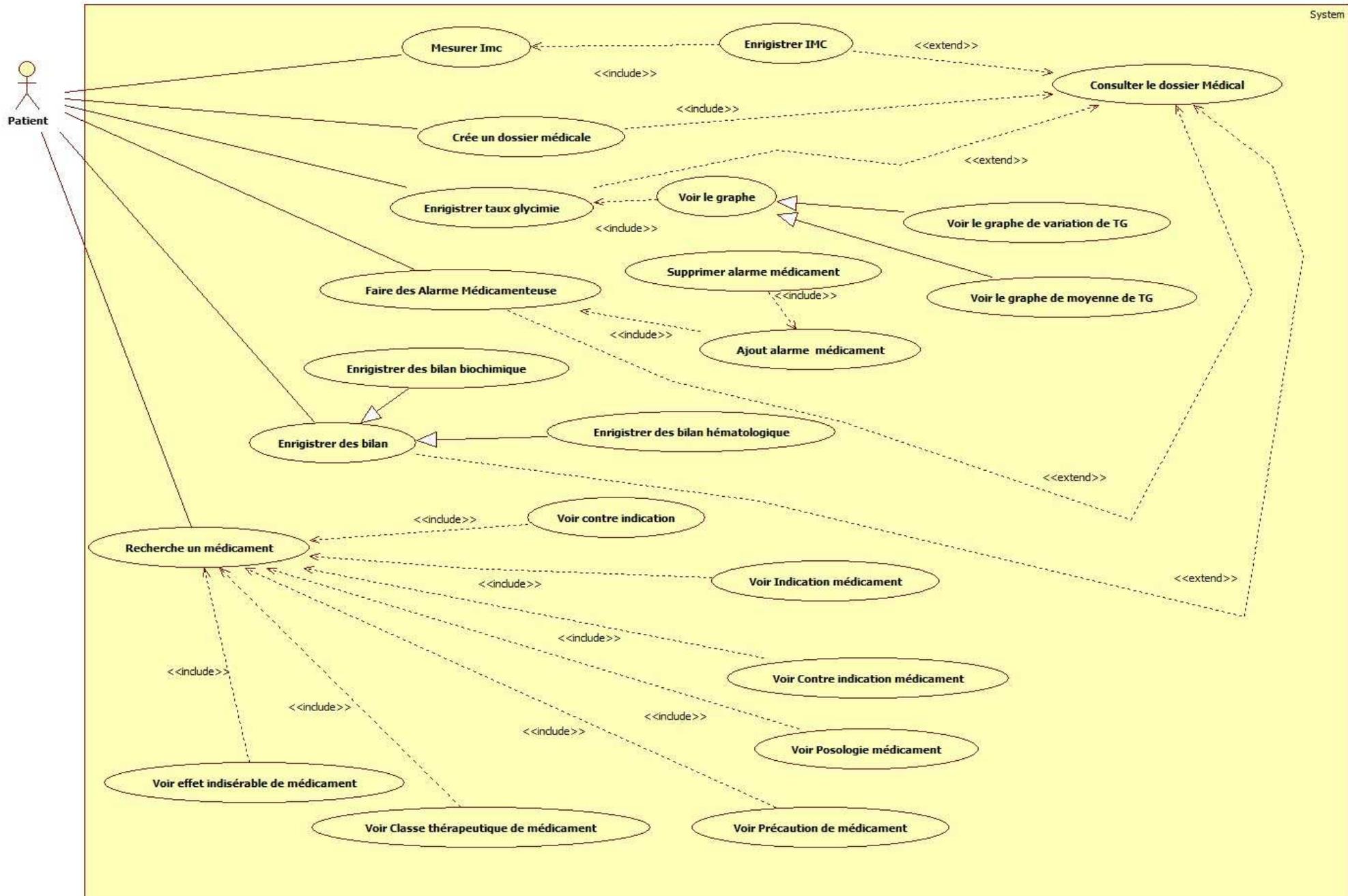


Fig. 2.1 diagramme cas d'utilisation.

### **III. Diagrammes 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.

Ce diagramme met en scène une interaction. En particulier, il montre aussi les objets qui participent à cette même interaction par leur "ligne de vie" et les messages qu'ils échangent présentés sous forme de séquence dans le temps.

Ci-dessous ; une description détaillée des différents diagrammes de séquences de chaque cas utilisation avec leurs scénarios respectifs.

#### **III.1 Cas "Mesure IMC" :**

##### **III.1.1 Scénario :**

1. Le patient saisit son poids et sa taille.
2. L'application vérifiée automatiquement les valeurs introduit.
3. Le patient demande le calcul de l'IMC.
4. l'application contrôle les valeurs fournies par le patient.
5. l'application affiche la valeur calculée de d'IMC et ainsi le poids idéal pour être dans la norme.
6. L'application génère des consignes selon le résultat obtenu.
7. Le patient enregistre la valeur obtenue.
8. L'application complète automatiquement le dossier médical.

Ci dessous le diagramme explique le déroulement de différentes séquences du scénario :

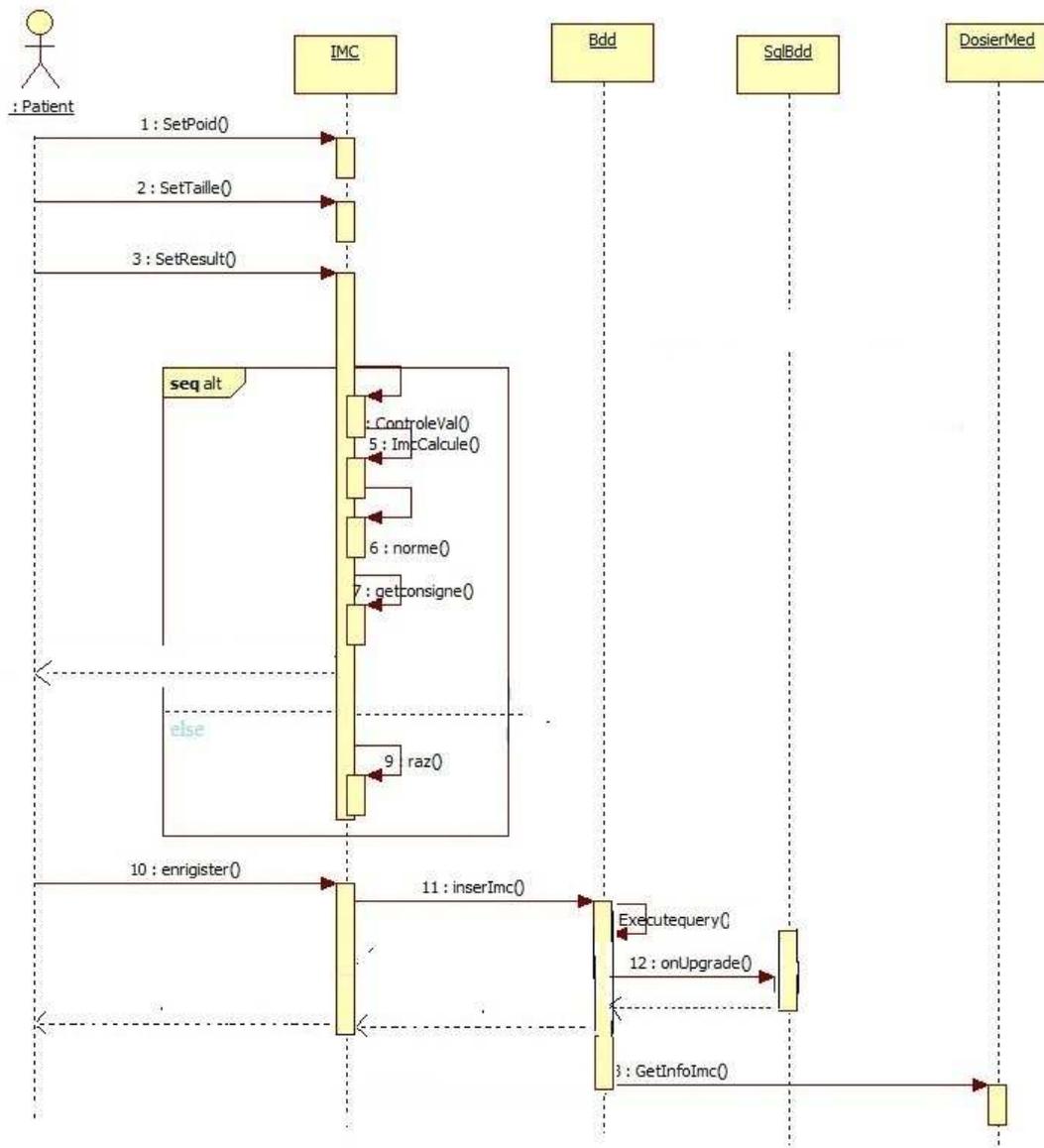


Fig 2.2 Diagramme de séquence "mesurer l'IMC".

### III.2 Cas "création d'un dossier médical" :

#### III.2.1 Scénario :

1. Le patient demande de créer un dossier médical.
2. L'application affiche le formulaire de création du dossier médical.
3. Le patient remplit les champs.
4. L'application contrôle les valeurs entrées par le patient.
5. L'application crée un dossier médical.
6. Le patient peut à tout moment modifier son dossier médical.

Ci dessous le diagramme explique le déroulement de différentes séquences du scénario :

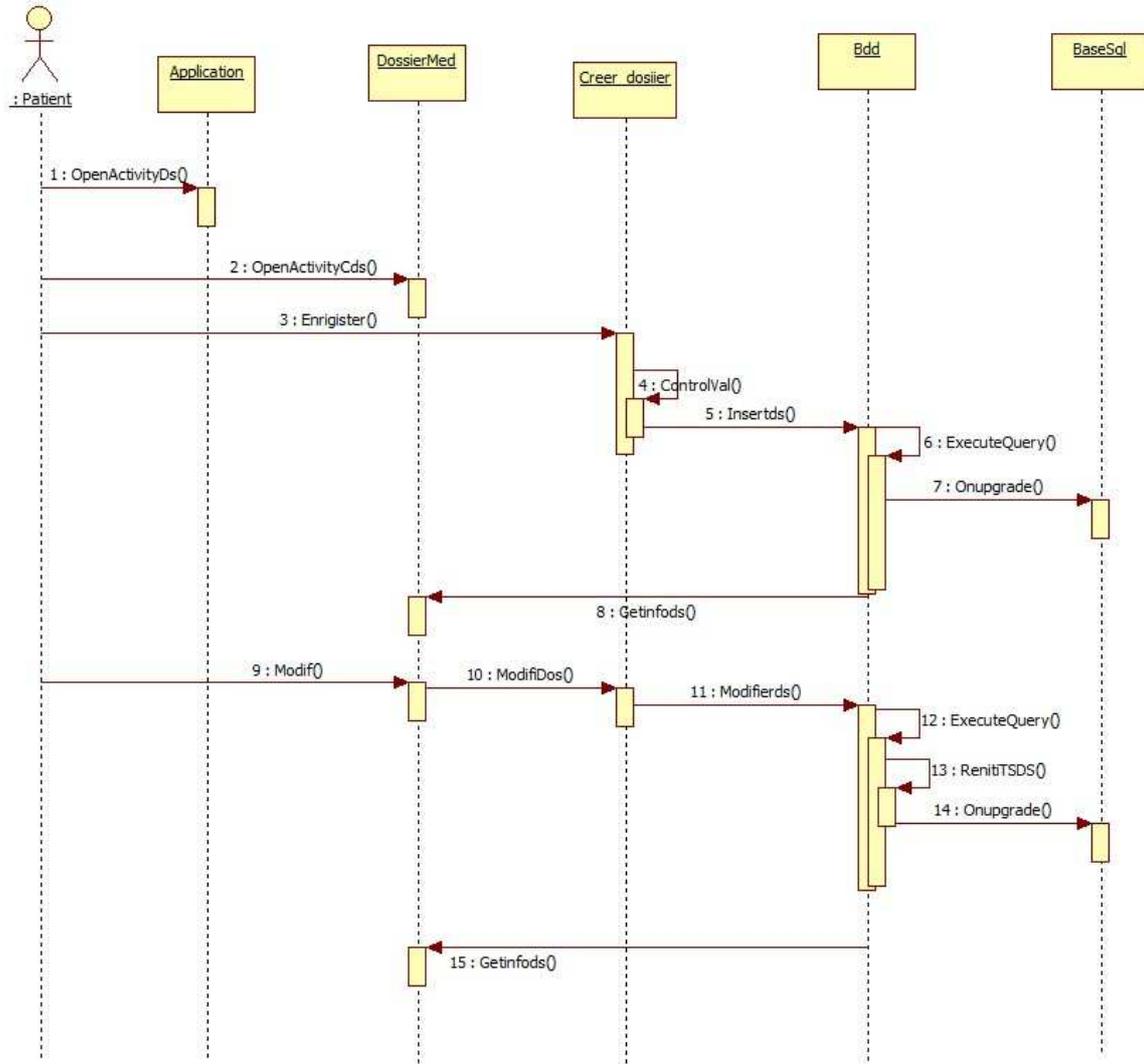


Fig 2.3 Diagramme de séquence "Création d'un dossier médical".

