

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE



UNIVERSITÉ ABDERRAHMANE MIRA DE BÉJAIA
FACULTÉ DES SCIENCES EXACTES

Mémoire de fin de cycle

*En vue de l'obtention du diplôme de Master
Professionnel en Informatique*

Option : Génie Logiciel

THEME

Conception et Réalisation d'une Application
Desktop de Gestion d'Un Cabinet Dentaire

Présenté par : M^{lle} Oulmi Malika

Encadrée par : Dr Bachiri Lina
Dr Zidani Ferroudja

Devant le jury composé de :

Présidente : Dr Ait Abdelouahab Karima

Examineur 1 : Dr Djebari Nabil

Examinatrice 2 : Dr Chibani Samia

Examinatrice 3 : Dr Chaaban Sarah

Promotion 2024 — 2025

TABLE DES MATIÈRES

Introduction	10
1 Technologies,Méthodes et Organisme d’Accueil	11
1.1 Introduction	11
1.2 Technologie des Applications Desktop	11
1.2.1 Définition simple d’une application desktop	11
1.2.2 Frameworks populaires pour le développement d’applications Desktop en Java	12
1.2.3 Définition de JAVA	12
1.2.3.1 Java Development Kit (JDK)	13
1.2.3.2 Java Virtual Machine (JVM)	14
1.2.3.3 Java Runtime Environment (JRE)	14
1.2.4 Caractéristiques de Java	15
1.2.5 Architecture d’une application desktop	16
1.3 Méthodes d’analyse et de conception	16
1.3.1 Définition du processus unifié	17
1.3.2 Caractéristiques du processus unifié	17
1.3.3 Cycle de vie du processus unifié	17
1.4 Langage de modélisation	19
1.4.1 Unified Modeling Language (UML)	19
1.4.2 Les diagrammes	19
1.5 Présentation de l’organisme d’accueil	20

1.5.1	Présentation de Cabinet Dentaire "Diamond Dental Centre" :	20
1.5.2	Les Services de Cabinet : [B1]	21
1.5.2.1	Implantologie Flapless	21
1.5.2.2	Pédodontie	21
1.5.2.3	Blanchement Dentaire	21
1.5.2.4	Couronne Dentaire et Facette Dentaire	22
1.5.2.5	Dentisterie Restauratrice	22
1.5.2.6	Orthodontie	22
1.5.2.7	Prothèses Amovibles	23
1.5.2.8	Hollywood Smile	23
1.5.2.9	Chirurgie Dentaire	24
1.6	Conclusion	24
Introduction		24
2 Cahier des Charges et Analyse des Besoins		25
2.1	Introduction	25
2.2	Problématique et objectif du projet	26
2.2.1	Problématique	26
2.2.2	Contexte	26
2.2.3	Objectif du projet	27
2.3	Cahier des Charges	28
2.3.1	Définition des Besoins Fonctionnels	28
2.3.2	Définition des Besoins non Fonctionnels	29
2.4	Analyse des Besoins	30
2.4.1	Identification des acteurs	30
2.4.2	Diagramme de Contexte	30
2.4.3	Identification des Cas d'Utilisation	31
2.4.4	Diagramme global de cas d'utilisations	32
2.4.5	Description des cas d'utilisation	33
2.4.6	Modélisation des diagrammes de séquence	38
2.4.6.1	Diagramme de séquence du cas d'utilisation « S'authentifier »	39

2.4.6.2	Diagramme de séquence du cas d'utilisation « Gestion des Patients »	40
2.4.6.3	Diagramme de séquence du cas d'utilisation «Gestion des Rendez-Vous»	41
2.4.6.4	Diagramme de séquence du cas d'utilisation «Gestion des Traitements»	42
2.4.6.5	Diagramme de séquence du cas d'utilisation«Ajouter des Prescriptions »	43
2.4.6.6	Diagramme de séquence du cas d'utilisation«Editer des ordonnances »	44
2.5	Conclusion	45
Introduction		45
3 Conception		46
3.1	Introduction	46
3.2	Réalisation du diagramme de classes	46
3.2.1	Présentation du diagramme de classes	46
3.2.2	Spécification des règles de gestion pour le cabinet dentaire	47
3.2.3	Le diagramme de classes	48
3.3	Le dictionnaire des données	49
3.4	Le modèle relationnel	51
3.4.1	Modèle relationnel pour notre cas	51
3.5	Conclusion	51
Introduction		51
4 Réalisation		52
4.1	Introduction	52
4.2	Langages et environnements de développement	52
4.2.1	Draw.io	52
4.2.2	NetBeans IDE 8.0.1	53
4.2.3	SQLite	53
4.2.4	SQL	53

4.2.5	Java 8	54
4.3	Présentation des interfaces de l'application	54
4.3.1	Interface de connexion	54
4.3.2	Interface menu principal.	55
4.3.3	Interface de gestion des patients	55
4.3.4	Interface de gestion des rendez-vous	56
4.3.5	Interface de gestion des traitements	56
4.3.6	Interface de gestion des prescriptions	57
4.4	Conclusion	57

LISTE DES TABLEAUX

2.1	Description des Acteurs	30
2.2	Cas d'utilisation associés aux acteurs	31
2.3	Description détaillée du cas d'utilisation « S'authentifier »	33
2.4	Description détaillée du cas d'utilisation « Gestion des Patients »	34
2.5	Description détaillée du cas d'utilisation « Gestion des Rendez-Vous»	35
2.6	Description détaillée du cas d'utilisation « Gestion des Traitements »	36
2.7	Description détaillée du cas d'utilisation « Ajouter des Prescriptionst»	37
2.8	Description détaillée du cas d'utilisation « Editer une Ordonnance»	38
3.1	Table utilisateur	49
3.2	Table patient	49
3.3	Table rendez_vous	49
3.4	Table ordonnance	50
3.5	Table prescription	50
3.6	Table traitement	50

TABLE DES FIGURES

1.1	JAVA.	13
1.2	Java Development Kit.	13
1.3	Java Runtime Environment.	14
1.4	Cycle de Vie de UP.	18
1.5	Implantologie Flapless.	21
1.6	Couronne Dentaire et Facette Dentaire.	22
1.7	Prothèse Amovible.	23
1.8	Hollywood Smile.	23
1.9	Chirurgies Dentaires.	24
2.1	Diagramme de Contexte	31
2.2	Diagramme global de cas d'utilisation	32
2.3	Diagramme de séquence du cas d'utilisation « S'authentifier »	39
2.4	Diagramme de séquence du cas d'utilisation « Gestion des Patients »	40
2.5	Diagramme de séquence du cas d'utilisation « Gestion des Rendez-Vous »	41
2.6	Diagramme de séquence du cas d'utilisation « Gestion des Traitements »	42
2.7	Diagramme de séquence du cas d'utilisation « Ajouter des Prescriptions ».	43
2.8	Diagramme de séquence du cas d'utilisation « éditer des ordonnances ».	44
3.1	Diagramme de classes.	48
4.1	Interface connexion utilisateur.	54
4.2	Interface du menu principal de l'application.	55
4.3	Interface de gestion des patients.	55

4.4	Interface de gestion des rendez-vous.	56
4.5	Interface de gestion des traitements.	56
4.6	Interface de gestion des prescriptions.	57

Remerciements

Je souhaite exprimer ma gratitude et mes dédicaces à tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail, car sans leur soutien, il n'aurait pas été possible.

En premier lieu, je rends grâce à Dieu tout-puissant pour m'avoir permis d'atteindre ce jour tant attendu et pour m'avoir accordé la force et la persévérance nécessaires pour mener à bien ce projet.

J'adresse mes plus sincères remerciements à mes parents, qui ont été une source constante de soutien, d'encouragement et de motivation. Leur amour inconditionnel et leur exemple de travail acharné sont une inspiration pour moi. Que Dieu les protège et les bénisse pour leur amour et leur dévouement.

Je tiens à remercier Mme BACHIRI LINA d'avoir accepté de me guider tout au long de ce travail, pour sa disponibilité, sa gentillesse et sa contribution à l'aboutissement de ce projet.

Je tiens à exprimer ma profonde gratitude à Mme Zidani Ferroudja, mon co-encadrant pour son accompagnement tout au long de ce travail, ses conseils avisés.

À tous les enseignants qui ont cultivé en moi l'amour de l'Informatique, qu'ils trouvent dans ce modeste travail un petit geste de reconnaissance qui ne sera jamais à la hauteur de la grandeur de leurs âmes.

Dédicace

*À la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma
vie et mon bonheur :*

Ma grand-mère OUIZA.

*À toi, prunelle de mes yeux, qui ne m'as jamais refusé un rêve, un souhait, une
volonté...*

Je suis fier d'être ta fille...

Papa Houcine.

*À mon oncle LARBI et ma tante LILA, qui m'ont soutenue tout au long de mon
travail.*

À mon cher frère SABER,

À mon adoré petit frère SAMY, qui m'a encouragée,

À ma sœur de cœur SABRINA.

*À mon cher ami, la source de mes efforts,
qui a toujours été un soutien précieux par ses conseils, son aide matérielle et
morale...*

À qui je souhaite plus de réussite :

AZZEDDINE.

À toutes les personnes qui ont partagé ma souffrance et ma joie,

Je vous dédie ce travail.

Introduction Générale

Actuellement, le monde connaît une avance technologique considérable dans tous les secteurs et cela grâce à l'informatique qui est une science qui étudie les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement de travail.

Avant l'invention de l'ordinateur, on enregistrait toutes les informations manuellement sur des supports en papier ce qui engendrait beaucoup de problèmes tel que la perte de temps considérable dans la recherche de ces informations ou la dégradation de ces dernières...etc.

Ainsi, jusqu'à présent, l'ordinateur reste le moyen le plus sûr pour le traitement et la sauvegarde de l'information. Cette invention a permis d'informatiser les systèmes de données des entreprises, ce qui est la partie essentielle dans leur développement aujourd'hui.

Les cliniques font partie intégrante des établissements que l'informatique pourra beaucoup aider. Jusqu'à ce jour, la manière de gérer manuellement est encore dominante, d'où la nécessité d'introduire l'informatique dans les gestions des cliniques.

Ce projet a pour objectif la conception et le développement d'une application desktop complète ,intuitive et sécurisée, destinée à faciliter la gestion quotidienne des professionnels de la dentisterie.L'application permet la gestion des patients et les rendez-vous, enregistrements des services et des traitements qui offre la clinique, ajout des prescriptions pour le bon suivi des patients et édition des ordonnances. Cette application offre un ensemble d'avantages :

- *Elle est hors ligne pas de dépendance à une connexion Internet.
- *Interface ergonomique :prise en main rapide même pour les non-techniciens.
- *Modulaire : Possibilité d'ajouter des extensions (comptabilité, imagerie dentaire, etc.).
- * Coût réduit : Alternative économique aux logiciels professionnels onéreux.

Le projet est organisé en 4 chapitres , on commence par les technologies,les méthodes et la présentation de l'organisme d'accueil dans le premier chapitre , ensuite on passe au cahier des charges et analyse des besoins dans le deuxième chapitre, après la conception dans le troisième chapitre ,enfin la réalisation et les outils utilisés dans le dernier chapitre.

CHAPITRE 1

TECHNOLOGIES, MÉTHODES ET ORGANISME D'ACCUEIL

1.1 Introduction

Les applications desktop restent incontournables grâce à leurs performances, leur autonomie et leur accès direct aux ressources système. Ce chapitre présente les principes fondamentaux de ce type d'applications et les atouts du langage Java pour leur développement.

Nous aborderons également les méthodes d'analyse et de conception, notamment le Processus Unifié (UP) et la modélisation UML, outils clés pour structurer des solutions logicielles efficaces. Enfin, nous contextualiserons notre étude en présentant le cabinet dentaire "Diamond Dental Centre", dont les besoins spécifiques ont orienté la conception de notre application.

1.2 Technologie des Applications Desktop

1.2.1 Définition simple d'une application desktop

Une application desktop est un logiciel qui s'installe et s'exécute directement sur un ordinateur (Windows, macOS ou Linux), sans nécessiter de connexion Internet pour fonctionner. Contrairement aux applications web qui tournent dans un navigateur, elle est autonome, plus rapide et offre un meilleur accès aux ressources de la machine (disque dur, imprimante, etc.). Ses caractéristiques principales sont : installation locale (Pas besoin de serveur externe), performances élevées (Optimisée pour le système d'exploitation), accès hors ligne (Fonctionne sans Internet) et une interface riche (Expérience utilisateur fluide et personnalisable) .

1.2.2 Frameworks populaires pour le développement d'applications Desktop en Java

Les frameworks Java simplifient le développement d'applications desktop en offrant des composants pré-construits, des outils de gestion d'interface et des bonnes pratiques architecturales. Voici les plus utilisés :

- 1-JavaFX : Successeur de Swing, JavaFX permet de créer des interfaces modernes et dynamiques avec des effets graphiques avancés (animations, CSS intégré).
- 2- Swing : Bibliothèque historique pour les GUIs Java, basée sur des composants légers (widgets).
- 3- Apache Pivot : Framework open-source combinant interfaces desktop et technologies web (JSON, REST).
- 4- Griffon (pour les projets Groovy/Java) : Framework inspiré de Grails, adapté aux applications desktop en Groovy/Java.
- 5- Eclipse RCP (Rich Client Platform) : Environnement modulaire pour des applications d'entreprise complexes (ex. IDE Eclipse).

