SOMMAIRE :

1. CONTEXT DU PROJET :
   1. PRÉSENTATION DU PROJET
   2. LES OBJECTIFS DU SITE
   3. LES CIBLES
   4. BESOIN FONCTIONNELLE
   5. BESOIN NON FONCTIONNELLE
2. CONCEPTION  :
   1. DIAGRAMME DE CAS D'UTILISATION
   2. DIAGRAMME DE SÉQUENCE
   3. DIAGRAMME DE CLASS
   4. DIAGRAMME D’ETAT TRANSITION
3. SPÉCIFICITÉS ET LIVRABLES
   1. TECHNOLOGIE
   2. LES LIVRABLES
4. CONTEXT DU PROJET
5. PRÉSENTATION DU PROJET :

Dans le cadre de faciliter le suivi des dossiers des patients au Maroc, ce projet vise à créer une application console qui va être exploitée dans tous le réseau des agences CNSS Maroc. L’application MaCNSS va permettre aux agents CNSS de suivre en toute simplicité les dossiers de remboursement de chaque patient qui sont inscrits à la CNSS.

L’ACTIVITÉ PRINCIPALE :

Gestion des dossiers et remboursement des patients.

DURÉE DE RÉALISATION :

DU 29 Septembre 2022

AU 06 October 2022

L'ANALYSE DE L'EXISTANT :

* Ce projet ne contient pas un existant

PROBLÉMATIQUE :

Ce projet règle une problématique trouvé souvent dans ce processus de travaille tels que :

* Un traitement très long pour gérer les informations nécessaire pour valider le remboursement.
* Durée de remboursement assez prolongée.
* Possibilité de perte des documents.

SOLUTION :

La solution est de mettre en œuvre une application console pour objectif d’organiser d’une manière fiable et facile les informations nécessaire pour automatiser le processus de remboursement des clients.

pour un rendement plus rapide et efficace.

1. . LES OBJECTIFS DE L’APPLICATION :

Les objectifs de cette application se divisent en deux parties.

OBJECTIFS QUALITATIFS :

* Améliorer l’offre de service au client
* Traitement de donnés assez rapide
* Développer une stratégie d’évitement de contacts
* Améliorer les processus de traitement des demandes clients

OBJECTIFS QUANTITATIFS :

* Optimisez le temps de traitement des données.
* Contourné les erreures au niveau de calcule de remboursement

1. LES CIBLES :

Cette appli vise des cibles uniquement professionnelles pour gérer leur travail et l’automatiser d’une manière fiable et simple, ainsi qu’un public client qui peut à tout moment consulter ces données et leurs progression.

1. BESOIN FONCTIONNELLE

MaCNSS application viendra à l’existence pour satisfaire plusieurs besoins de différents utilisateurs :

- Agent CNSS :

* Chaque Agent CNSS a un compte dans l'application, pour qu'il puisse accéder à l'application, il doit saisir l'email + mot de passe avec u**n code vérification envoyé sur son boite email valable (5min).**
* L’agent va vérifier manuellement chaque document et par la suite il saisit toutes les informations sur l'application.
* Le système va indiquer automatiquement à l'agent le montant de remboursement de chaque document déposé par le patient.
* Le patient peut consulter à tout moment l’historique de ses dossiers de remboursement :
  + Le dossier du remboursement est soit : en attente, refusé ou validé
    - En attente : le dossier est en traitement par l’agent CNSS
    - Refusé : le dossier du patient est incomplet, **un email est envoyé automatiquement au patient concerné avec le motif de refus**
    - Validé : c’est un dossier terminé et traité par l’agent CNSS, **dans ce sens un email est envoyé automatiquement au patient avec le montant du remboursement**

- Patient :

* Le patient arrive à l'agence, il dépose son dossier de remboursement, qui est constitué obligatoirement d'un formulaire CNSS, ce dossier est identifié par le matricule du patient.
* Le patient peut joindre le code barre (si nécessaire) de chaque médicament prescrit sur l'ordonnance
* Le patient peut joindre les analyses de laboratoire.
* Le patient peut joindre une copie des radios ou scanner.