### III.3 Cas "Enregistrement du taux de glycémie" :

#### III.3.1 Scénario :

1. Le patient saisit la valeur du taux de glycémie obtenu ainsi que l'état du patient lors de ses prises de mesures.
2. Le patient enregistre la valeur obtenue.
3. L'application contrôle la valeur fournie par le patient.

4. Le patient peut demander la représentation graphique des dernières mesures effectuées.
5. Le patient peut demander la représentation graphique des moyennes des valeurs enregistrées.

Ci dessous le diagramme explique le déroulement de différentes séquences du scénario :

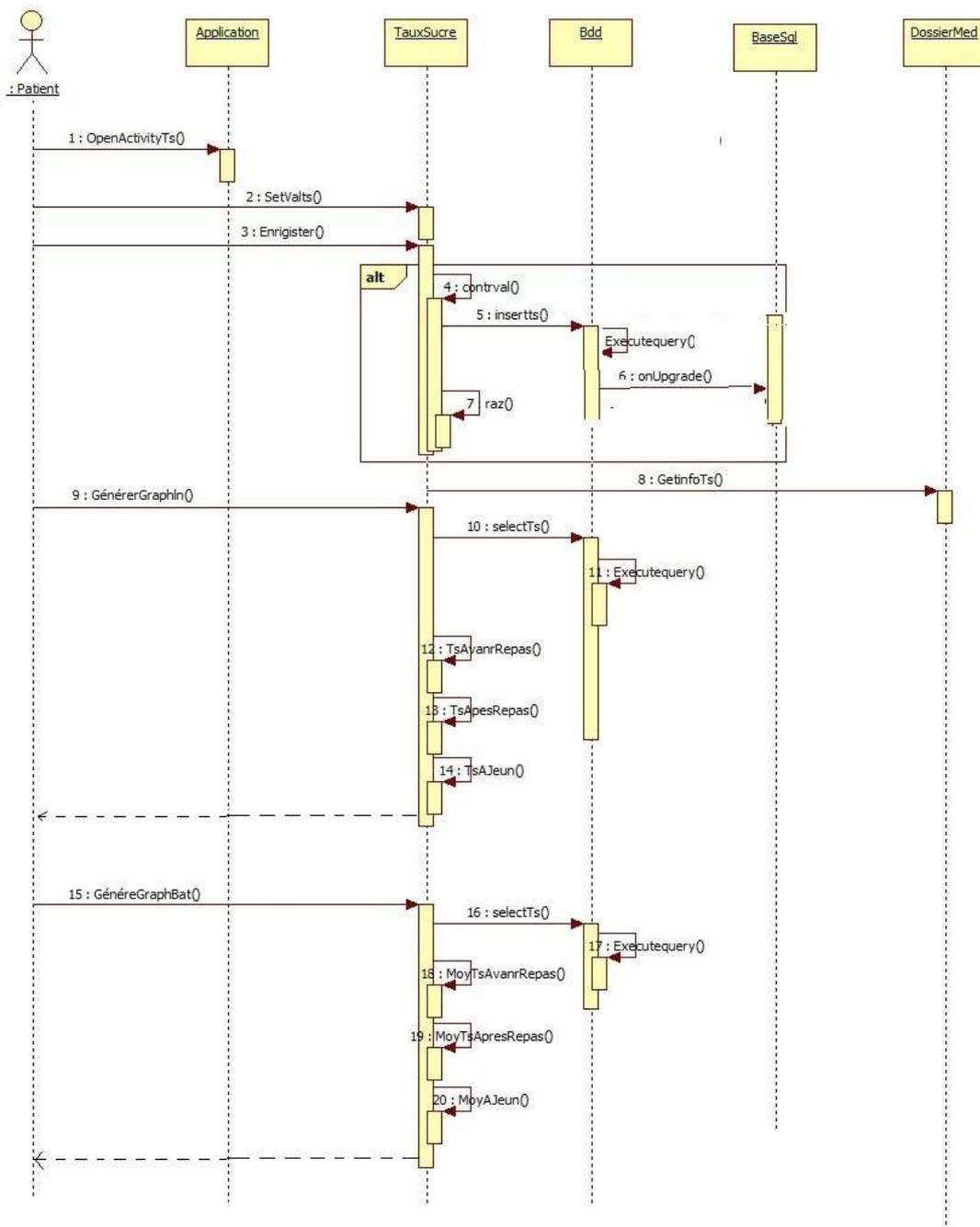


fig2.4Diagramme de séquence " Enregistrer le taux de glycémie ".

### **III.4 Cas "rappel des prises médicamenteuses " :**

#### **III.4.1 Scénario :**

1. Le patient demande l'ajout d'une nouvelle alarme.
2. L'application affiche le formulaire d'ajout d'une alarme.
3. Le patient remplit ce formulaire.
4. L'application génère des nouveaux champs pour entrer l'heure de l'alarme.
5. Le patient remplit les heures et valide le formulaire.
6. l'application contrôle les valeurs entrées par le patient et enregistre les données fournies.
7. L'application compare la date actuelle avec minuit et commence à compter le jour.
8. L'application déclenche des notifications si le jour j est venu.
9. Le patient peut supprimer une alarme inutile.
10. Le patient peut modifier une alarme selon ses besoins (changement de dose ou de durée...)

Ci dessous le diagramme explique le déroulement de différentes séquences du scénario :

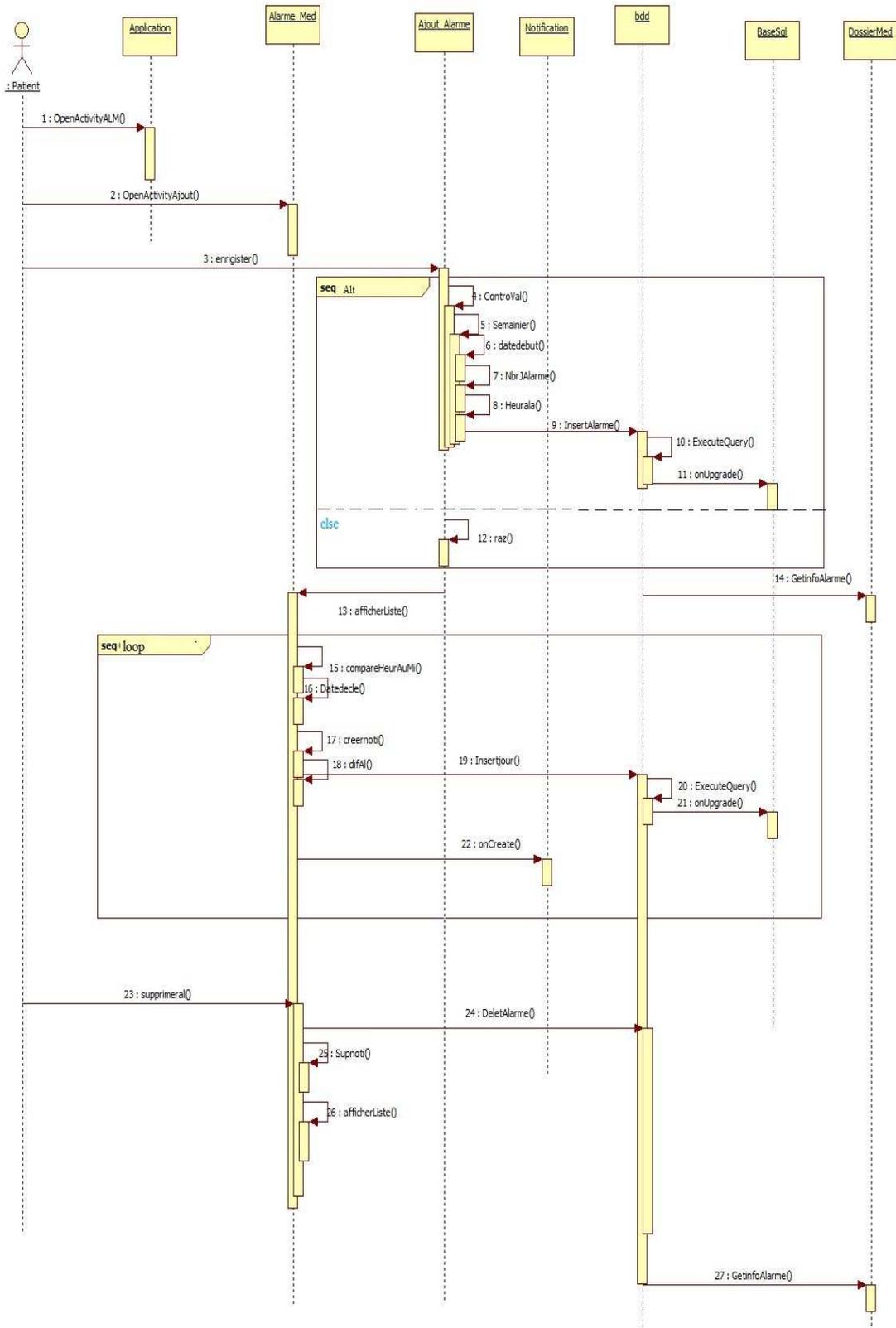


Fig 2.5 Diagramme de séquence " Ajout d'une alarme des prises médicamenteuses".

### III.5 Cas : "Enregistrement des bilans sanguins" :

#### III.5.1 Scénario :

1. Le patient enregistre ses bilans sanguins.
2. L'application contrôle à chaque fois les valeurs entrées par le patient et enregistre les données.
3. L'application complète automatiquement le dossier médical.

Ci dessous le diagramme explique le déroulement de différentes séquences du scénario :

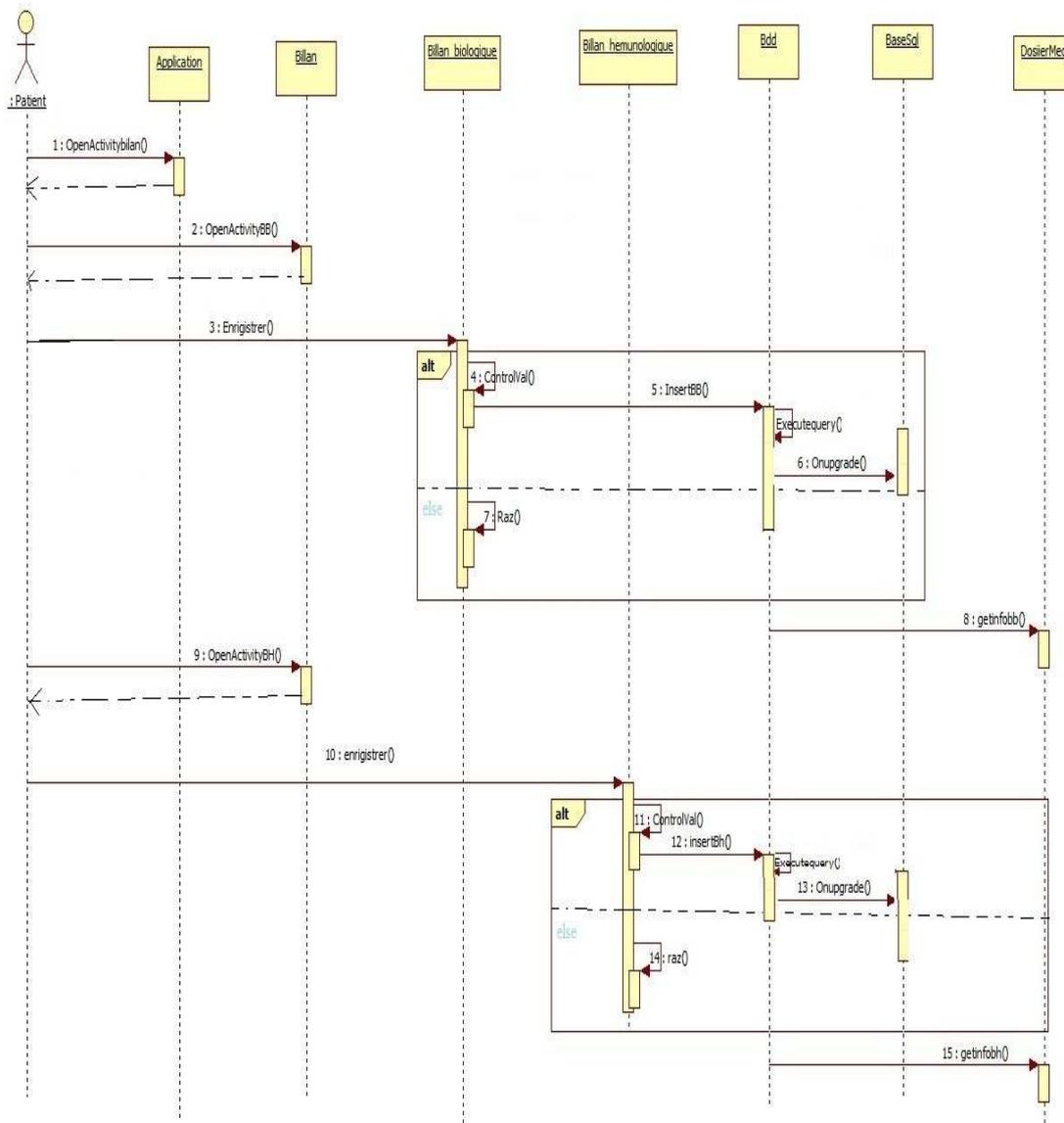


Fig2.6 Diagramme de séquence " Enregistrement des bilans sanguins ".