1.2.3 Définition de JAVA

Java est un langage de programmation orienté objet de haut niveau, développé par James Gosling et Patrick Naughton chez Sun Microsystems (aujourd'hui Oracle) et lancé en 1995. Conçu pour être portable, sécurisé et performant, il repose sur la JVM (Java Virtual Machine), permettant d'exécuter le même bytecode sur différentes plateformes (Windows, Linux, macOS) grâce au principe "Write Once, Run Anywhere" (WORA). Sa syntaxe, inspirée du C++ mais simplifiée, en fait un langage accessible tout en restant puissant, avec des fonctionnalités clés comme le garbage collection (gestion automatique de la mémoire), le multithreading (programmation concurrente) et une riche bibliothèque standard couvrant les réseaux, les bases de données (JDBC), les interfaces graphiques (JavaFX, Swing) et plus encore.

Java est largement utilisé dans le développement web (Spring, Jakarta EE), les applications Android (bien que Kotlin soit désormais privilégié), les systèmes embarqués, le Big Data (Hadoop, Spark) et les applications d'entreprise. Malgré la concurrence de langages plus récents tels que (Kotlin, Python, Go), il reste extrêmement populaire, se classant régulièrement dans le

top 3 des langages les plus utilisés (TIOBE, Stack Overflow), notamment grâce à sa stabilité, sa modularité et son écosystème robuste (IDE comme IntelliJ IDEA et Eclipse, outils comme Maven et Gradle). Les versions majeures, telles que Java 8 (introduction des lambdas) et Java 17 (LTS), ont renforcé sa modernité, faisant de Java un pilier durable de l'informatique.



FIGURE 1.1 – JAVA.

Pour bien comprendre les bases du langage Java, il est essentiel de se familiariser avec certains concepts clés, tels que le Java Development Kit (JDK), le Java Runtime Environment (JRE) et le Java Virtual Machine (JVM). Dans cette section, nous explorerons ces composants fondamentaux et expliquerons comment ils interagissent pour permettre l'exécution des programmes Java.

1.2.3.1 Java Development Kit (JDK)

Le kit de développement Java (JDK) est un ensemble d'outils essentiels pour le développement d'applications Java. Il permet de fournir un environnement complet pour écrire, compiler, déboguer et exécuter des programmes Java (cf. I.B). Les outils fournis par le JDK sont généralement rencontrés dans tous les environnements de développement logiciel. Il se compose de :

1. Le compilateur Java (javac).
2. Les bibliothèques standard (Java Class Library).
3. Les outils de développement.



FIGURE 1.2 – Java Development Kit.

1.2.3.2 Java Virtual Machine (JVM)

Le (JVM) est un composant essentiel de l'environnement d'exécution Java. Son rôle principal est d'exécuter les programmes Java en convertissant le bytecode Java en instructions exécutables pour la machine sous-jacente. Le JVM permet la portabilité des programmes Java, ce qui signifie qu'ils peuvent être exécutés sur différentes plates-formes sans nécessiter de modifications majeures. Elle fournit une couche d'abstraction entre le code Java et le système d'exploitation, permettant aux programmes Java d'être indépendants de la plateforme.

1.2.3.3 Java Runtime Environment (JRE)

Le Java Runtime Environment (JRE) est un environnement d'exécution nécessaire pour exécuter des applications Java. Il comprend la Java Virtual Machine (JVM), les bibliothèques de classes et d'autres composants essentiels. Il prend en charge l'exécution des programmes Java en fournissant une plateforme d'exécution pour la JVM. Il charge et exécute les fichiers bytecodes Java et fournit l'accès aux bibliothèques de classes nécessaires. Il se compose de :

1. La Java Virtual Machine (JVM).
2. Les bibliothèques de classes du JRE.



FIGURE 1.3 – Java Runtime Environment.

1.2.4 Caractéristiques de Java

1. Distribué

- Gestion native des protocoles réseau (TCP/IP, HTTP, FTP) via des bibliothèques intégrées.
- Accès transparent aux ressources distantes (traitement des URL aussi simple que l'accès aux fichiers locaux).

2. Fiabilité

- Vérifications dynamiques pendant l'exécution pour prévenir les erreurs.
- Modèle de pointeurs sécurisé :
Pas d'accès direct à la mémoire (évite les corruptions de données). Protection contre l'écrasement mémoire.

3. Orienté Objet

- Programmation centrée sur les objets (données + méthodes).
- Encapsulation, héritage, polymorphisme pour une modularité accrue.

4. Sécurité

- Conçu pour les environnements distribués (serveurs, réseaux).
- Protection contre les virus :Sandboxing (exécution isolée dans la JVM).
- Pas d'accès direct au système hôte.

5. Architecture Neutre (Portabilité)

- Bytecode indépendant de l'architecture :
Compilé une fois, exécutable partout (principe Write Once, Run Anywhere).
- Interprété par la JVM (adaptation au processeur cible).

1.2.5 Architecture d'une application desktop

L'architecture d'une application desktop est la structure interne qui définit comment les différents éléments d'une application interagissent et comment l'application est organisée pour fonctionner. En d'autres termes, l'architecture d'une application desktop désigne l'organisation structurelle de ses composants principaux. Voici ses caractéristiques essentielles :

*Modèle 3 couches standard :

- Présentation : Interface utilisateur (fenêtres, formulaires).
- Logique métier : Traitement des données et règles applicatives.
- Accès aux données : Communication avec la base de données.

*Composants typiques :

- Interface graphique (GUI).
- Moteur de traitement principal.
- Connecteurs de base de données.
- Module de gestion des fichiers.

1.3 Méthodes d'analyse et de conception

Une méthode d'analyse et de conception a pour objectif de permettre la formalisation des étapes préliminaires du développement d'un système, afin de rendre ce développement plus fidèle aux besoins du client.

- La phase d'analyse permet de lister les résultats attendus, en termes de fonctionnalités, de performance, de robustesse, de maintenance, de sécurité, d'extensibilité, etc.
- La phase de conception permet de décrire de manière non ambiguë, en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

Exemples de Méthodes Agiles

Il existe plusieurs méthodes agiles, chacune ayant ses propres caractéristiques :

* Scrum : Développement en sprints (cycles courts de 2 à 4 semaines) avec un Scrum Master qui guide l'équipe et des réunions quotidiennes pour ajuster le travail.

* Kanban : Système de gestion visuelle avec un tableau de tâches permettant un suivi en temps réel du travail en cours.

* Extreme Programming (XP) : Mise en avant de la qualité du code avec des pratiques comme le développement par paires, l'intégration continue et les tests fréquents.

Nous avons opté pour le Processus Unifié : (UP-Unified Process).

1.3.1 Définition du processus unifié

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

1.3.2 Caractéristiques du processus unifié

1/ Centré sur l'architecture : L'architecture peut être considérée comme l'ensemble de vues du système qui vont provenir des besoins de l'entreprise et des différents intervenants.

2/ Piloté par les cas d'utilisation : Le modèle des cas d'utilisations guide le processus unifié et décrit les fonctionnalités du système.

3/ Itératif et incrémental : Les itérations se succèdent dans un ordre logique permettant de donner lieu à un incrément et donc d'établir un développement plus optimisé et efficace.

1.3.3 Cycle de vie du processus unifié

Le processus unifié se déroule en quatre phases :

1) Analyse des besoins : Établir une vision globale du projet où on spécifie les besoins et on étudie la faisabilité du projet.

2) Elaboration : On reprend les éléments de l'analyse des besoins et on développe une architecture de référence, les risques et la plupart des besoins sont identifiés.

3) Construction : Finaliser l'analyse, la conception, l'implémentation et les tests puis transformer l'architecture de référence en produit exécutable tout en veillant à respecter son intégrité.

4) Transition : Livraison du produit au client afin d'effectuer des essais pour détecter d'éventuelles anomalies.

Chaque phase répète un nombre de fois une série d'itérations et chaque itération est composée de cinq activités :

- Expression des besoins : Compréhension et expression des besoins et des exigences du client qu'elle soit fonctionnelle ou non fonctionnelle.
- Analyse et Conception : Permet d'acquérir une compréhension approfondie des contraintes liées aux outils de réalisation en prenant en compte le choix d'architecture technique retenu pour le développement et l'exploitation système.
- Implémentation : On implémente le système sous forme de composants, bibliothèques et de fichiers. Elle a pour objectif de planifier l'intégration.
- Tests : Permettent de vérifier les résultats de l'implémentation de toutes les exigences et de s'assurer de la bonne intégration de tous les composants dans le logiciel.
- Déploiement : Livraison et exploitation du produit.[B9]

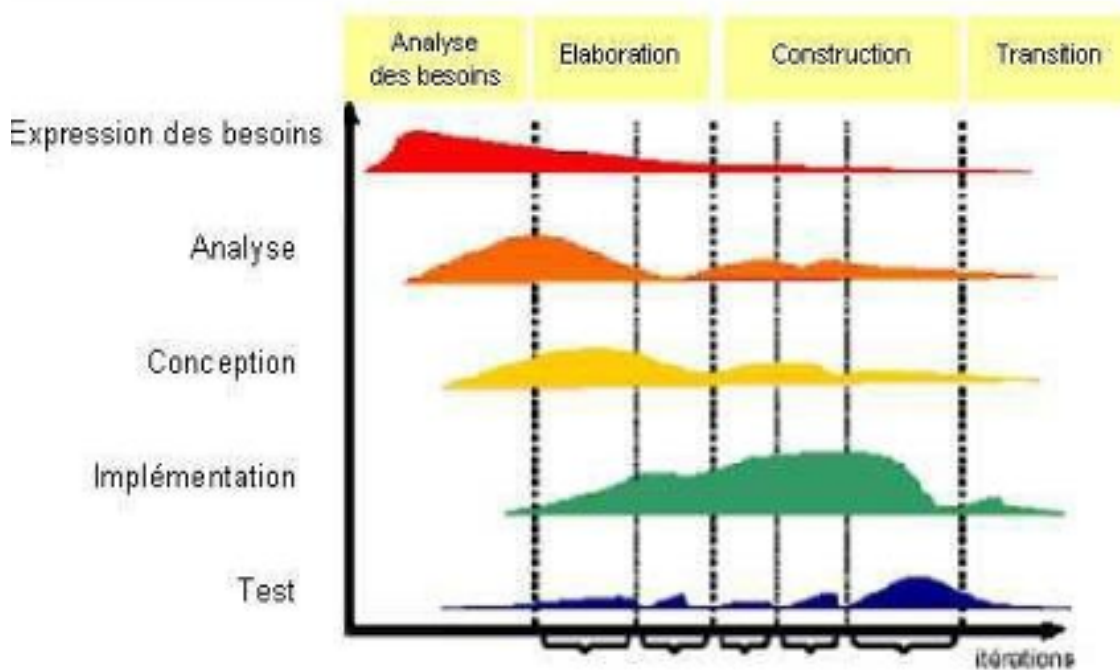


FIGURE 1.4 – Cycle de Vie de UP.

1.4 Langage de modélisation

L'UP s'appuie sur UML qui est un langage de modélisation graphique et textuel destiné à comprendre et à d'écrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vues.

1.4.1 Unified Modeling Language (UML)

C'est un langage de modélisation graphique à base de pictogrammes, conçu pour représenter et spécifier les artefacts de systèmes logiciels, de plus il est destiné à comprendre et décrire des besoins spécifiés et documentés des systèmes, esquissé des architectures logicielles, concevoir des solutions et communiquer des points de vue, comme il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique. UML résulte de l'unification de techniques ayant fait leurs preuves pour l'analyse et la conception de grands logiciels et de systèmes complexes .[B10]

1.4.2 Les diagrammes

1. Définition des diagrammes : Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle. Chaque type de diagramme UML possède une structure et véhicule une sémantique précise .

2. Les différents types de diagrammes : Il existe 14 diagrammes UML, pour notre projet, nous présenterons trois diagrammes : cas d'utilisation, de séquence et de classe.

A. Diagramme de cas d'utilisation (use case diagram) : Les diagrammes de cas d'utilisation représentent un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils représentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système

B. Diagramme de séquence (sequence diagram) : Un diagramme de séquence met en évidence le classement des messages par ordre chronologique. On forme un diagramme de séquence en plaçant d'abord les objets qui participent à l'interaction en haut du diagramme sur l'axe des abscisses. En générale, on place l'objet qui débute l'interaction à gauche, puis on continue en

progressant vers la droite, les objets les plus subordonnés étant tout à fait à droite. On place ensuite les messages envoyés et reçus par ces objets le long de l'axe des ordonnées, par ordre chronologique, du haut vers le bas. Cela donne au lecteur une indication visuelle claire du mot de contrôle dans le temps.

C. Diagramme de classe (class diagram) : Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ses classes. Outre les classes, ils représentent un ensemble d'interfaces et de paquetages, ainsi que leurs relations.

