

INSTITUT NATIONAL DE STATISTIQUE ET
D'ÉCONOMIE APPLIQUÉE



The use of Swing Framework to implement graphical to manage the vaccination process

M'chichi Boutayna

10 octobre 2021

MOTS CLÉS

- ☐ POO
- ☐ Java
- ☐ Eclipse
- ☐ Star UML
- ☐ Gestion
- ☐ Vaccination

TABLE DES MATIÈRES

1	Objectif	6
2	Modélisation	7
1	Digramme de classe	7
2	Digramme de cas d'utilisation	8
2.1	Identification des acteurs	8
2.2	Description textuelle des cas d'utilisation	8
3	Digramme de séquence	10
3	Developpement de notre application	13
1	Choix du langage	13
1.1	Java	13
1.2	UML	14
4	Présentation de l'application et discussion	15
1	Le frontend de l'application	15
1.1	Une première vue	15
1.2	Les composants de notre interface	16
1.3	Un premier essai	16
2	Le backend de l'application	20
A	Le code source	21

LISTE DES FIGURES

2.1	Diagramme de cas d'utilisation	7
2.2	Diagramme de cas d'utilisation	9
2.3	Diagramme de séquence 1	10
2.4	Diagramme de séquence 2	11
2.5	Diagramme de séquence 3	12
3.1	Symbole du langage java	13
3.2	Symbole de StarUML	14
4.1	Interface de notre application	15
4.2	La saisi des informations	16
4.3	Ajouter un patient à la liste des sujets	17
4.4	Vérification de l'âge	17
4.5	Vérification de l'âge	18
4.6	Consultation du médecin	18
4.7	Mise à jour du stock	19
4.8	Essai application	19
4.9	Arborescence du projet	20

Le monde a connu cette année un bouleversement comme l'on n'a jamais vu auparavant ; les écoles fermées, les magasins ont baissé leurs rideaux, les rues devenues dénudées de toute trace humaine . . . Il s'agit bien de la pandémie qui hante nos vies depuis Mars dernier et qui nous hante toujours jusqu'à l'heure d'aujourd'hui, à croire qu'on n'en sortira jamais de ce fléau qui ôte la vie à plusieurs d'entre nous ; des connaissances, des membres de la famille . . . Malgré la fatalité de ce fléau et sa vitesse de propagation, nous sommes enfin arrivés à une solution qui pourrait mettre fin à ce cauchemar : Le vaccin. De nos jours, la course n'est plus après le pétrole, les pays se battent afin d'obtenir le plus grand nombre possible de vaccin afin de mettre fin à cette pandémie. Or, afin de profiter au maximum du nombre limité de vaccin que reçoit chaque pays, il s'ensuit une bonne préparation au préalable. Notre travail donc consiste à la réalisation d'un projet sous Java qui permet de gérer la campagne de vaccination de la manière la plus optimale possible.

CHAPITRE 1

OBJECTIF

Ce projet vise l'amélioration de la gestion des prises de rendez-vous de vaccination contre COVID19 dans les établissements de santé. . Cela permettra la prise d'un rendez-vous, son report ou son annulation tant par le patient que par le médecin 24h/24 et 7jr/7. Il sera ainsi question de réduire le flux de coups de fil dans les établissements de santé ainsi que le nombre de rendez-vous non honorés ; situation qui pénalise aussi bien le patient que le médecin. . Il s'agira de permettre au patient de consulter les dates de ses deux rendez-vous (première vaccination et la deuxième).



1 Diagramme de classe

Le diagramme de classe montre la structure interne d'un système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation.