## III.6 Cas : "Recherche" :

### III.6.1 Scénario :

1. Le patient remplit le champ de recherche.
2. L'application complète le texte automatiquement pour faciliter la recherche.
4. Le patient lance une recherche de la notice du médicament ciblé.
5. L'application accède au code source HTML de ce médicament et filtre les balises du code et le transforme en texte plat.
6. L'application remplace les caractères spéciaux HTML par le caractère adéquat.
7. L'application affiche les informations relatives à ce médicament (Posologie, Indication, Contre-indication .....).

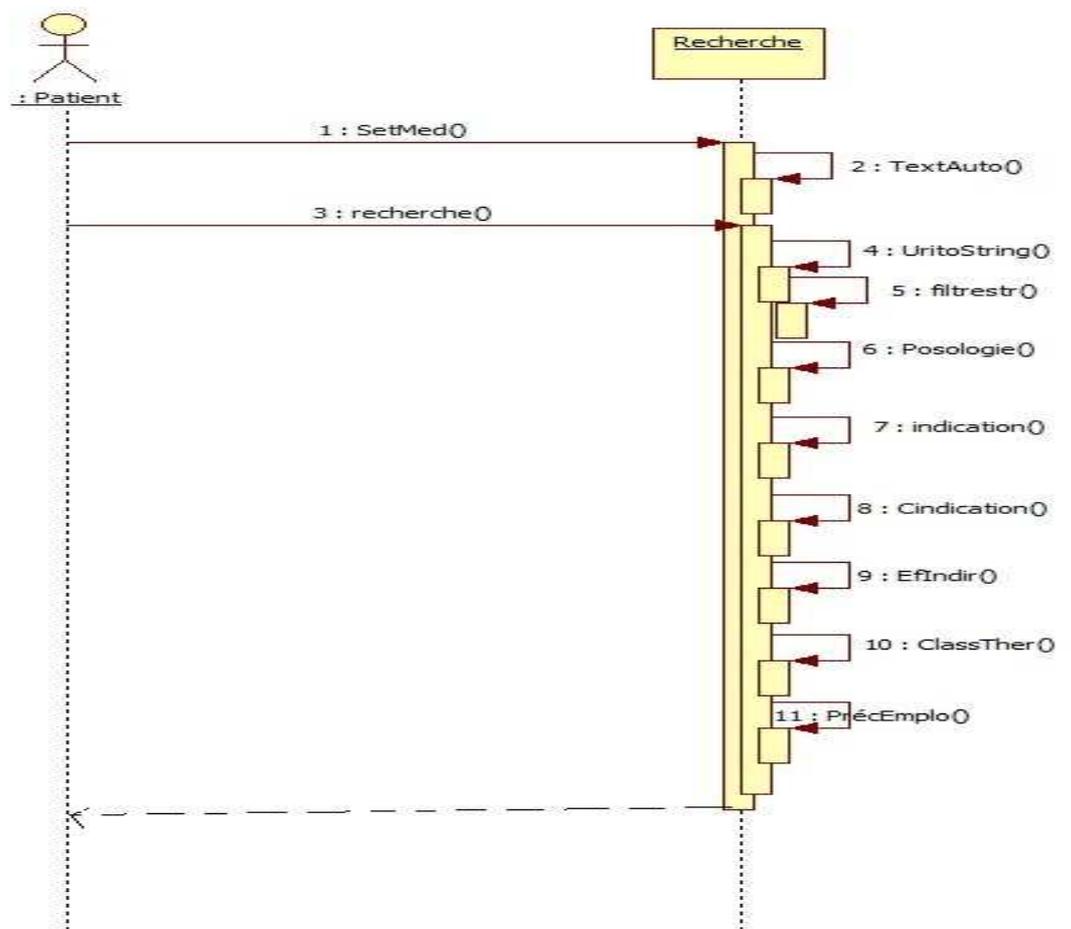


Fig 2.7 Diagramme de séquence " Recherche".

## IV. Diagrammes de classes :

Le diagramme de classes est le point central dans un développement orienté objet. Il a pour objectif de décrire la structure des entités manipulées par les patients. En conception, le diagramme de classes représente la structure d'un code orienté objet.

### IV.1 Cas : "Mesure de l'IMC" :

Dans ce diagramme, on distingue quatre classes principales:

1. Classe « IMC » hérite de la classe « Activity ». Elle contient les méthodes représentées dans le tableau 2.1 :

Méthode	Description
<b>ControlVal()</b>	Sert à contrôler les valeurs entrées par l'utilisateur.
<b>Norme (int p,int imc)</b>	Sert à retourner la norme si l'IMC est anormale.
<b>Enregistrer()</b>	Sert à enregistrer les valeurs (poids, taille, IMC).
<b>GénèreConsigne(int imc)</b>	Sert à générer des consignes si IMC est anormale.
<b>ImcCalcule(int p ,int t)</b>	Sert à calculer IMC a partir des arguments p (poids) et t (taille).
<b>Raz ()</b>	Sert à vider tous les champs entrés.

Tab.2.1 Classe IMC

2. Classe « BaseSql » hérite de « SQLiteOpenHelper » Elle contient les méthodes représentées dans le tableau 2.2 :

Méthode	Description
<b>OnCreate(SqlDataBase db)</b>	Sert à créer une base de données.
<b>Onupgrade(SqlLiteDatabase db)</b>	Elle est exécutée si il y'a un changement dans la création ou modification de la base de données.

Tab.2.2 Classe BaseSql

3. Classe « Bdd » est appelée en cas d'enregistrement ou en cas de modification. Elle contient les méthodes représentées dans le tableau 2.3 :

Méthode	Description
<b>Open()</b>	Sert à ouvrir la base de données.
<b>Close()</b>	Sert à fermer la base de données.
<b>ExecuteQuery(String Query)</b>	Sert à exécuter des requête SQL, la requête est passée comme paramètre.
<b>InsertImc(int p,int t,intimc)</b>	Sert à insérer les valeurs (imc, poids, taille dans la table Bdd_imc.
<b>InsertDs(String nom,String prenom ...)</b>	Sert à insérer les valeurs (nom, prénom, sexe, âge...) dans la table Bdd_DS.
<b>InsertJour(String date)</b>	Sert a enregistrée la date du prochaine alarmes.
<b>InsertTs(int V_mesurer,String etat)</b>	Sert à insérer les valeurs (valeur mesurée et l'état de mesure) dans la table Bdd_Ts.
<b>insertBH()</b>	Sert à insérer les valeurs (taux de globules rouge, taux de globules blanc,...) dans la table Bdd_Bh.
<b>InsertBb()</b>	Sert à insérer les valeurs (LDL, Vldl,...) dans la table Bdd_Tb.
<b>SelectAl(int id)</b>	Sert à charger les informations de l'alarme à partir de son id.
<b>InsertJour(String date)</b>	Sert à insérer la date de la prochaine alarme dans la BDD.
<b>SelectDs(int id)</b>	Sert à charger l'information de dossier médical à partir de son id.
<b>SelectBb(int id)</b>	Sert à charger les informations du bilan biochimique à partir de son id.
<b>SelectBh(int id)</b>	Sert à charger les informations de bilan hymnologique à partir de son id.
<b>RenitiTsDS(int idim,int idts,...)</b>	Sert à réinitialiser tous les coordonnées d'un patient en cas changement de propriétaire d'Alsanté.
<b>SelectImc(int id)</b>	Sert à charger les informations de l'IMC à partir de son id.
<b>ModifDs(int id)</b>	Sert à modifier les informations de dossier médical du patient à partir de son id.
<b>Conter()</b>	Sert à retourner le dernier id enregistré de la table Bdd_ts.
<b>Conter Imc()</b>	Sert à retourner le dernier id enregistré de la table Bdd_Imc.
<b>ConterBb ()</b>	Sert à retourner le dernier id enregistré de la table Bdd_Bb.
<b>SelectBh()</b>	Sert à retourner le dernier id enregistré de la table Bdd_Bh.
<b>ConterAl()</b>	Sert à retourner le dernier id enregistré de la table Bdd_al.
<b>SupprimerAlarme(String nommed)</b>	Sert à supprimer alarme avec son nom de médicament.

Tab.2.3 Classe BDD.

4. Classe « DossierMed » gère le dossier médical du patient. Elle contient les méthodes représentées dans le tableau 2.4:

Méthode	Description
<b>GetInfoImc()</b>	Sert à retourner le dernier enregistrement des valeurs (poids, taille, IMC).
<b>GetInfoTs()</b>	Sert à retourner l'état diabétique du patient.
<b>GetInfoPer()</b>	Sert à retourner les coordonnées du patient (nom, prénom).
<b>GetInfoAl()</b>	Sert à retourner le nom des médicaments pris en cours.
<b>GetInfofobb()</b>	Sert à retourner le rapport du dernier bilan biochimique.
<b>GetInfofobh()</b>	Sert à retourner le rapport du dernier bilan hématologique.
<b>Modifields()</b>	Sert à lancer la fonction de modification.

Tab.2.4 Classe DossierMed.

La Fig.2.8 Représente du diagramme de classe du module "mesurer l'IMC" :

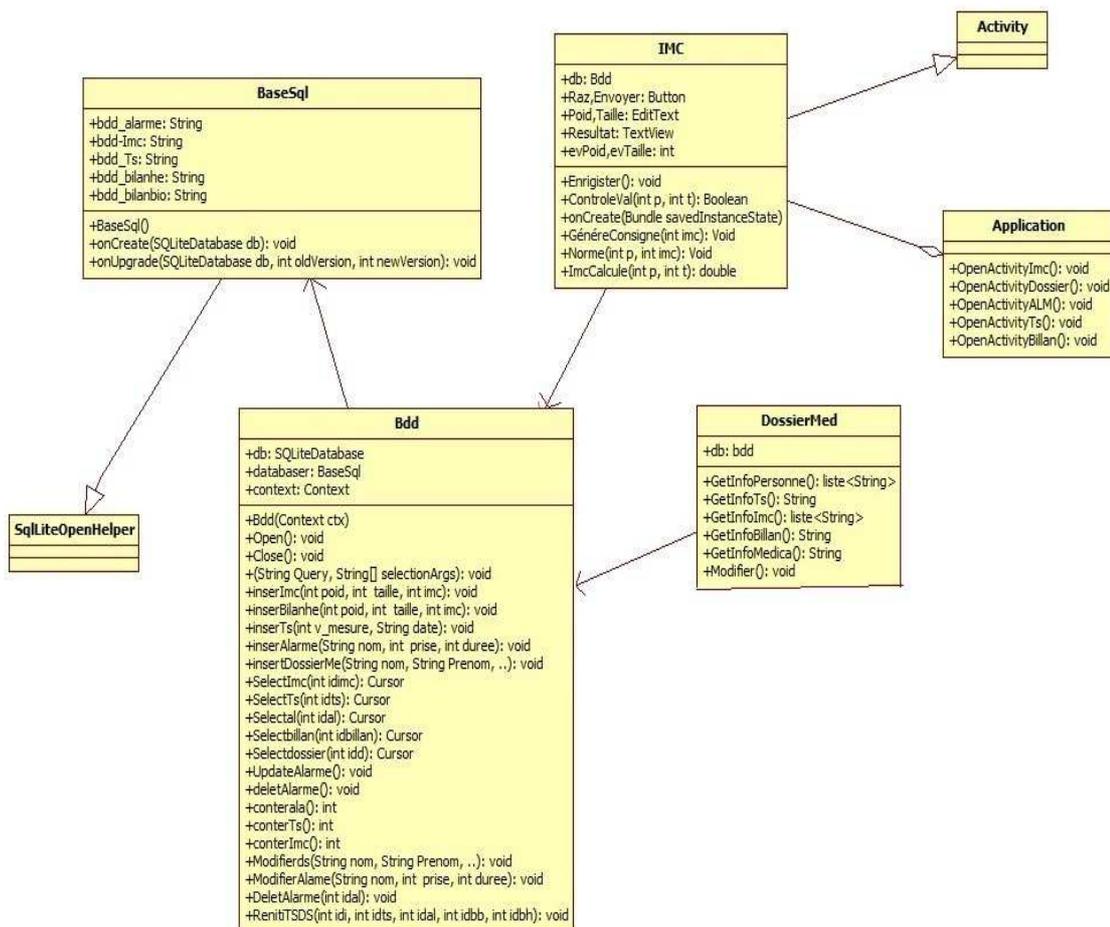


Fig 2.8 Diagramme de classe " mesurer l'IMC ".

