## RESUME

L’objectif de ce projet est de mettre à la disposition du centre médical interentreprises(CMI) de Orange cote d’Ivoire Telecom(OCIT) une nouvelle application de gestion médicale. Cette nouvelle application prend en compte les besoins du centre qu’ils soient de la gestion des activités du circuit curatif, des activités du circuit préventif, de la gestion du stock pharmaceutique et la gestion des statistiques.

L’étude globale de la solution existante nous a permis de proposer des solutions appropriées. Nous avons procédé dans un premier temps à la formalisation des besoins du CMI qui s’est aboutie à la modélisation de ceux-ci. Ensuite nous avons procédés à la réalisation du projet en s’inspirant des modèles. Enfin des tests ont été réalisés sur la nouvelle solution avant la mise en production et la formation des Agents du CMI.

Soulignons que cette application permettra à Orange côte Télécom de suivre en temps Réel l’état de santé de ses agents, et quant au CMI elle lui permettra de gérer aisément le traitement d’un patient.

# 

# ETUDE CONCEPTUELLE

ents modèles qui décrieront notre c

hode d’analyse et de c

## CHAPITRE III- METHODE D’ANALYSE ET DE CONCEPTION

Une méthode d’analyse est un ensemble de démarches que suit l'esprit et l'arrangement qui en résulte*.* C’est donc un ensemble formé de modèles ou de langage et d’une démarche ou processus qui utilise ces modèles. En clair il s’agit de représenter le domaine à étudier ou à automatiser à l’aide de langage abstrait, en suivant des étapes bien précises, avec chaque étapes un modèles bien défini à élaborer. Son objectif étant de formaliser les étapes préliminaires de la conception d’un système informatique afin de rendre son développement plus fidèles aux besoins du client. Il existe plusieurs méthodes d’analyse et de conception : les plus utilisées sont la méthode MERISE et le processus Unifié que nous tenterons de décrire dans la suite.

### III.1- Présentation des méthodes d’analyse et de conception

#### III.1.1- Présentation du processus unifié(UML)

Le processus unifié est une méthode d’analyse et de conception générique de développement de logiciel. C’est une méthode :

* Pilotée par les cas d’utilisation d’UML
* Elle s’appuie sur un cycle de vie itératif et incrémental, Chaque itération porte sur un niveau d’abstraction de plus en plus précis.
* Centrée sur l’architecture
* Orientée vers la diminution des risques
* Il est construit autour de la création et de la maintenance d’un modèle, plutôt que de la production de montage de documents.
* Il est orienté composant.
* Il est orienté utilisateur.

De plus le processus unifié utilise une démarche en quatre (4) phases

* Démarche(Lancement) : phase d’initialisation du projet où l’on mène une étude d’opportunité et de faisabilité du système à construire, consiste aussi à l’indentification des cas d’utilisation.
* Élaboration : spécification du plan du projet, des exigences et des bases de l’architecture. Le plan est spécifié en un plan d’itérations.
* Construction : réalisation du produit
* Transition. Test Validation et déploiement.

Il utilise comme langage de modélisation **l’UML** (Unified Modeling Language) qui propose 13 diagrammes dans sa version 2.0 consigné dans le tableau suivant.

|  |  |  |
| --- | --- | --- |
| Diagrammes structurels(6) | Diagrammes fonctionnels et comportementaux(7) | |
| * Digramme de classe * Digramme d’objet * Diagramme de composant * Digramme de déploiement * Digramme de paquetage (UML 2) * Digramme de structure composite UML 2 | | * Digramme de cas d’utilisation * Digramme de séquences * Digramme d’activités * Digramme de communication * Digramme d’état transition * Digramme de temps UML 2 * Diagramme global d’interaction UML 2 |

Tableau 2: Diagrammes UML 2

Le processus unifié étant générique, il faut l’adapter au projet et à l’environnement de travail. En ce sens il existe plusieurs implémentations dont les principaux sont: RUP, XUP et 2TUP. Nous les présenterons dans le tableau récapitulatif suivant.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Description | Points fort | Point faible |
| Cascade | -Propose de dérouler les phases projet de  Manière séquentielle  - Cité pour des  raisons historiques | - Distingue  clairement les phases projet | - Non itératif  - Ne propose pas de  modèles de documents |
| RUP : Rational Unified Process | - Promu par  Rational.  - Le RUP est à la  fois une méthodologie et un outil prêt à l'emploi (documents types partagés dans  un référentiel Web)  - Cible des projets de  plus de 10 personnes | - Itératif  - Spécifie le dialogue  entre les différents  intervenants du  projet : les livrables,  les plannings, les  prototypes…  - Propose des modèles de documents, et des canevas pour des projets types  - Propose des modèles de documents, et des  canevas pour des  projets types | - Coûteux à  personnaliser  - Très axé processus, au détriment du  développement : peu de place pour le code et la technologie |
| XP eXtreme  Programming | - Ensemble de «  Bests Practices » de  développement  (travail en équipes, transfert de  compétences…)  - Cible des projets de  moins de 10  personnes | - Itératif  - Simple à mettre en  œuvre  - Fait une large place  aux aspects  techniques :  prototypes, règles de  développement, tests… | - Ne couvre pas les  phases en amont et en  aval au développement :  capture des besoins,  support, maintenance,  tests d'intégration…  - Elude la phase  d'analyse, si bien qu'on  peut dépenser son  énergie à faire et défaire  - Assez flou dans sa  mise en œuvre: quels  intervenants, quels  livrables ? |
| 2TUP :Two Track  Unified Process | - S'articule autour de  l'architecture  - Propose un cycle de  développement en Y  - Détaillé dans "UML  en action" (voir  références)  - Cible des projets de  toutes tailles | - Itératif  - Fait une large place  à la technologie et à la  gestion du risque  - Définit les profils  des intervenants, les  livrables, les  plannings, les prototypes | Plutôt superficiel sur  les phases situées en  amont et en aval du  développement : capture des besoins, support,  maintenance, gestion du changement…  - Ne propose pas de  documents types |

Tableau 3: Description des implémentations du PU

#### III.1.2- Présentation de la méthode Merise

La méthode MERISE (Méthode d’Etude et de Réalisation Informatique pour les Systèmes d’Entreprise), créée dans les années 70, a pour objectif de fournir à la fois une démarche, des modèles, des formalismes et des normes pour la conception et la mise en place du système d’information d’une entreprise. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. Cette séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment. La puissance de la méthode réside dans le fait qu’elle permet de schématiser les niveaux d’abstraction et offre un niveau de granularité adaptable à tous les besoins.

La méthode MERISE utilise :

* **un modèle fonctionnel** basé sur les diagrammes de flux (flux d’informations ou messages) ;
* **un modèle statique** basé sur l’Entité-Association enrichi de méthodes de traitement. Il décrit les tâches à effectuer à la réception ou pour l’émission d’un flux d’informations;
* **un modèle dynamique** des objets explicitant le contrôle et les interactions des objets. C’est la structure de mémorisation des informations représentée sous une forme qui permet un passage aisé vers les enregistrements informatiques.

##### III.1.2.1- Les niveaux

L’informatique consiste à mettre à disposition de l’utilisateur des moyens ou des outils de gestion informatique. Avant de spécifier les moyens informatiques, il est nécessaire de définir le travail de ce ou de ces utilisateurs finaux, de définir l’organisation, l’analyse des objectifs et des fonctions majeures de l’entreprise doit être menée. Ainsi, l’informatisation est conçue en fonction de l’organisation et l’organisation en fonction des objectifs à atteindre. Or l’enchainement de l’informatique, de l’organisation et de la fonction nécessite un découpage en niveaux de la démarche d’informatisation.