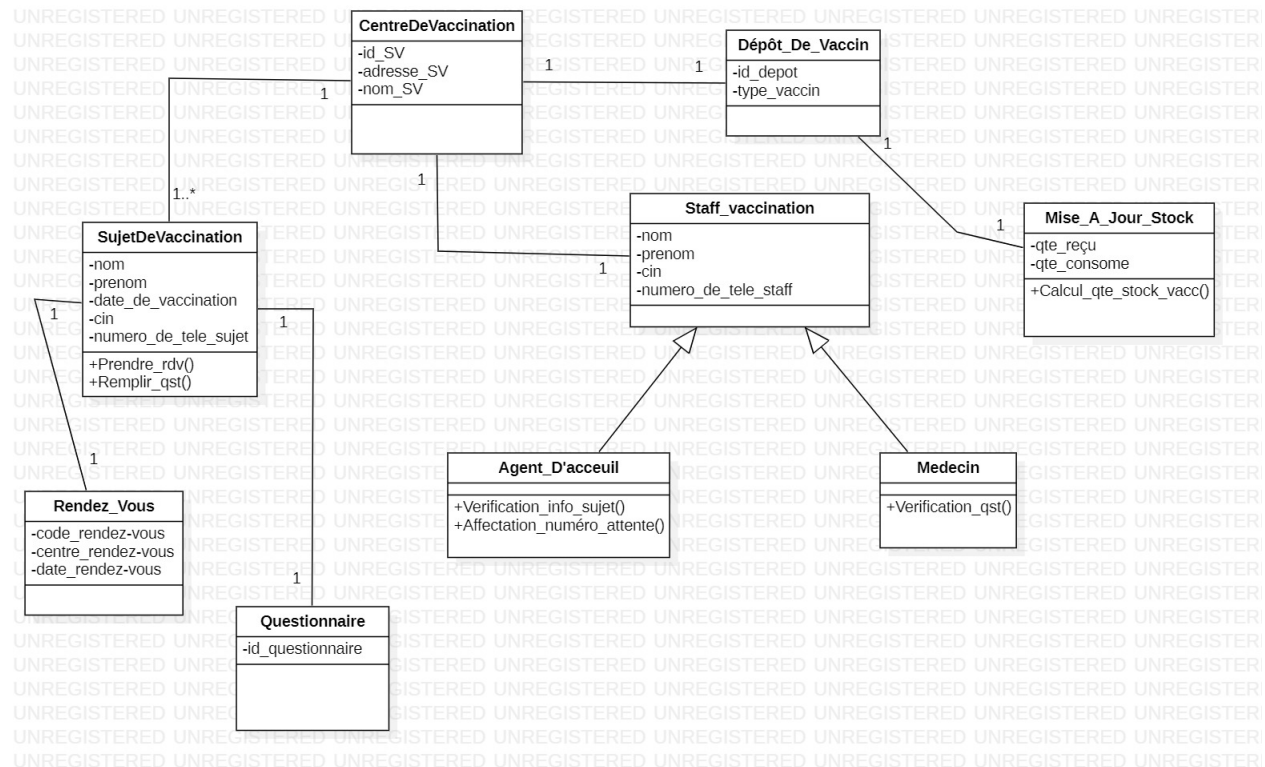


FIGURE 2.1 – Diagramme de cas d'utilisation

2 Digramme de cas d'utilisation

Cette partie de notre modélisation consistera à présenter les différents acteurs du système ainsi que les différentes interactions qu'ils pourront effectuer avec celui-ci.

2.1 Identification des acteurs

En UML, on entend par acteur toute idéalisation d'un rôle joué par une personne externe, un processus ou une entité qui interagit avec le système. Dans le cadre de notre système nous avons retenu :

2.2 Description textuelle des cas d'utilisation

Nous décrirons ci-dessous les cas d'utilisation pour l'application de gestion de prise de rendez-vous :

Le patient va se présenter au centre de vaccination :

- Acteur : Patient
- Description ; Toute personne se présentant au centre de vaccination a soit déjà reçu un SMS qui contient la date de rendez-vous, soit il vient par lui-même(en suivant la tournure actuelle de la procédure de vaccination au Maroc)
- Pré-condtion ; Soit il a envoyé CIN au 1717(pour le premier cas cité auparavant), oui bien il s'est présenté par lui même.

Le patient doit en premier remplir un questionnaire qui se compose des éléments suivants :

- ☐ Nom
- ☐ Prénom
- ☐ Age
- ☐ CIN

Par la suite, l'agent d'accueil va vérifier ces informations et ensuite décider selon la condition de l'âge > 18 ans décider comme suit :

- ◊ Le patient respecte la conditon, on lui affecte un numéro d'attente qui s'affiche sur l'interface de l'application
- ◊ Le cas échéant, il n'a pas de numéro et donc ne pas terminer le procesuss de vaccination

Sur place, son tour arrivé, le patient se dirige vers le médecin, qui, à son tour vérifie avec lui d'autres conditions supplémentaires.

Au fur et à mesure, un infirmier comptabilise le nombre de doses administrés et calucle le stock restant, ces infos sont également affichés sur l'interface