## IV.2 Cas: "Création d'un dossier médical" :

Dans ce diagramme, on distingue une seule classe :

La classe « CreeDossier » à pour rôle de créer un dossier médical. Elle hérite de la classe Activity et elle contient les méthodes représentées dans le tableau 2.5 :

Méthode	Description
<b>ControlVal()</b>	Sert à contrôler les valeurs entrées par utilisateur.
<b>Raz()</b>	Sert à vider tous les champs entrés.
<b>Enrigistrer()</b>	Sert à enregistrer les valeurs entrées par l'utilisateur dans la base de données.

Tab.2.5 Classe CreeDossier.

La Fig.2.9 représente le diagramme de la classe « CreeDossier » ainsi que les différentes classes mères.

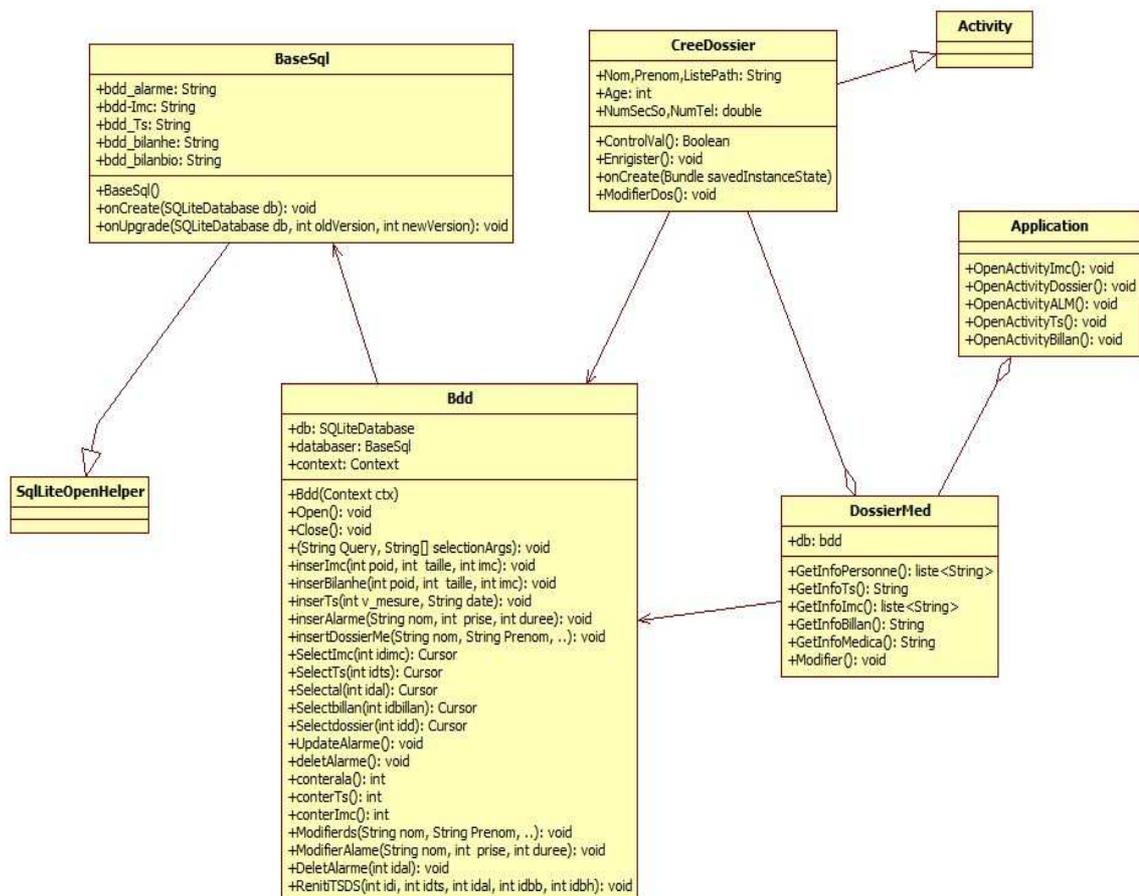


Fig 2.9 Diagramme de classe " CreeDossier ".

### IV.3 Cas: "Enregistrement de taux de glycémie" :

Dans ce diagramme, on distingue une seule classe :

- Classe « TauxSucre » : elle hérite de la classe « Activity » et elle contient les méthodes représentées dans le tableau 2.6 :

Méthode	Description
<b>ControlVal()</b>	Sert à contrôler les valeurs d'entrées par le patient.
<b>Enrigistrer()</b>	Sert à enregistrer les valeurs entrées par le patient dans la base de données.
<b>générerGraphligne()</b>	Sert à fournir un graphe de type ligne, des différentes valeurs enregistrer dans la base données et cela avec un filtrage auparavant (avant repas, après repas, à jeune) en utilisant Google Charting API (Annexe H).
<b>générerGraphbat()</b>	Sert à donner un graphe de type bâtons des différentes valeurs enregistrer dans la base données après le calcul de la moyenne des valeurs filtrées par (avant repas, après repas, à jeun) en utilisant toujours Google Charting API.
<b>TsApresrepas()</b>	Sert à retourner une chaine de caractère contenant les différentes valeurs enregistrées après repas.
<b>TsAvantrepas()</b>	Sert à retourner une chaine de caractère contenant les différentes valeurs enregistrées avant repas.
<b>TsAjeune()</b>	Sert à retourner une chaine de caractère contenant les différentes valeurs enregistrées à jeun.
<b>MoyTsApresrepas()</b>	Sert à calculer la moyenne des différents taux de glycémie après repas.
<b>MoyTsAvantrepas()</b>	Sert à calculer la moyenne des différents taux de glycémie avant repas.
<b>MoyTsAjeune()</b>	Sert à calculer la moyenne des différents taux de glycémie à jeun.

Tab.2.6 Classe TauxSucre.

La Fig.2.10 représente le diagramme de la classe « TauxSucre » ainsi que les différentes classes appelées, composites et mères.

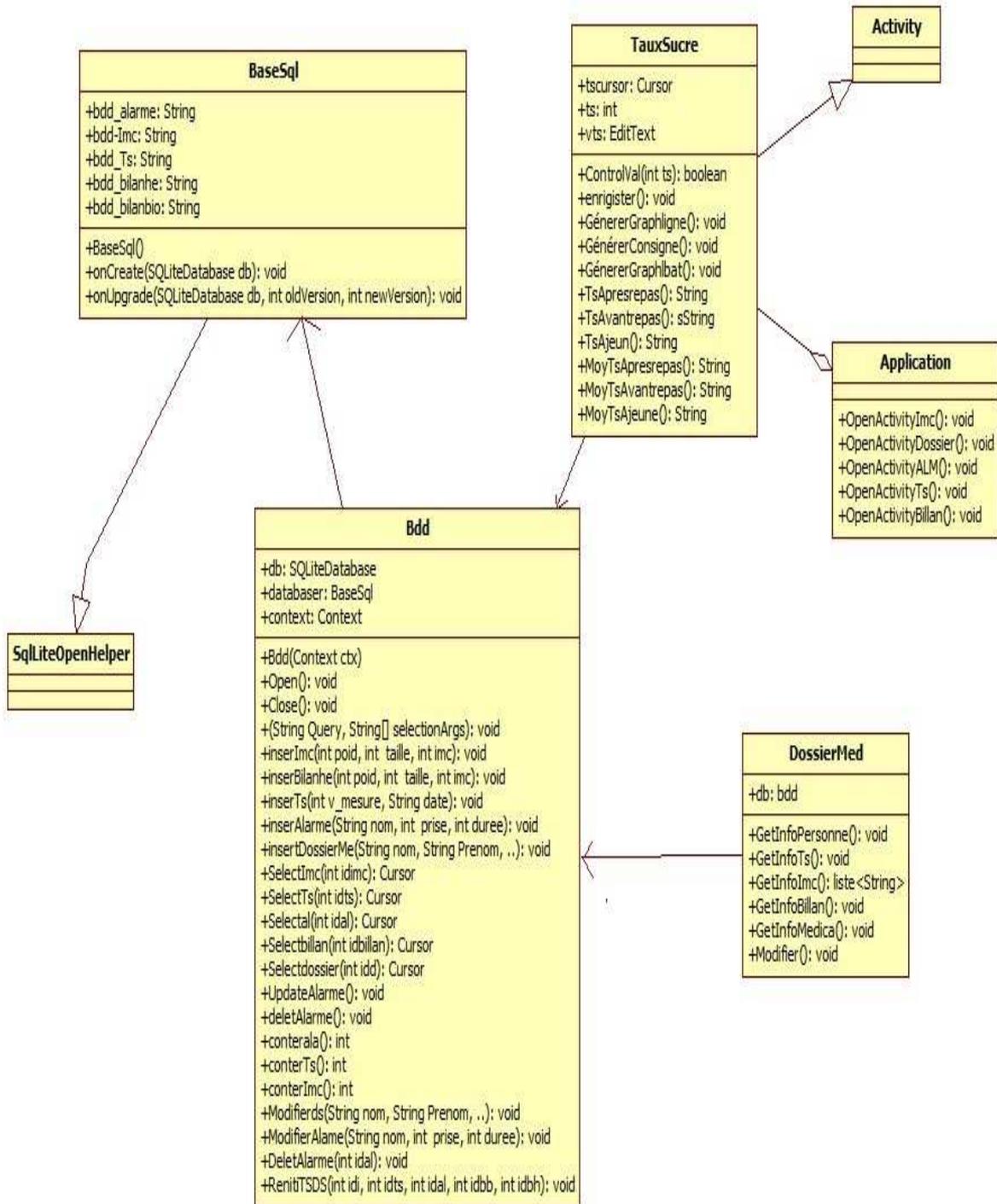


Fig 2.10 Diagramme de classe " Enregistrement de taux de glycémie".

#### IV.4 Cas : "Rappel des prises médicamenteuse":

Ce diagramme contient deux principales classes :

1. Classe « Alarme » contient les méthodes représentées dans le tableau 2.7:

Méthode	Description
<b>AfficherListe()</b>	Sert à afficher la liste des alarmes programmées par le patient.
<b>diffAl(String heural)</b>	Retourne en milliseconde la différence entre heur alarme et minuit.
<b>creernoti(String nomme, long dif)</b>	Sert à supprimer des notifications
<b>prochaineDate(String date)</b>	Retourne la date de prochaine alarme
<b>CompareDate(String datedec)</b>	Retourne ture si la date aujourd'hui égale la date passée en paramètre
<b>Datedecle(): string</b>	Date de déclenchement de la prochaine alarme.
<b>supnoti(String nom)</b>	Sert a supprimer les notification déclencher par son nom.
<b>supprimeral(String nom)</b>	Sert à supprimer l'alarme dans la BDD.

Tab.2.7 Classe Alarme.

2. La classe « Ajout\_Alarme » contient les méthodes représentées dans le tableau 2.8 :

Méthode	Description
<b>ControlVal()</b>	Sert à Contrôler les valeurs d'entrées par le patient.
<b>Enregistrer()</b>	Sert à enregistrer les valeurs entrées par le patient dans la base de données.
<b>Semainier()</b>	Retourne la fréquence de l'alarme sur les jours.
<b>DateDebut()</b>	Retourne la date courante.
<b>Heurala()</b>	Retourne l'ensemble des heurs d'alarmes.
<b>NbrJAlarme()</b>	Retourne le nombre du jour de l'alarme.

Tab.2.8 Classe Ajout\_ Alarme.

La classe « Notification » contient la méthode Oncreate() qui sert a crée une interface alerte . Le diagramme dans la Fig.2.11 représente le diagramme de classe " rappel des prises médicamenteuses ".

