

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ FERHAT ABBAS 1



FACULTÉ DES SCIENCES
DÉPARTEMENT D'INFORMATIQUE

MÉMOIRE DE FIN D'ÉTUDE

En vue de l'obtention du diplôme de LICENCE en
Informatique

Thème

Outil d'aide à la préparation de solutions
médicamenteuses

RÉALISÉ PAR :
Chougui Abdeldjalil
Moussaoui Achraf

ENCADRÉ PAR :
Benmahmoud Sabrina

Remerciement

Au nom d'Allah le tout puissant, un grand merci lui revient pour nous avoir donnée la foi, la volonté, le courage et surtout, de nous avoir permis d'en arriver là;

Nous tenons aussi à adresser nos vifs remerciements à notre encadreur

Madame Benmahmoud Sabrina qui par ses encouragements renouvelés, ses remarques pertinentes, ses conseils, sa disponibilité malgré la mauvaise période que nous traversons, et son soutien qui ne nous ont jamais fait défaut, nous avons puachever notre travail dans les meilleures conditions;

Nos remerciements vont également nos parents qui nous ont toujours aidés et soutenu .

Nous tenons aussi à remercier tous ceux qui ont contribué à ce modeste travail.

Résumé

Ce travail est réalisé dans le cadre d'un mémoire de fin d'étude pour l'obtention du diplôme de licence académique en informatique, il consiste à créer une application mobile sous Android qui permet aux opérateurs pharmaciens de préparer une solution de médicaments pour perfusion d'un malade, à partir d'une base de données de médicaments (locale).

Dans cette mémoire vous pourrez suivre les étapes pris pour la création de cette application mobile, la première étape de la création est l'analyse des besoins qui consiste a citer l'objectif de l'application et les besoins fonctionnels qui devrait être réalisé, et puis l'étape de conception qui inclut quelques diagrammes UML pour réaliser l'étude conceptuelle de système, ensuite l'étape de réalisation dans laquelle on transforme nos diagrammes en code source et tester notre application sur plusieurs appareilles.

TABLE DES MATIÈRES

Remerciement	1
Résumé	2
Table Des Figures	6
Introduction Générale	7
1 Analyse des Besoins	8
1.1 Introduction	9
1.2 Objectif Du Projet	9
1.3 Spécification Des Besoins	9
1.3.1 Les besoins fonctionnels	9
1.3.2 Les besoins non fonctionnels	10
1.4 Conclusion	11
2 Conception	12
2.1 Introduction	13
2.2 Langage de modélisation unifie(UML)	13
2.2.1 Définition	13
2.2.2 Les différents types diagrammes d'UML	14
2.2.3 Les avantages d'UML	14
2.3 Diagramme de cas d'utilisation	15

2.3.1	Définition	15
2.3.2	Identification des acteurs	15
2.3.3	Diagramme de cas D'utilisation globale	16
2.4	Diagramme De Classe	16
2.4.1	Définition	16
2.4.2	Diagramme De Classe De L'application	17
2.5	Conclusion	18
3	Réalisation	19
3.1	Introduction	20
3.2	Environnement de développement :	20
3.2.1	Environnement matériel :	20
3.2.2	Environnement logiciel :	21
Android Studio	21	
Framework Flutter	21	
Language Dart	22	
SQLite	22	
SideSync	23	
Star UML	23	
Latex	24	
TexStudio	25	
3.3	Présentation des Interfaces de l'application :	26
3.4	Une partie de code source <Fonction de Calcul>	35
3.4.1	Explication De La Fonction De Calcul	36
3.5	Conclusion	37
Conclusion Générale		38
Bibliographie		39

TABLE DES FIGURES

2.1	Logo d'UML	13
2.2	Les différents types diagrammes d'UML	14
2.3	Diagramme de cas D'utilisation globale	16
2.4	Diagramme De Classe De L'application	17
3.1	Logo D'Android Studio	21
3.2	Logo De Flutter	22
3.3	Logo De Language Dart	22
3.4	Logo De SQLite	23
3.5	Logo De SideSync	23
3.6	Interface De Star UML	24
3.7	Logo De Latex	25
3.8	Interface De TexStudio	25
3.9	Page D'accueil	26
3.10	Les Interfaces Introduction	26
3.11	Sign Up	27
3.12	Authentification	27
3.13	La Liste Des Médicaments	28
3.14	Gestion De La Liste Des Médicaments	28
3.15	Préparer Solution	29
3.16	Ajouter Patient	29
3.17	Choix de médicament	30
3.18	Remplir Posologie et Voir Résultat	30

3.19	Bilan Du Jour	31
3.20	Liste Des Reliquats	31
3.21	Suppression De Reliquat Périme	32
3.22	Interface Paramétrés	33
3.23	Changer Nom D'utilisateur et Le Mot de Passe	33
3.24	Changer Le Thème	34
3.25	Rapport de fin du journée PDF	34
3.26	Le Thème Sombre	35
3.27	Code Source Partie 1	35
3.28	Code Source Partie 2	36

Introduction Générale

Le marché de la téléphonie portable connaît actuellement une véritable révolution ,d'un simple téléphone portable pour émettre des appels à un téléphone évolué doté de capacités proches d'un véritable ordinateur appelé Smartphone.

Aujourd’hui . . . le monde connaît une progression rapide dans l'utilisation des Smartphone, grâce aux applications mobiles .Ces dernières sont capables de satisfaire les besoins actuels des utilisateurs dans plusieurs domaines dans nos vies quotidienne , et facilitent l'accomplir des tâches et on offrant plusieurs services.

Cette progression d'utilisation des Smartphones a permis de faire une collaboration entre le domaine de la santé et les application mobiles qui peut aider a augmenter les capacités des services réalisée, un de ces services est la préparation des solution médicamenteuses.

La préparation des solutions médicamenteuses est une tache très intéressante dans le processus de traitement d'une maladie , L'utilisation de tout médicament expose à des risques multiples, liés au médicament lui-même (effets indésirables) ou aux conditions entourant sa préparation et son administration (erreurs, interactions, risque infectieux...), et pour minimiser le taux des erreurs lors de la préparation (faute des calculs , choix de médicaments . .) on trouve que la solution c'est d'Informatiser tout ce qui concerne la préparation de solutions médicamenteuse par la création d'une application mobiles destiné aux opérateurs pharmaciens.

Notre projet de fin d'études s'inscrit dans un cadre général de conception et de développement d'une application mobile qui permet aux opérateurs pharmaciens de préparer une solution de médicaments pour perfusion d'un malade, à partir d'une base de données de médicaments (locale).

Nous allons détailler le projet dans ce rapport sur trois chapitres :

1. Le premier chapitre se focalise sur l'analyse des besoins et la présentation générale du cadre de notre projet.
2. Le deuxième chapitre sera réservé pour la Conception qui nous permet de décrire sans ambiguïté l'application à développer avec une modélisation formelle à travers des diagrammes de langage UML.
3. Le troisième chapitre est dédié à la réalisation, où nous trouvons la présentation de l'environnement de développement, l'explication de fonctionnement de notre application en présente des interfaces finies.

CHAPITRE 1

ANALYSE DES BESOINS

1.1 Introduction

La première étape de la réalisation d'un projet consiste à analyser la situation pour tenir compte des contraintes répondant aux besoins d'utilisateur.

Dans ce chapitre nous présentons les objectifs de notre application, ce qui nous amène à identifier les possibilités et les besoins fonctionnels et non fonctionnels de l'application. ces derniers donnent une idée générale sur le fonctionnement de notre application.

1.2 Objectif Du Projet

L'objectif de ce projet est la conception et la création d'une application Mobile Android, qui permet aux opérateurs pharmaciens de préparer une solution de médicaments pour perfusion d'un malade, à partir d'une base de données de médicaments (locale).

1.3 Spécification Des Besoins

L'application doit satisfaire les besoins fonctionnels qui seront exécutés par le système et les besoins non fonctionnels qui perfectionnent la qualité logicielle du système.

1.3.1 Les besoins fonctionnels

L'application à réaliser doit offrir un ensemble de fonctionnalités qui doivent être mises en relation avec un ensemble de besoins utilisateur. Ces derniers définissent les services que les utilisateurs s'attendent à voir fournis par cette application. Notre application doit couvrir principalement les besoins fonctionnels suivants :

- Les besoins fonctionnels attendus :**

- Calcule la dose et le volume à administrer.
- L'indication de type de poche de sérum le plus adaptée à la préparation.
- Calcule les reliquats (restes).
- Calcule le prix des reliquats jetés.
- Calcule la quantité consommé des médicaments.
- La Gestion et La mise à jour de la base de données de médicaments (ajout, recherche, modification, suppression).

- Effectuer un bilan du jour.
- Enregistre le bilan du jour sous forme d'un PDF.