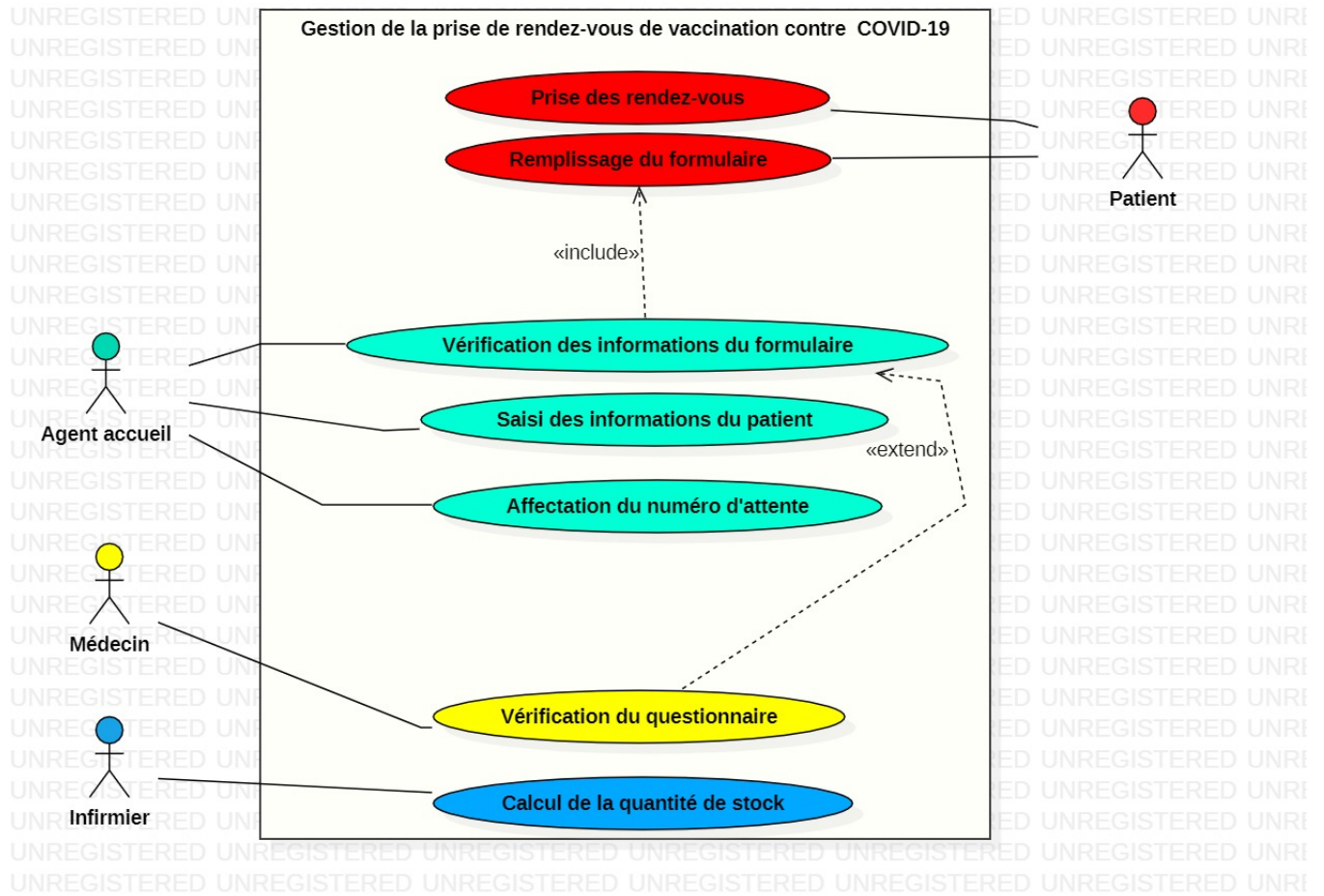


FIGURE 2.2 – Diagramme de cas d'utilisation

3 Digramme de séquence

Le diagramme de séquence est une représentation intuitive lorsque l'on souhaite concrétiser des interactions entre deux entités (deux sous-systèmes ou deux classes d'un futur logiciel). Ils permettent à l'architecte ou le designer de créer au fur et à mesure sa solution. Cette représentation intuitive est également un excellent vecteur de communication dans une équipe d'ingénierie pour discuter cette solution. Ils permettent de décrire COMMENT les éléments du système interagissent entre eux et avec les acteurs :

- ◇ Les objets au cœur d'un système interagissent en s'échangeant des messages.
- ◇ Les acteurs interagissent avec le système au moyen d'IHM (Interfaces Homme-Machine).