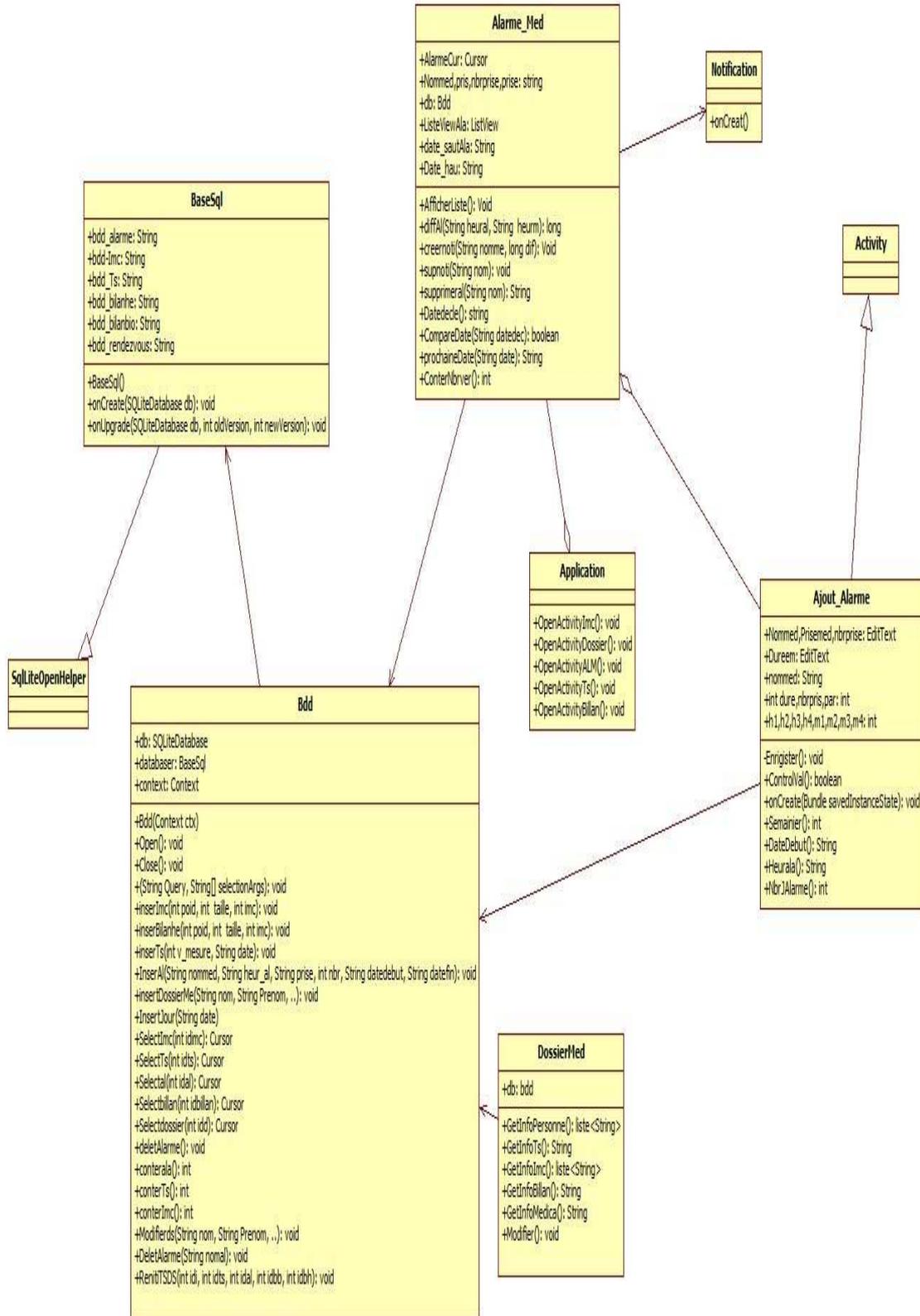


Fig. 2.11 Diagramme de Classe " rappel des prises médicamenteuses".



## IV.6 Cas: "Recherche" :

Dans ce diagramme, on distingue une seule classe "Recherche" qui contient les méthodes décrites dans le tableau 2.10.

Méthode	Description
textAuto()	Permet un auto-remplissage de champs basés sur plus que 3000 noms de médicaments intègrent dans le fichier Resource Android (Annexe A).
Uritostring()	Sert à convertir le code source HTML en texte plat (sans balises HTML).
Indication ()	Retourne les indications à propos du médicament recherché.
Cindication()	Retourne les contre-indications à propos du médicament recherché.
préEmplo ()	Retourne les précautions d'emploi à propos du médicament recherché.
Classether()	Retourne la classe thérapeutique à propos du médicament recherché.
Posologie()	Retourne la posologie du médicament recherché.

Tab.2.10 Classe Recherche.

Le diagramme dans la Fig.2.13 représente le diagramme de classe " Recherche":

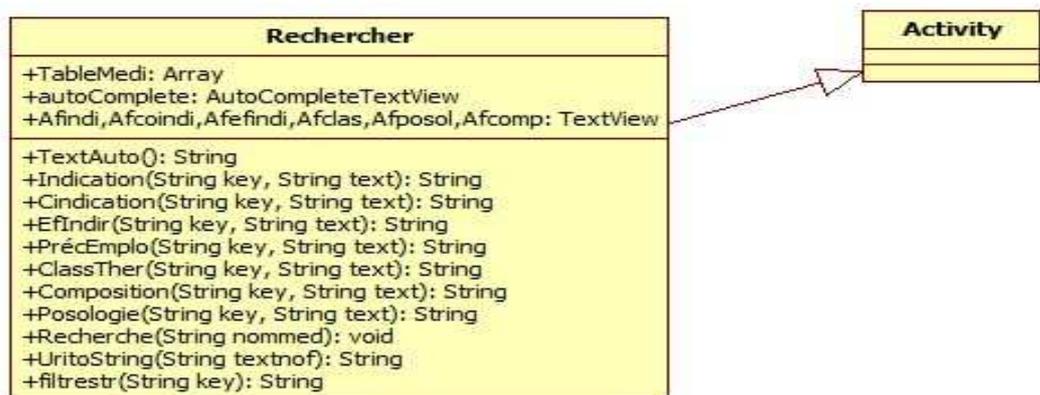


Fig. 2.13 Diagramme de Classe "recherche ".

## IV.7 Diagramme de classe générale :

Le diagramme de classe général de l'application d'*AISanté* est présenté dans la Fig.2.14

## Chapitre 2 : Conception

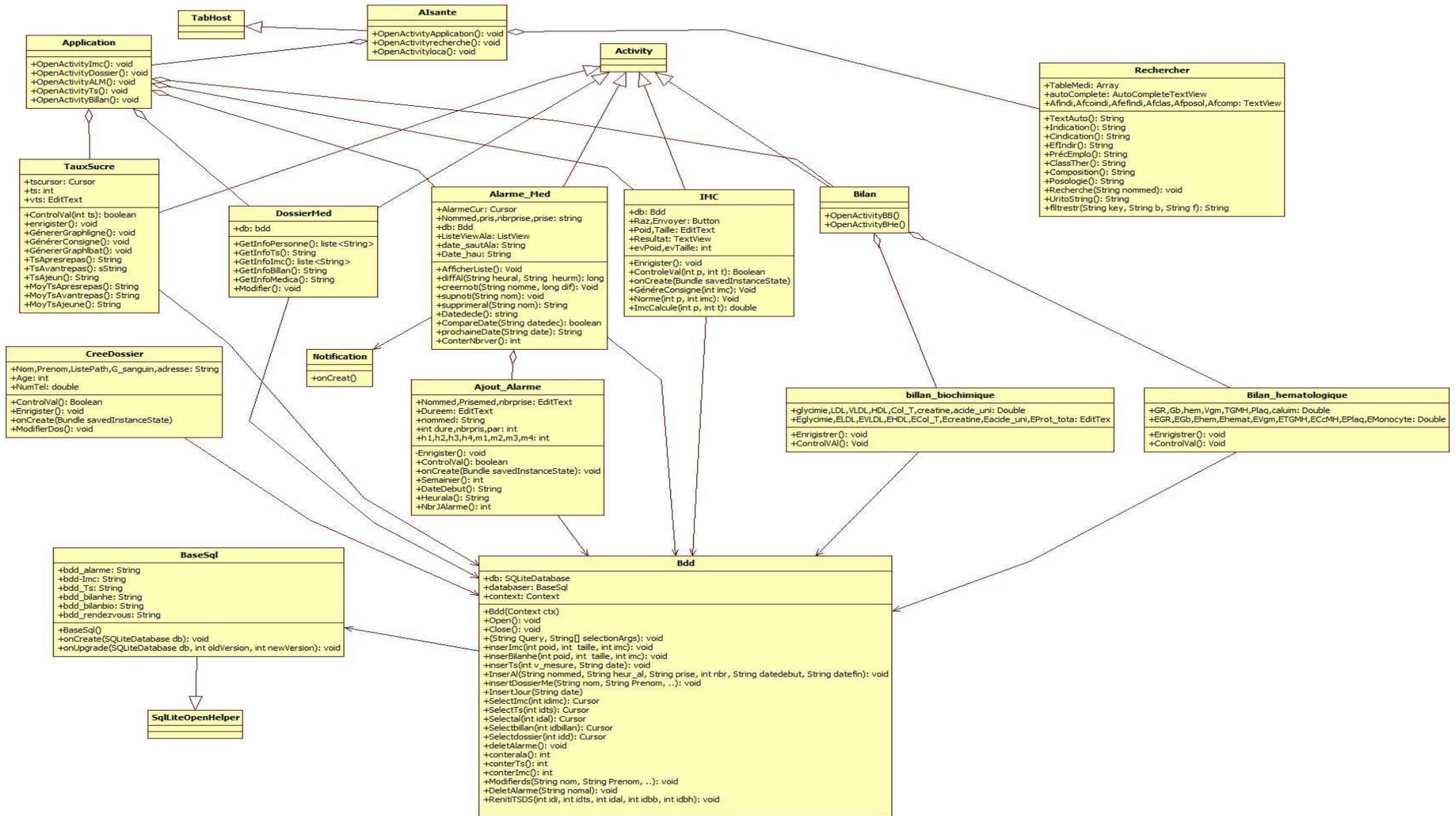


Fig. 2.14 Diagramme de Classe AISanté.

## V.OCL:

OCL [6] est utilisé pour gérer à bien les contraintes liées à une classe *C* et se distingue en trois catégories d'OCL qui peuvent être applicables à une contrainte :

- **Inv** : spécifications d'invariants, i.e. une condition qui doit être vraie pour chaque instance de la classe en permanence.
- **pre** : spécifications de pré-conditions des opérations, i.e. expressions qui permettent de spécifier des contraintes qui doivent être vérifiées avant l'appel d'une opération.
- **post** : spécifications de post-conditions des opérations, i.e. expressions qui permettent de spécifier des contraintes qui doivent être vérifiées après l'exécution d'une opération.

Les contraintes décrites ci-dessous ont été appliquées pour *AI santé*.

### V.1 Contrainte N° 1 :

Le poids d'une personne est compris dans un intervalle qui prend en charge des personnes dont le poids varie de 40kg (poids minimal) jusqu'à 600kg (poids maximal).

```
Context IMC Inv (evpoid)>40 and (evpoid <600)
```

### V.2 Contrainte N° 2 :

La taille d'une personne est comprise dans un intervalle qui prend en charge des personnes dont la longueur varie de 150cm (valeur minimal) jusqu'à 270cm (valeur maximal).

```
Context IMC inv. (evtaille>150) and (evtaille<270)
```

### V.3 Contrainte N° 3 :

Calcul de la valeur de l'IMC d'une personne avec les paramètres : (poids, taille) selon l'équation décrite ci-dessous :

```
Context IMC ::ImcCalcule(int p,int,t) :double  
Post result=evpoid/(evtaille*evtaille)
```

## V. 4 Contrainte N° 4 :

La valeur de l'IMC suivra un conditionnement en cas d'une valeur anormale par le biais d'une norme ; norme qui doit être fixée dans un intervalle qui lui est propre.

```
Context IMC ::norme (double imc,int p) :String  
def : norme_inf=(18.5*imc)/p , (norme_sup=25*imc)/p
```

## V. 5 Contrainte N° 5 :

L'âge d'une personne est toujours positif et il ne doit pas dépasser 140ans.

```
Context CreeDossier Inv (age)>15 and (age <130)
```

## V.6 Contrainte N° 6 :

Le nombre de médicaments pris par jour est obligatoirement compris entre les valeurs [1,4].

```
Context Ajout_Alarme inv (par>0) and (par<5)
```

## V.7 Contrainte N° 7:

Les heurs alarmes introduit sont obligatoirement compris entre les valeurs [0,23] et pour les minutes entre [0,60]

```
Context Ajout_Alarme inv (h1>=0) and (h1<=23)  
Context Ajout_Alarme inv (m1>=0) and (m1<=60)
```

## V.8 Contrainte N° 8:

Le semainier doit être obligatoirement inférieur ou égale a la durée du traitement

```
Context Ajout_Alarme inv (par<dure)
```

## V.9 Contrainte N° 9 :

Le taux de glycémie doit être supérieur à 0 et inferieur à 6

```
Context Tauxsucre inv tg>0 and (tg<6)
```

## V.10 Contrainte N° 10:

Les champs nom et prénom, adresse, âge et groupe sanguin doit être obligatoirement saisie

```
Context d :creer_dossier
Inv : (d.nom<>'') and (d.prénom<>'') and (d.age<>'') and (G_sanguin<>''), and
(adresse<>'')
```

## V.2.11 Contrainte N° 11:

Le nombre de jour de traitement est le résultat de durée de traitement sur le semainier

```
Context Ajout_Alarme:: NbrJAlarme(): int
```

```
Post : resultat= duree/par
```

## V.2.12 Contrainte N° 12:

Les valeurs saisie dans les champs de bilan hématologie doit être contrôler dans des intervalles respective

```
Context bilan_biochimique
Inv : ((glycimie>0 )and (glycimie<6)) and ((LDL>0)and (LDL<5))and((VLDL>0) and
(VLDL <3 ))and ((HDL>0) and (HDL <3 ))and ((Col_T>0) (Col_T <7)) and
((creatine>0) and (creatine <30))and((acide_uni>0) and( acide_uni <200))
```

## V.2.13 Contrainte N° 13:

Les valeurs saisie dans les champs de bilan hématologie doit être contrôler dans des intervalle respective

```
Context Bilan_hématologique
inv :(((GR >2 )and (GR <8) )and( (Gb>0)and (Gb<30000))and((hem >0) and (hem <50 ))and
((Vgm >0) and (Vgm <3 ))and ((TGMH >0) (TGMH <200)) and ((Plaq >0) and (Plaq <1000000))
```

## VI. Conclusion :

La phase de conception a été établie et dont l'objectif principal était de réaliser les différents diagrammes UML. le chapitre suivant décrira la phase de développement et implémentation de l'application de *AISanté* .