1.5 Présentation de l'organisme d'accueil

1.5.1 Présentation de Cabinet Dentaire "Diamond Dental Centre" :

Le Diamond Dental Centre est un cabinet dentaire situé à Béjaïa en Algérie, est composé de plusieurs chirurgiens dentistes, chapeauté par docteur Kemouche Rassim, implanté en plein centre de Béjaïa, assure l'établissement d'un diagnostic d'expert. La prise en charge des soins dentaires traditionnels ainsi que les missions de sauvetage des situations de délabrement dentaire extrêmes. Le cabinet assure une réhabilitation et une restauration de votre sourire. Son organisation est un système très bien tissé, elle privilège une coordination exemplaire entre les membres du staff. Elle combine une entente mutuelle en interne ainsi qu'une parfaite communication avec les patients en externe, le tout dans un cadre agréable et jovial, leurs site officiel est : <https://diamonddentalcenterdz.com> .

1.5.2 Les Services de Cabinet : [B1]

1.5.2.1 Implantologie Flapless

Implant dentaire sans chirurgie*

L'implantologie représente la branche de la médecine dentaire qui s'adonne au remplacement de la racine par une autre artificielle, celle-ci portera le nom d'implant dentaire. Le dispositif prothétique possède la forme d'une vis qui s'insère au niveau de l'os de la mâchoire. L'implant dentaire est une solution efficace pour remplacer une dent manquante.



FIGURE 1.5 – Implantologie Flapless.

1.5.2.2 Pédodontie

La pédodontie est une spécialité de la dentisterie qui se concentre sur les soins dentaires des enfants. Les dentistes spécialisés en pédodontie prennent en charge les jeunes patients en utilisant une approche douce et empathique pour s'assurer que les enfants se sentent en sécurité et à l'aise lors des soins dentaires. Au Diamond Dental Center, une prise en charge exceptionnelle offerte aux enfants, en veillant à ce que leur expérience chez le dentiste soit positive et agréable.

1.5.2.3 Blanchement Dentaire

Vous souhaitez avoir des dents étincelantes et éclatantes ? le brossage minutieux et méticuleux ne vous a pas donné le résultat que vous souhaitiez ? Aux grands maux, les grands remèdes ! Il est temps d'opter pour le traitement par blanchiment dentaire.

1.5.2.4 Couronne Dentaire et Facette Dentaire

Les facettes dentaires et couronnes dentaires sont des solutions populaires en dentisterie esthétique pour améliorer l'apparence des dents. Les facettes dentaires sont des coques minces qui sont fixées sur la surface des dents pour améliorer leur forme, leur couleur ou leur alignement. Les couronnes dentaires, quant à elles, recouvrent entièrement la dent pour la protéger et restaurer sa forme et sa fonction. Ces traitements sont souvent utilisés pour corriger les dents cassées, ébréchées, décolorées ou mal alignées.

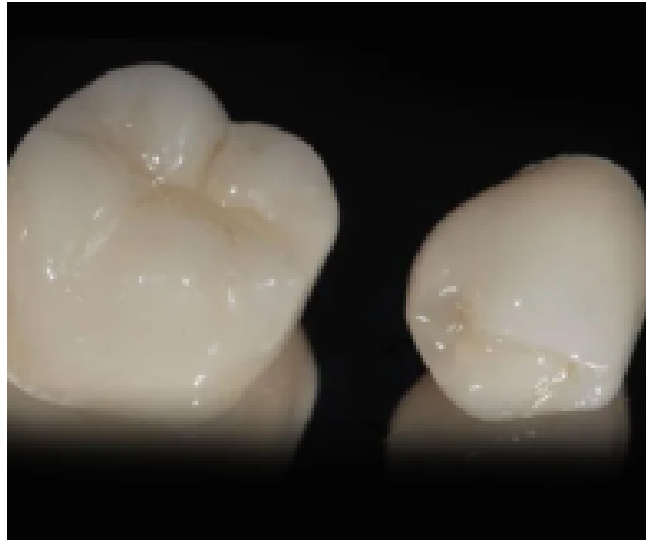


FIGURE 1.6 – Couronne Dentaire et Facette Dentaire.

1.5.2.5 Dentisterie Restauratrice

La dentisterie restauratrice désigne le fait de remettre en état les dysfonctionnements et les soucis bucco-dentaires via une prise en charge spécialisée. Le maître mot est la conservation de la dent originale tant que c'est possible, le remplacement des structures ne se fait que pour les tissus endommagés qu'il n'est pas possible de restaurer.

1.5.2.6 Orthodontie

Appareils dentaires ODF et gouttières dentaires transparentes. L'orthodontie est une spécialité dentaire qui vise à corriger les problèmes de positionnement des dents et des mâchoires. Elle peut être réalisée à l'aide d'un appareil dentaire traditionnel (ODF) ou de gouttières dentaires transparentes. Que ce soit pour des raisons esthétiques ou médicales, l'orthodontie permet d'améliorer la santé bucco-dentaire et d'obtenir un sourire harmonieux.

1.5.2.7 Prothèses Amovibles

Les prothèses dentaires amovibles sont des dispositifs qui servent à substituer au vide laissé par les dents manquantes. L'appareillage est composé d'un ensemble de dents artificielles surplombées d'une gencive.



FIGURE 1.7 – Prothèse Amovible.

1.5.2.8 Hollywood Smile

Le Hollywood Smile est une tendance qui consiste à avoir un sourire “parfait” et éclatant, comme celui des stars d'Hollywood. Pour cela, différents traitements dentaires sont proposés, tels que les facettes dentaires, les couronnes dentaires et le blanchiment des dents. Cette technique permet d'améliorer considérablement l'apparence des dents et de donner à chacun un sourire éclatant de blancheur. Si vous souhaitez vous offrir un sourire de star.



FIGURE 1.8 – Hollywood Smile.

1.5.2.9 Chirurgie Dentaire

La chirurgie dentaire englobe diverses interventions qui peuvent être nécessaires pour corriger des problèmes de santé bucco-dentaire tels que les caries, les maladies de gencive, les infections dentaires ou encore les traumatismes dentaires. La chirurgie dentaire peut également être utilisée à des fins esthétiques, comme pour la correction d'un sourire gingival ou la modification de la forme des gencives.

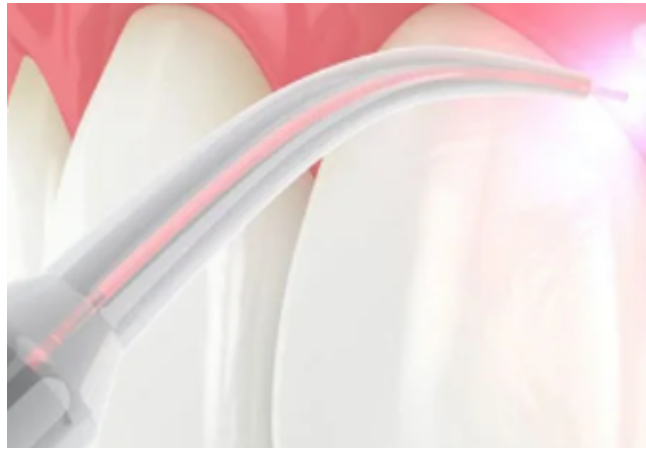


FIGURE 1.9 – Chirurgies Dentaires.

1.6 Conclusion

Les applications desktop s'appuient sur des architectures robustes et des méthodologies rigoureuses pour allier performance, fiabilité et autonomie. Leur conception exige une approche structurée, combinant des technologies éprouvées comme Java avec des méthodes telles que le Processus Unifié et UML. Ces éléments sont déterminants pour développer des solutions logicielles stables, en particulier dans le domaine dentaire où la précision et l'efficacité opérationnelle sont cruciales.

La présentation du Diamond Dental Centre illustre concrètement l'utilité de ce projet, qui vise à moderniser la gestion clinique et à optimiser la prise en charge des patients. Une telle application desktop représente ainsi un levier stratégique pour améliorer à la fois la qualité des soins et l'organisation du cabinet.

CHAPITRE 2

CAHIER DES CHARGES ET ANALYSE DES BESOINS

2.1 Introduction

Ce chapitre a pour objectif de formaliser les exigences et les attentes du projet à travers l'élaboration du cahier des charges et l'analyse des besoins. Il s'agit d'une étape cruciale qui permet de cerner précisément les contours du projet, d'identifier les fonctionnalités requises et de définir les contraintes techniques, organisationnelles et budgétaires.

La première partie, consacrée au cahier des charges, précise les objectifs du projet, son périmètre, ses contraintes (techniques, budgétaires, temporelles) , ainsi que les fonctionnalités attendues. Ce document servira de référence tout au long du développement pour aligner les attentes des différentes parties prenantes.

La seconde partie, dédiée à l'analyse des besoins, modélise et structure les exigences à l'aide de diagrammes. Cette approche visuelle permet de clarifier les interactions utilisateur-système, les processus métiers et les flux d'informations, facilitant ainsi la conception future.

Ensemble, ces deux volets assurent une compréhension commune du projet et posent les bases techniques nécessaires à sa réalisation.

2.2 Problématique et objectif du projet

2.2.1 Problématique

Dans le domaine dentaire, la gestion manuelle des patients, des rendez-vous et des traitements engendre plusieurs difficultés :

*Gestion fastidieuse des rendez-vous : Les annulations, reports ou modifications de rendez-vous sont difficiles à gérer manuellement, entraînant des erreurs et une perte de temps.

*Surcharge administrative : La secrétaire doit gérer les dossiers papiers, les prescriptions et les traitements, ce qui ralentit le processus de soin.

*Accès limité aux dossiers patients : Les dossiers papiers sont vulnérables (perte, détérioration) et leur consultation est peu pratique.

*Manque de traçabilité : Le suivi des traitements et des prescriptions n'est pas optimisé, ce qui peut nuire à la qualité des soins.

Cette application vise à automatiser et centraliser la gestion du cabinet dentaire pour améliorer l'efficacité, la sécurité et la qualité des soins.

2.2.2 Contexte

Avec la digitalisation croissante des systèmes de santé, les professionnels médicaux, y compris les dentistes, ont besoin d'outils performants pour optimiser leur pratique clinique et administrative. Une application desktop dédiée à la gestion d'un cabinet dentaire se présente comme une solution robuste, sécurisée et efficace, permettant de répondre aux défis quotidiens liés à :

*La gestion des patients : Remplacement des dossiers papiers par des fiches numériques complètes, consultables en un clic.

*L'organisation des rendez-vous : Automatisation des prises de rendez-vous, gestion des annulations /modifications.

*Faciliter la prescription et le suivi des traitements.

*Edition et impression des ordonnances.

*Sécuriser les données médicales (accès contrôlé, sauvegarde automatisée).

Cette application s'adresse aux dentistes, secrétaires et assistants, en remplaçant les méthodes traditionnelles par une solution numérique intégrée. Contrairement aux solutions web, une application desktop offre une meilleure stabilité, des temps de réponse plus rapides et une indépendance vis-à-vis d'une connexion Internet. Elle s'intègre parfaitement dans l'environnement de travail d'un cabinet dentaire, tout en garantissant simplicité d'utilisation, productivité accrue et réduction des erreurs administratives.

2.2.3 Objectif du projet

Notre projet consiste à réaliser une application desktop dédiée à la gestion d'un cabinet dentaire.

La conception et le développement de notre application vise à atteindre les objectifs suivants :

*Une gestion centralisée des patients :Création, modification et archivage des dossiers patients/Historique des consultations, traitements et prescriptions.

*Une gestion efficace des rendez-vous : Prise,Annulation et modification des rendez-vous.

*Un suivi des traitements et prescriptions :Enregistrement des diagnostics et des plans de traitement,Génération automatique de prescriptions,Suivi de l'évolution des soins.

*Edition des ordonnances :génération , enregistrement dans la base de donnée et impression des ordonnances.

* Une sécurisation des données :Accès sécurisé par authentification,Sauvegarde des données pour éviter toute pertes.

2.3 Cahier des Charges

Notre cahier des charges est sous forme d'un ensemble de besoins fonctionnels et de besoins non fonctionnels, présenté comme suit :

2.3.1 Définition des Besoins Fonctionnels

Les besoins fonctionnels expriment les actions que doit effectuer le système en réponse à une demande (sorties qui sont produites pour un ensemble donné d'entrées).

L'application à réaliser doit satisfaire l'ensemble des fonctionnalités définies par les utilisateurs.

Cette plate-forme doit répondre aux besoins fonctionnels suivants :

*Authentification et Sécurité

-Page de login avec accès différencié (dentiste, secrétaire).

-Mot de passe chiffré et gestion des sessions.

*Gestion des Patients

-Ajout/modification/suppression de fiches patients.

-Consultation de l'historique médical (radiographies, traitements passés).

-Recherche et filtrage des patients (nom, date, numéro de dossier).

*Gestion des Rendez-vous

-Calendrier interactif avec plages horaires.

-Prise de rendez-vous avec sélection du motif (consultation, contrôle, urgence).

-Gestion des conflits de disponibilités.

*Gestion des traitements et des prescriptions

-Enregistrement des diagnostics et des plans de traitement.