- **Les besoins fonctionnels additifs :**

- Authentification d'utilisateur.
- L'affichage de la liste des médicaments.
- Traitement d'une ordonnance qui contient jusqu'à 5 médicaments à la fois.
- Changer le nom d'utilisateur.
- Changer le mot de passe.
- Changer le Thème.

1.3.2 Les besoins non fonctionnels

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système, telles que les exigences en matière de performances, les dépendances du projet, de facilité de maintenance, d'extensibilité et de fiabilité.

Dans le cadre de ce travail, l'application devra répondre à ces besoins :

- **Les Contraintes techniques :**

- **la sécurité** : tous les accès d'utilisateurs doivent être protégés par un login et un mot de passe pour parvenir à la sécurité de l'application.
- **Validité** : réaliser exactement les taches définies dans la spécification.
- **la performance** : temps de réponse court.
- **la fiabilité** : Les données fournies par l'application doivent être fiables et la solution doit rendre des résultats corrects.

- **Les Contraintes ergonomiques :**

- **Brièveté** : Limiter le travail de lecture, d'entrée et les étapes par lesquelles doivent passer les usagers.
- **Guidage** : L'ensemble des moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'application.
- **prise en compte de l'expérience de l'utilisateur** : Le système doit respecter le niveau d'expérience de l'utilisateur donc il doit être simple, compréhensible et facile à utiliser.
- **Groupement/Distinction entre Items** : groupement des différents éléments visuels de façon cohérente et ordonnée.

1.4 Conclusion

Dans ce chapitre, nous avons essayé d'établir une étude théorique de l'application par la définition de notre objectif et puis la spécification des besoins fonctionnels et non fonctionnels du système résultant, ce qui correspondait aux différentes activités de la première phase du cycle de développement du notre système. Dans le chapitre suivant, nous étudierons la phase de conception.

CHAPITRE 2

CONCEPTION

2.1 Introduction

La conception est l'étape la plus importante dans le cycle d'implémentation des applications informatiques. Elle consiste à façonner le système et à lui donner une forme répondant à tous les besoins et les exigences.

Dans ce chapitre nous allons entamer une partie cruciale de développement de l'application et qui consiste un pont entre la spécification des besoins et la réalisation. Elle comporte une modélisation conceptuelle suivant la méthodologie UML, D'abord, on présente le diagramme de cas d'utilisation et de séquences pour spécifier les besoins fonctionnels de notre système. Après, nous avons réalisé le diagramme de classes pour la conception de la base de données. Nous avons utilisé StarUML comme outil de modélisation de nos trois diagrammes choisis dans notre conception.

2.2 Langage de modélisation unifiée(UML)

2.2.1 Définition

UML (Unified Modeling Language) est un langage de modélisation unifié permet de modéliser une application logicielle d'une façon standard dans le cadre de conception orienté objet. Il définit un ensemble de diagrammes permettant de représenter un système ou un programme informatique et son utilisation prévue, UML n'est pas limité à l'ingénierie du logiciel, mais est également utilisé dans l'ingénierie des systèmes. La représentation des structures organisationnelles peut être unifiée par la modélisation du langage UML. La mise en place de logiciels de conception intégrée pour un projet peut aussi être à l'acte (Diagrammes) d'un code initial du programme.



FIGURE 2.1 – Logo d'UML

2.2.2 Les différents types diagrammes d'UML

Les diagrammes sont dépendants hiérarchiquement (structure, comportement et interaction) et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. Il en existe quatorze depuis UML 2.3.

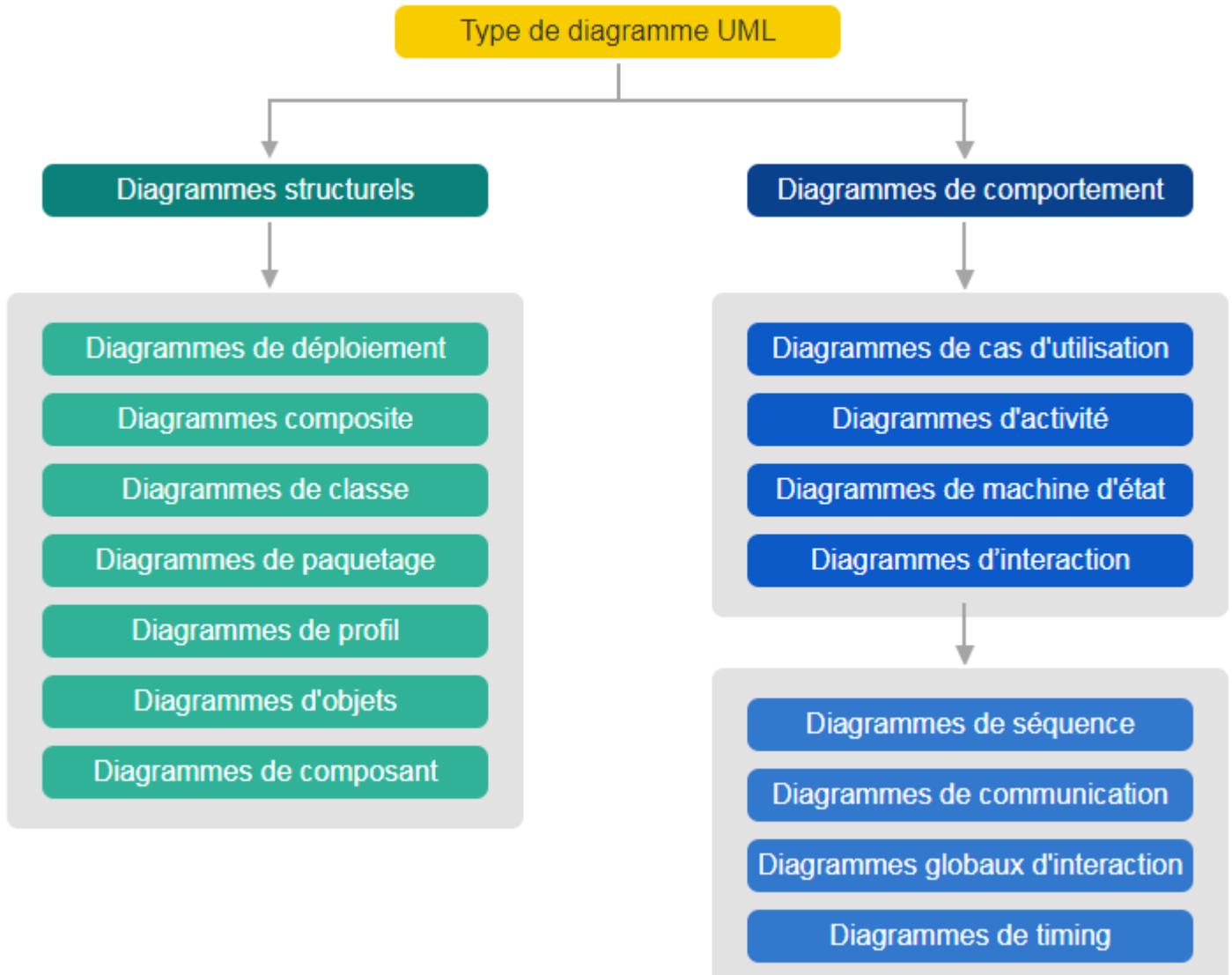


FIGURE 2.2 – Les différents types diagrammes d'UML

2.2.3 Les avantages d'UML

- UML est un langage formel et normalisé : il permet un gain de précision et de stabilité.
- UML est un support de communication performant : il permet grâce à sa représentation graphique, d'exprimer visuellement une solution objet, de faciliter la comparaison et l'évolution de solution.
- Son caractère polyvalent et sa souplesse en font un langage universel.

2.3 Diagramme de cas d'utilisation

2.3.1 Définition

Les diagrammes de cas d'utilisation sont utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour la présentation de toutes les interactions des utilisateurs avec le système.

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation.

2.3.2 Identification des acteurs

Les acteurs sont des entités externes d'un rôle joué par une personne, un processus ou une chose interagit avec le système.

Dans notre système, nous pouvons identifier le Pharmacien comme un acteur.

- Le Pharmacien : Il peut accéder à l'application par la saisie d'un identifiant et un mot de passe. Il peut préparer une solution médicamenteuse, gérer la base de données des médicaments et d'autres fonctionnalités.