- Admin :

* Le seul administrateur dans le système va gérer les comptes des agents CNSS

1. BESOIN NON FONCTIONNELLE

— Fiabilité : fonctionner d’une façon cohérente sans erreurs ;

— Sécurité : respectez surtout la confidentialité des données ;

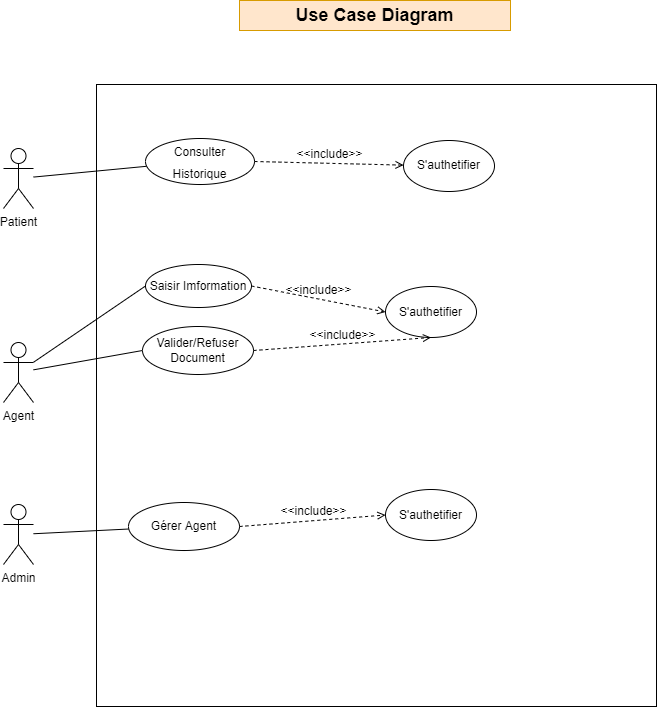
— Performance : c’est-à-dire à travers ses fonctionnalités, répond à toutes les exigences des usagers d’une manière optimale ;

— Les erreurs : Les ambiguïtés doivent être signalées par des messages d’erreurs bien organisés pour bien guider l’utilisateur et le familiariser avec l’application ;