-Génération de prescriptions (modèles pré-enregistrés).

-Suivi de l'avancement des soins.

*Edition des ordonnances.

-Enregistrement des ordonnances dans la base de données.

-Impression des ordonnances.

2.3.2 Définition des Besoins non Fonctionnels

Ce sont des exigences techniques spécifiques au système. Elles doivent être prise en compte tout au long du processus de développement de l'application, à savoir :

* Sécurité : tous les accès des utilisateurs (dentist,secrétaire) doivent être protégés par un identifiant et un mot de passe.

* Validité : l'application desktop doit réaliser exactement les taches spécifiées dans le cahier des charges.

*Performance : le temps de réponse doit être court.

* L'ergonomie et la convivialité : interface intuitive (menus clairs, formulaires simplifiés)/Compatibilité avec les résolutions d'écran courantes.

* Maintenabilité : Code modulaire pour faciliter les mises à jour/Documentation technique et utilisateur/ Le code doit être clair pour permettre de futures évolutions ou améliorations .

Cette application permettra au cabinet dentaire d'améliorer son efficacité, de réduire les erreurs et d'offrir un meilleur suivi des patients, grâce à une gestion numérique intégrée.

2.4 Analyse des Besoins

2.4.1 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Autrement dit un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Dans le cadre de notre étude, nous avons distingué deux acteurs principaux :

Acteur	Description
Dentiste	Le dentiste est l'utilisateur principal de l'application. Il gère les patients, leurs traitements et prescriptions, les rendez-vous ainsi que l'édition des ordonnances.
Assistant	L'assistant est un utilisateur secondaire de l'application. Il gère les patients et leurs rendez-vous.

TABLE 2.1 – Description des Acteurs

2.4.2 Diagramme de Contexte

Le diagramme de contexte a pour but de représenter les flux d'information entre le système et les acteurs externes selon une représentation standard dans laquelle chaque entité (acteur ou système) est identifié par un nom.

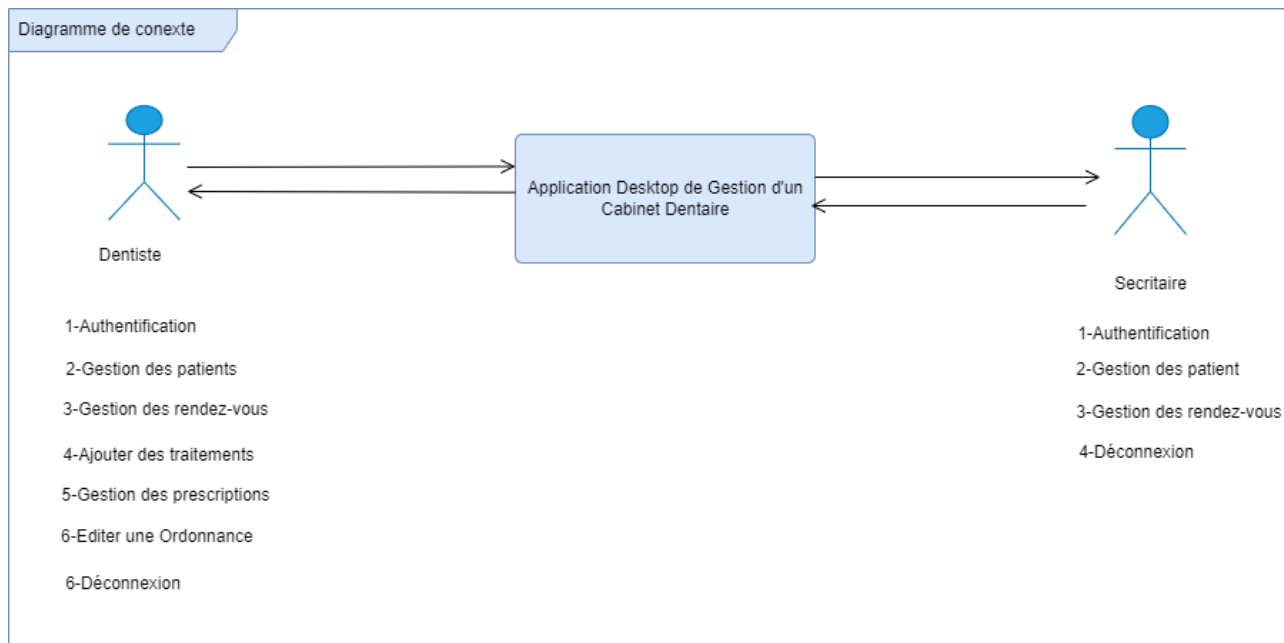


FIGURE 2.1 – Diagramme de Contexte .

2.4.3 Identification des Cas d'Utilisation

Les cas d'utilisation sont des outils formels utilisés pour identifier et modéliser les besoins fonctionnels du système. Chaque cas d'utilisation représente une unité fonctionnelle cohérente, visible de l'extérieur.

Le tableau suivant illustre l'ensemble d'actions réalisées par le système :

N°	Cas d'utilisation	Acteur(s)
1	Authentification	Dentiste/Assistant
2	Gestion des patients	Dentiste/Assistant
3	Gestion des rendez-vous	Dentiste/Assistant
4	Gestion des traitements	Dentiste
5	Gestion des prescriptions	Dentiste
6	Édition des ordonnances	Dentiste
7	Déconnexion	Dentiste/Assistant

TABLE 2.2 – Cas d'utilisation associés aux acteurs

2.4.4 Diagramme global de cas d'utilisations

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont particulièrement utiles lors de présentation auprès de la direction ou des acteurs d'un projet.

Pour notre système, le diagramme global des cas d'utilisation est représenté dans la figure

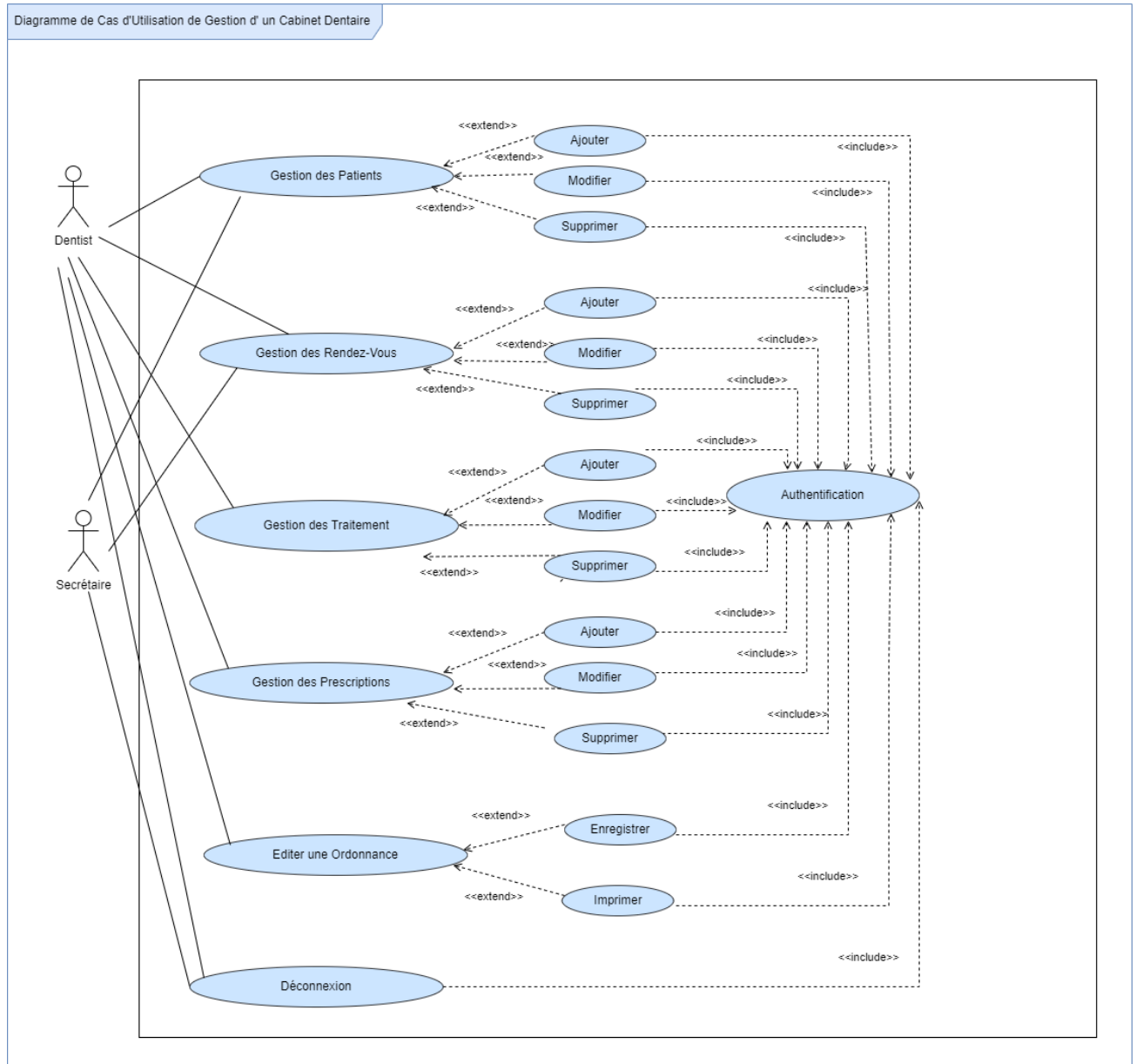


FIGURE 2.2 – Diagramme global de cas d'utilisation .

2.4.5 Description des cas d'utilisation

Chaque cas d'utilisation d'un système doit être défini textuellement. Cela consiste à :

- * Identifier le cas : résumer son objectif et les acteurs impliqués.
- * Décrire un scénario nominal : il s'agit d'un ensemble de messages échangés entre les acteurs et le système, décrivant le déroulement idéal des actions, où tout se passe sans erreur .
- * Décrire un scénario alternatif : un ensemble d'actions qui s'exécutent si les conditions dans le scénario nominal ne sont pas validées.

Dans les tableaux ci-dessous, nous décrirons les cas d'utilisation les plus pertinents :

Cas d'utilisation n°1	S'authentifier
Résumé	Vérifier l'identité de l'utilisateur avant d'accéder au système.
Acteur(s)	Dentiste / Assistant
Pré-condition	Avoir accès à l'application (connexion réseau, application installée).
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur saisit ses identifiants (login et mot de passe) 2. L'utilisateur valide les informations 3. Le système vérifie les informations 4. Le système affiche le menu principal
Scénario alternatif	<p>A. Si les informations sont incorrectes :</p> <ul style="list-style-type: none"> — Le système affiche un message d'erreur — Le système propose de ressaisir les informations

TABLE 2.3 – Description détaillée du cas d'utilisation « S'authentifier »

Cas d'utilisation n°2	Gestion des Patient
Résumé	ajouter,modifier ou supprimer les dossiers des patients.
Acteur(s)	Dentiste / Assistant
Pré-condition	utilisateur authentifié avec les droits nécessaires.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur sélectionne :Gestion des Patients 2. Le systeme affiche l'interface des dossiers patients 3. L'utilisateur choisi d'ajouter,modifier ou supprimer un dossier patient 4. Le système enregistre ou affiche les informations
Scénario alternatif	<ul style="list-style-type: none"> — si le patient existe déjà,le systeme propose de mettre a jour — si les informations sont incompatible,le systeme demande le resaisie

TABLE 2.4 – Description détaillée du cas d'utilisation « Gestion des Patients »

Cas d'utilisation n°3	Gestion des Rendez-Vous
Résumé	planifier,annuler ou modifier un rendez-vous.
Acteur(s)	Dentiste / Assistant
Pré-condition	utilisateur authentifié /patient enregistré.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur sélectionne :Gestion des rendez-vous 2. L'utilisateur sélectionne un patient et clique pour récupérer ses informations 3. L'utilisateur un rendez-vous(date/heure) 4. L'utilisateur confirme et le systeme enregistre le rendez-vous
Scénario alternatif	<ul style="list-style-type: none"> — si le créneau est indisponible,le systeme propose des alternatives — si le patient a déjà un rendez-vous ,le systeme avertit l'utilisateur

TABLE 2.5 – Description détaillée du cas d'utilisation « Gestion des Rendez-Vous»

Cas d'utilisation n°4	Gestion des Traitements
Résumé	ajouter,modifier ou supprimer les traitements disponible à la clinique.
Acteur(s)	Dentiste
Pré-condition	dentiste authentifié .
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur sélectionne l'interface :Gestion des Traitements 2. Il ajoute un traitement avec ses détails(dose,nom,prix) 3. Le système valide et enregistre le traitement
Scénario alternatif	<ul style="list-style-type: none"> — si le traitement est déjà exist,le systeme n'effectue pas l'enregistrement — si les champs des informations sont incompatible ,le systeme demande le resaisie

TABLE 2.6 – Description détaillée du cas d'utilisation « Gestion des Traitements »

Cas d'utilisation n°5	Ajouter des Prescriptions
Résumé	ajouter /modifier/supprimer des prescriptions appliqué au patients.
Acteur(s)	Dentiste
Pré-condition	dentiste authentifié /patient enregistré/traitement enregistré.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur sélectionne :Gestion des traitements 2. L'utilisateur sélectionne un patient et clique pour récupérer ses informations 3. L'utilisateur sélectionne un traitement et clique pour récupérer ses détails 4. Le systeme valide et sauvegarde le traitement
Scénario alternatif	— si le traitement est existe déjà ,le systeme n'effectue pas l'enregistrement

TABLE 2.7 – Description détaillée du cas d'utilisation « Ajouter des Prescriptionst»

Cas d'utilisation n°6	Edition des Ordonnances
Résumé	Générer une ordonnance imprimable.
Acteur(s)	Dentiste
Pré-condition	dentiste authentifié /patient enregistré.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur clique sur :Editer une Ordonnance 2. Le systeme affiche un modèle pré-rempli 3. L'utilisateur récupère le patient ses info 4. L'utilisateur remplit les médicaments nécessaire 5. clique sur enregistrer 6. clique sur PDF après confirme l'impression
Scénario alternatif	— si l'un des champs pas remplie,le systeme ne valide pas l'enregistrementet bloque l'édition

TABLE 2.8 – Description détaillée du cas d'utilisation « Editer une Ordonnance»

2.4.6 Modélisation des diagrammes de séquence

Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont exécutées. Il fournit une représentation graphique des interactions entre les acteurs et le système, selon un ordre chronologique, conformément à la notation UML.