Ces niveaux sont :

* **Le niveau conceptuel**

C’est le niveau le plus invariant, il correspond aux finalités de l’entreprise. Il s’agit de décrire le « Quoi » en faisant abstraction des contraintes d’organisation et techniques. Les modèles utilisés pour la description conceptuelle du système d’information sont :

* le **Modèle Conceptuel de Communication (MCC)** qui représente les échanges de flux de produits, d’énergie, de personne, de valeur ou d’information entre système.
* le **Modèle Conceptuel de Données (MCD)** qui est une description des données et des relations et est réalisé à l’aide des trois concepts du formalisme entité-association : entité (ou objet) – relation – propriétés.
* le **Modèle Conceptuel des Traitements (MCT)** qui est la description de la partie dynamique du système d’information et est réalisé à l’aide des concepts suivants : événements/résultat – synchronisation – opération - processus.
* **Le niveau organisationnel**

Les choix d’organisation sont pris en compte à ce niveau comme la répartition des traitements entre l’homme et la machine, le mode de fonctionnement (temps réel ou différé) et l’affectation des données et des traitements. Ce niveau répond aux questions « Qui fait Quoi et Où ».

Trois modèles sont associés à ce niveau :

* le **Modèle Organisationnel de Communication (MOC)** qui représente les échanges qui ont lieu entre les sites de traitements et de données. Il ne concerne que les communications entre sites. Il n’existe pas s’il n’existe qu’un site ;
* le **Modèle Organisationnel de Données (MOD)** qui reprend le formalisme utilisé dans le MCD et qui représente aussi l’ensemble des données par type de site organisationnel ;
* le **Modèle Organisationnel des Traitements (MOT)** qui représente par procédures les phases et les tâches exécutées par chaque poste de travail.
* **Le niveau logique**

Ce niveau décrit le système d’information au plan informatique sans choix de matériel ou de logiciel précis. Nous avons trois modèles reliés à ce niveau :

* le **Modèle Logique de Communication (MLC)** qui est une représentation des messages échangés entre site et base de données. Il provient du MLD et de l’utilisation des outils en temps différé.
* le **Modèle Logique des Données (MLD)** qui est une représentation des données décrit dans le MCD en tenant compte du type de base de données.
* le **Modèle Logique des Traitements (MLT)** qui permet de décrire la conception technique qui traite principalement de la structuration en unités de traitement de type temps réel ou de type temps différé.
* **Le niveau physique**

Ce niveau décrit le résultat de la méthode ou l’informatisation finale. Il dépend des logiciels de développement nécessaires à la programmation et à la manipulation des données. La méthode laisse place aux normes du réel.

Nous avons encore trois modèles liés :

* **le Modèle Physique des Communication (MPC)** qui comprend la télématique entre sites informatiques ;
* **le Modèle Physique des Données (MPD)** qui n’est qu’un modèle de la base de données;
* **le Modèle Physique des Traitements (MPT)** qui comprend les programmes techniques et leur environnement d’exploitation, moniteur temps réel, traitement par lot, temps partagé.

##### I.1.2.2- La démarche

La méthode MERISE propose une démarche de construction de système d’information en 6 étapes:

* **le schéma directeur** : définition de la politique de l’entreprise ;
* **l’étude préalable** : analyse de l’existant et étude des différentes solutions possible puis choix de la solution appropriée ;
* **l’étude détaillée** : étude approfondie de la solution choisie par la direction ;
* **l’étude technique** : spécifications techniques complètes, seuls les informaticiens interviennent ;
* **la réalisation** : écriture des programmes, tests, essais, formation utilisateur ;
* **la maintenance** : suivi et évolution.

Cette démarche s’appuie sur un cycle de vie en cascade qui date de 1970 et est l’œuvre de Royce. Il s’agit d’un mode linéaire où toute étape est censée n’avoir de rétroaction que sur l’étape qui la précède.

L’activité d’une étape se réalise avec les résultats fournis par l’étape précédente ; ainsi, chaque étape sert de contrôle du travail effectué lors de l’étape précédente. L’utilisateur attend le déroulement complet du cycle de vie du logiciel pour vérifier, lors de la dernière étape, l’adéquation entre ses exigences et le produit livré.

Schéma Directeur

Etude Détaillé

Etude Préalable

Etude Technique

Etude Réalisation

Maintenance

**Cycle de vie**

Figure 2: la démarche Merise

### III.2- Comparaison processus unifié et Merise

|  |  |  |
| --- | --- | --- |
| CRITERES | MERISE | PROCESSUS UNIFIE |
| Données et traitements | Séparation des données et des traitements. | Regroupement des données et méthodes au sein des classes. Application du principe de l’encapsulation. |
| Niveaux | Plusieurs niveaux : conceptuel, logique, organisationnel, physique avec plusieurs types de modèles : données, traitements, communication. Existence de règles de passage entre les différents niveaux. | Niveau unique mais plusieurs types de modèles en fonction de l’aspect à décrire. Affinement des modèles lors des différentes étapes de l’analyse et de la conception. Continuité entre les différentes phases d'élaboration de l'application (traçabilité). |
| Gestion de projet | Concentré principalement sur la conception, sans prise en compte de la phase de programmation (modèle de déploiement et de composant absent) | Prise en compte de tous les stades de la conduite d'un projet.  Cohérence de la phase d’analyse jusqu’au code du programme. |

Tableau 4: Etude comparative entre la méthode MERISE et Processus unifié

### III.3- Choix de la méthode

L’analyse du tableau ci-dessus permet de faire ressortir plusieurs points. Mais l’élément majeur est que, MERISE offre une démarche d’analyse cohérente et rigoureuse, et mieux orientée vers les BD relationnelles. Elle est généralement utilisée dans l’informatique de gestion. Alors que le processus unifié utilisant UML qui est basé sur l’approche objet, est très souvent utilisé dans l’informatique technique (temps réel) et dans les projets évolutifs c’est à dire pouvant avoir différentes versions, formalise les interactions entre le système et l’utilisateur. Vu les besoin exprimé dans le cahier de charge, celui de favoriser l’ergonomie, la simplicité d’utilisation, la livraison de l’application en plusieurs version et vu l’interaction utilisateur/système, nous choisissons le processus unifié comme méthode de conception après avoir modélisé notre problème avec le langage UML.

* **Choix de la variante du PU**

Pour ce projet nous avons choisirons le Two Track Unified Process (2TUP) car il offre les possibilités suivantes :

* Itératif ;
* Offre des privilèges aux interactions utilisateurs ;
* Prise en compte les contraintes liées aux changements des SI
* Possibilité de décomposer le problème suivant un axe fonctionnel et un axe technique.
* Adapter à la représentions systèmes sous forme de module ;

Voici décrit schématiquement le processus 2TUP.



Figure 3 : Le processus 2TUP

Dans la suite nous nous servirons de cette méthode pour faire la modélisation métier de notre système.

## CHAPITRE IV- MODELISATION METIER DU SYSTEME

Dans ce chapitre nous présenterons les différents modèles du système formalisé à partir des diagrammes d’UML. Il s’agira particulièrement du diagramme contexte statique, le diagramme de cas d’utilisation, du diagramme de séquence.