2.3.3 Diagramme de cas D'utilisation globale

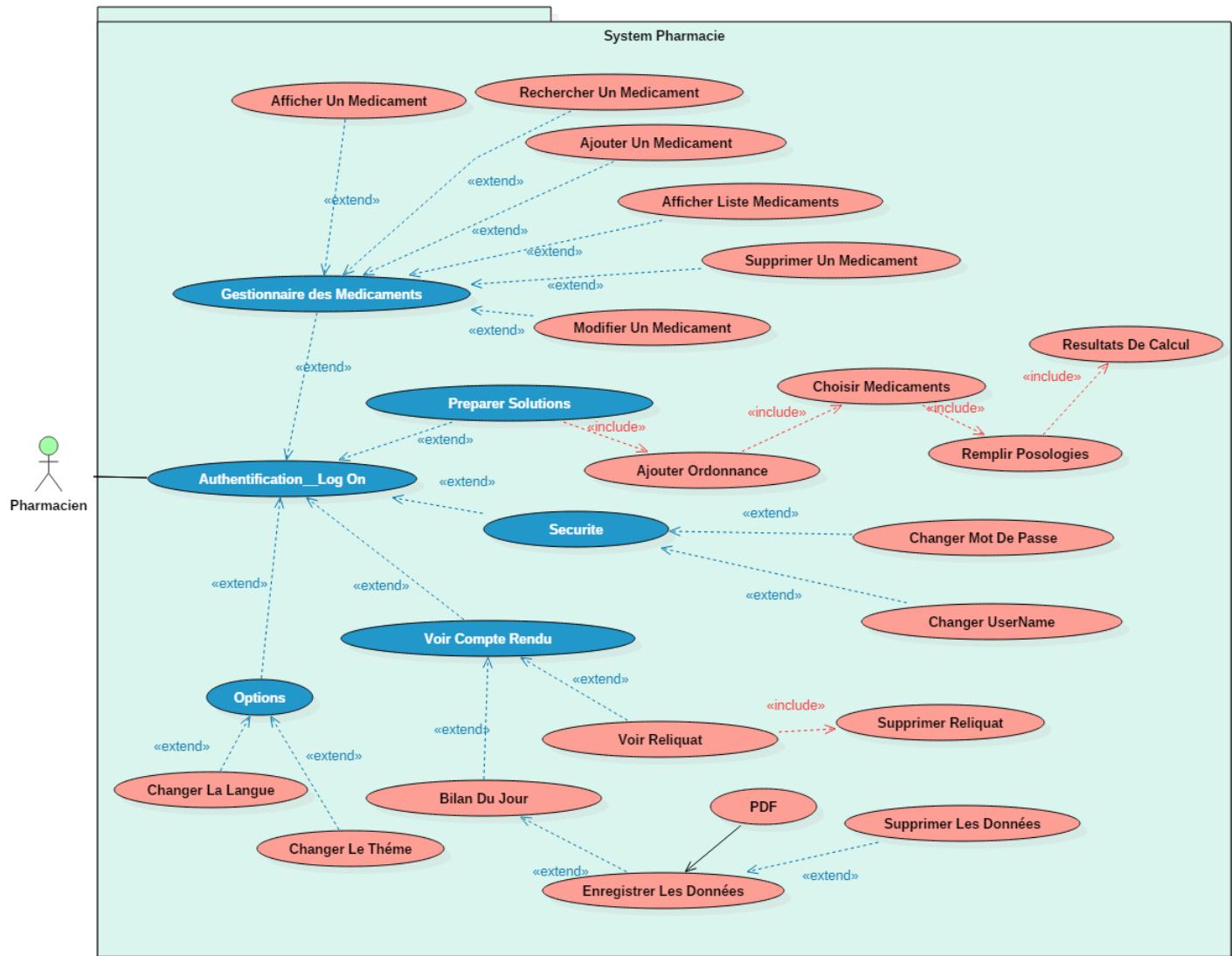


FIGURE 2.3 – Diagramme de cas D'utilisation globale

2.4 Diagramme De Classe

2.4.1 Définition

Il représente les classes intervenant dans le système. Le diagramme de classe est une représentation statique des éléments qui composent un système et de leurs relations.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont

possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes. Elles sont finalement instanciées pour créer des objets

2.4.2 Diagramme De Classe De L'application

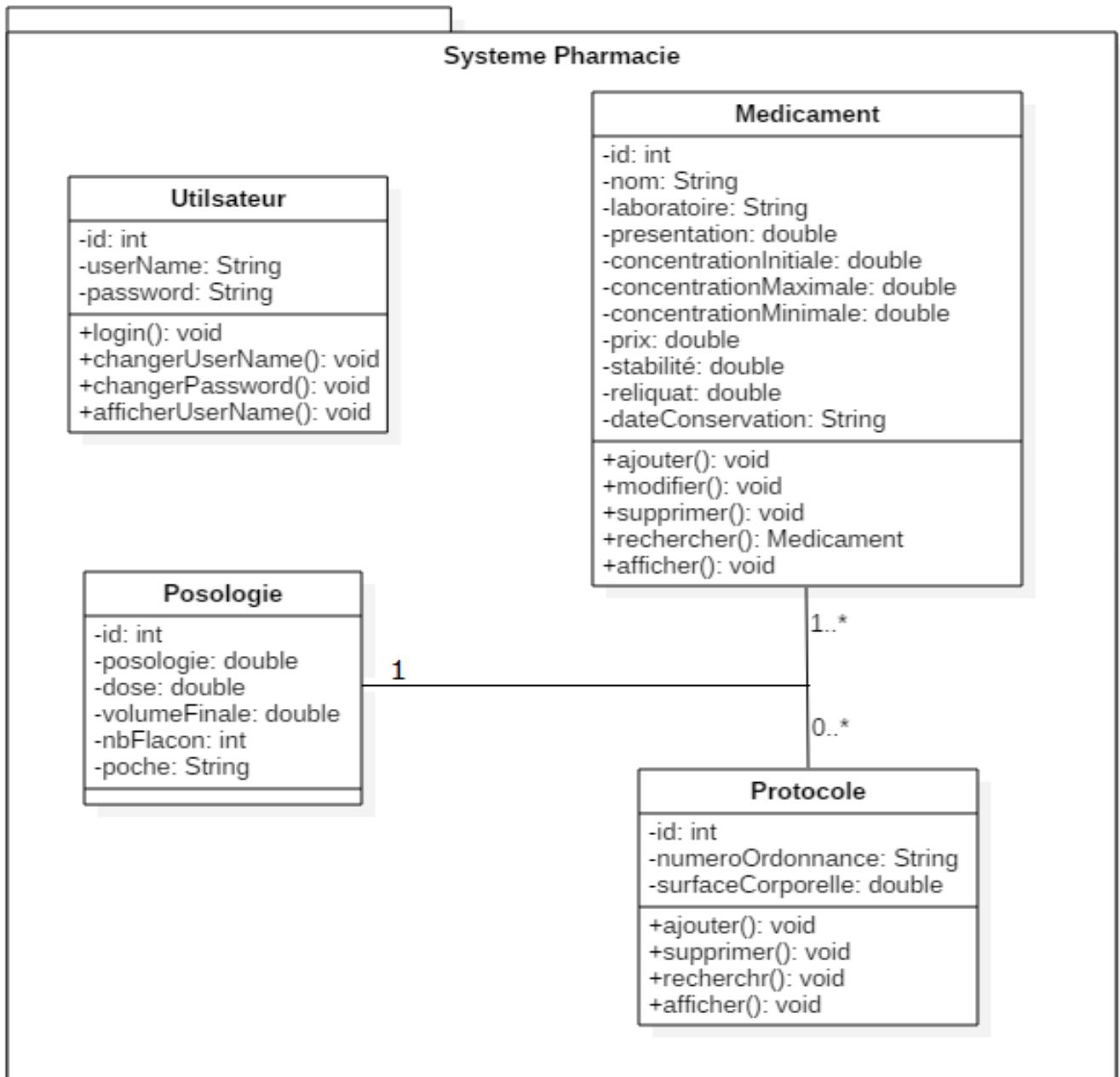


FIGURE 2.4 – Diagramme De Classe De L'application

2.5 Conclusion

La phase de conception était importante pour pouvoir visualiser le fonctionnement de notre application d'une façon abstraite. En clair, comme nous venons de le voir, ce chapitre était consacré, à la conception des diagrammes d'UML pour associer au cette application et des descriptions pour mieux comprendre la fonctionnalité de cette application.

Le chapitre suivant fera le thème de la phase de réalisation, cette dernière se concentre sur l'élaboration de l'application ainsi que l'observation des résultats obtenus.

CHAPITRE 3

RÉALISATION

3.1 Introduction

Après avoir achevé l'étape de conception de l'application, on va entamer dans ce chapitre la partie réalisation et implémentation en exposant l'environnement du travail et les outils de développements choisis ainsi que les interfaces de notre application.

A la fin de ce chapitre, les objectifs doivent avoir été atteints et le système doit être prêt pour être exploité par les utilisateurs finaux.

3.2 Environnement de développement :

Il y'a deux environnements précisés sont : Environnement logiciel et matériel.