2.4.6.1 Diagramme de séquence du cas d'utilisation « S'authentifier »

L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification des informations associées à un utilisateur (un nom d'utilisateur et un mot de passe). Lors d'une authentification deux cas se présentent : les informations introduites par l'utilisateur sont incomplètes, dans ce cas un message d'erreur s'affiche, ou les informations saisies sont complètes et le système procède à leur vérification.

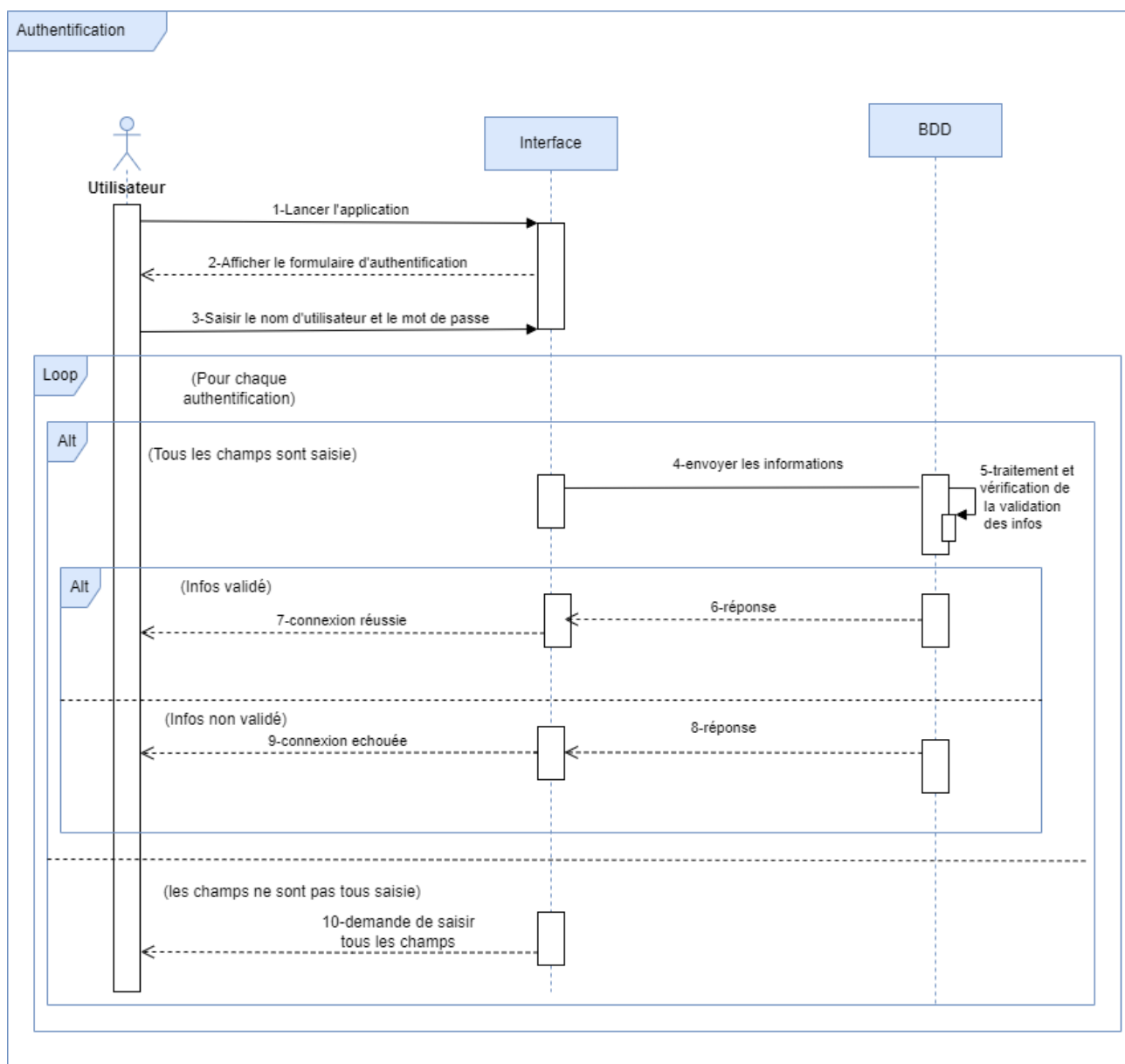


FIGURE 2.3 – Diagramme de séquence du cas d'utilisation « S'authentifier » .

2.4.6.2 Diagramme de séquence du cas d'utilisation « Gestion des Patients »

Après que l'utilisateur s'authentifie, il remplit les champs nécessaire et enregistrer le dossier de patient, il peut modifier ou supprimer un dossier.

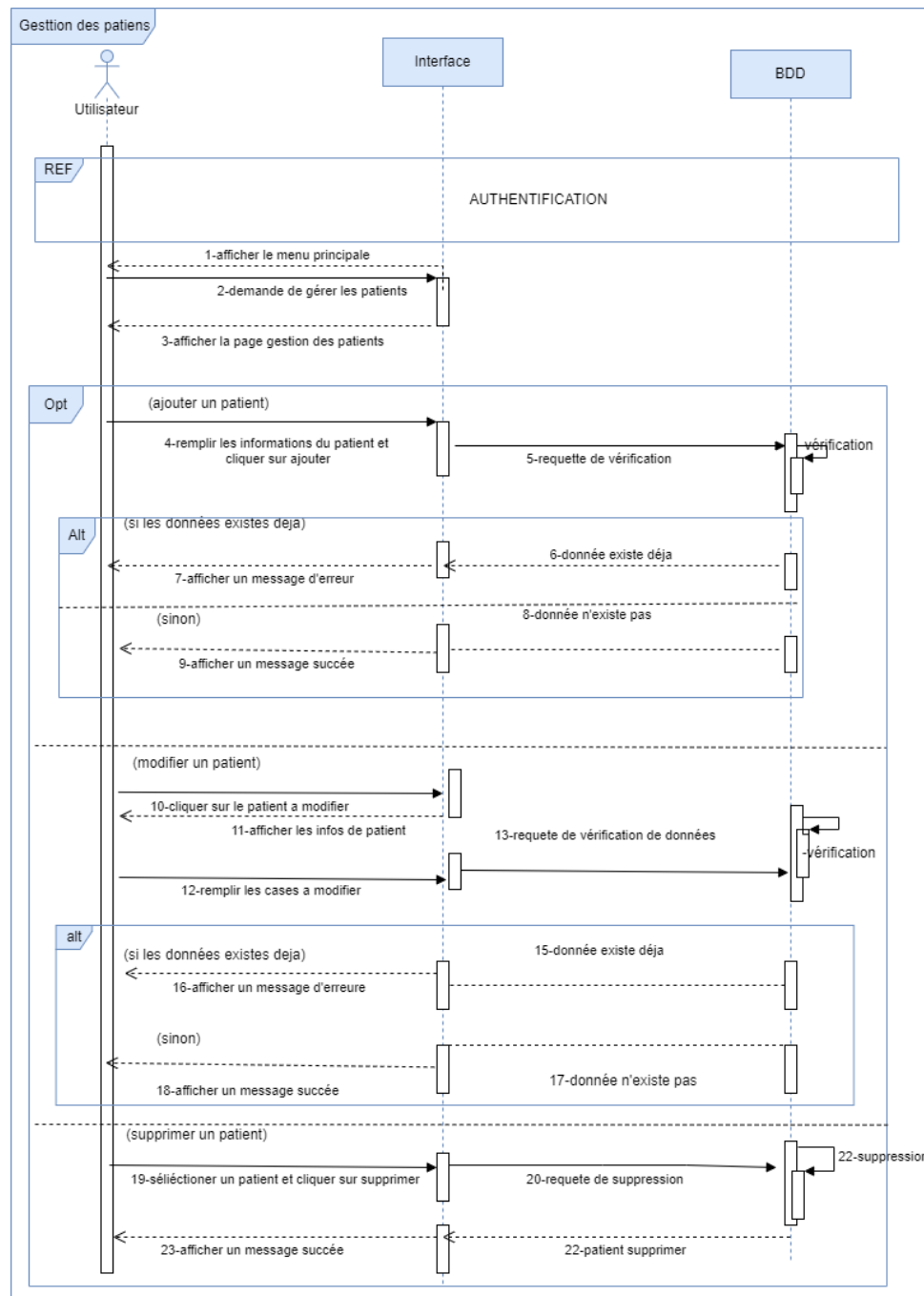


FIGURE 2.4 – Diagramme de séquence du cas d'utilisation « Gestion des Patients » .

2.4.6.3 Diagramme de séquence du cas d'utilisation «Gestion des Rendez-Vous»

Après authentification il récupère le nom de patient enregistré, après il choisi un rendez vous et enregistrer le rendez-vous.

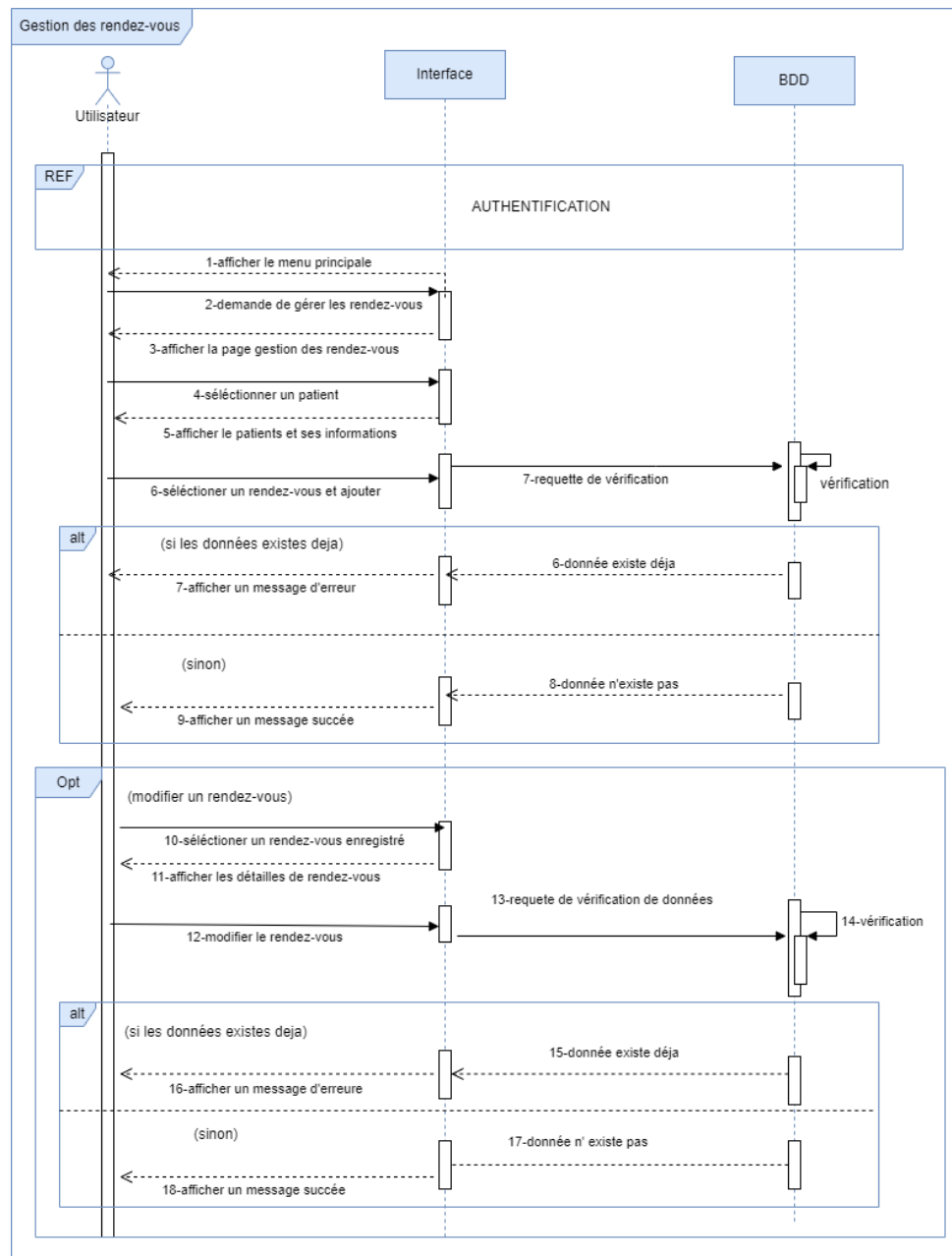


FIGURE 2.5 – Diagramme de séquence du cas d'utilisation « Gestion des Rendez-Vous » .