# **Chapitre 3 :**

# **Réalisation**

## I. Introduction :

Le déroulement logique qui succède la conception est le développement ; la concrétisation de cette application reposera sur la théorie de toutes les phases, les schémas et les diagrammes précédemment étudiés.

Dans cette partie, il sera possible d'évoquer toutes les fonctionnalités réalisées à ce jour.

## II. L'application :

### II.1 La Fenêtre principale :

La fenêtre principale Fig.3.1 est composée de plusieurs icônes qui permettent un accès rapides aux différents modules proposés par l'application :

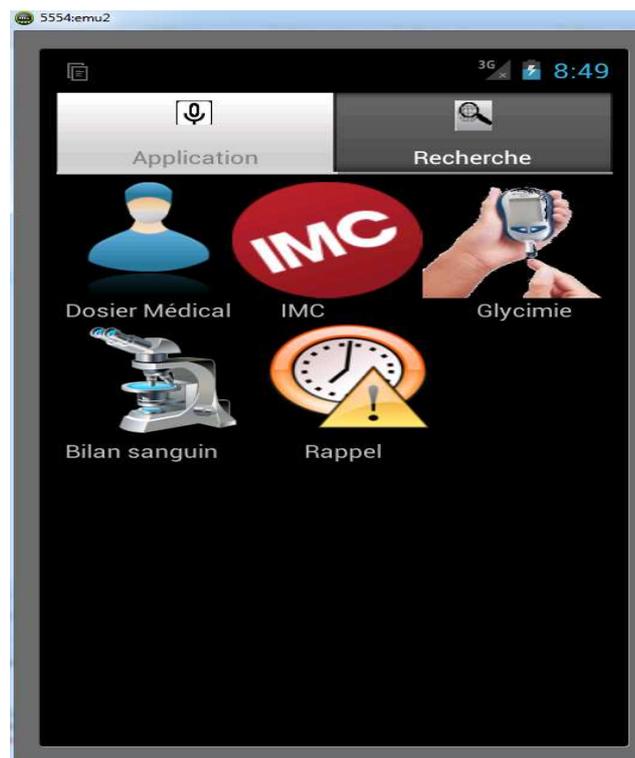


Fig. 3.1 Fenêtre principale.

## II.2 La Création d'un dossier médical :

L'accès au dossier médical pour la toute première fois une icône + est disponible (voir Fig.3.2). Le patient doit cliquer dessus pour pouvoir créer un son dossier médical.

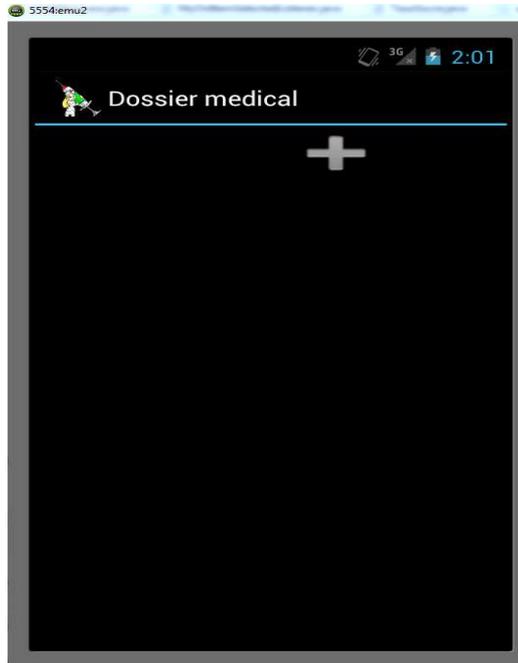


Fig. 3.2 Création d'un dossier médicale.

Un formulaire apparaîtra aussitôt, il doit être soigneusement rempli par le patient(voir Fig.3.3)

Nom:	kherbouche
Prenom:	abdessamade
Sexe:	homme
Age:	25
Adresse:	dine imama tlemcen
Groupe sanguin:	A+
Num tel personne a contacter:	83 32
pathologie	<input checked="" type="checkbox"/> Alergie <input type="checkbox"/> Asthme
	<input type="checkbox"/> Ultaire <input checked="" type="checkbox"/> Gastrique
	<input type="checkbox"/> Diabète

Fig. 3.3 Formulaire de création d'un dossier médicale.

Après la validation du formulaire un écran (voir Fig.3.4) apparaît.



Fig. 3.4 Validation du formulaire de création.

### II.3 Calculer l'indice de la masse corporelle IMC :

Quand on accède au module IMC à partir de la fenêtre principale (voir Fig.3.5), le patient doit fournir sa taille en cm et son poids en Kg et l'application s'en charge de calculer son IMC.

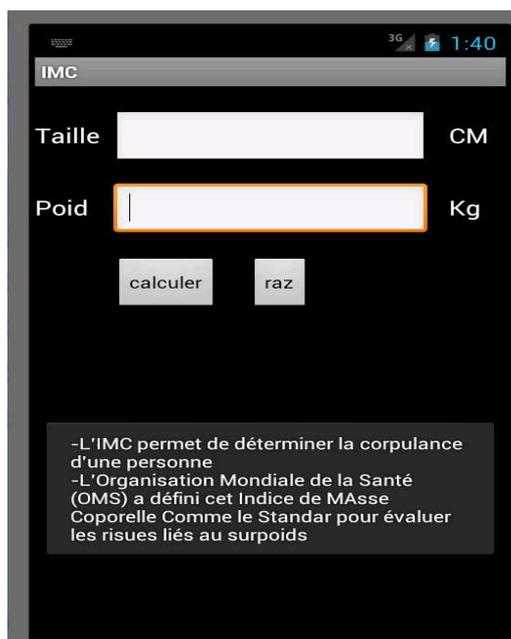


Fig. 3.5 Fenêtre calcule de l'IMC.

L'application calcule l'IMC du patient et affiche un message d'information visuel et audio si il est en dehors de la norme pour lui prévenir de sa non-conformité avec les normes et le conseille des valeurs d'IMC normatifs comme la Fig.3.6 le montre.



Fig. 3.6 Résultat de l'IMC.

## II.4 Taux de glycémie :

Une fois que le patient accède au module à partir de la fenêtre principale, ce dernier doit fournir le taux de glycémie mesuré ainsi que la période quotidienne de la prise (voir Fig.3.7). Cela facilitera le tri, et la présentation des différentes valeurs prises auparavant.



Fig. 3.7 Taux de glycémie.

Le patient peut obtenir un graphe de variation de son taux de glycémie au fil du temps. Pour cela il doit cliquer sur l'icône 'graphe' qui proposera un graphe de variation du taux de glycémie et de la moyennes des différents taux enregistrés (voir Fig.3.8 et Fig.3.9) :



Fig. 3.8 Graphe de variation du taux de glycémie.



Fig. 3.9 Moyenne des taux de glycém

## II.5 Bilans sanguins :

Le module « bilans sanguins » permettra d'enregistrer les différents résultats des derniers bilans sanguins effectués par le patient à l'aide du formulaire représenté dans la Fig.3.10 et Fig. 3.11. On peut distinguer deux onglets ; le premier pour remplir les résultats du bilan biochimique et l'autre pour remplir les résultats du bilan hématologique.

Paramètre	Valeur	Unité
Glycémie:	1.1	g/l
Chole-total:	3	g/l
HDL:	0.5	g/l
LDL:	1	g/l
VLDL:	0.1	g/l
Acide urique:	70	mg/l
Créatine:	16	mg/l

Fig. 3.10 Bilans biochimiques.

Paramètre	Valeur	Unité
Globules Rouge:	2	millions / ml
Globules Blancs:	5000	millions / ml
Hémoglobine:	15	g / 100 ml
VGM:	70	/m3
TGMH:	35	picogramm
Plaquettes:	150000	/ mm3
VS:	1.2	mm

Fig. 3.11 Bilans hématologiques.

## II.6 Rappel des prises médicamenteuses :

Pour accéder à ce module, il faut cliquer sur l'icône 'Horloge ' dans la fenêtre principale (voir Fig.3.12). la possibilité de consulter toute les rappels en cours de programmation est possible tout comme le fait d'ajouter ou de supprimer les rappels non valide. Quant à l'accès au module pour la première fois une icône + apparait (voir Fig.). Le patient doit cliquer dessus pour pouvoir créer une nouvelle alarme.

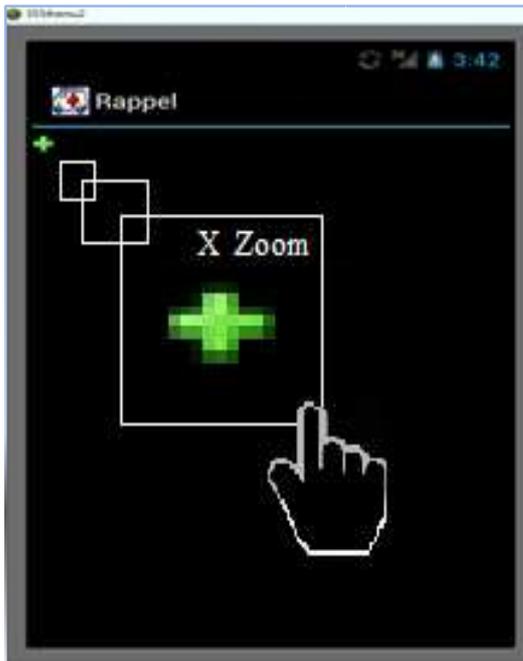


Fig 3.12 Création d'une alarme.

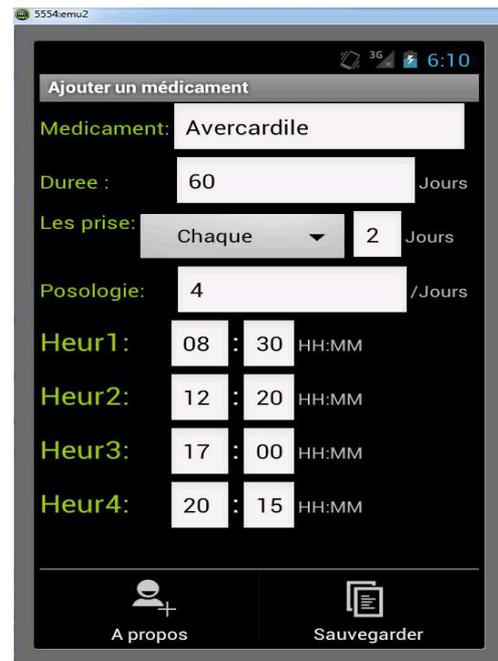


Fig.3.13 formulaire détaillant l'ajout d'un nouveau rappel.

Il faut cliquer sur la croix rouge et saisir le nom du médicament destiné dans la zone de remplissage qui est apparue.



Fig. 3.14 Supprimer un rappel.

Pendant ce temps, le dossier médical personnel continue à s'enrichir au fur et à mesure que le patient enregistre les différentes informations relatives à son état de santé (voir Fig.3.14).



Fig. 3.15 Dossier médical personnel.

## II.7 Recherche :

Pour accéder à ce module, il existe dans la fenêtre principale un onglet « recherche » qui permettra d'effectuer une recherche et proposera un médicament *M*.

Le patient commence par saisir le nom du médicament dans le champ recherche tout en utilisant un système d'auto-remplissage qui facilitera la saisie. Puis il cliquera sur le bouton 'recherche'.

Les résultats seront triés, filtrés et affichés. Les écrans dans la Fig.3.15, Fig.3.16 et Fig.3.17 montrent ces différentes explications :



Fig. 3.16 Recherche.



Fig. 3.17 Auto remplissage.