3.2.1 Environnement matériel :

L'application Pharmedic est réalisé par les ordinateurs et les Smartphones suivants :

Ordinateur 01 :

- La marque :DELL
- Processeur : intel®core™ i5-4200U CPU .
- Mémoire : 4 Go RAM.
- Système d'exploitation : Windows 8.1.
- Type du système : système d'exploitation 64 bits.

Ordinateur 02 :

- La marque :HP
- Processeur : intel®core™ i5-4200U CPU.
- Mémoire : 4 Go RAM.
- Système d'exploitation : Windows 7.
- Type du système : système d'exploitation 64 bits.

Smartphone 01 :

- La marque :Samsung.
- Modèle : Galaxy J5 Pro.
- Android : 6.0.1.

Smartphone 02 :

- La marque :Samsung.
- Modèle : Galaxy S6 Edge.
- Android : 6.0.1.

3.2.2 Environnement logiciel :

Pour la réalisation de notre application, nous avons au recours différents outil et logiciel de développement, outil pour la gestion de la base de données , logiciel de modélisation , logiciel pour la rédaction de ce rapport.

Android Studio :

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux. Il propose aussi des outils pour gérer le développement d'applications multilingues et permet de visualiser la mise en page des différents types et tailles d'écrans avec des résolutions variées simultanément, Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur.

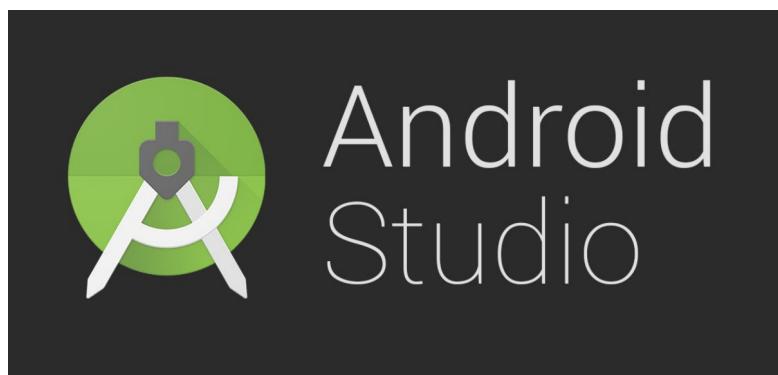


FIGURE 3.1 – Logo D'Android Studio

Framework Flutter :

Flutter est un framework de développement d'interface utilisateur gratuit et open source créé par Google. Jusque-là, il était utilisé pour développer des applications pour Android et iOS.

En effet ce framework est utilisé pour tout ce qui est interface utilisateur. Mais aujourd’hui Flutter se fait surtout connaître pour sa capacité à concevoir des applications natives multiplateforme pour Android et iOS (Windows/Mac/Linux sont également supportés).

L’équipe de Flutter provient essentiellement du Web (plus particulièrement de Chrome) et a essayé d’adapter sa philosophie au monde du mobile. Ils se sont appuyés pour cela sur Skia, le moteur qui fait partie intégrante de Chrome ou encore de toute la gestion du texte d’Android.



FIGURE 3.2 – Logo De Flutter

Language Dart :

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web.

Dart est un langage orienté objet, basé sur la classe, récupérateur de mémoire avec une syntaxe de type C1 .Il prend en charge les interfaces, les mixins, les classes abstraites.



FIGURE 3.3 – Logo De Language Dart

SQLite :

SQLite est une base de données open source, qui supporte les fonctionnalités standards des bases de données relationnelles comme la syntaxe SQL, les

transactions et les prepared statement. La base de données nécessite peu de mémoire lors de l'exécution (env. 250 ko), ce qui en fait un bon candidat pour être intégré dans d'autres environnements d'exécution.

SQLite est intégrée dans chaque appareil Android. L'utilisation d'une base de données SQLite sous Android ne nécessite pas de configuration ou d'administration de la base de données. Vous devez uniquement définir les instructions SQL pour créer et mettre à jour la base de données. Ensuite, celle-ci est gérée automatiquement pour vous, par la plate-forme Android.



FIGURE 3.4 – Logo De SQLite

SideSync :

Samsung SideSync pour Android est une application de synchronisation de données proposée par Samsung mais fonctionnant sur tous les smartphones et tablettes Android, quelle que soit leur marque. Pour l'utiliser avec votre ordinateur, vous devez au préalable télécharger Samsung SideSync pour Windows ou Mac. Le logiciel de synchronisation de Samsung vous offre la possibilité de partager l'écran et les données entre votre PC et votre appareil mobile. Vous pouvez ainsi passer des appels ou encore envoyer des SMS directement depuis votre PC, en utilisant la carte SIM insérée dans votre smartphone.
Dans Notre cas on utilise l'application SideSync pour exécuter notre application Pharmedic Sur une appareil réel (Physique) et remplacer l'émulateur AVD(Android Virtual Device) qui nécessite un peu plus de ressources matériels.



FIGURE 3.5 – Logo De SideSync

Star UML :

StarUML est un logiciel de modélisation UML, qui a été "cédé comme open source" par son éditeur, à la fin de son exploitation commerciale (qui visiblement continue ...), sous une licence modifiée de GNU GPL.

StarUML est écrit en Delphi1, et dépend de composants Delphi propriétaires (non open-source).

StarUML gère la plupart des diagrammes spécifiés dans la norme UML 2.0, Parmi ces Diagrammes . . les Diagrammes qui on a Modélisé dans Notre Projet (Cas D'utilisation+Diagramme de Classe).

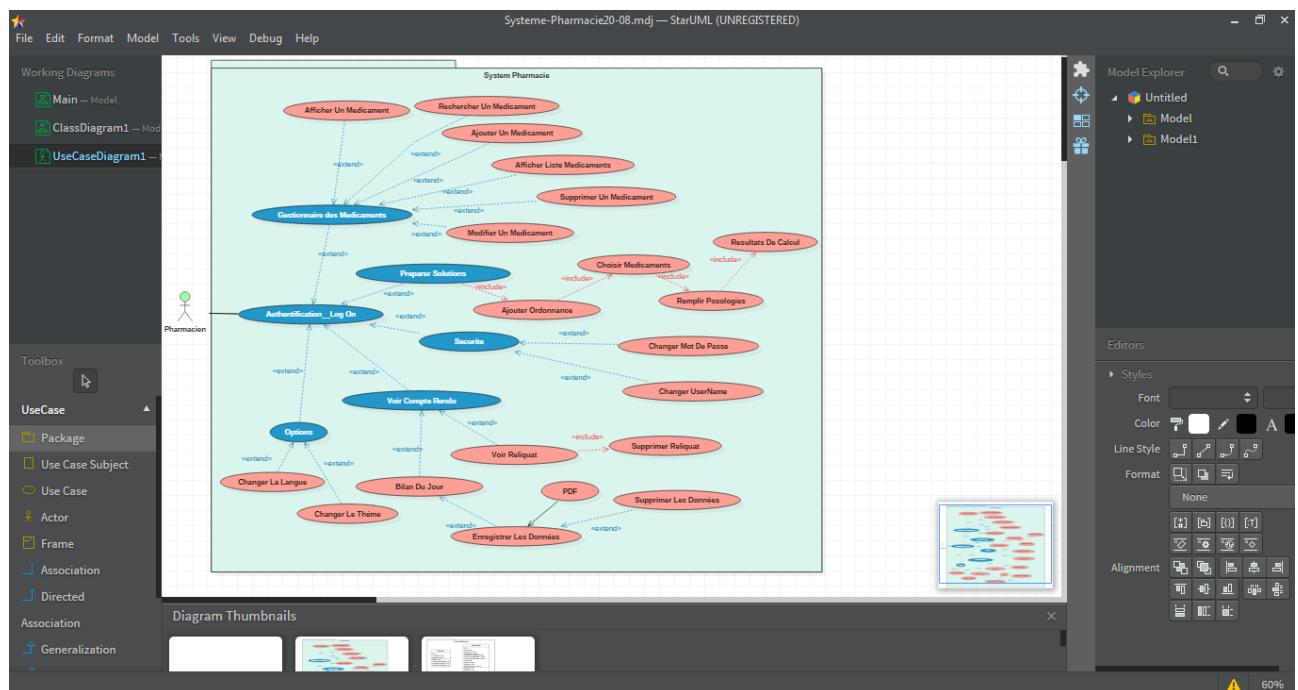


FIGURE 3.6 – Interface De Star UML

Latex :

LaTeX est un langage et un système de composition de documents. LaTeX permet de rédiger des documents dont la mise en page est réalisée automatiquement en se conformant du mieux possible à des normes typographiques. Une fonctionnalité distinctive de LaTeX est son mode mathématique, qui permet de composer des formules complexes.