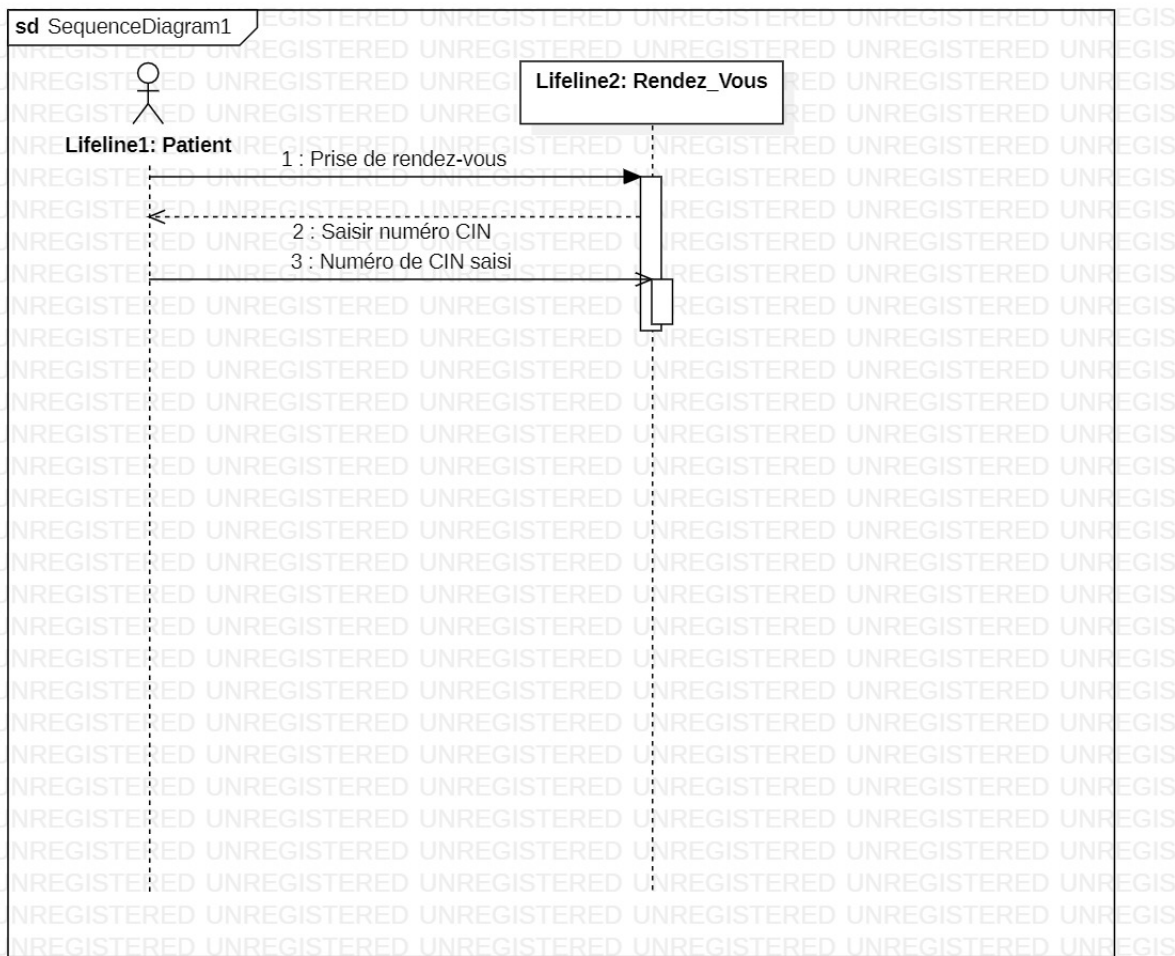


FIGURE 2.3 – Diagramme de séquence 1

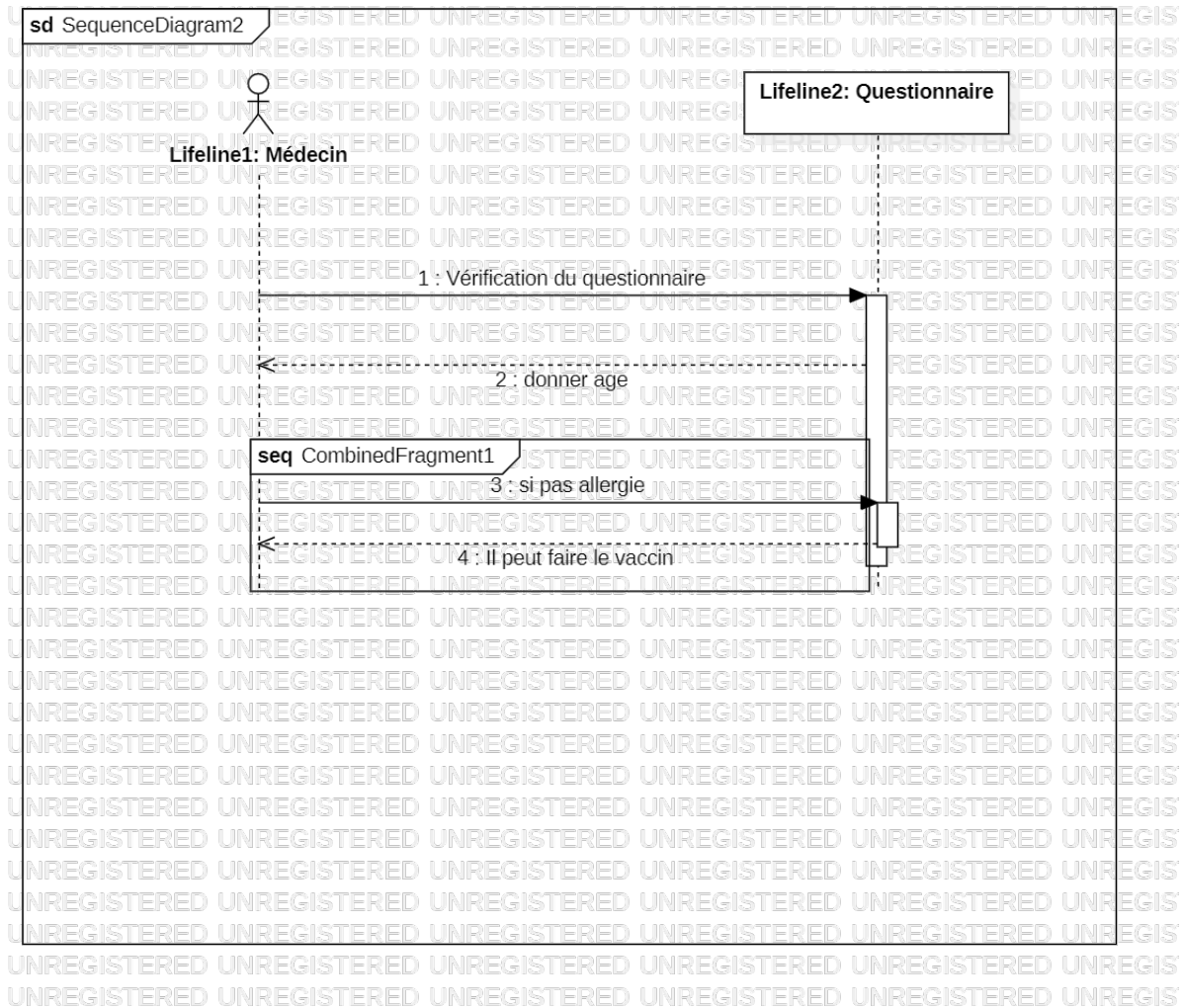


FIGURE 2.4 – Diagramme de séquence 2

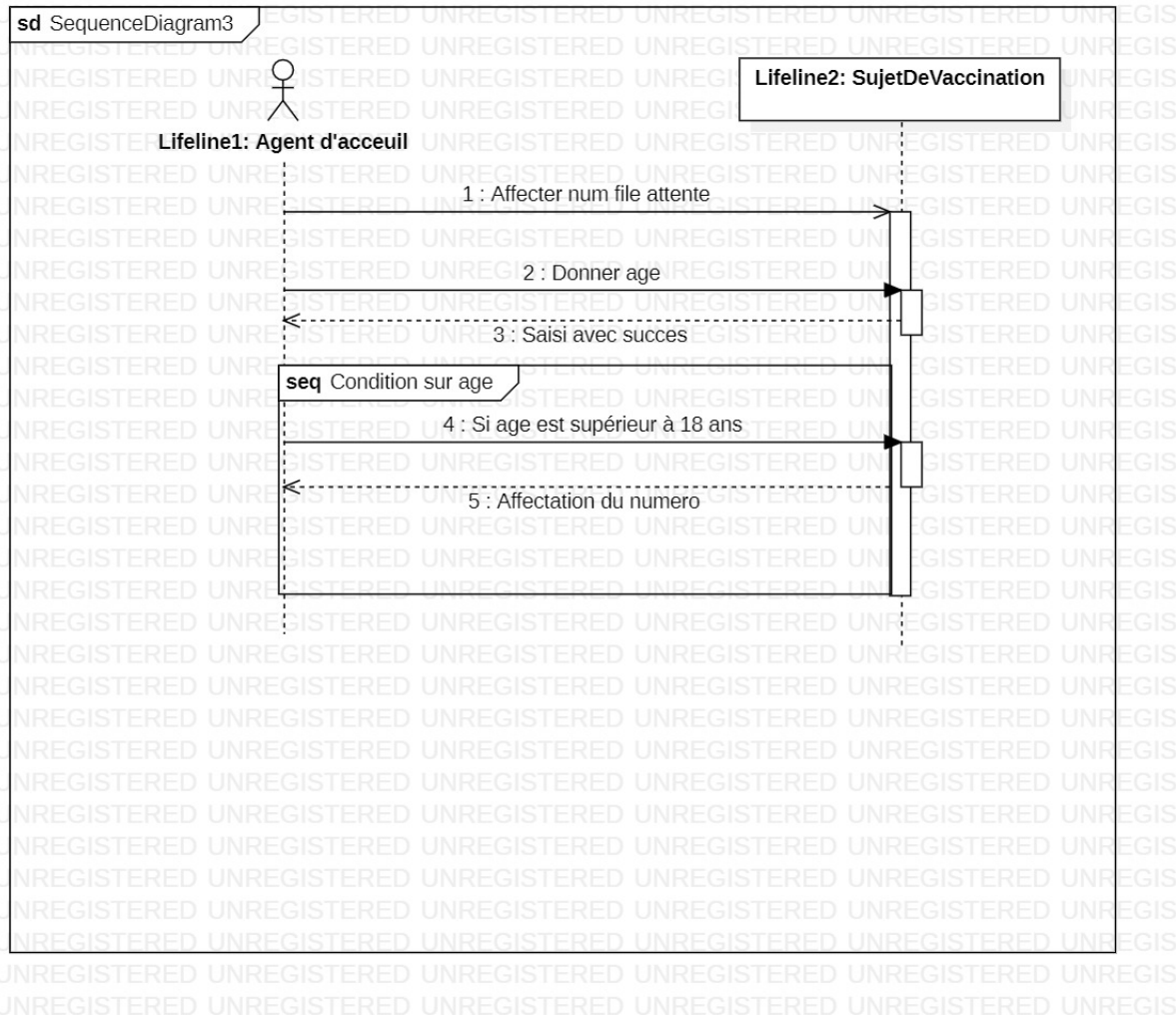


FIGURE 2.5 – Diagramme de séquence 3

1 Choix du langage

1.1 Java

Un langage de modélisation est un langage artificiel qui peut être utilisé pour exprimer de l'information ou de la connaissance ou des systèmes dans une structure qui est définie par un ensemble cohérent de règles. Dans le cadre de l'évaluation du module **Programmation orientée Objet** sous la direction de **Mme Hind LAMHARHAR**, nous avons programmé avec le langage JAVA avec l'éditeur Eclipse.



FIGURE 3.1 – Symbole du langage java

1.2 UML

L'UML est un langage nous permettant de modéliser nos classes et leurs interactions. Autrement c'est un ensemble d'annotations graphiques s'appuyant sur des diagrammes et permettant de spécifier, visualiser et de documenter les systèmes logiciels orientés-objets. Notre choix s'est porté sur le langage UML pour ses nombreux avantages. En effet, avec UML nous avons le concept des objets auxquels associés les principes d'héritages de composition et transition d'état.



FIGURE 3.2 – Symbole de StarUML

CHAPITRE 4