### IV.1- Capture des besoins

Les besoins sont essentiellement des besoins fonctionnels et des besoins techniques.

#### II.1.1- Capture des besoins fonctionnels

Les besoins fonctionnelles sont les fonctions du système ainsi que l’interaction avec les utilisateurs.

II.1.1.1- Etude préliminaire

* **Identification des acteurs**

Un acteur est l’idéalisation d’un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système.

Les utilisateurs directs de cette application sont les agents du CMI à savoir les :

* Agents d’accueil : c’est l’initiateur d’un dossier, il accueille un patient l’inscrit et le dirige vers un infirmier ou un médecin.
* Agents infirmiers : Agent charger de prendre les constantes d’un patient et de proposer et de lui administrer le premier soin.
* Médecins : c’est l’agent chargé d’examiner un patient, de diagnostiquer une pathologie et de prescrire des ordonnances.
* Pharmaciens : c’est agent chargé de la gestion du stock pharmaceutique.
* Archivistes : c’est l’agent chargé de la gestion des dossiers médicaux.
* Archivistes : c’est l’agent chargé du paramétrage du système.
* **Identification des messages**

Dans cette partie, nous allons citer les différents messages échangés entre le système et l’extérieur mais d’abord il serait profitable de définir un message. En effet, un message représente la spécification d’une communication unidirectionnelle entre les objets qui transporte l’information avec l’intention de déclencher une activité chez le récepteur.

Comme **messages émis** par le système, nous avons :

* La liste des rendez-vous
* Les listes des patients en attente dans la corbeille d’un archiviste, un infirmier, un médecin ou un spécialiste ;
* La liste des ayants droit ;
* Les informations sur un ayant droit ou un patient (nom, prénom, photo….);
* L’Historiques des consultations ;
* L’Historiques des arrêts externes ;

Concernant les messages que reçoit le système, nous pouvons citer :

* L’ajout, la modification d’un patient ;
* L’ajout, la modification d’un ayant droit ;
* L’enregistrer d’un acte de décès ;
* Assigner une carte d’assurance
* Sortir un dossier physique
* Ranger un dossier
* Saisir les constantes et actes
* Saisir les actes
* Saisir un accident de travail
* Ajouter un arrêt de travail
* **Diagramme de contexte statistique**

Ce diagramme permet de définir notre système et son interaction avec les objets extérieurs.

**SYSTEME**

Administrateur

Utilisateur

**Paramétrer**

**Enregistrer, modifier, , imprimer**

Figure 4: Diagramme de contexte statique de l’application de suivi des cours

II.1.1.2- Construction du diagramme de cas d’utilisation

Ce diagramme permet de recueillir, d’analyser et d’organiser les besoins, et de recenser les grandes fonctionnalités du système. Nous allons organiser et analyser ces différentes fonctionnalités selon les modules suivant :

Le module accueil, le module archiviste, le module infirmier, le module médecin, le module pharmacie, le module administration.