2.4.6.4 Diagramme de séquence du cas d'utilisation «Gestion des Traitements»

Le dentiste enregistre les services disponible dans le cabinet .

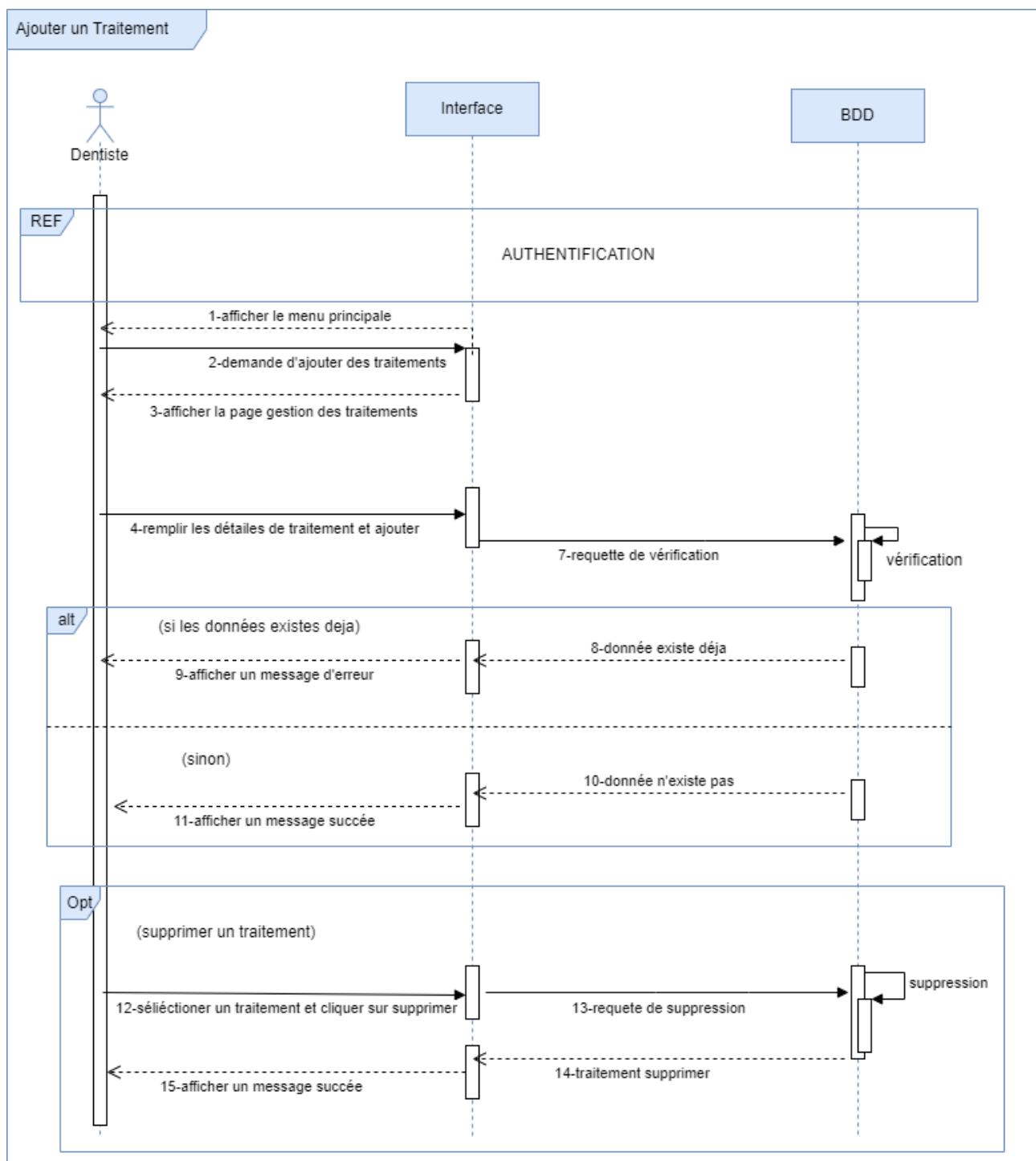


FIGURE 2.6 – Diagramme de séquence du cas d'utilisation « Gestion des Traitements » .

2.4.6.5 Diagramme de séquence du cas d'utilisation «Ajouter des Prescriptions »

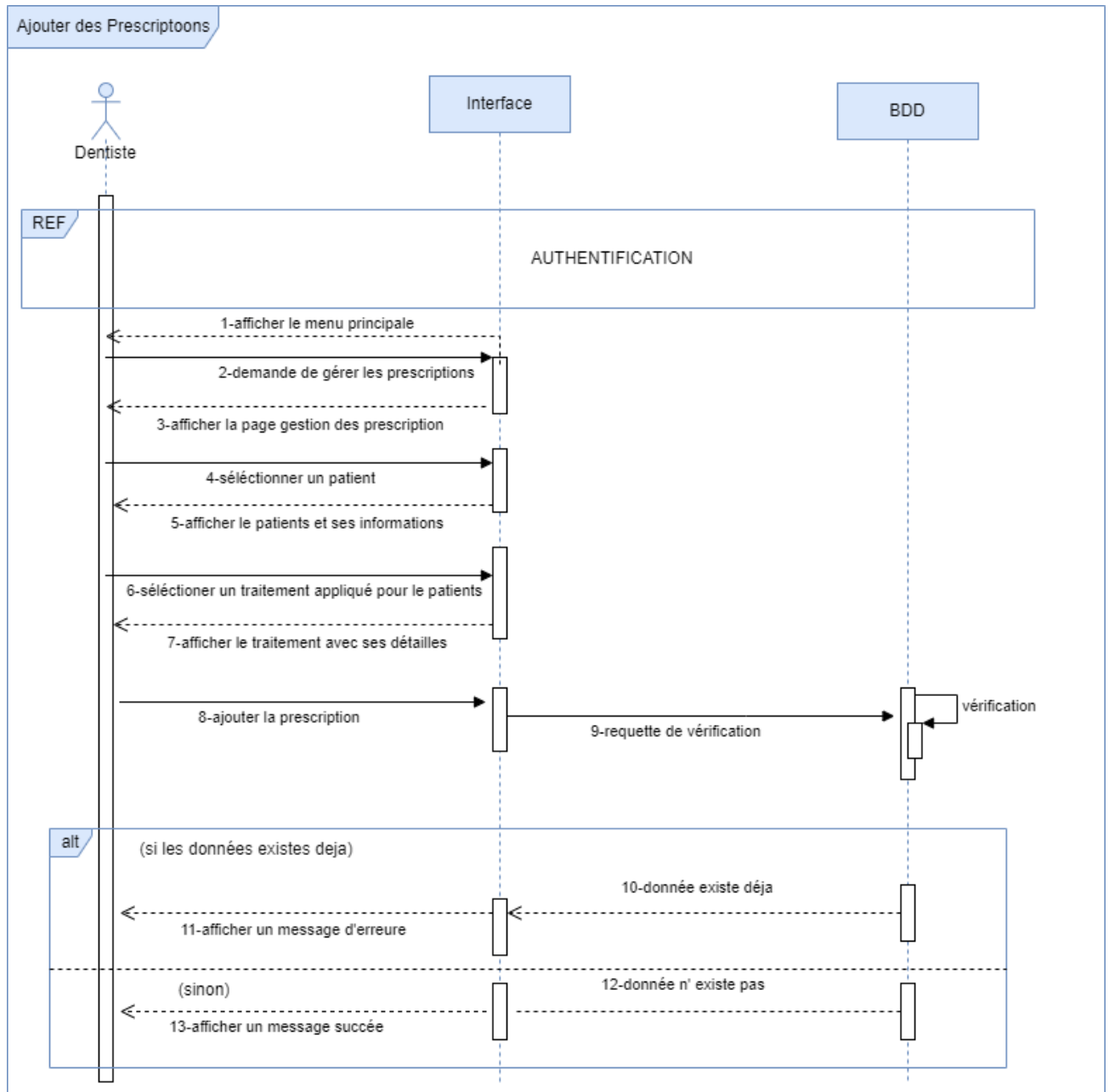


FIGURE 2.7 – Diagramme de séquence du cas d'utilisation « Ajouter des Prescriptions ».

2.4.6.6 Diagramme de séquence du cas d'utilisation «Editer des ordonnances »

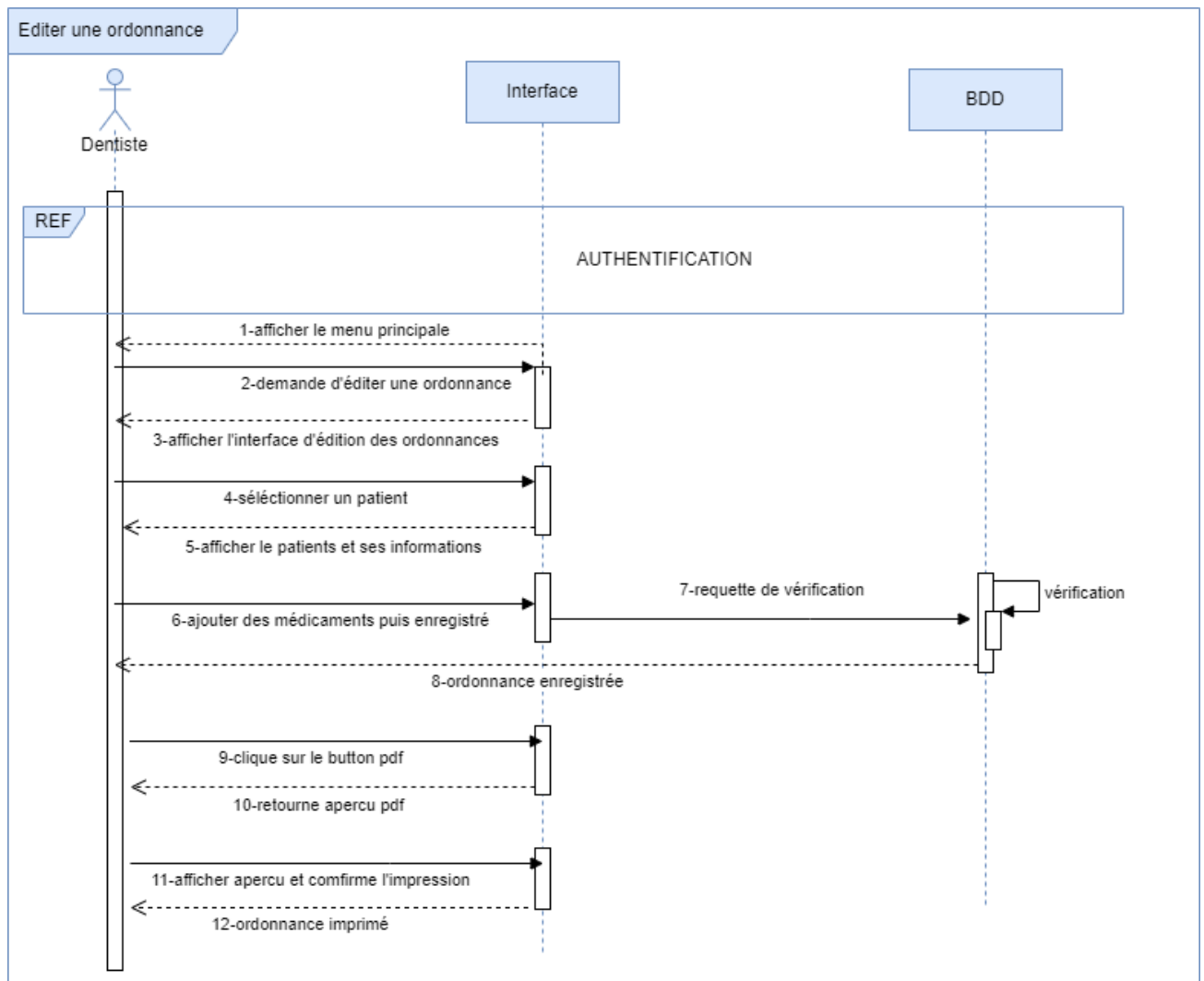


FIGURE 2.8 – Diagramme de séquence du cas d'utilisation « éditer des ordonnances ».