PRÉSENTATION DE L'APPLICATION ET DISCUSSION

1 Le frontend de l'application

1.1 Une première vue

Dans ce qui suit, nous allons expliquer pas à pas les différentes options que présentent notre application.

The screenshot shows a web application interface for 'Gestion de rendez-vous' (Appointment Management). The interface is divided into three main sections:

- Saisir les informations (Enter information):** This section contains input fields for patient details: 'Nom citoyen' (Citizen name), 'Prenom citoyen' (Citizen first name), 'Age citoyen' (Citizen age), and 'CIN'. There is also a dropdown menu for 'Maladie chronique' (Chronic disease) with 'Oui' (Yes) selected. To the right of these fields are three buttons: 'Ajouter un patient(e)' (Add patient(s)), 'Age requis' (Required age), and 'Affecter num d'attente' (Assign waiting number). Below these buttons is a button labeled 'Tester aptitude vaccin' (Test vaccine suitability).
- Liste des sujets (List of subjects):** This section displays a table with the following columns: 'Nom patient', 'Prenom Patient', 'CIN', 'Âge', 'Maladie chronique', 'Âge requis pour se faire va...', 'Numéro d'attente', and 'Aptitude à faire vaccin'. The table body is currently empty.
- Mise à jour du stock (Stock update):** This section contains two buttons: 'Calcul nombre de doses donnés' (Calculate number of doses given) and 'Mise à jour du stock du vaccin' (Update vaccine stock). Each button is followed by an input field.

At the bottom right of the interface, there is a watermark that reads 'Activer Windows' and 'Accédez aux paramètres pour activer Windows.' (Access settings to activate Windows).

FIGURE 4.1 – Interface de notre application

1.2 Les composants de notre interface

Comme le montre l'image ci-dessus, notre interface se compose de :

- ◊ Trois JPanel : Saisir mes informations, Liste des sujets, Mise à jour du stock
- ◊ Six boutons :
 1. Ajouter patient
 2. Âge requis
 3. Affecter numéro d'attente
 4. Tester aptitude vaccin
 5. Calcul nombre de doses administrés
 6. Mise à jour du stock
- ◊ Un tableau où s'affiche les données saisies par l'utilisateur ainsi que le résultat des boutons cités auparavant

1.3 Un premier essai

Saisir

Nous allons commencer par saisir les informations demandés dans le JPanel "Saisir les informations" :

The screenshot shows a window titled "Gestion de rendez-vous" with a sub-header "Saisir les informations". The form has the following fields and buttons:

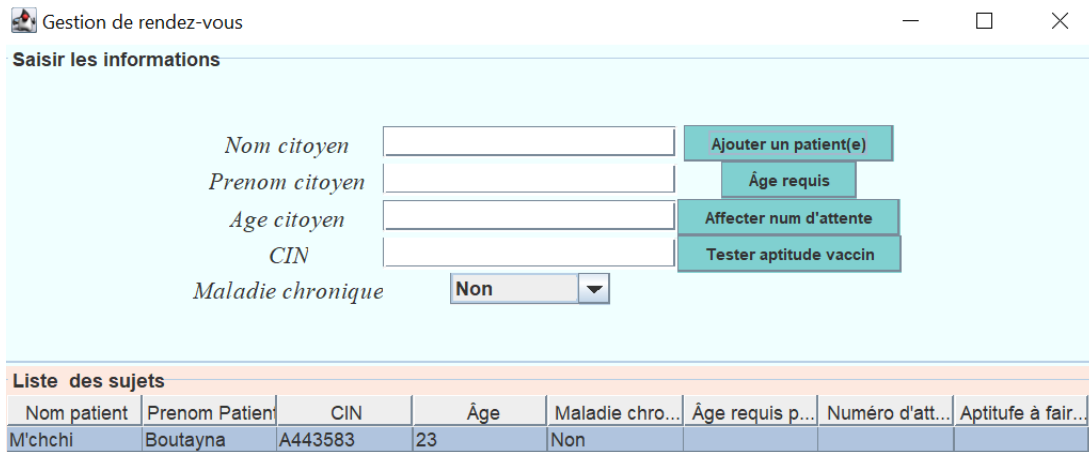
Label	Value	Action Button
Nom citoyen	M'chchi	Ajouter un patient(e)
Prenom citoyen	Boutayna	Âge requis
Age citoyen	23	Affecter num d'attente
CIN	A443583	Tester aptitude vaccin
Maladie chronique	Non	

FIGURE 4.2 – La saisi des informations

Ajouter

Par la suite, nous allons cliquer sur le bouton ajouter pour avoir le résultat suivant :

N.B : Pour des raisons de facilité d'utilisations, à chaque fois que l'utilisateur clique sur le bouton **Ajouter un patient(e)**, les zone de texte s'iniliasent à nouveau.



Gestion de rendez-vous

Saisir les informations

Nom citoyen **Ajouter un patient(e)**
Prenom citoyen **Âge requis**
Age citoyen **Affecter num d'attente**
CIN **Tester aptitude vaccin**
Maladie chronique

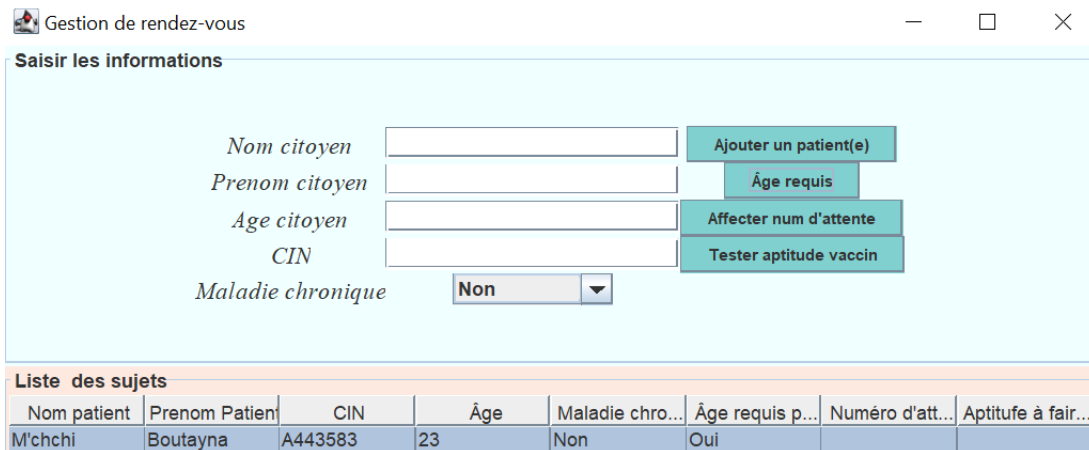
Liste des sujets

Nom patient	Prenom Patient	CIN	Âge	Maladie chro...	Âge requis p...	Numéro d'att...	Aptitufe à fair...
M'chchi	Boutayna	A443583	23	Non			

FIGURE 4.3 – Ajouter un patient à la liste des sujets

Vérifier âge

Comme cité auparavant, le première étape à faire est la vérification de l'âge par un agent d'accueil et ce en cliquant sur le bouton "Âge requis" :



Gestion de rendez-vous

Saisir les informations

Nom citoyen **Ajouter un patient(e)**
Prenom citoyen **Âge requis**
Age citoyen **Affecter num d'attente**
CIN **Tester aptitude vaccin**
Maladie chronique

Liste des sujets

Nom patient	Prenom Patient	CIN	Âge	Maladie chro...	Âge requis p...	Numéro d'att...	Aptitufe à fair...
M'chchi	Boutayna	A443583	23	Non	Oui		

FIGURE 4.4 – Vérification de l'âge

Affecter numéro file d'attente

Si la condition de l'âge est vérifiée, on va affecter au patient un numéro de file d'attente pour avoir une consultation avec le médecin avant l'injection de la dose du vaccin (la 1ère ou la 2ème)