LaTeX est particulièrement utilisé dans les domaines techniques et scientifiques pour la production de documents de taille moyenne (tels que des articles) ou importante (thèses ou livres, par exemple).

Dans Notre cas on a utilisé Latex pour la rédaction de ce rapport.



FIGURE 3.7 – Logo De Latex

Et pour L'utilisation De latex il s'agit d'installer TexStudio.

TexStudio :

TeXstudio (anciennement TeXMakerX) est un environnement de développement intégré (IDE) très puissant pour écrire des documents LaTeX en format PDF ou autre, qui s'affichent de la même manière quelque soit le système d'exploitation. L'utilisation de TeXstudio exige la présence d'une distribution Latex : c'est la partie la plus importante, qui comporte tous les composants de LATEX et sera chargée de transformer le code tapé dans l'éditeur LATEX en un document PDF ou PostScript.

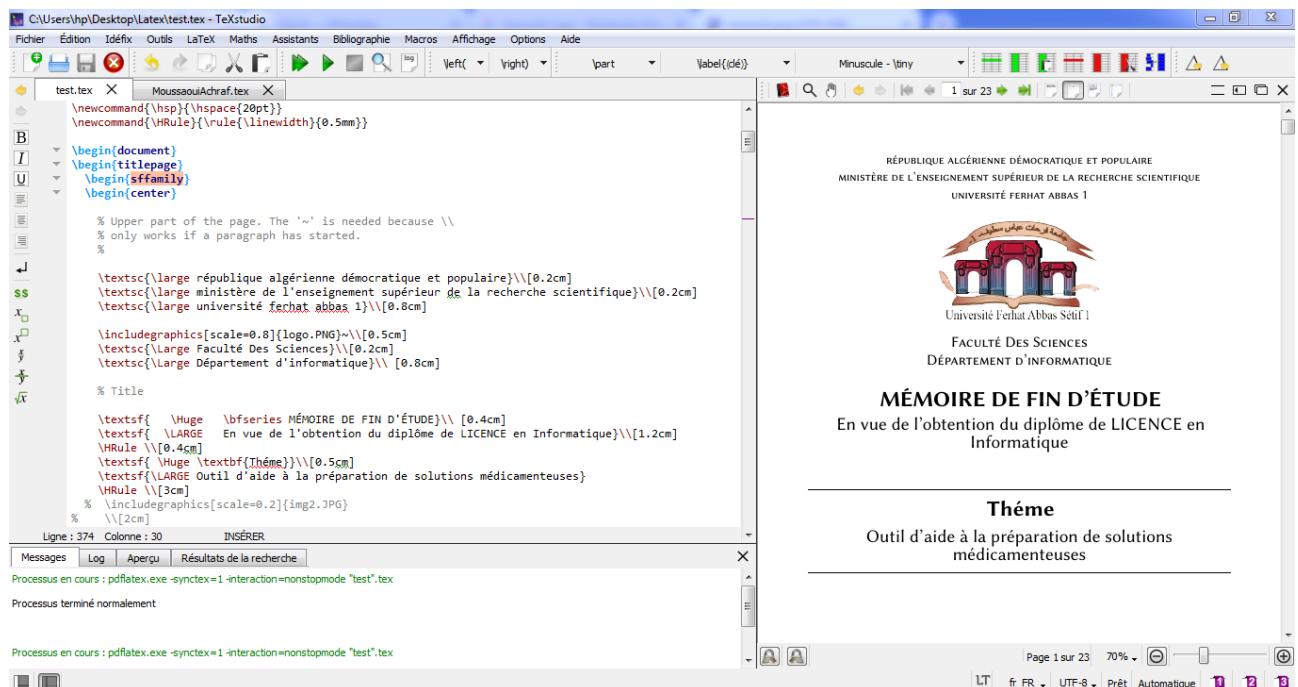


FIGURE 3.8 – Interface De TexStudio

3.3 Présentation des Interfaces de l'application :

Dans cette partie on va représenter les différentes interfaces de notre application et on va faire une petite explication de chaque étape.

La première interface qui s'affiche c'est une interface qui contient un simple logo avec le nom de l'application.



FIGURE 3.9 – Page D'accueil

Ensuite Il s'affiche Les Pages Introduction (Introduction Screen) qui Présente les fonctionnalité de notre application.

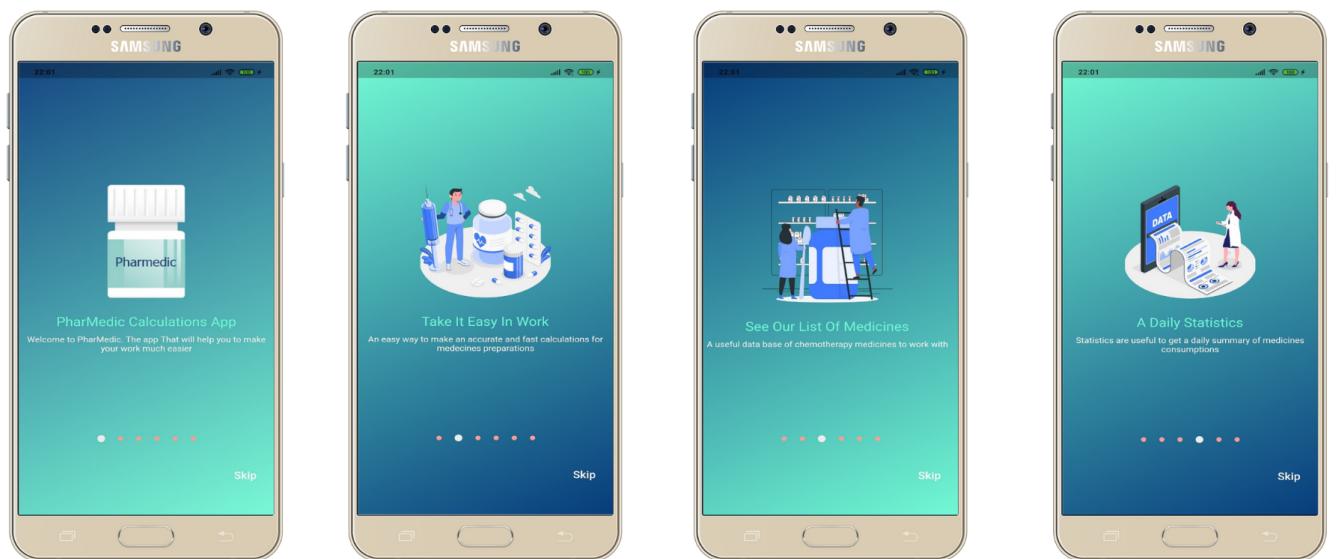


FIGURE 3.10 – Les Interfaces Introduction

Dans le cas si vous êtes un nouveau utilisateur il devrait s'inscrire dans notre application.

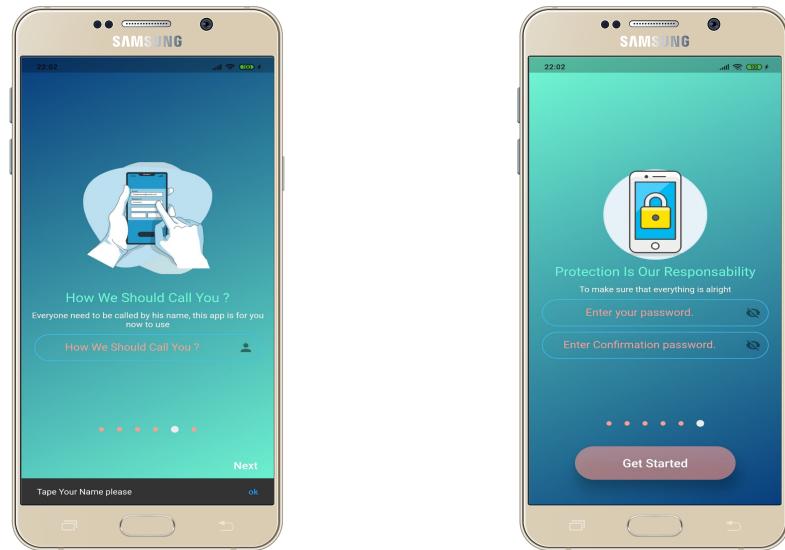


FIGURE 3.11 – Sign Up

Si Vous êtes Déjà inscrit il devrait juste entre le mot de passe pour accéder a l'application.