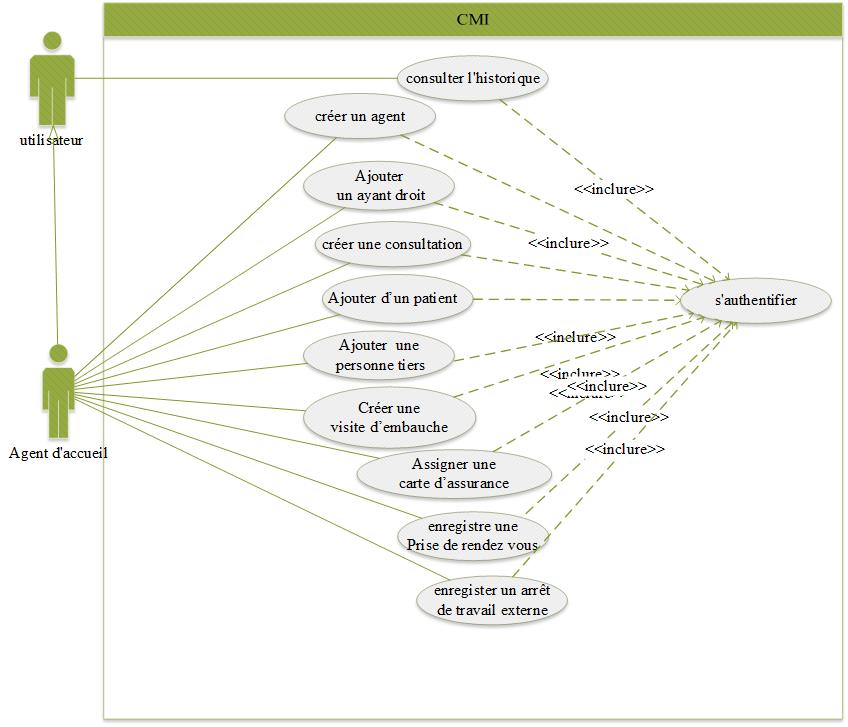


Figure 5 : diagramme de cas d’utilisation de l’agent d’accueil

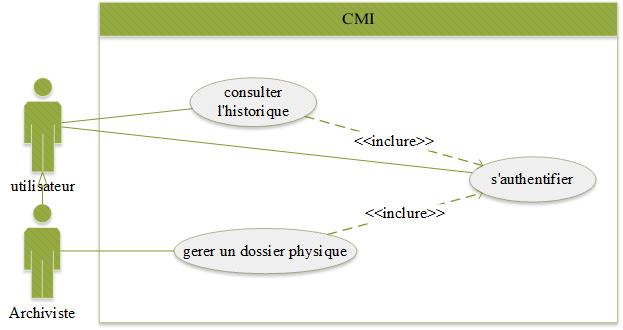


Figure 6 : diagramme de cas d’utilisation de l’archiviste

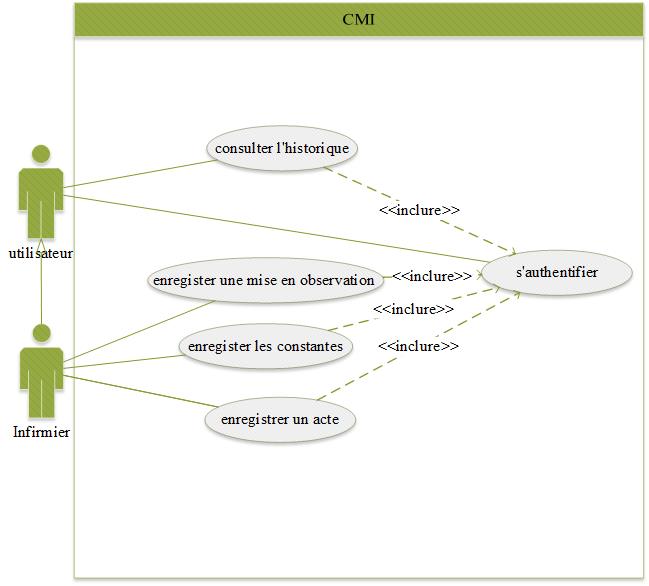


Figure 7 : diagramme de cas d’utilisation infirmier

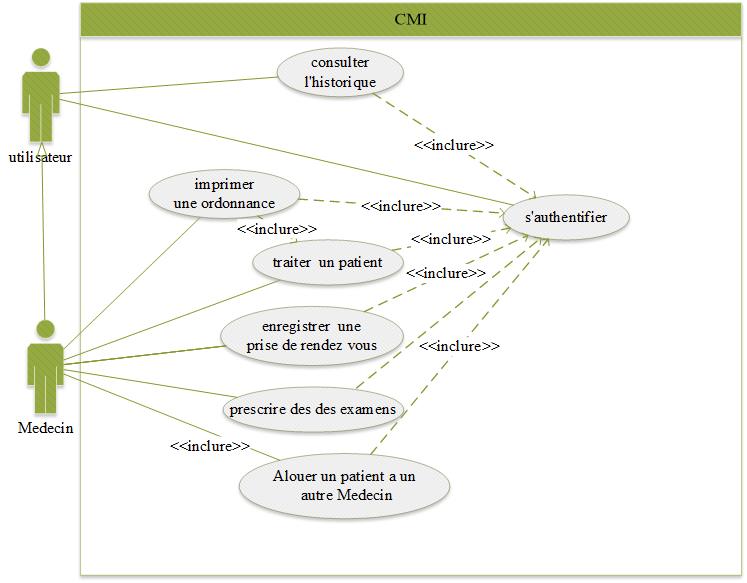


Figure 8 : Diagramme de cas d’utilisation d’un médecin

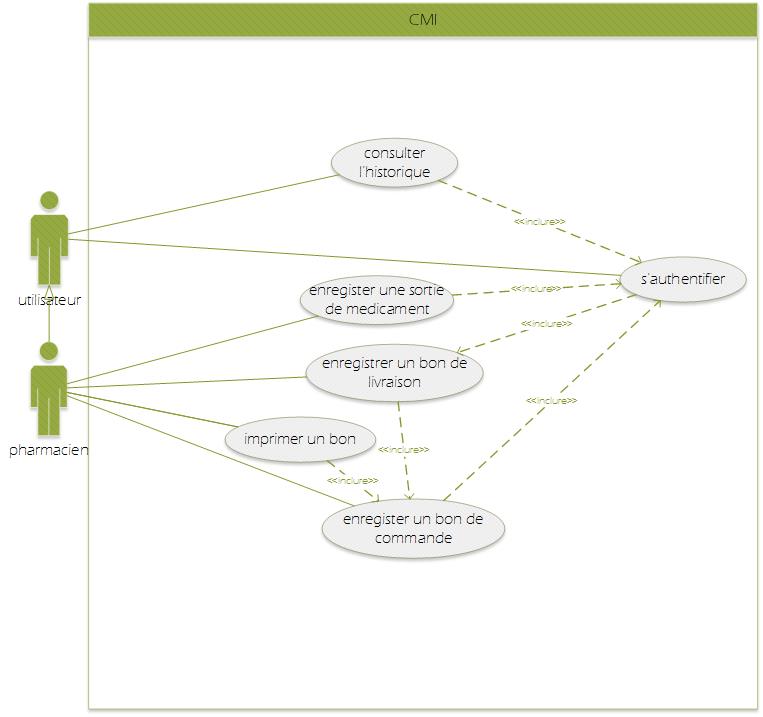
**

Figure 9: Diagramme de cas d’utilisation d’un pharmacien

II.1.1.3- Description textuelle des cas d’utilisation

* **LES CAS D’UTTILISATION DE L’ACCUEIL**

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | S’authentifier | | |
| ACTEURS PRINCIPAUX | utilisateur | | |
| PRECONDITIONS | Le système est fonctionnel | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | L’utilisateur soumet ses identifiants de connexion ; | |
|  | Le système vérifie l’authenticité du couple login password ; | |
|  | Le système génère es patients les patients dans la corbeille de l’utilisateur ; | |
|  | Le système affiche la corbeille sur la page d’accueil ; | | Le système affiche la corbeille sur la page d’accueil ; |
| SENARIO ALTERNATIF | | |
| 2. le couple fourni  est erroné | | Le système redemande le login et le mot de passe |
| POSTCONDITTION | L’utilisateur accède à la page d’accueil du système | | |

Tableau 5 : description textuelle de l’authentification

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | consulter l’historique des patients | | |
| ACTEURS PRINCIPAUX | utilisateur | | |
| PRECONDITIONS | L’authentification est un succès | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | L’utilisateur sélectionne l’onglet pour voir l’historique ; | |
|  | Le système génère les patients et affiche | |
|  | Le système affiche l’historique de consultation | |
| SENARIO ALTERNATIF | | |
| 2. aucun patient n’est encore enregistrer | | Le système affiche le message l’historique des patients est vide |
| POSTCONDITTION | L’utilisateur accède à l’’ensemble des patients | | |

Tableau 6: description textuelle de consulter l’historique

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Ajouter un patient | | |
| ACTEURS PRINCIPAUX | Agent d’accueil | | |
| PRECONDITIONS | L’authentification est un succès et le patient n’est pas enregistrer | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | L’agent d’accueil soumet le formulaire de création d’un patient au système ; | |
|  | Le système enregistre les données du patient | |
|  | Le système affiche un message de succès | |
| SENARIO ALTERNATIF | | |
| 2. l’enregistrement échoue | | Le système affiche un message le cas d’utilisation reprend |
| POSTCONDITTION | Le système contient les données du patient | | |

Tableau 7: description textuelle de ajouter un patient

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Créer une consultation | | |
| ACTEURS PRINCIPAUX | Agent d’accueil | | |
| PRECONDITIONS | L’authentification est un succès et le patient est déjà enregistrer | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | L’agent d’accueil soumet le formulaire de création d’une consultation; | |
|  | Le système enregistre les données du patient | |
|  | Le système affiche un message de succès | |
|  | Le dossier du patient devient visible dans la corbeille de l’archiviste | |
| SENARIO ALTERNATIF | | |
| 2. l’enregistrement échoue | | Le système affiche un message d’erreur le cas d’utilisation reprend |
| POSTCONDITTION | Le système contient un dossier de consultation lié à un patient | | |

Tableau 8: description textuelle de créer une consultation

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Créer un arrêt de travail externe | | |
| ACTEURS PRINCIPAUX | Agent d’accueil | | |
| PRECONDITIONS | L’authentification est un succès et le patient est déjà enregistrer | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | L’agent d’accueil soumet le formulaire de création d’un arrêt de travail externe | |
|  | Le système enregistre les données du patient | |
|  | Le système affiche un message de succès | |
|  | Le dossier du patient devient visible dans la corbeille de l’archiviste | |
| SENARIO ALTERNATIF | | |
| 2. l’enregistrement échoue | | Le système affiche un message d’erreur le cas d’utilisation reprend |
| POSTCONDITTION | Le système contient un dossier d’arrêt externe lié à un patient | | |

Tableau 9: description textuelle de créer un arrêt de travail externe

* LES CAS D‘UTILISATION DE L’ARCHIVISTE

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | gérer un dossier un dossier physique | | |
| ACTEURS PRINCIPAUX | Archiviste | | |
| PRECONDITIONS | L’authentification est un succès et il existe un patient dans la corbeille de l’archiviste | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | L’archiviste sélectionne le patient concerné | |
|  | Le système affiche les choix (créer, ranger, faire sortir un dossier physique) | |
|  | L’archiviste choisit une option et valider | |
|  | Le système affiche un message de succès | |
|  | Le patient devient visible dans la corbeille de l’infirmier et disparait de celui de l’archiviste | |
| SENARIO ALTERNATIF | | |
| 3. la validation échoue | | Le système affiche un message d’erreur le cas d’utilisation reprend |
| POSTCONDITTION | Le système reconnait la sortie du dossier et le dossier du patient apparait dans la corbeille de l’infirmier | | |