c. CONCEPTION UML

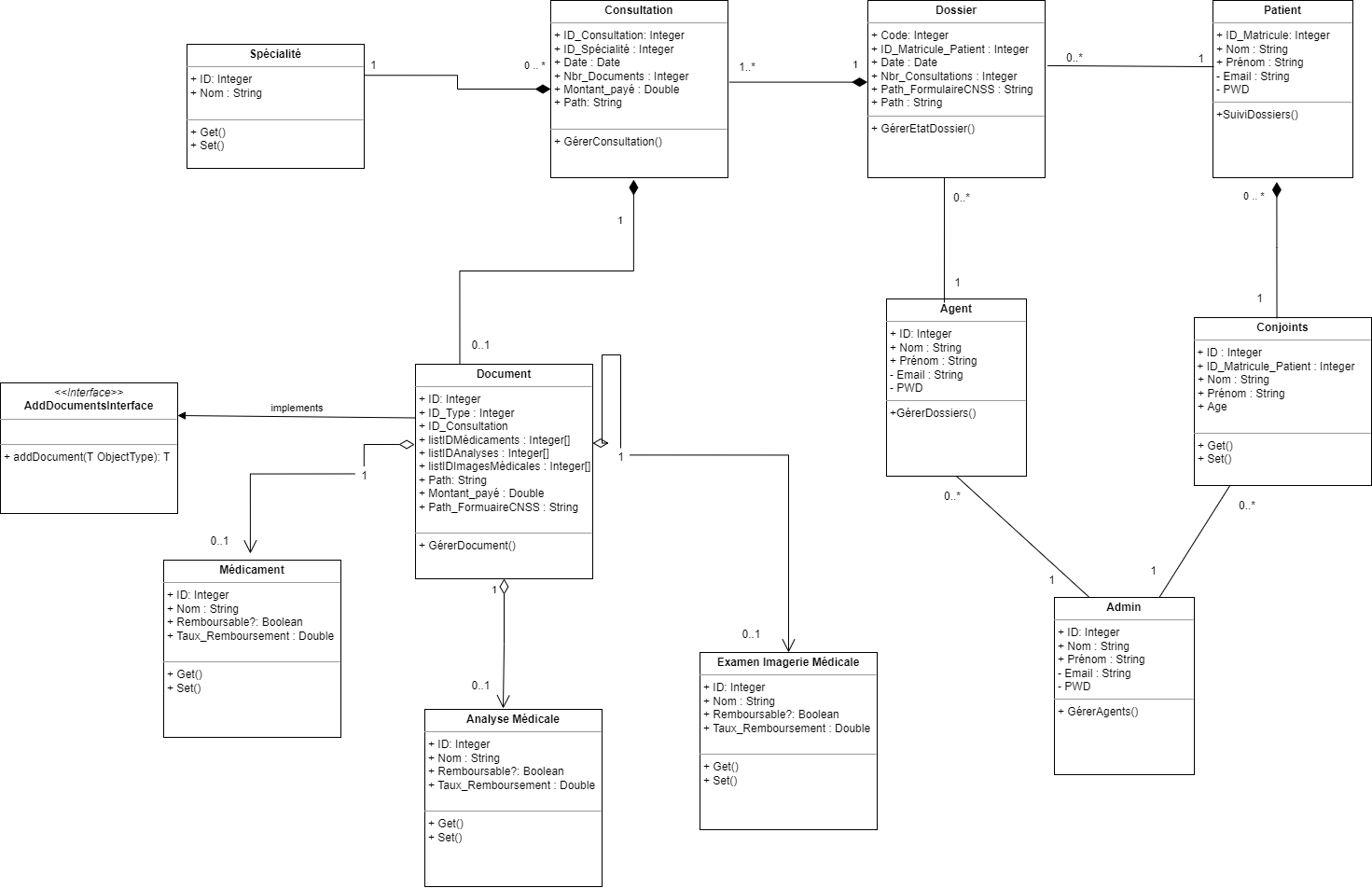
Le **langage de modélisation unifié** (UML) est un langage de modélisation de développement à usage général dans le domaine du génie logiciel qui vise à fournir un moyen standard de visualiser la conception d'un système.

* Diagramme de cas d’utilisation

Les **diagrammes de cas d'utilisation** (DCU) sont des **diagrammes** UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les **cas d'utilisation** sont plus appropriés.