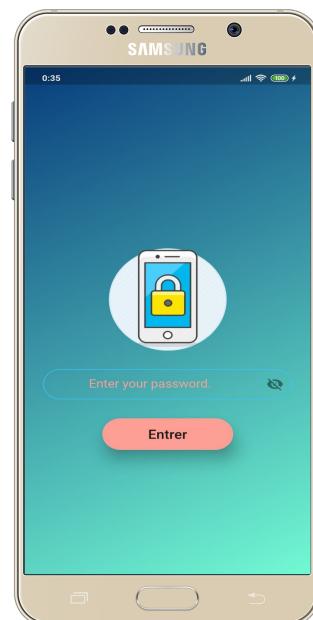


FIGURE 3.12 – Authentication

Lorsque on entre dans l'application on a 5 interface principaux on va découvrir une après l'autre.

La première Interface Principale est la liste des médicaments . . on peut ajouter, supprimer , modifier et rechercher un médicament avec la fonction de **AutoComplete**.

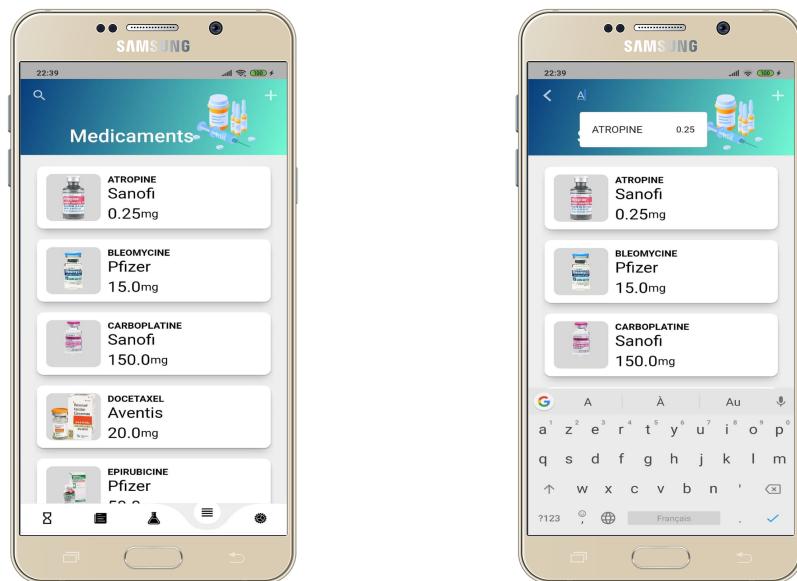


FIGURE 3.13 – La Liste Des Médicaments

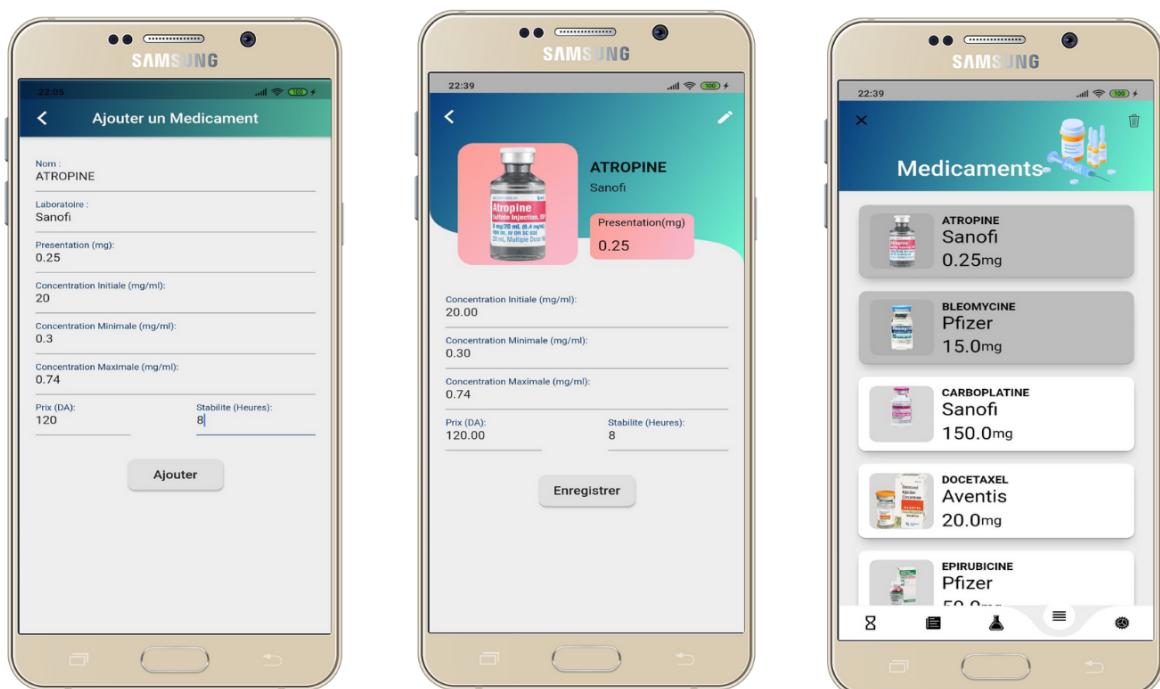


FIGURE 3.14 – Gestion De La Liste Des Médicaments

La deuxième Interface principale c'est L'interface Préparer Solution.



FIGURE 3.15 – Préparer Solution

Pour Préparer Une solution Il faut passer par 3 étapes : Ajouter Patient, Choix des médicaments, remplir les posologies.

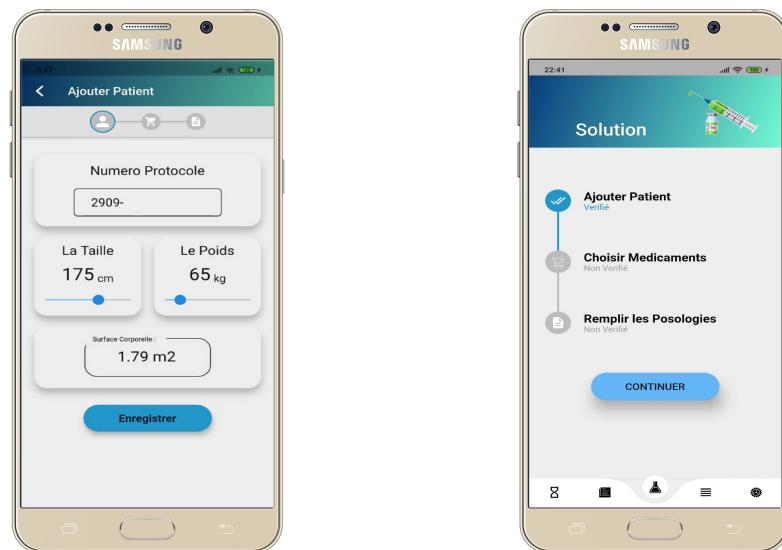


FIGURE 3.16 – Ajouter Patient

Ici On choisit les médicaments a utilisé dans une ordonnance on peut utiliser de 1 a 5 médicaments a la fois.

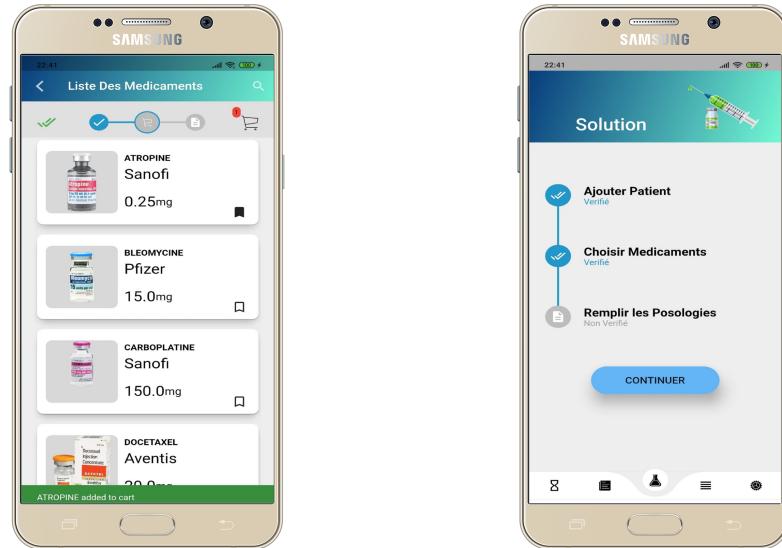


FIGURE 3.17 – Choix de médicament

La dernière étape c'est de remplir les posologie et clique sur Au Résultat pour voir les résultats.



FIGURE 3.18 – Remplir Posologie et Voir Résultat

La Troisième Interface Principale C'est L'interface de Bilan Du Jour qui s'affiche les ordonnances traité et la quantité consommé de chaque médicaments en volume et en nombre de flacon.



FIGURE 3.19 – Bilan Du Jour

La Quatrième Interface principale est celle de reliquat , il affiche le reliquat de chaque médicament avec le temps reste de stabilité et le prix de perte en cas ou le reliquat était périmé il s'affiche en premier avec une couleur rouge.