Tableau 10: description textuelle de gérer un dossier physique

* LES CAS D’UTILISATION DE L’INFIRMIER

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Enregistrer les constantes | | |
| ACTEURS PRINCIPAUX | Infirmier | | |
| PRECONDITIONS | L’authentification est un succès, il existe le dossier d’un patient dans la corbeille de l’infirmier | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | L’infirmier sélectionne le dossier du patient concerné | |
|  | Le système génère l'interface de saisie des informations | |
|  | L’infirmier saisie les champs du formulaire et valide | |
|  | Le système enregistre les informations renseignées | |
|  | Le système affiche un message de succès | |
|  | Le patient disparait de la corbeille de l’infirmier | |
| SENARIO ALTERNATIF | | |
| 3. l’enregistrement échoue | | Le système affiche un message d’erreur : le cas d’utilisation reprend |
| POSTCONDITTION | Le dossier du patient disparait de la corbeille de l’infirmier et apparait dans celle d’un médecin | | |

Tableau 11: description textuelle de enregistrer des constantes

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Enregistrer les Actes | | |
| ACTEURS PRINCIPAUX | Infirmier | | |
| PRECONDITIONS | L’authentification est un succès, il existe le dossier d’un patient dans la corbeille de l’infirmier | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | L’infirmier sélectionne le dossier du patient concerné | |
|  | Le système génère l'interface de saisie des informations | |
|  | L’infirmier saisie les champs du formulaire et valide | |
|  | Le système enregistre les informations renseignées | |
|  | Le système affiche un message de succès | |
|  | Le patient disparait de la corbeille de l’infirmier | |
| SENARIO ALTERNATIF | | |
| 2. l’enregistrement échoue | | Le système affiche un message d’erreur : le cas d’utilisation reprend |
| POSTCONDITTION | Le dossier du patient disparait de la corbeille de l’infirmier et apparait dans celle d’un médecin | | |

Tableau 12: description textuelle de enregistrer des constantes

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Mettre en observation | | |
| ACTEURS PRINCIPAUX | Infirmier | | |
| PRECONDITIONS | L’authentification est un succès, il existe le dossier d’un patient dans la corbeille de l’infirmier | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | L’infirmier sélectionne le dossier du patient concerné | |
|  | Le système génère l'interface de saisie des informations | |
|  | L’infirmier saisie les champs du formulaire et valide | |
|  | Le système enregistre les informations renseignées | |
|  | Le système affiche un message de succès | |
|  | Le patient disparait de la corbeille de l’infirmier | |
| SENARIO ALTERNATIF | | |
| 3. l’enregistrement échoue | | Le système affiche un message d’erreur : le cas d’utilisation reprend |
| POSTCONDITTION | Le dossier du patient disparait de la corbeille de l’infirmier et apparait dans celle d’un médecin | | |

Tableau 13: description textuelle de mettre en observation

* **Les cas d’utilisation du module médecin**

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Traiter un patient | | |
| ACTEURS PRINCIPAUX | Médecin | | |
| PRECONDITIONS | L’authentification est un succès, il existe le dossier d’un patient dans la corbeille du médecin | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | Le médecin sélectionne le dossier du patient concerné | |
|  | Il choisit un diagnostic | |
|  | le système génère les pathologies liées aux diagnostics | |
|  | Il choisit les pathologies | |
|  | le système génère les pathologies liées aux diagnostics | |
|  | Le médecin prescrit des médicaments | |
|  | générer l'interface de saisie des médicaments | |
|  | Le médecin renseigne la posologie et le nombre de jour d’utilisation du médicament | |
|  | Le médecin valide et enregistrer | |
|  | Le patient disparait de la corbeille de l’infirmier | |
| SENARIO ALTERNATIF | | |
| 6. le médicament n’existe pas dans pas dans le stock de pharmacie | | Changer et prendre un autre médicament de même famille  Le cas d’utilisation reprend 7 du scénario nominal. |
| POSTCONDITTION | Le dossier du patient disparait de la corbeille du médecin et apparait dans celle du pharmacien | | |

Tableau 14: description textuelle de traiter un patient

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Traiter un patient | | |
| ACTEURS PRINCIPAUX | Médecin | | |
| PRECONDITIONS | L’authentification est un succès, il existe le dossier d’un patient dans la corbeille du médecin | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | Le médecin sélectionne le dossier du patient concerné | |
|  | Il choisit un diagnostic | |
|  | Il choisit les pathologies | |
|  | Le médecin saisit posologie | |
|  | Le médecin valide et enregistrer | |
| SENARIO ALTERNATIF | | |
| 4. le médicament n’existe pas dans pas dans le stock de pharmacie | | Changer et prendre un autre médicament de même famille  Le cas d’utilisation reprend 5 du scénario nominal. |
| POSTCONDITTION | Le dossier du patient disparait de la corbeille du médecin et apparait dans celle du pharmacien | | |

Tableau 15: description textuelle de enregistrer des constantes

* LES CAS DE LA PHARMACIE

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Enregistrer une sortie de médicament | | |
| ACTEURS PRINCIPAUX | pharmacien | | |
| PRECONDITIONS | L’authentification est un succès, il existe le dossier d’un patient dans la corbeille du pharmacien | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | Le pharmacien sélectionne le dossier du patient concerné | |
|  | Il choisit de faire sortir un médicament et valide | |
|  | Le système enregistre la sortie du médicament | |
|  | Le Système envoie un message de succès | |
|  | Le patient disparait de la corbeille du pharmacien | |
| SENARIO ALTERNATIF | | |
| 3. le médicament n’existe pas dans pas dans le stock de pharmacie | | Le système envoie un message d’erreur. Le cas d’utilisation reprend. |
| POSTCONDITTION | Le dossier du patient disparait de la corbeille du pharmacien et apparait dans celle du l’archiviste, le stock diminue d’un médicament. | | |

Tableau 16: description textuelle de enregistrer une sortie de médicament

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Enregistrer un bon de commande | | |
| ACTEURS PRINCIPAUX | pharmacien | | |
| PRECONDITIONS | L’authentification est un succès | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | Le pharmacien renseigne les entêtes du bon (numéro de bon, fournisseur) | |
|  | Le pharmacien ajoute un médicament (Dénomination, quantité, ...) | |
|  | Le pharmacien valide la commande | |
|  | Le système enregistre le bon de commande | |
|  | Le Système envoie un message de succès | |
| SENARIO ALTERNATIF | | |
| 4. l’enregistrement échoue | | Le système envoie un message d’erreur. Le cas d’utilisation reprend au point du scénario nominal. |
| POSTCONDITTION | Le système contient un nouveau bon de commande. | | |

Tableau 17: description textuelle de enregistrer un bon de commande

|  |  |  |  |
| --- | --- | --- | --- |
| NOM | Enregistrer un bon de livraison | | |
| ACTEURS PRINCIPAUX | pharmacien | | |
| PRECONDITIONS | L’authentification est un succès et un bon de commande a été enregistrer | | |
| SCENARII |  | SCENARIO NOMINAL | |
|  | Le pharmacien renseigne les entêtes numéro de bon, fournisseur | |
|  | Le système génère le bon de commande | |
|  | Le système afficher les informations de la commande | |
|  | Le pharmacien renseigne le formulaire de livraison et valider | |
|  | Le système enregistre le bon de livraison | |
|  | Le Système envoie un message de succès | |
| SENARIO ALTERNATIF | | |
| 4. l’enregistrement échoue | | Le système envoie un message d’erreur. Le cas d’utilisation reprend. |
| POSTCONDITTION | Le système contient un nouveau bon de livraison et le stock de médicament a augmenté | | |

Tableau 18: description textuelle de enregistrer un bon de commande

#### II.1.2- Capture des besoins techniques

Les besoins techniques sont nécessairement les équipements réseaux, des ordinateurs pour les utilisateurs, un serveur web, un serveur de base de données.