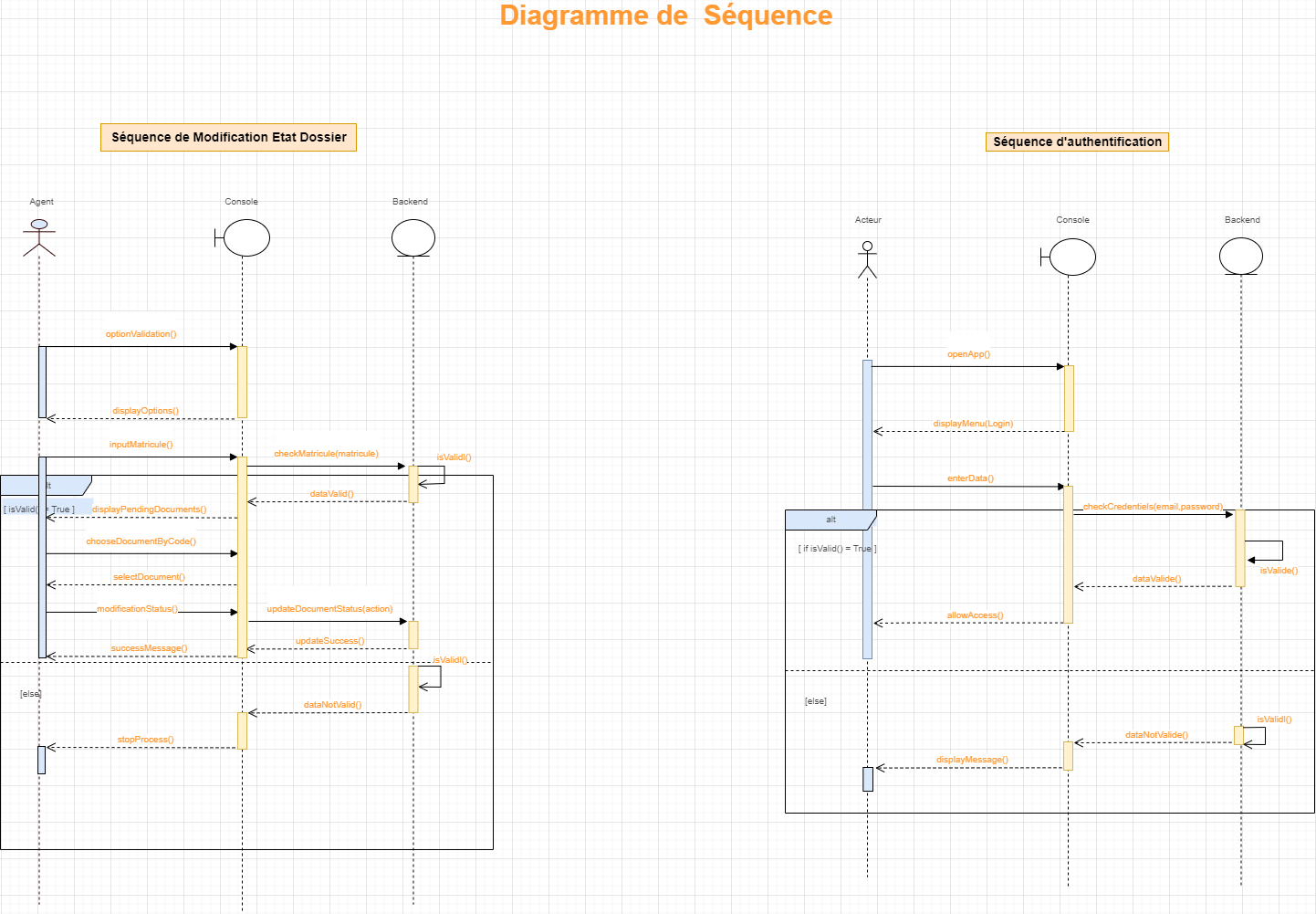
* Diagramme de class

 Un **diagramme de classe** montre la structure statique du modèle d'information, particulièrement les choses qui existent, leur structure interne, et leurs relations aux autres choses. Un **diagramme de classe** ne doit présenter aucune information de nature temporelle.