The screenshot shows a window titled 'Gestion de rendez-vous'. Inside, there's a form titled 'Saisir les informations' with the following fields and buttons:

- Nom citoyen*: text input
- Prenom citoyen*: text input
- Age citoyen*: text input
- CIN*: text input
- Maladie chronique*: dropdown menu with 'Oui' selected
- Buttons: 'Ajouter un patient(e)', 'Âge requis', 'Affecter num d'attente', 'Tester aptitude vaccin'

Below the form is a table titled 'Liste des sujets' with the following data:

Nom patient	Prenom Patient	CIN	Âge	Maladie chro...	Âge requis p...	Numéro d'att...	Aptitufe à fair...
M'chici	Boutayna	A443584	23	Oui	Oui	1	

FIGURE 4.5 – Vérification de l'âge

Consultation médecin

En dernier lieu s'impose un passage chez le médecin, qui va revérifier la fiche du malade et entretenir d'autres questions supplémentaires avec le patient ; des questions qui concernent des maladies chroniques, des allergies...Par la suite, il décidera si le patient est apte à faire le vaccin ou pas.

The screenshot shows the same 'Gestion de rendez-vous' window. The form 'Saisir les informations' is identical to the previous one. The 'Liste des sujets' table now has an additional column 'Aptitufe à faire vaccin' with the value 'Le patient ne peut pas fair...' for the patient M'chici.

Nom patient	Prenom Patient	CIN	Âge	Maladie chronique	Âge requis pour se faire va...	Numéro d'attente	Aptitufe à faire vaccin
M'chici	Boutayna	A443584	23	Oui	Oui	1	Le patient ne peut pas fair...

FIGURE 4.6 – Consultation du médecin

Mise à jour du stock

Enfin, au fur et à mesure, un infirmier s'occupe de calculer le nombre de doses administrées et de faire la mise à jour du stock

Liste des sujets							
Nom patient	Prenom Patient	CIN	Âge	Maladie chro...	Âge requis p...	Numéro d'att...	Aptitude à fair...
M'chichi	Boutayna	A443584	22	Non	Oui	1	Le patient pe...

Mise à jour du stock

Calcul nombre de doses donnés

1

Mise à jour du stock du vaccin

49

FIGURE 4.7 – Mise à jour du stock

Un aperçu sur plusieurs patients présents dans le système

Dans l'image ci-dessous, nous avons essayé de combiner tout les cas possibles afin de voir le rendu de l'application ;

Gestion de rendez-vous

Saisir les informations

Nom citoyen

Prenom citoyen

Age citoyen

CIN

Maladie chronique

Ajouter un patient(e)

Âge requis

Affecter num d'attente

Tester aptitude vaccin

Oui

Mise à jour du stock

Calcul nombre de doses donnés

2

Mise à jour du stock du vaccin

48

FIGURE 4.8 – Essai application

2 Le backend de l'application

Pour ce qui est du Backend de l'application, il s'agit en somme toute de la définition des classes, des instances,des méthodes... Le schéma ci-dessous représente l'arborescence de notre projet :

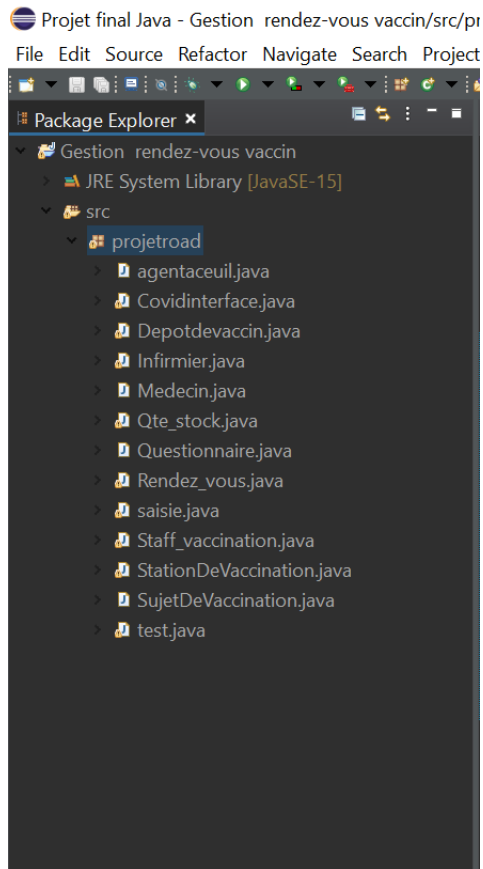


FIGURE 4.9 – Arborescence du projet

ANNEXE A

LE CODE SOURCE

Pour ne pas allourdir ce document avec plusieurs pages de code, vous allez trouver le projet tout entier inclus avec le dossier **Projet Final java**