Fig. 3.18 Résultats de recherche

Un autre exemple de recherche (PRANEVITAMINE C):

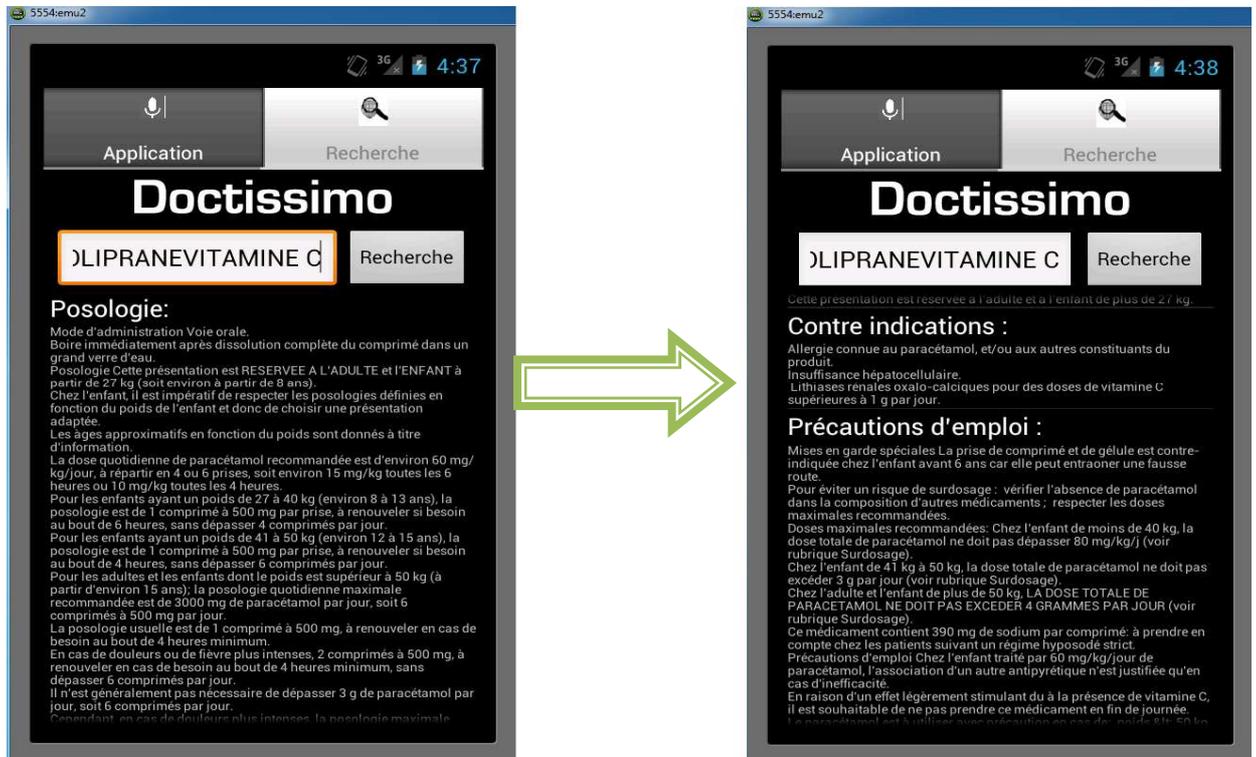


Fig. 3.19 Résultats de recherche 2.

### III. Conclusion :

Ce segment s'est matérialisé par une présentation de la phase de réalisation dont l'objectif principal était de détailler l'implémentation de cette application tout en se basant sur les données des chapitres antérieurs.

# **Conclusion et perspectives**

Les objectifs de ce travail s'étaient avérés multiples pour les plans personnels de leurs concepteurs respectifs qui ont recherché à fructifier et améliorer leur application pour pouvoir avoir un impact qui se voudrait « positif » pour faciliter professionnellement parlant leur application et apporter un plus non négligeable dans le domaine « médico-virtuel ». Il a été permis de consolider toutes les connaissances acquises et de matérialiser cette application mobile sous *Android*.

La concrétisation de ce projet à mener au conclure une réalisation de taches à 90% qui à engendrait une incapacité dans fonctionnement de certains modules (le module de rappel).

Cette application pourra être sujette à une amélioration et une évolution en intégrant différents points qui n'ont pas pu être intégrés dans ce projet et qui devraient bonifier l'expérience et optimiser l'utilisation de l'application, les points cités en sont quelques exemples :

- Dans un souci de précision ; une extension de la recherche pour les différentes pathologies.
- Travailler en collaboration avec des médecins et des biologistes pour performer l'analyse des bilans sanguins
- Travailler en association avec des ingénieurs de génie-bio-médicale et électroniques pour développer des kits pour mesurer le taux de glycémie automatiquement ainsi que les pulsations cardiaques.
- Elargir le module de bilan sanguin en intégrant d'autres bilans (Le bilan hépatique .Le bilan rénal .....).
- L'ajout d'autres modules pour compléter le dossier médical (Module de carnet vaccins, un module gestionnaire du rythme cardiaque )

- l'ajout d'un nouveau module de géo localisation qui permettra de localiser le patient ainsi que les centres de soins, les pharmacies et les hôpitaux les plus proches de ce dernier.

De plus, il est envisageable d'adapter cette application pour fonctionner sur un réseau. En effet, cela permettra de centraliser les données des différents patients, de faciliter l'accès aux corps médical à distance sans utiliser le support mobile pour venir en aide rapidement au patients et d'établir aussi des statistiques médicales relatives aux informations enregistrées sur les différents patients (nombre de diabétiques ; nombre d'obèses ou d'anorexiques; Problèmes cardiaques ...).

# **Annexes**

## Annexe A: Doctissimo[7]

Doctissimo est un site Web francophone consacré à la santé et au bien-être, propose plusieurs grandes catégories relatives à la santé : « santé », « médicament », « psychologie », « grossesse/bébé », « nutrition »

L'objectif de doctissimo.fr est d'informer les gens au sujet de leur santé. C'est pour cela que les sujets traités sont très diversifiés allant des petits maux quotidiens aux grandes maladies et médicaments. Des conseils sont alors disponibles : des médecins spécialistes écrivent des articles et donnent des conseils en fonction des questions des visiteurs. Leurs contenus sont très riches et ils sont faciles à comprendre

## Annexe B : Android

### La plateforme[8]

Android est la réponse de Google à la volonté de l'Open Handset Alliance de réaliser des normes ouvertes pour les terminaux mobiles.

Android est un système d'exploitation basé sur un noyau Linux pour terminaux mobiles. Il était initialement développé par la société Android Inc., cette start-up a été rachetée par Google. La première version d'Android a été publiée le 5 novembre 2007.

### Architecture d'Android

Le schéma suivant illustre les principaux composants du système d'exploitation Android. Chaque section est décrite plus en détail ci-dessous.

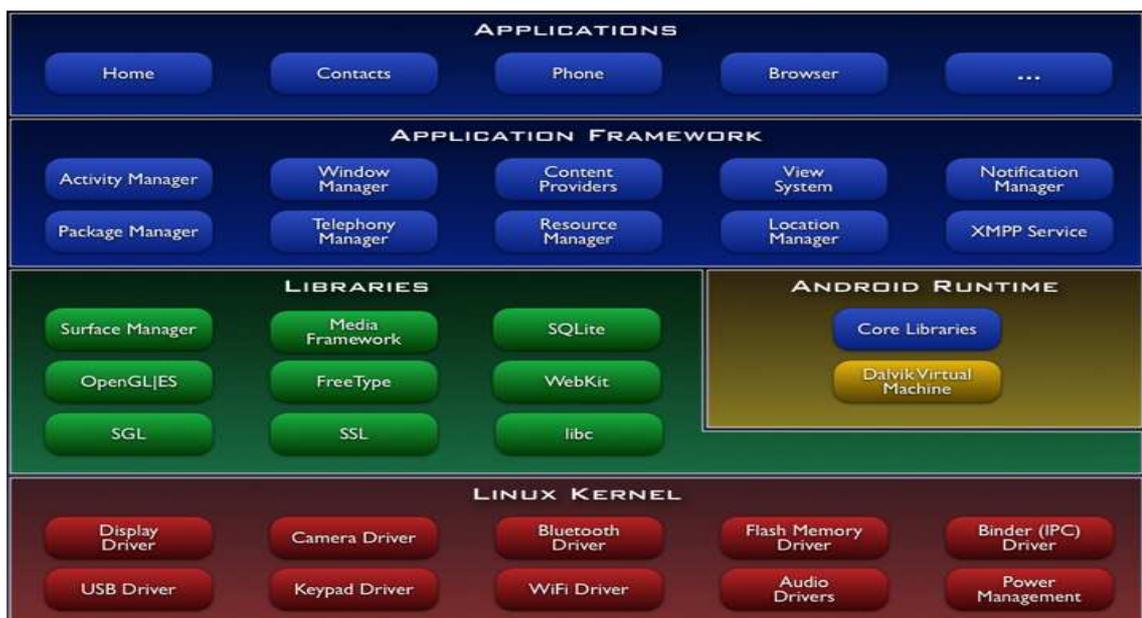


Fig. B.1 Fig. Architecture du système d'exploitation Android

- **Linux Kernel** : le noyau Linux 2.6 pour les services système de base.
- **Android Runtime** : Chaque application Android s'exécute dans un processus, avec sa propre instance de la machine virtuelle Java, appelée Dalvik Virtuel machine. Dalvik a été écrit pour optimiser l'exécution d'une multitude d'instances de la machine virtuelle, avec une empreinte mémoire réduite.
- **Librairies**: Android fournit un ensemble de librairies C/C++ utilisées par différents composants du système. Ces fonctionnalités sont rendues disponibles aux développeurs au travers du framework d'application d'Android. On trouve parmi ces librairies: librairie C standard, moteurs d'affichage 2D et 3D, SQLite, rendu des polices de caractères etc.
- **Application Framework** : Le framework d'application est la couche qui nous intéresse tout particulièrement. C'est elle qui fait le lien, grâce à un ensemble d'APIs Java, entre le système et l'application. Étant un système ouvert, Android permet aux développeurs de concevoir des applications très riches et de tirer partie d'un maximum de fonctionnalités. Les développeurs ont donc accès aux mêmes fonctionnalités que celles utilisées par les applications fournies avec Android.

## **Le SDK Android [9]**

Le SDK Android fournit des outils et des API nécessaires pour développer des applications sur sa plate-forme en utilisant le langage de programmation Java.

SDK Android est le l'outil le plus important Facile à installer, il permet de télécharger tous les outils indispensables au développement d'applications.

Le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android. C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel.



Fig. B.2 L'Emulateur Android

Le SDK est fournit avec encore d'autres outils afin de favoriser la productivité on peut citer :

- **apkbuilder** :pour créer des « livrables », il s'agit d'un paquet prêt à être installé sur un appareil
- **mksdcard** :permet de créer une « fausse » carte SD pour notre émulateur

## Les concepts de base Android [10]

Une caractéristique de base d'Android est la possibilité pour une application d'utiliser des éléments d'une autre application (à condition que cette dernière le permette). Par exemple, si votre application a besoin d'afficher une liste d'images et une autre application possède une liste de défilante élégante, et autorise son utilisation, vous pouvez alors utiliser cette liste pour faire votre travail au lieu de développer la votre. Votre application n'est pas liée et n'intègre pas le code de l'application étrangère. Simplement, le système démarrera cet élément quand votre application en aura besoin.

Pour ce faire, le système doit être capable de démarrer le processus d'une application quand n'importe qu'elle partie de celle-ci est demandée, et instancier les objets Java nécessaire. Pour pouvoir effectuer ceci, les applications Android, contrairement à de nombreuses applications dur d'autres systèmes, ne possède pas un unique point d'entrée pour toute l'application (pas de fonction `main ()` par exemple). A

la place, les applications possèdent de composants que le système peut instancier et exécuter au besoin. Android utilise quatre composants de base

- **Activité (Activity)** : Une activité présente une interface utilisateur pour un objectif. Par exemple, une activité peut présenter une liste d'items pour que le patient puisse choisir un élément ou afficher une image avec la possibilité de zoomer. Chaque activité doit hériter de la classe « Activity ».
- **Services (Services)** : Un service n'a pas d'interface utilisateur, il s'agit plutôt d'instructions s'effectuant en tâche de fond pour une période indéfinie. Les services à développer doivent hériter de la classe « Service ».
- **Les écouteurs de messages (Broadcast Receivers)** : il s'agit qui ne fait rien d'autres que recevoir et réagir aux signaux diffusés dans tout le système. De nombreux signaux sont déjà intégré au système, par exemple une notification lorsque la batterie est faible ou lors d'un appel entrant ou alerter l'utilisateur. Les applications peuvent aussi envoyer leurs propres signaux, par exemple pour prévenir les autres applications qu'on vient se déplacer de 100 mètres ou une .
- **Les fournisseurs de contenu (content providers)** : un fournisseur de contenu permet de publier un ensemble d'information pour d'autres applications

## **Interfaces graphique Android[11]**

On peut construire une Interface graphique (UI) dans Android en utilisant deux méthodes, soit en définissant un fichier XML ou par du code Java. La première méthode est largement préférable parce sa respecte l'architecture Modèle-Viewer-contrôle. En plus l'adaptation d'une résolution d'un écran est beaucoup plus facile. La définition d'une interface en XML est très similaire à la création d'un document HTML, exemple,

Pour crée un XML-Layouts. Tout est bien structuré et peut être exprimé en arborescence : Le fichier AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent" android:layout_height="fill_parent">
    <TextView android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"/>
</LinearLayout>
```

## Fichier Android Manifest.xml :

Le fichier Android Manifest.xml est un fichier indispensable pour chaque application Android, il se situe dans la racine de l'application, il décrit les valeurs globales de notre package, y compris les composants de l'application (activités, services, etc.).

Le fichier AndroidManifest.xml déclare l'ensemble des éléments de l'application exemple :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="andro.jf"

    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon"
        android:label="@string/app_name">
        <activity android:name=".Main"
            android:label="@string/app_name">
            <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <service>...</service>
        <receiver>...</receiver>
        <provider>...</provider>
    </application>
</manifest>
```

## Les ressources :[12]

Les ressources sont des fichiers externes (fichiers non-code) qui sont utilisés par notre code et compilé dans notre application au moment de la compilation. Android supporte un certain nombre de différents types de fichiers de ressources, y compris XML, PNG et JPEG. Les fichiers XML sont des formats très différents en fonction de ce qu'ils décrivent. Les ressources sont externalisées du code source. Les fichiers XML sont compilés dans un format binaire pour des raisons d'efficacité. Les chaînes de caractères sont compressées dans une forme de stockage plus efficace.