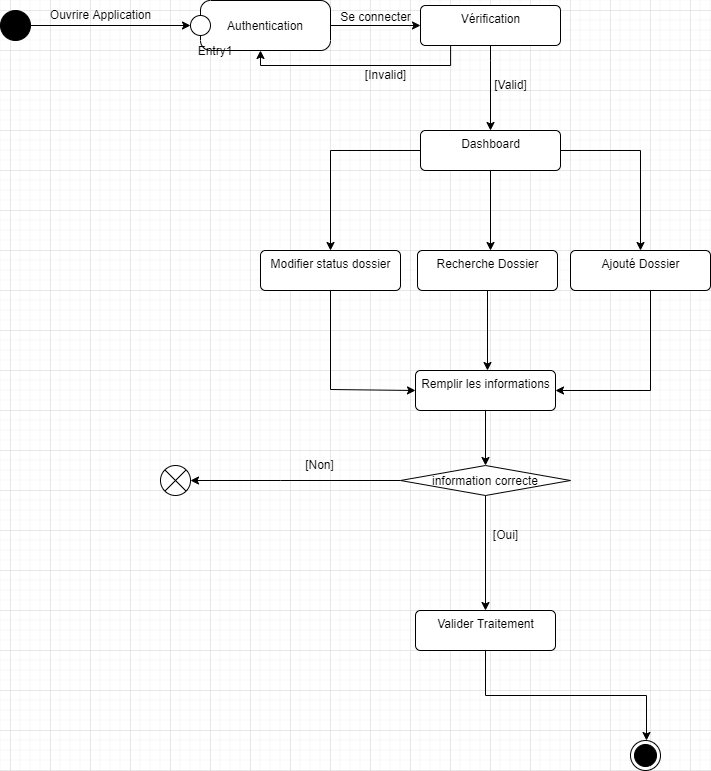


* Diagramme de séquence :

Un **diagramme de séquence** est un **diagramme** UML (Unified Modeling Language) qui représente la **séquence** de messages entre les objets au cours d'une interaction. Un **diagramme de séquence** comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction.



* Diagramme d’état de transition

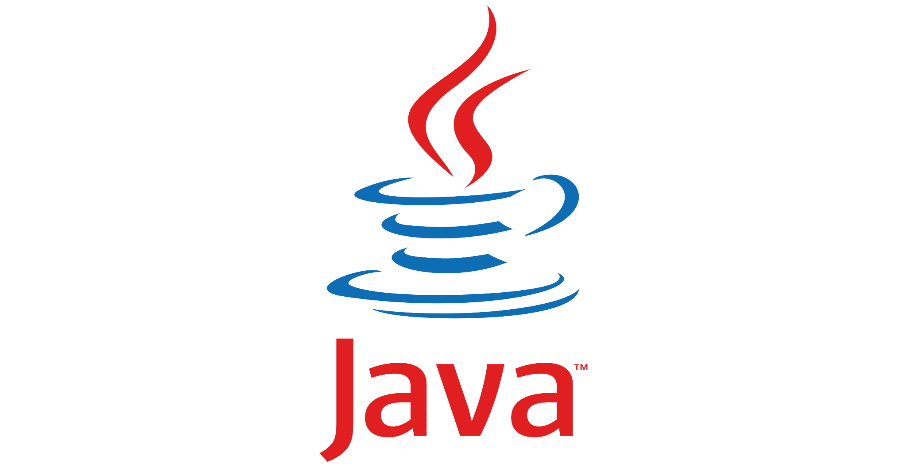
Un diagramme états-transitions est un schéma utilisé en génie logiciel pour représenter des automates déterministes. Il fait partie du modèle UML et s'inspire principalement du formalisme des state charts et rappelle les grafcets des automates. S'ils ne permettent pas de comprendre globalement le fonctionnement du système, ils sont directement transposables en algorithme. En effet, contrairement au diagramme d'activité qui aborde le système d'un point de vue global, le diagramme états-transitions cible un objet unique du système. Tous les automates d'un système s'exécutent parallèlement et peuvent donc changer d'état de façon indépendante.

C. SPÉCIFICITÉ ET LIVRABLE

a. TECHNOLOGIQUE

La technologie choisie pour la réalisation de ce projet et Java native :

* Java :



Le Java est un langage de programmation orientée objet développé par Sun Microsystems en 1995, et racheté depuis par Oracle. Le principal avantage de Java est son interopérabilité : la technologie fonctionne aussi bien sur Windows que Mac ou Linux, et sur une myriade d'appareils : centres de données, ordinateur, téléphone mobile, lecteur Blu-ray, périphériques TV, consoles de jeux, appareils connectés... Un autre avantage est son caractère universel : le même système peut être utilisé pour une grande variété d'applications. Le langage Java est basé sur le C++, mais avec une approche simplifiée et des fonctionnalités plus avancées.

b. LIVRABLE

Un livrable final est prévu à la fin de la réalisation du travail demandé.