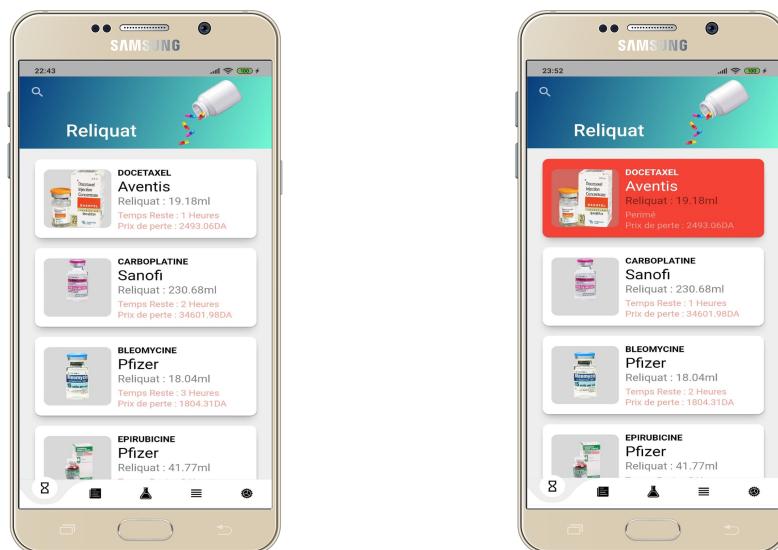


FIGURE 3.20 – Liste Des Reliquats

On a un cas très intéressant lorsque on essaye de préparer une solution avec un médicament qui a un reliquat périme il s'affiche une boîte de dialogue pour choisir entre une suppression **automatique** du reliquat périme ou une suppression **manuelle** qu'il va dirigé l'utilisateur vers la page reliquat pour supprimer le reliquat périme (en rouge) et identifier avec un signe jaune.

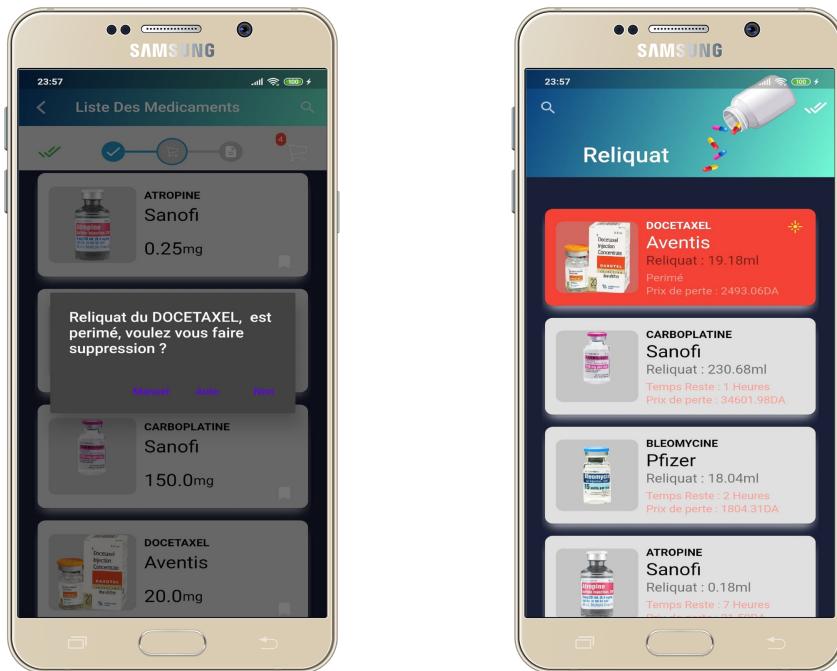


FIGURE 3.21 – Suppression De Reliquat Périme

La cinquième Interface principale est l'interface de paramétrés, tu peux changer le nom d'utilisateur, le mot de passe, le thème (clair ou sombre), et même voir le rapport de fin du journée et le télécharger en format PDF.



FIGURE 3.22 – Interface Paramétrés

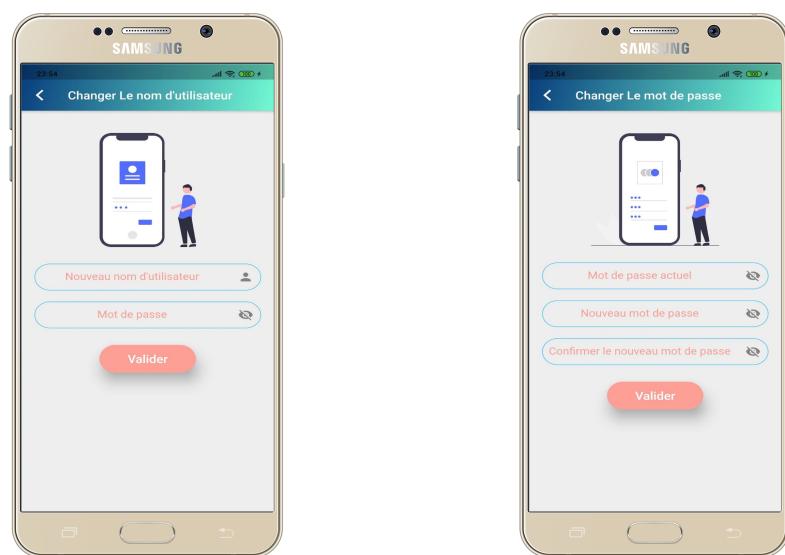


FIGURE 3.23 – Changer Nom D'utilisateur et Le Mot de Passe

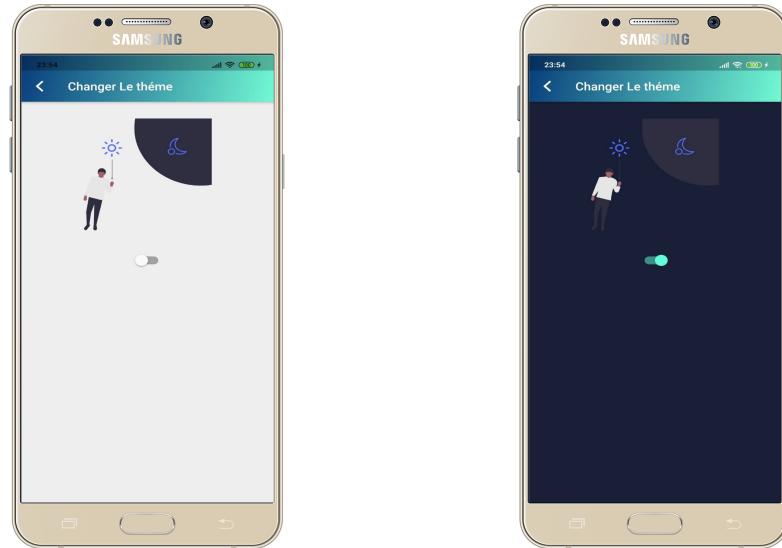


FIGURE 3.24 – Changer Le Thème



FIGURE 3.25 – Rapport de fin du journée PDF

Quelques Interface avec le thème sombre.