2.5 Conclusion

Dans ce chapitre, nous avons présenté la problématique et défini les objectifs du projet, suivi avec un cahier de charge qui définit les besoins fonctionnels et non fonctionnels de notre projet. Ensuite nous avons analysé ces besoins via les diagrammes de cas d'utilisation, suivi par les diagrammes de séquence système. Cette phase d'analyse nous a permis de décrire de manière globale les attentes des utilisateurs et le fonctionnement souhaité du système afin d'en faciliter la réalisation et la maintenance futures.

Dans le chapitre suivant, nous aborderons une phase cruciale du développement : la conception, où nous décrirons en détail comment les besoins identifiés seront concrètement implémentés dans notre application..

3.1 Introduction

La phase de conception suit immédiatement la phase d'analyse, elle propose une solution aux problèmes identifiés précédemment. Autrement dit, cette étape vise à lever les abstractions définies lors de l'analyse et à fournir une vision claire, structurée et simplifiée du système avant l'implémentation.

3.2 Réalisation du diagramme de classes

3.2.1 Présentation du diagramme de classes

Le diagramme de classes constitue l'un des pivots essentiels de la modélisation avec UML. Il permet de représenter la structure statique du système à développer, en décrivant les entités principales et leurs relations..

La description du diagramme de classe est fondée sur :

- * Le concept d'objet.
- * Le concept de classe ,comprenant les attributs et les opérations.
- * Les différents types d'associations entre les classes (agrégation, composition, héritage, etc.).

3.2.2 Spécification des règles de gestion pour le cabinet dentaire

1. Un dentiste peut gérer plusieurs patients :

Le dentiste a accès à tous les dossiers patients et peut effectuer toutes les opérations médicales.

2. Un assistant peut gérer les rendez-vous et les fiches patients :

L'assistant peut créer/modifier/supprimer des rendez-vous et mettre à jour les informations de base des patients.

3. Un dentiste peut créer plusieurs prescriptions :

Chaque dentiste peut établir des prescriptions médicales pour ses patients selon leurs besoins..

4. Un dentiste peut générer et imprimer des ordonnances :

Le système permet au dentiste de produire et d'imprimer des ordonnances médicales.

5. Un dentiste peut fixer des rendez-vous :

Le dentiste a la capacité de planifier des consultations pour ses patients.

6. Un patient peut recevoir plusieurs prescriptions :

Un patient peut bénéficier de plusieurs prescriptions au fil des différentes consultations.

7. Un patient peut avoir plusieurs traitements :

Un même patient peut suivre plusieurs traitements , simultanément ou de manière successive.

8. Une prescription contient au moins un traitement :

Toute prescription médicale doit spécifier un ou plusieurs traitements.

3.2.3 Le diagramme de classes

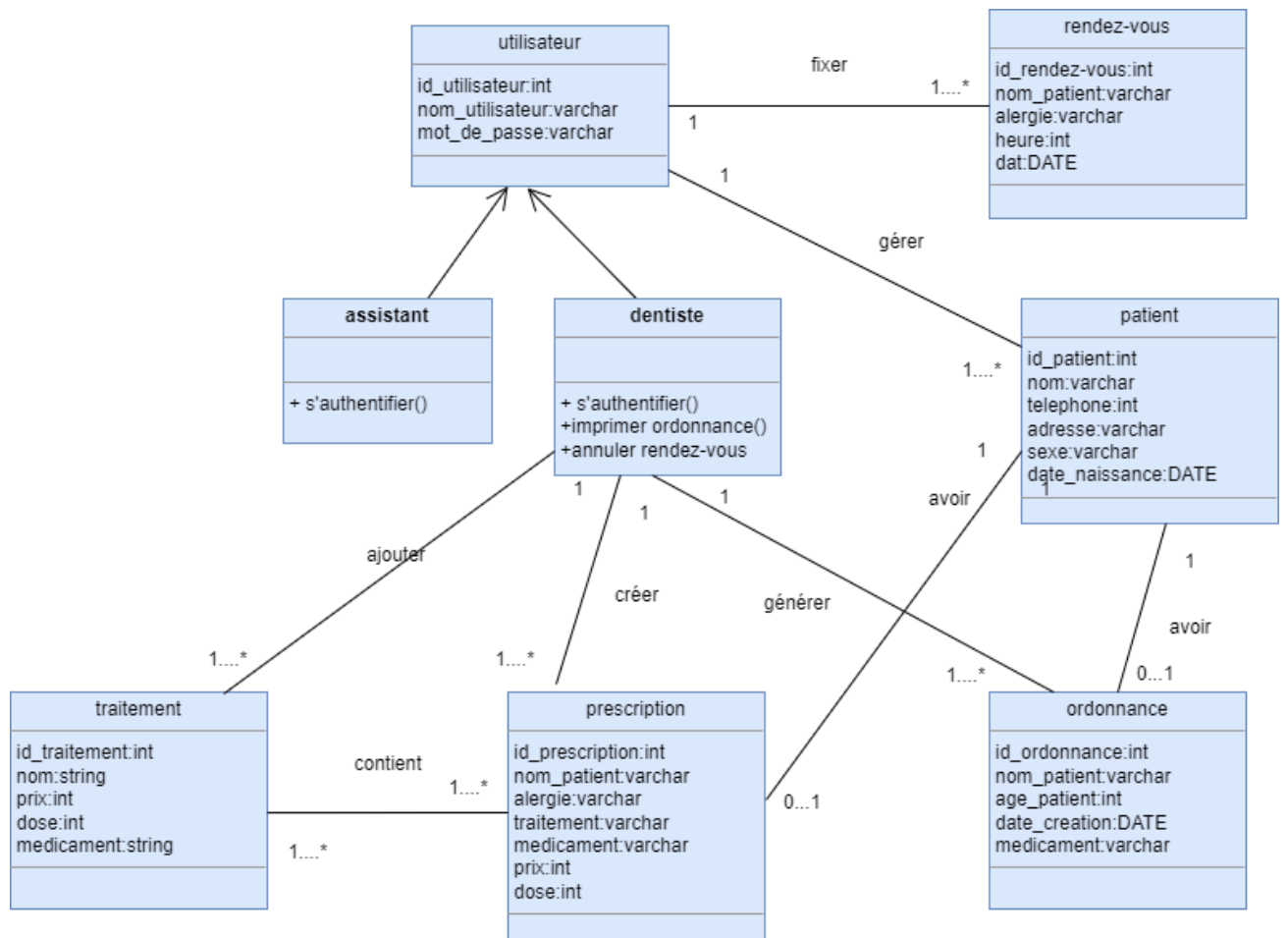


FIGURE 3.1 – Diagramme de classes.

3.3 Le dictionnaire des données

Le dictionnaire de données recense les différentes informations manipulées par l'application, classées par catégories. Il s'agit d'un ensemble de tableaux présentant les attributs identifiés lors de l'analyse préalable.

Nom de l'attribut	Désignation	Type	Dimension
id_utilisateur	Identifiant unique de l'utilisateur	integer	10
nom	le nom complet de l'utilisateur	VARCHAR	20
mot_de_passe	le mot de passe de l'utilisateur	VARCHAR	20

TABLE 3.1 – Table utilisateur

Nom de l'attribut	Désignation	Type	Dimension
id_patient	Identifiant unique de patient	integer	10
nom	le nom complet de patient	VARCHAR	20
telephone	le numero de téléphone de patient	integer	20
adresse	l'adresse complet de patient	VARCHAR	50
sexe	le sexe de patient	VARCHAR	50
date_de_naissance	la date de naissance de patient	date	30

TABLE 3.2 – Table patient

Nom de l'attribut	Désignation	Type	Dimension
id_rendezvous	Identifiant unique de rendez vous	integer	10
nom_patient	le nom complet de patient	VARCHAR	20
alergie	l'allergie sur un médicament de patient	VARCHAR	20
heure	l'heure de rendez vous	integer	10
dat	la date complet de rendez vous	date	20

TABLE 3.3 – Table rendez_vous

Nom de l'attribut	Désignation	Type	Dimension
id_ordonnance	Identifiant unique de l'ordonnance	integer	10
nom_patient	le nom complet de patient	VARCHAR	20
age_patient	l'age de patient	integer	20
date_creation	la date de creation de l'ordonnance	date	10
medicament	le médicament precrit par le dentiste	VARCHAR	20

TABLE 3.4 – Table ordonnance

Nom de l'attribut	Désignation	Type	Dimension
id_prescription	Identifiant unique de la prescription	integer	10
nom_patient	le nom complet de patient	VARCHAR	20
alergie	l'allergie sur la quelle souffre le patient	VARCHAR	20
traitement	le traitement appliqué pour le patient	VARCHAR	20
prix	le prix de traitement appliqué pou le patient	integer	20
dose	la dose de traitement appliqué pour le patient	integer	20

TABLE 3.5 – Table prescription

Nom de l'attribut	Désignation	Type	Dimension
id_traitement	Identifiant unique de traitement	integer	10
nom	le nom de traitemnet	VARCHAR	20
prix	le prix de type traitement	integer	20
dose	la dose de médicament utilisé pour le traitement	integer	20
medicament	le médicament utilisé pour le traitement	VARCHAR	20

TABLE 3.6 – Table traitement

3.4 Le modèle relationnel

Le concepteur d'une base de données relationnelle doit élaborer un schéma relationnel qui décrit les différentes relations de la base ainsi que leurs attributs..

Cette activité consiste à définir toutes les relations de la base de données et leurs attributs.

3.4.1 Modèle relationnel pour notre cas

```
dentist(id_dentist,nom,mot_de_passe)
assistant(id_assistant,nom,mot_de_passe)
rendezvous(id_rendezvous,nom,allergie,heure,dat,#id_dentist,#id_assistant)
traitement(id_traitement,nom,prix,dose,médicament, #id_dentist)
prescription(id_prescription,nom_patient,allergie,traitement,médicament, prix,dose, #id_dentist,
#id_patient)
contient(#id_traitement,#id_prescription)
patient(id_patient,nom,telephone,adresse,date_naissance,sexe,allergie,#id_dentist,
#id_assistant)
ordonnance(id_ordonnance, nom_patient, age_patient, date_creation, médicament, #id_dentist,
#id_patient)
```

3.5 Conclusion

Dans ce chapitre, nous avons réalisé le diagramme de classes de notre application, ainsi que le modèle relationnel de données qui en découle. Ce dernier permet de structurer efficacement la base de données en identifiant les différentes entités, leurs attributs, ainsi que les relations entre elles..

Le chapitre suivant sera dédié à la réalisation, où nous présenterons l'implémentation de l'application et les outils utilisés . Des copies d'écran seront ajoutées pour montrer les fonctionnalités de notre application.

4.1 Introduction

Après avoir terminé la phase de conception, nous allons entamer la dernière phase de notre projet, qui consiste à la mettre en pratique de tous ce qui a été déjà présenté précédemment. Dans ce chapitre nous présenterons l’environnement et les outils de développement utilisés pour la réalisation de notre application. Nous expliquerons également son fonctionnement global, illustré par un organigramme d’interaction ainsi que quelques interfaces représentatives.

4.2 Langages et environnements de développement

4.2.1 Draw.io

Draw.io est un logiciel de diagramme en ligne simple et puissant qui permet de créer facilement des diagrammes, des organigrammes, des schémas et bien d’autres représentations visuelles. . Avec une interface intuitive et conviviale, Draw.io offre une large gamme d’outils de dessin et de formes pré-définies pour vous aider à donner vie aux idées de manière claire et structurée. Que vous soyez un professionnel à la recherche d’un outil de visualisation de données ou un étudiant qui souhaite créer des présentations claires et attrayantes, Draw.io vous permet de créer des diagrammes de qualité professionnelle sans effort. Grâce à sa compatibilité avec diverses plates-formes et sa facilité de partage, vous pouvez collaborer en temps réel avec d’autres utilisateurs et travailler ensemble sur des projets. Avec Draw.io, vous disposez de tous les outils

nécessaires pour donner vie à vos concepts et communiquer efficacement vos idées visuelles.[B2]

4.2.2 NetBeans IDE 8.0.1

NetBeans IDE 8.0.1 est un environnement de développement intégré (IDE) open source et gratuit, conçu pour faciliter la création d'applications Java, ainsi que d'applications web, mobiles, et dans d'autres langages comme PHP et JavaScript. Il offre un support complet pour le cycle de développement, de la création du projet au débogage, au profilage et au déploiement. NetBeans 8.0.1 apporte des améliorations significatives pour le développement web avec HTML5 et JavaScript, incluant le support de RequireJS, Grunt, Karma, et Bower. Il offre également des améliorations pour Java, PHP, et Git. En résumé, NetBeans IDE 8.0.1 est une version mise à jour de l'IDE NetBeans qui se concentre sur l'amélioration du développement web avec des technologies comme HTML5, JavaScript, et CSS3.[B3]

4.2.3 SQLite

SQLite est une bibliothèque écrite en langage C qui propose un moteur de base de données relationnelle accessible par le langage SQL. Elle implémente en grande partie le standard SQL-92 ainsi que les propriétés ACID.

Contrairement aux serveurs de bases de données traditionnels, tels que MySQL, MariaDb ou PostgreSQL, SQLite ne produit pas le schéma habituel client-serveur mais elle est directement intégrée aux programmes. L'intégralité de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme.