Ci-dessous la liste des ressources-types et ou on doit les leurs emplacement dans l'arborescence du projet Android :

- lay-out-files (fichier plans) → “/res/layout/”
- animations → “/res/anim/”
- styles, strings and arrays → “/res/values/”
- ‘arrays.xml’ pour définir les tableaux
- ‘colors.xml’ pour définir les couleurs
- ‘dimens.xml’ pour définir les dimensions
- ‘strings.xml’ pour définir les chaînes de caractères

## Android Développement Tools :[13]

Android Développement Tools (ADT) est un plugin pour l'IDE Eclipse est conçu pour donner au développeur un puissant environnement intégré dans lequel la création d'applications Android il à comme fonctionnalités :

- permettre de rapidement mettre en place de nouveaux projets Android,
- créer une interface d'application
- ajouter des packages basés sur l'API Android,
- déboguer vos applications en utilisant les outils du SDK Android,

Développement dans Eclipse avec ADT est fortement recommandé et aussi le moyen le plus rapide pour commencer. Avec la mise en place de projet guidée qu'elle fournit, ainsi que l'intégration des outils, éditeurs XML personnalisés, et le volet de sortie de débogage, ADT vous donne un élan incroyable dans le développement d'applications Android.

## Annexe C : IMC [14]

L'indice de masse corporelle est un outil inventé par l'Organisation Mondiale de la Santé (OMS) pour évaluer les risques sur la santé individuelle liés au surpoids. L'indice de masse corporelle est lui calculé en divisant le poids par la taille au carré.

$$IMC = Poids / Taille^2$$

L'interprétation de l'IMC se fait selon les critères définis par l'Organisation mondiale de la Santé est représenté dans ce tableaux.

IMC (kg.m-2)	Interprétation (D'après l'OMS)
moins de 16,5	Dénutrition
16,5 à 18,5	Maigreur
18,5 à 25	Corpulence normale
25 à 30	Surpoids
30 à 35	Obésité modérée
plus de 40	Obésité morbide ou massive

Tab C.1 Interprétation de IMC

## Annexe D: Taux de glycémie [15]

L'indice glycémique est un critère de classement des aliments contenant des glucides, basé sur leurs effets sur la glycémie (taux de glucose dans le sang) durant les deux heures suivant leur ingestion. Il permet de comparer le pouvoir glycémiant de chaque aliment, mesuré directement lors de la digestion.

Le glucose apporte l'énergie aux différents tissus de l'organisme. Si le taux de glucose dans le sang reste stable même après un repas ou après un effort physique, c'est qu'il existe un système régulateur complexe dans lequel l'insuline joue un rôle primordial.

La glycémie est le taux de sucre dans le sang.

- Valeur moyenne : 1 gramme par litre .

Elle varie entre 1 et 1,4 g/l deux heures après un repas.

Elle varie entre 0.8 et 1,26 g/l à jeun le matin.

- Selon les critères de l'OMS (Organisation mondiale de la santé), il y a diabète quand la glycémie à jeun est supérieure ou égale à au moins deux reprises à 1,26 g/l.

- L'hypoglycémie correspond à une glycémie inférieure à 0,45 g/l.

## Annexe E: Analyse Résultats sur l'hématologie : [16]

Elle explore les différentes cellules présentes dans le sang. À savoir : les globules rouges (hématies), toutes les familles de globules blancs (leucocytes), et les plaquettes, ou d'éventuelles cellules anormales.

Nom	Définition	Norme	Interprétation anormale (probable)
<b>Globules rouges</b>	Un globule rouge est une cellule sanguine qui permet le transport de l'oxygène.	4,5 – 5,5 millions / mm <sup>3</sup>	+Syndrome inflammatoire.
<b>Globules blancs</b>	sont présentes dans le sang. jouent un rôle important dans la défense de l'organisme contre les agents extérieurs, qu'elles peuvent détruire directement ou indirectement en créant des anticorps.	4000 – 10000 /mm <sup>3</sup>	+Dépister une infection : hyperleucocytose : infection bactérienne.
<b>Hémoglobine</b>	une substance contenue dans chaque globule rouge qui permet de transporter l'oxygène des poumons à tous les tissus de l'organisme, et le gaz carbonique des tissus aux poumons.	14 – 18 g / 100 ml : chez l'homme. 12 – 16 g / 100 ml : chez la femme.	+animée polyglobulie -anémie.
<b>VGM</b>	VGM (L) = Hématocrite (nombre compris entre 0 et 1) / Nombre de Globules Rouges par litre	80 – 95 / mm <sup>3</sup>	-il à une anémie par manque de fer
<b>TGMH</b>	est la quantité moyenne d'hémoglobine contenue dans un globule rouge.	27 à 32 picogramme	+une anémie macrocytaire -une anémie microcytaire
<b>Plaquettes</b>	fabriquer par la moelle osseuse et jouent un rôle important dans la coagulation.	150 000 – 450 000 / mm <sup>3</sup>	+ Dépister une thrombocytes - Dépister une thrombopénie
<b>VS</b>	La vitesse de sédimentation des éléments figurés du sang est mesurée dans une colonne calibrée et graduée.	150 000- 400 000/mm	+ l'existence d'une inflammation

## Annexe F : Analyse Résultats Biochimiques [17]

Étude des différentes substances chimiques du plasma (lipide, urée, créatine, etc.). Ainsi que leur norme et interprétation sont citées et classés dans le tableau ci-dessous :

Nom	Définition	Norme	Interprétation anormale (probable)
<b>Glycémie</b>	Le taux de sucre dans le sang	0,7 - 1,10 g/l	+ diabète -Hypoglycémie
<b>Cholestérol total</b>	un lipide qui joue un rôle central dans de nombreux processus biochimiques.	3,2 - 5,8 g/l	+maladie cardiovasculaire
<b>HDL</b>	produit par le corps pour circuler dans le sang, le cholestérol adhère aux protéines et forme une lipoprotéine.	0,4 - 0,9 g/l	+ est un facteur de protection dans l'évaluation du risque cardiovasculaire - est un facteur de risque supplémentaire.
<b>LDL</b>	mauvais cholestérol Les lipoprotéines de basse densité	1,0 - 1,6 g/l	-un bon signe
<b>VLDL</b>	lipoprotéines responsables du transfert des lipides endogènes de leur lieu de synthèse	0,2 - 0,4g/l	+ infarctus du myocarde attaque cérébrale.
<b>Acide urique</b>	Provenant de la destruction des acides nucléiques contenus dans l'organisme. Cet acide provient également de la digestion de certains aliments riches en acides nucléiques	28 - 65 mg/l	+ Personnes souffrant d'hyper uricémie
<b>Créatinine</b>	La valeur plus fiable du fonctionnement rénal	6,8 - 13,6mg/l	-insuffisance rénale.

## Annexe G : SQLite[18]

SQLite est une bibliothèque écrite en C qui propose un moteur de base de données relationnelle accessible par le langage SQL. L'intégralité de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme.

D. Richard Hipp, le créateur de SQLite, a choisi de mettre cette bibliothèque ainsi que son code source dans le domaine public, ce qui permet son utilisation sans restriction aussi bien dans les projets open source que dans les projets propriétaires.

SQLite est le moteur de base de données le plus distribué au monde, grâce à son utilisation dans certains produits d'Apple, d'Adobe et de McAfee et dans les bibliothèques standards de nombreux langages comme PHP ou Python. De par son extrême légèreté (moins de 300 KiloOctets), elle est également très populaire sur les systèmes embarqués, notamment sur la plupart des Smartphones modernes : Android, les systèmes d'exploitation mobiles Symbian et l'iPhone utilisent comme base de données embarquée.

Il existe plusieurs types de données dans SQLite, ceux-ci définissant la façon dont SQLite va travailler lors de l'entrée des nouvelles données.

- **TEXT** : enregistre la donnée comme une chaîne de caractères, sans limite de taille. Si un nombre est entré dans une colonne de ce type, il sera automatiquement converti en une chaîne de caractères.
- **INTEGER** : enregistre la donnée comme un entier si celle-ci peut être encodée sans perte, mais peut utiliser les types REAL ou TEXT si ça ne peut être fait .
- **REAL** : enregistre la donnée comme un réel, même s'il s'agit d'un entier. Si la valeur est trop grande, la donnée sera convertie en chaîne de caractères ;
- **NONE** : la donnée est enregistrée telle quelle, sans conversion.

## Annexe H:Google Charting API: [19]

L'utilisation de Google Charting est très simple. Cette API permet de créer des graphiques par l'intermédiaire d'une URL. Vous n'avez qu'à passer les bons arguments à celle-ci pour récupérer votre graphique. Comme ce service est basé sur une URL, vous devez disposer d'une connexion internet pour que celle-ci fonctionne. Si votre application s'exécute hors ligne, passez votre chemin.

L'appel à l'API Google Charting s'effectue par l'intermédiaire d'URL ressemblant à celle-ci :

`http://chart.apis.google.com/chart?cht=p3&chs=250x100&chd=t:30,60,10&chl=cars|bikes|trucks`

Dans cet exemple, voici comment lire les paramètres :

- `cht=t3`, type du graphique, `t3` = camembert 3D,
- `chs=250x100`, la résolution,
- `chd=t:30,60,10`, les données,
- `chl=cars|bikes|trucks`, les libellés.

## **Annexe I:IDE Eclipse : [20]**

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions. La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plugin : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

Dans le cadre de notre projet, nous avons utilisé la version Eclipse INDIGO.

# Référence bibliographiques

- [1] S.SFARTZ, Processus de développement en y, Cours, 2004
- [2] : Louise Michel, S. Laporte, Introduction au Modèle conceptuel des données (MCD) ,2002
- [3] : B. Meyer, Conception et Programmation orientées objet. Eyrolles. 2000.
- [4] :N.Pitman, UML 2 en concentré, O'Reilly. 2006.
- [5] : Frédéric di Gallo, méthodologie des système d information,aix-en-provence 2001
- [6] Ocl : Laurent Audibert, UML 2, 01/12/2009.
- [8]:Jean-Francois Lalande, Développement sous Android, ENSI Bourges.2011.
- [9] :Kherbouche Mouhamed Oussama, Initiation à Android,Cours.2011-2012.
- [10]: Damien Guignard, Julien Chable, Emmanuel Robles Programmation Android, de la conception au déploiement avec le SDK Google
- [11] :Nicolas Druet, Utiliser les services sous Android, France, Février 2010
- [12] :Jean-Francois Lalande, Développement sous Android,ENSI Bourges,2011
- [13]:M. Dalmau, Développement d'applications pour Android, IUT de Bayonne-Pays Basque
- [16]: Cl.audigié,G.Zonszain ,Pricipes des méthodes d'analyse biochimique tome 2.
- [17] :A.Franciscus, interpréter un rapport de laboratoire,Publication, AVRIL 2004
- [18]: Victor Matos, Android Persistency: SQL Databases,Cleveland State University,2011

# Référence web-graphique

[7] :Doctissimo, <http://www.doctissimo.fr/>,11/09/2012

[15]: Dr. Denise Cazivassilio [Médecin Généraliste],<http://www.docteurlic.com/examen/glycemie.aspx>, 07/02/2008

[19]:Erone.alliaume,comment intégrer des graphiques dans une application android,21/10/2010.

[20]: eclipse,<http://www.eclipse.org/>,10/09/12