### IV.2- Analyse

Dans **la phase d’analyse**, nous cherchons d’abord à bien comprendre et à décrire de façon précise les besoins des utilisateurs ou des clients, les fonctionnalités souhaitées et comment fonctionne les actions sur le système.

#### II-2.1- Développement du modèle statique

Dans cette partie nous présenterons le diagramme de dialogue

* **Diagramme de classe**

Le diagramme de classes montre la structure interne du système. Il permet de fournir une représentation abstraite des objets du système qui vont interagir pour réaliser les cas d'utilisation.

Nous allons regrouper ses tables en trois selon qu’elles sont liées à la gestion de la personne, à la gestion de la consultation et à la gestion de la pharmacie.



Figure 10 : Diagramme de dialogue de la gestion des personnes



Figure 11 : Diagramme de dialogue de la gestion d’une consultation



Figure 12 : Diagramme de dialogue de la gestion de la pharmacie

#### II.2.2- Développement du modèle dynamique

Il consiste en la représentation des aspects dynamique du système. De ce fait nous élaborons dans cette partie, les diagrammes de séquence, d’activité et d’état transition.

* Diagramme de séquence

Les principales informations contenues dans un diagramme de séquence sont les messages échangés entre les lignes de vie, présentés dans un ordre chronologique. Ainsi, contrairement au diagramme de communication, le temps y est représenté explicitement par une dimension (la dimension verticale) et s’écoule de haut en bas.