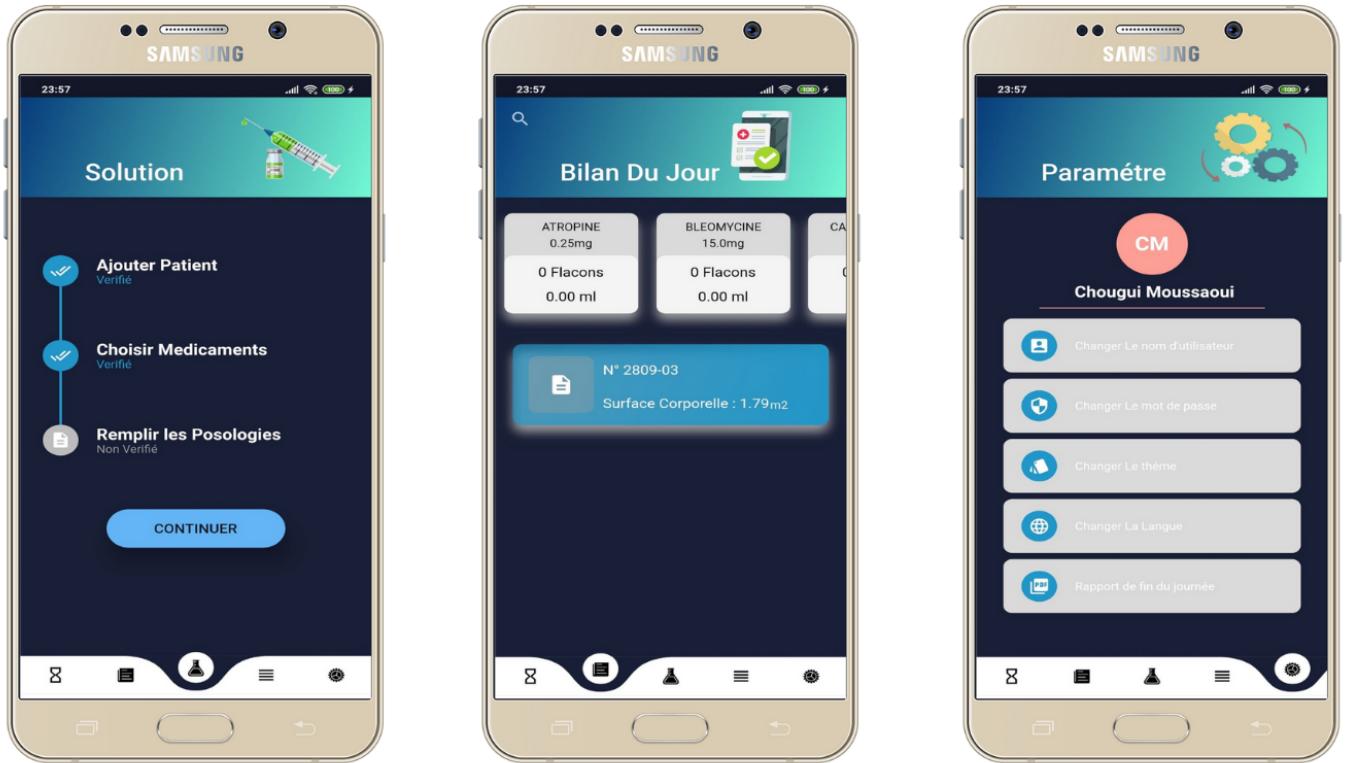


FIGURE 3.26 – Le Thème Sombre

3.4 Une partie de code source <Fonction de Calcul>

```

129     calculer() async {
130         for (int i = 0; i < listeDesMedics.length; i++) {
131             listesDesPosologies.add(double.parse(posologieController[i].text));
132
133             listeDesDoses.add(listesDesPosologies[i] * listeDesProtocoles[listeDesProtocoles.length - 1].surfaceCorporelle);
134
135             listeDesVolumesFinales.add(listeDesDoses[i] / listeDesMedics[i].concentrationInitiale);
136
137             if (listeDesMedics[i].reliquat == null) {
138                 listeDesMedics[i].dateDeConservation = (DateTime.now().day).toString() + "-" + (DateTime.now().hour).toString();
139                 listeDesNbFlacons.add((listeDesVolumesFinales[i] ~/ listeDesMedics[i].presentation).floor() + 1);
140                 listeDesMedics[i].reliquat = ((listeDesNbFlacons[i] * listeDesMedics[i].presentation) - listeDesVolumesFinales[i]);
141             }
142             else {
143                 int jourDeConservation = int.parse(listeDesMedics[i].dateDeConservation.substring(0, listeDesMedics[i].dateDeConservation.indexOf("-")));
144                 int heureDeConservation = int.parse(listeDesMedics[i].dateDeConservation.substring(listeDesMedics[i].dateDeConservation.indexOf("-") + 1));
145                 int differenceJour, differenceHeure;
146
147                 // if we are in the same day of save
148                 if (jourDeConservation == DateTime.now().day) {
149                     if (heureDeConservation <= DateTime.now().hour) {
150                         differenceHeure = DateTime.now().hour - heureDeConservation;
151                         differenceJour = 0;
152                     }
153                 }
154                 else {
155                     // if we are in a different day and we make 24 hours
156                     if (heureDeConservation <= DateTime.now().hour) {

```

FIGURE 3.27 – Code Source Partie 1

```

55         // if we are in a different day and we make 24 hours
56     if (heureDeConservation <= DateTime.now().hour) {
57         differenceJour = DateTime.now().day - jourDeConservation;
58         differenceHeure = DateTime.now().hour - heureDeConservation;
59     }
60     else {
61         differenceJour = DateTime.now().day - jourDeConservation - 1;
62         differenceHeure = 24 + DateTime.now().hour - heureDeConservation;
63     }
64 }
65
66 int differenceFinale = differenceJour * 24 + differenceHeure;
67
68 if (listeDesMedics[i].stabilite > differenceFinale) {
69     listeDeNbFlacons.add(((listeDesVolumesFinales[i] - listeDesMedics[i].reliquat) / listeDesMedics[i].presentation).floor().toInt() + 1);
70     listeDesMedics[i].reliquat = (listeDeNbFlacons[i] * listeDesMedics[i].presentation) - (listeDesVolumesFinales[i] - listeDesMedics[i].reliquat);
71     listeDesMedics[i].dateDeConservation = (DateTime.now().day).toString() + "-" + (DateTime.now().hour.round()).toString();
72 }
73 else {
74     listeDeNbFlacons.add(((listeDesVolumesFinales[i] / listeDesMedics[i].presentation).floor().toInt() + 1);
75     listeDesMedics[i].reliquat = ((listeDeNbFlacons[i]*listeDesMedics[i].presentation) - listeDesVolumesFinales[i]);
76     listeDesMedics[i].dateDeConservation = (DateTime.now().day).toString() + "-" + (DateTime.now().hour.round()).toString();
77 }
78 }
79

```

FIGURE 3.28 – Code Source Partie 2

3.4.1 Explication De La Fonction De Calcul

Au Début on fait une boucle for sur les médicaments sélectionné d'une ordonnance ca dépend de 1 a 5, pour chaque médicament on a listesDesPosologie qui est une liste de type double on va ajouter a cette liste les posologies écrite par l'utilisateur grâce a posologieController de notre textField. puis ListeDesDose on lui ajouter l'élément i (médicament i) de ListeDesPosologie X surface corporelle de médicament i de la dernière ordonnance (protocole). après on a ListeDesVolumesFinale de type double on lui ajouter dose/concentration Initiale. on sait déjà que les attributs reliquat et date de conservation sont des attribut de la classe médicament. la calcul de reliquat ca dépend le cas. le cas ou le reliquat est null. date de conservation reçoit le jour-heures de l'instant de conservation , puis on ajoute le nombre des flacons a ListeNombreFlacon par le calcul de volume finale / présentation on prend l'entier de résultat et on lui ajoutant 1, et puis l'attribut reliquat reçoit (nombre de flacon / présentation) - volume finale. dans le cas ou il ya une reliquat on test si cette reliquat est encore utilisable si oui lorsque on calcule le nombre de flacon ou la nouvelle reliquat en prend en considération le reliquat dans les calculs et soustraire cette reliquat a partir de volume finale, sinon si le reliquat est périmé donc on prend pas cette valeur en considération dans les calculs.

3.5 Conclusion

Ce chapitre a été consacré à la phase de réalisation , on a citer les outils logiciels et matériels qu'on a utiliser pour développer cette application , puis on a expliquer le processus d'utilisation de l'application avec des captures d'écran et a la fin on ajouter une partie de code source avec explication.

Conclusion Générale

Nous sommes parvenus, par le biais de ce projet, à réaliser une application qui fonctionne sous Android qui permet aux opérateurs pharmaciens de préparer une solution de médicaments pour perfusion d'un malade, à partir d'une base de données de médicaments (locale).

Au cours de la phase d'analyse nous avons structuré et défini les besoins du système. Il s'agit de formuler, d'affiner et d'analyser la pluparts des cas d'utilisations par les diagrammes UML. La phase de conception suit immédiatement la phase d'analyse, il s'agit alors d'étendre la représentation effectuée au niveau de l'analyse en y intégrant les aspects techniques les plus proches des préoccupations des besoins techniques. L'élément principal à livrer au terme de cette phase est le diagramme de classe. Enfin, nous avons entamé la réalisation en utilisant les outils de développement matériels et logiciels et présenter les différentes interfaces de notre application.

Malgré toutes les difficultés que nous avons rencontrées durant la réalisation de cette mémoire, ce travail nous a permis d'apprendre énormément de choses concernant le développement sous Android, et ce mémoire présente notre première expérience pour la mise en œuvre d'une application mobile, également nous avons appris à manipuler toute les d'outils : Android studio , Language de programmation Dart Avec La Framework Flutter , SQLite et d'autres outils . . .

Enfin, nous souhaitons que ce modeste travail satisfera les utilisateurs et apportera un maximum d'aide et pourra être bénéfique et utile.

BIBLIOGRAPHIE

- [1] <https://flutter.dev/>
- [2] <https://www.youtube.com/flutterdev>
- [3] <https://vogella.developpez.com/>
- [4] [https://fr.wikipedia.org/wiki/Dart_\(langage\)](https://fr.wikipedia.org/wiki/Dart_(langage))
- [5] <https://www.frandroid.com/>
- [6] <https://www.uml-sysml.org>
- [7] https://fr.wikipedia.org/wiki/Android_Studio#