D. Richard Hipp, le créateur de SQLite, a choisi de mettre cette bibliothèque ainsi que son code source dans le domaine public, ce qui permet son utilisation sans restriction aussi bien dans les projets open source que dans les projets propriétaires.[B5]

4.2.4 SQL

SQL (Structured Query Language) est un langage informatique standard pour la gestion des bases de données relationnelles et la manipulation des données. SQL est utilisé pour interroger, insérer, mettre à jour et modifier des données. La plupart des systèmes de gestion de bases de données relationnelles (SGBDR) prennent en charge SQL, ce qui constitue un avantage important pour les administrateurs de base de données, , notamment pour la gestion sur plusieurs plateformes.[B4]

4.2.5 Java 8

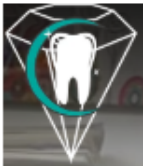
Java 8 est une version majeure du langage de programmation Java qui a introduit des changements significatifs, notamment l'introduction des expressions lambda, de l'API Streams, et des interfaces fonctionnelles, permettant une programmation plus fonctionnelle. Elle a également apporté une amélioration importante dans la gestion des dates et heures, ainsi que l'introduction de la classe Optional pour gérer les valeurs nulles de manière plus sûre. En résumé, Java 8 a apporté des améliorations majeures à la syntaxe et à la programmation fonctionnelle, faisant de lui un jalon important dans l'évolution du langage Java.[B6]

4.3 Présentation des interfaces de l'application

4.3.1 Interface de connexion

La page de connexion (figure 4.1) constitue la première interface visible de l'application. Elle permet aux utilisateurs autorisés (dentiste ou assistant) d'accéder au système à l'aide de leurs identifiants.

CABINET DIAMOND DENTAL



NOM D'UTILISTEUR

MOT DE PASSE

FERMERSE CONNECTER

FIGURE 4.1 – Interface connexion utilisateur.

4.3.2 Interface menu principal.

La page menu principal (figure 4.2) constitue la deuxième interface visible de l'application. Elle permet aux utilisateurs d'accéder aux différentes interfaces de l'application.

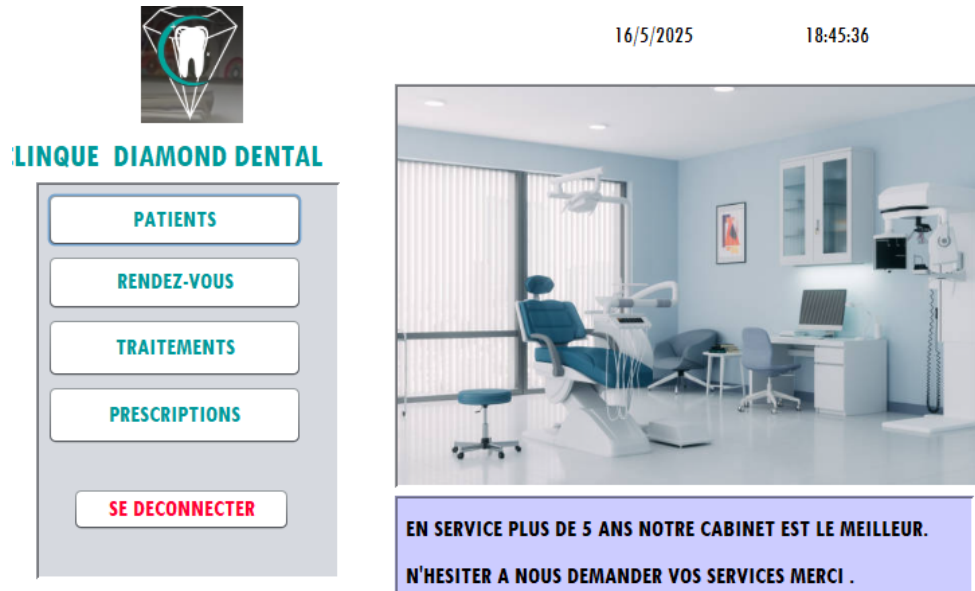


FIGURE 4.2 – Interface du menu principal de l'application.

4.3.3 Interface de gestion des patients

La page gestion des patients (figure 4.3) constitue la troisième interface visible de l'application. Elle permet d'ajouter, modifier, supprimer et consulter les dossiers des patients.

AJOUTER LES PATIENTS

NOM **DATE DE NAISSANCE** JJ/MOIN/ANNEE

TELEPHONE **SEXE**

ADRESSE **ALERGIE**

AJOUTER

MODIFIER

SUPPRIMER

ACTUALISER

MENU PRINCIPAL

CODE	NOM	TELEPHONE	ADRESSE	DATE	SEXE	ALERGIE
2	SAMY	0501024036	CHEHIMA	27/05/2016	HOME	LE SOLEIL
8	MALIKA	052562398	SEDDOUK	30/06/2001	FEMME	PANADOLE
10	RAMI RAP	0798618044	BEJAIA	12/08/2000	HOME	/

FIGURE 4.3 – Interface de gestion des patients.

4.3.4 Interface de gestion des rendez-vous

La page gestion des rendez-vous (figure 4.4) constitue la quatrième interface visible de l'application. Elle permet de planifier, déplacer ou annuler des consultations.

AJOUTER LES RENDEZ-VOUS

PATIENT

HEURE

ALERGIE

DATE

LES PATIENTS DOIVENT TOUJOURS PRESENTER LEURS TICKETS DE RENDEZ-VOUS MERCI.
N'HESITER A NOUS DEMANDER VOS SERVICES MERCI .

AJOUTER

MODIFIER

SUPPRIMER

ACTUALISER

MENUE PRINCIPAL

CODE	PATIENT	HEURE	DATE
2	SAMY	13H	25/05/2030
6	RAMI RAP	14H	01/06/2025

FIGURE 4.4 – Interface de gestion des rendez-vous.

4.3.5 Interface de gestion des traitements

La page gestion des traitements (figure 4.5) constitue la cinquième interface visible de l'application. Elle permet d'ajouter les différents traitements et services disponible dans le cabinet

AJOUTER DES TRAITEMENTS

NOM

MEDICAMENT

PRIX

DOSE

**FAUT TOUJOURS VERIFIER SI LE PATIENT
 N'EST PAS ALERGIE AVEC LE MEDICAMENT
 PRESCRIT POUR CE TRAITEMENT MERCI!**

CABINET DIAMOND DENTAL

AJOUTER

MODIFIER

SUPPRIMER

ACTUALISER

MENUE PRINCIPAL

CODE	NOM	PRIX	DOSE	MEDICAMENT
6	APPAREIL DENTAIRE	10000	/	LE CIR
7	PLOMBAGE	4000	1	COMPOSITE RÉSINE

FIGURE 4.5 – Interface de gestion des traitements.

4.3.6 Interface de gestion des prescriptions

La page gestion des prescriptions (figure 4.6) constitue la 6ème interface visible de l'application. Elle permet d'enregistrer le traitement appliquer pour le patient pour le bon suivi des patients au courant des consultations .

AJOUTER LES PRESCRIPTION

PATIENT

SAMY ▼ ■

ALERGIE

n'a pas

TRAITEMENT

PLOMBAGE ▼ ■

MEDICAMENT

COMPOSITE RÉSINE

4000

1

AJOUTER

MODIFIER

SUPPRIMER

ACTUALISER

MENUE PRINCIPAL

POUR LES PRESCRIPTIONS DEMANDE

AUX PATIENTS S'IL N'ONT PAS D'AUTRE TRUC A MODIFIER

CODE	PATIENT	ALERGIE	TRAITEMENT	MEDICAMENT	PRIX	DOSE
1	MALIKA	PANADOLE	NETTOYAGE DE...	GEL NETTOYANT	3000	3
3	RAMI RAP	/	APPAREIL DEN...	LE CIR	10000	
6	SAMY	N'A PAS	PLOMBAGE	COMPOSITE RÉ...	4000	1

FIGURE 4.6 – Interface de gestion des prescriptions.

4.4 Conclusion

La réalisation de notre application nous a permis de nous familiariser avec différents environnements de développement.

Au cours de ce chapitre nous avons présenté et expliqué d'une manière détaillée le fonctionnement de notre application à travers des interfaces explicatives.

Conclusion Générale

La conception et le développement d'une application desktop performante et intuitive requièrent une maîtrise approfondie des principes de programmation, d'architecture logicielle et des besoins métiers. Ce mémoire a porté sur la réalisation d'une application de gestion pour le cabinet dentaire "Diamond Dental Clinic", visant à moderniser et optimiser les processus administratifs et médicaux, tout en garantissant une expérience utilisateur fluide et sécurisée.

Notre démarche a débuté par une analyse approfondie des besoins du cabinet dentaire, incluant :

Une étude préliminaire pour identifier les différents acteurs qui interagissent avec le système à concevoir. Cette phase a été suivie par la spécification des besoins fonctionnels à travers le diagramme de cas d'utilisation et l'analyse des besoins en utilisant les diagrammes de séquence.

La phase de conception suit immédiatement la phase d'analyse. L'élément principal à livrer au terme de cette phase est le diagramme de classe ainsi que le schéma relationnel de la base de données.

Nous avons par la suite abordé la phase de réalisation en utilisant plusieurs outils de développement dédiés à la programmation d'applications desktop.

Ce projet a fait l'objet d'une expérience très bénéfique, il nous a conduit à nous confronter à des problèmes purement logiques mais aussi techniques, et sur le plan pédagogique il nous a permis d'approfondir les concepts vus en cours et de les appliquer au sein d'un travail concret pour en tirer des solutions, ajoutant à cela la mise en application des connaissances acquises dans la conception et la programmation.

En conclusion, cette application répond aux besoins actuels d'un cabinet dentaire tout en offrant une base évolutive pour l'ajout de fonctionnalités futures. Ce projet concrétise l'application des compétences acquises en génie logiciel et ouvre la voie à des innovations dans le domaine de la e-santé.

BIBLIOGRAPHIE

- [1] Diamond Dental Center. *Site officiel du cabinet dentaire*.
<https://diamonddentalcenterdz.com/>
- [2] Appvizer. *Guide complet sur Draw.io : création de diagrammes UML*. 2023.
<https://www.appvizer.fr/services-informatiques/diagramme/drawio>
- [3] Developpez.com. *NetBeans 8.0.1 : Renforcement du support HTML5/JavaScript/CSS*. 2014.
<https://www.developpez.com/actu/75376/NetBeans-8-0-1-renforce-le-support-de-HTML-5->
- [4] DataScientest. *MySQL : Tout comprendre sur ce SGBD relationnel*. 2022.
<https://datascientest.com/mysql-tout-comprendre>
- [5] Wikipédia. *SQLite : Base de données embarquée*. Mis à jour en 2023.
<https://fr.wikipedia.org/wiki/SQLite>
- [6] DataScientest. *Langage Java : Fonctionnalités et bonnes pratiques*. 2023.
<https://datascientest.com/langage-java-tout-savoir>
- [7] Université de Béjaia. *Conception d'un site web pour le centre d'imagerie médicale du CHU Khalil Amrane*. 2021.
<https://www.studocu.com/row/document/universite-de-bejaia/informatique/conception-e>
- [8] Scribd. *Projet de fin d'étude : Gestion d'un cabinet dentaire*. 2020.
<https://fr.scribd.com/document/577048863/pfe-gestion-cabinet-dentaire>
- [9] Google Search. *Cycle de vie du Processus Unifié (RUP)*. Consulté en 2023.
<https://www.google.fr/search?q=Cycle+de+vie+du+processus+unifié>
- [10] PIECHOCKI Laurent. *UML : Définition et principes de conception*. Dunod, 2005.

Résumé

L'informatique s'impose aujourd'hui comme un outil indispensable dans tous les secteurs d'activité, notamment dans le domaine médical où elle optimise la gestion des données et améliore la qualité des soins. Ce mémoire présente la conception et le développement d'une application desktop de gestion pour un cabinet dentaire, offrant une solution complète pour :

- La gestion des dossiers patients/gestion des traitements.
- La planification des rendez-vous/Édition des ordonnances.
- Le suivi des patients par l'ajout des prescriptions.

Ce travail a été réalisé en utilisant le processus de développement UP (Processus Unified) et le langage de modélisation UML(Unified ModelingLanguage) afin de schématiser la solution.

Les technologies clés utilisées incluent :

Java SE 8 (portabilité, sécurité), JavaFX (interface graphique moderne), SQLite (base de données embarquée),NetBeans (environnement de développement).

Mots clés : Application Desktop, UML, UP, SGBD, JAVA,SQLite.

Abstract

Information technology has become an indispensable tool across all professional sectors, particularly in the medical field where it optimizes data management and enhances care quality. This thesis presents the design and development of a desktop management application for dental practices, offering a comprehensive solution for :

- Patient files management / Treatment management.
- Appointment scheduling / Prescriptions editing.
- Patient follow-up through prescription additions.

This work was carried out using the UP (Unified Process) development process and the UML (Unified Modeling Language) modeling language to outline the solution.

Key technologies employed include : Java SE 8 (portability, security), JavaFX (modern graphical interface), SQLite (embedded database), NetBeans (development environment).

Keywords : Desktop Application , UML, UP, SGBD, JAVA,SQLite.