* Diagramme de séquence des cas d’utilisation de tous utilisateurs de système

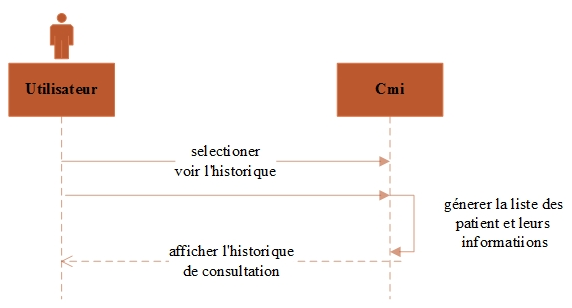


Figure 13 : Diagramme de séquence de s’authentifier

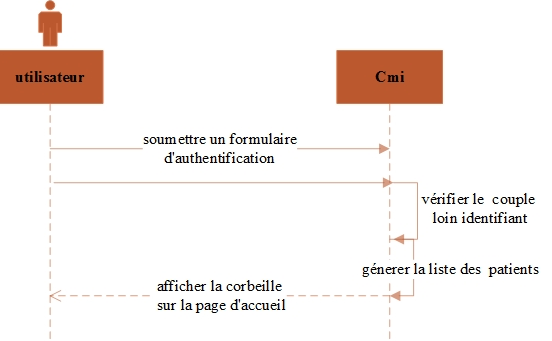


Figure 14 : Diagramme de séquence de s’authentifier

* Diagramme de séquence des cas d’utilisation du module Accueil

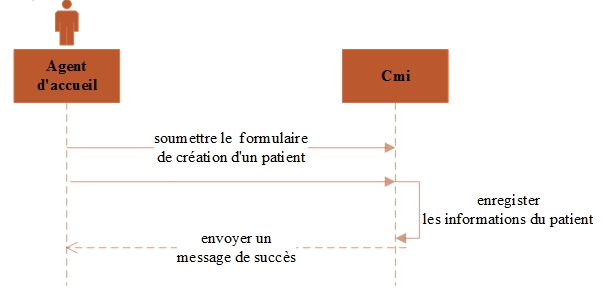


Figure 15 : Diagramme de séquence de créer un patient

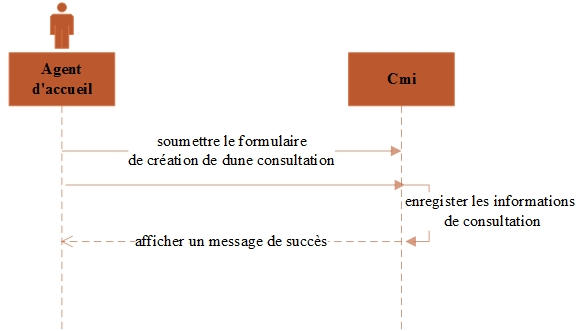


Figure 16 : Diagramme de séquence de créer une consultation

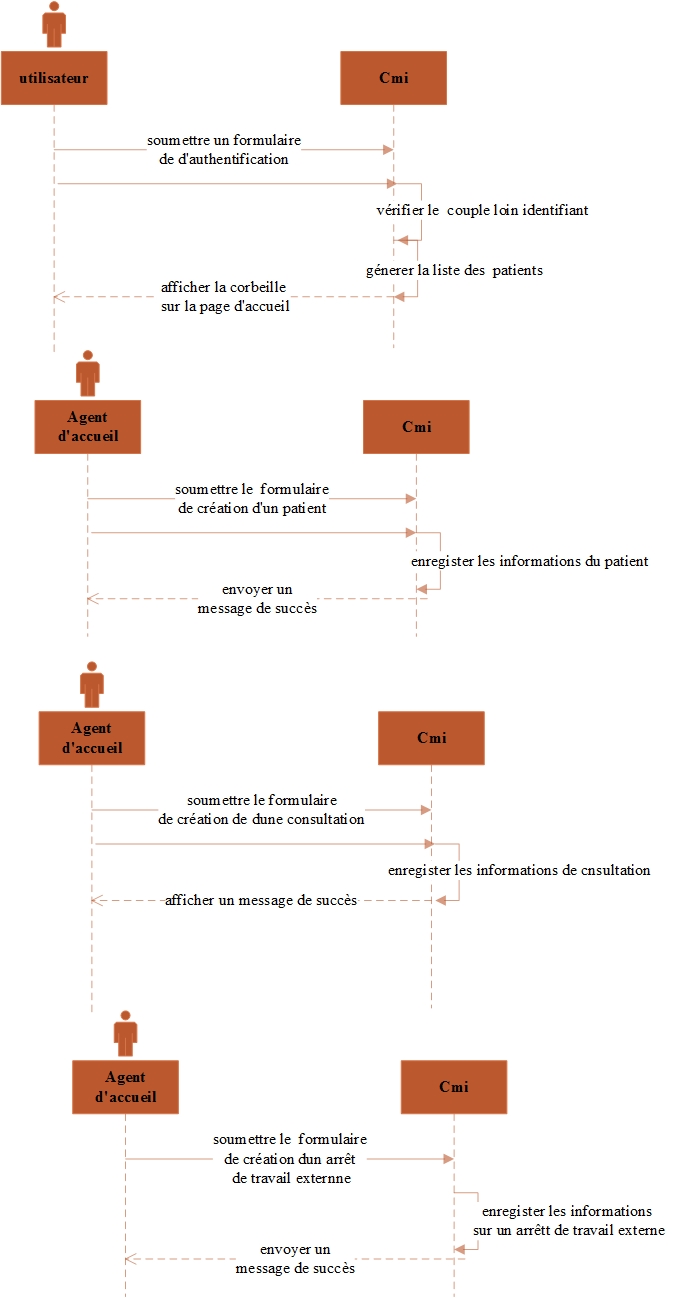


Figure 17 : Diagramme de séquence de créer un arrêt de travail externe

* Diagramme de séquence des cas d’utilisation du module Archiviste

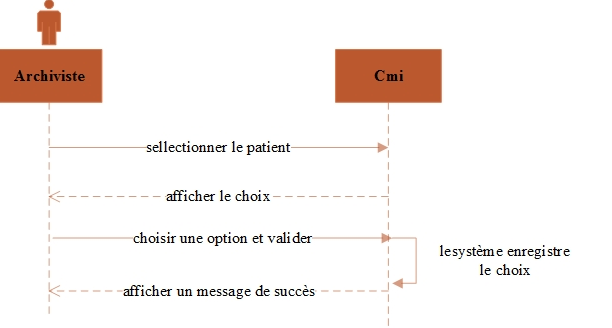


Figure 18 : Diagramme de séquence de gérer un dossier physique

* Diagramme de séquence des cas d’utilisation du module Infirmier

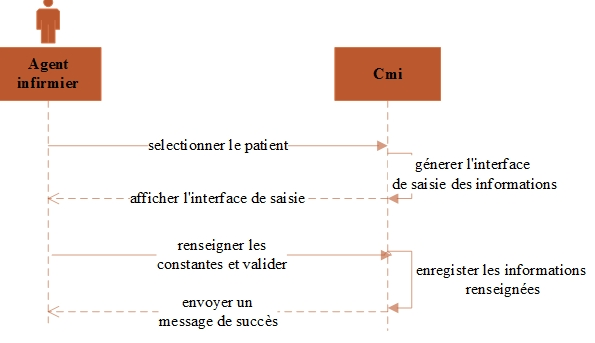


Figure 19 : Diagramme de séquence les constantes

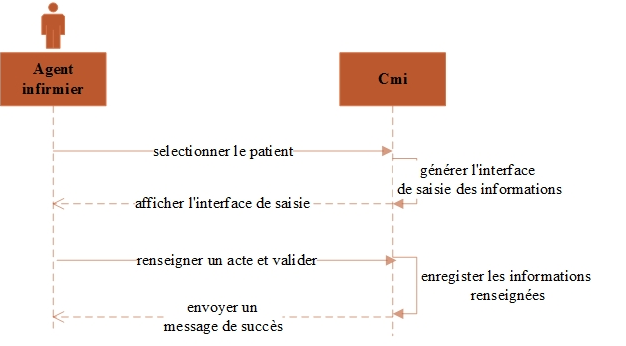


Figure 20 : Diagramme de séquence de enregistrer un actes

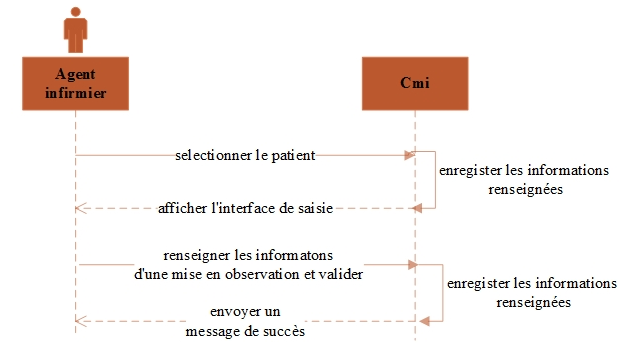


Figure 21 : Diagramme de séquence de enregistrer une mise en observation

* Diagramme de séquence des cas d’utilisation du module Médecin

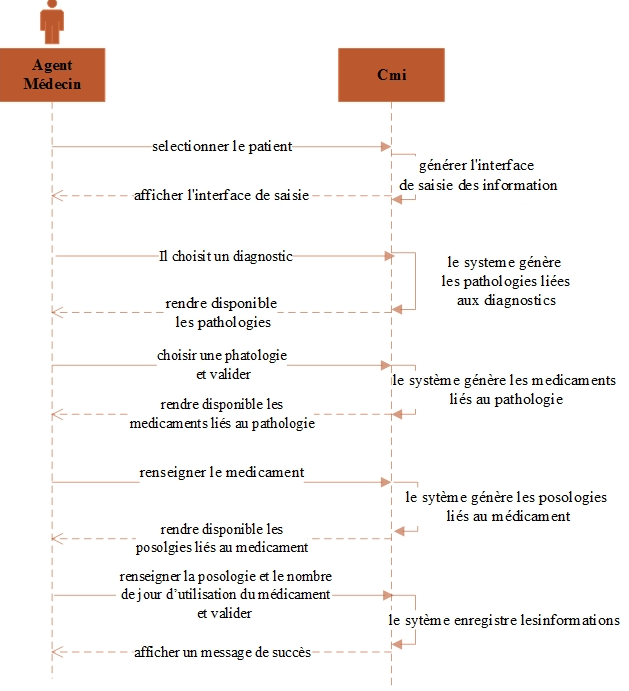


Figure 22 : Diagramme de séquence de enregistrer une mise en observation

* Diagramme de séquence des cas d’utilisation du module Pharmacie

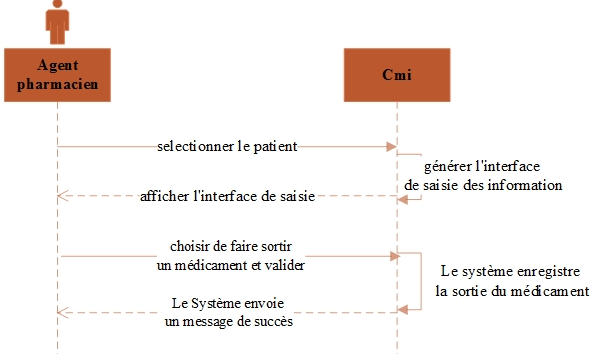


Figure 23 : Diagramme de séquence de enregistrer une sortie de médicament

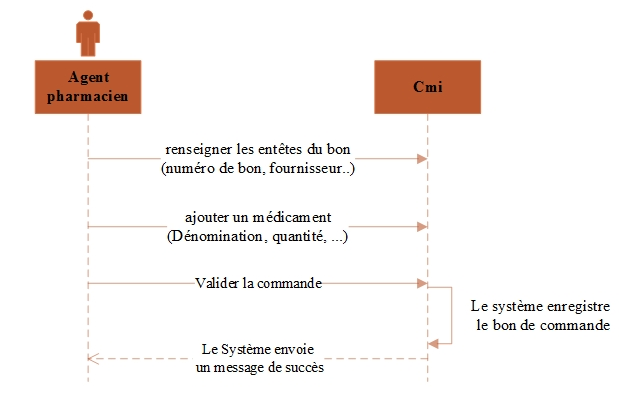


Figure 24 : Diagramme de séquence de enregistrer un bon de commande

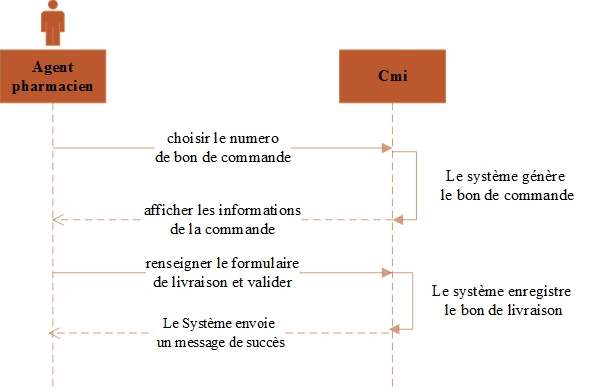


Figure 25 : Diagramme de séquence de enregistrer un bon de livraison

### IV.3- Conception préliminaire et détaillée

Dans décrirons les différents couche de notre solution ainsi que les éléments de son architecture de déploiement.

#### IV.3.1- Diagramme de composant

Les diagrammes de composants est un diagramme de vues statiques en UML. Ils décrit le système modélisé sous forme de composants réutilisables et mettent en évidence leurs relations de dépendance.

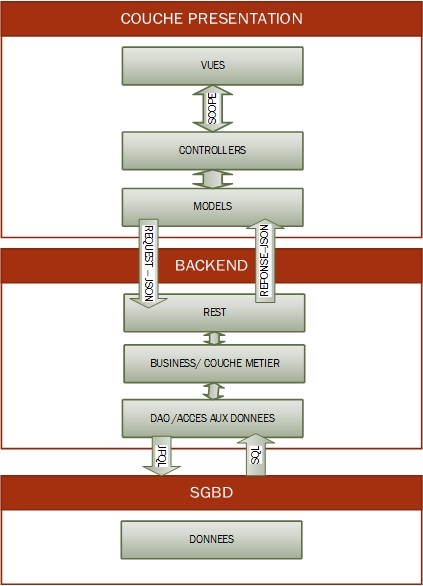


Figure 26: Diagramme de composant

#### IV.3.2- Diagramme de déploiement

Les diagrammes de déploiement tout comme le diagramme de composant est un diagramme de vues statiques en UML. Mais celui-ci se rapprochent encore plus de la réalité physique, puisqu’ils identifient les éléments matériels (PC, Modem, Station de travail, Serveur, etc.), leur disposition physique (connexions) et la disposition des exécutables (représentés par des composants) sur ces éléments matériels.

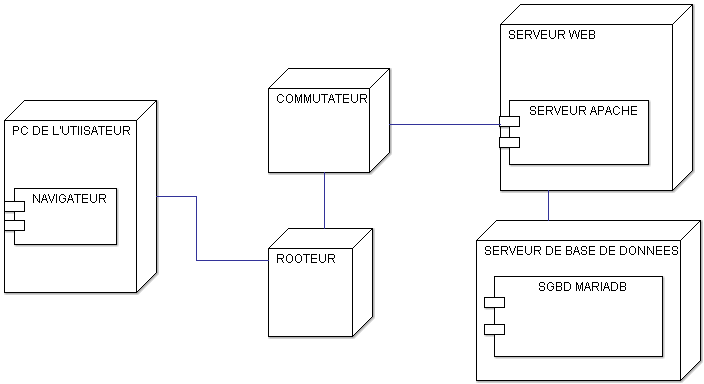


Figure 27: Diagramme de déploiement

#### IV.3.3- Diagramme de classe conception

Nous allons regrouper ses tables en trois selon qu’elles sont liées à la gestion de la personne, à la gestion de la consultation et à la gestion de la pharmacie.

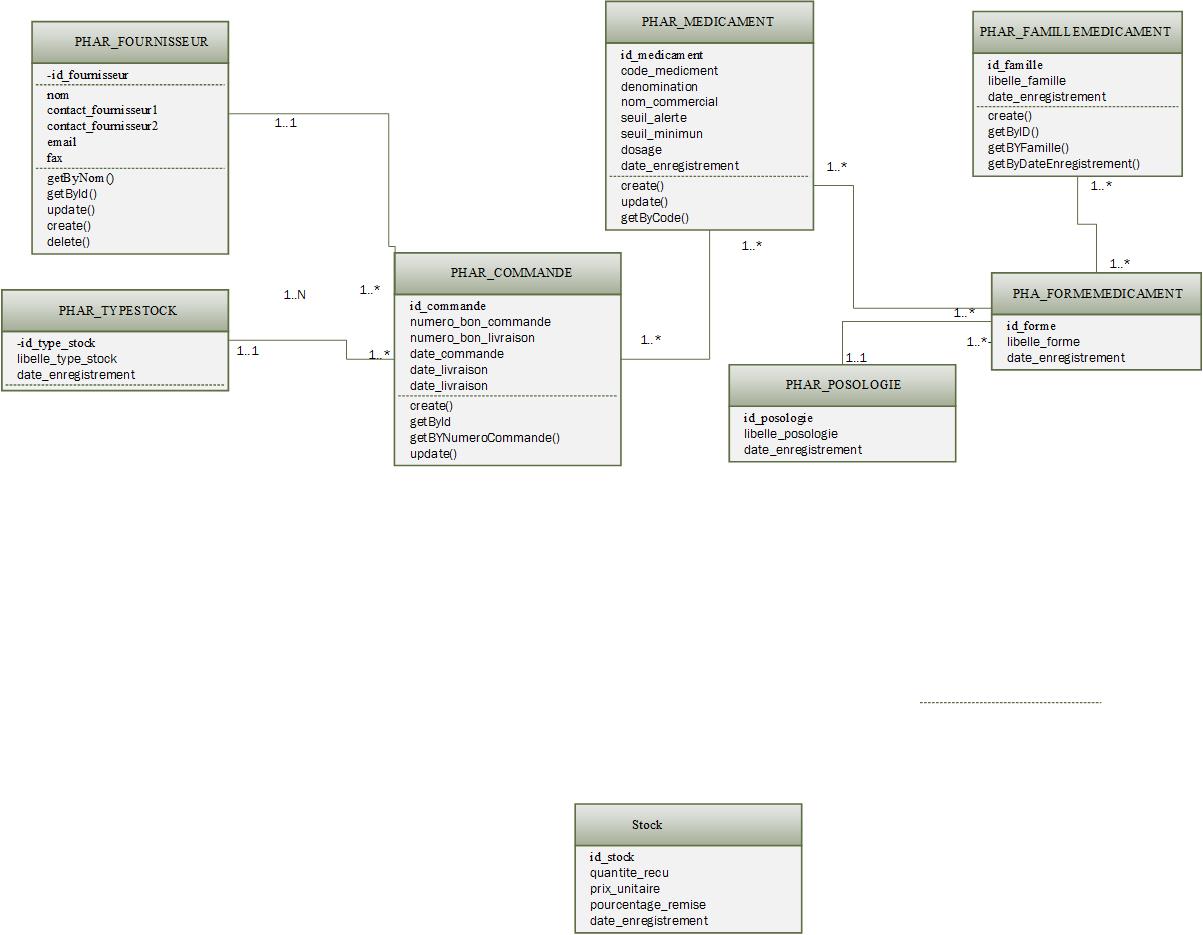


Figure 28 : Diagramme de classe conception

Dans cette deuxième partie nous avons modélisé les objets et les grandes fonctions du système à travers les diagrammes d’UML. Nous nous inspirerons de ces modèles dans la phase de réalisations dans la partie suivantes.

PARTIE



Dans cette partie nous évaluerons notre projet en termes de et de temps et par la suite présenterons l’application.