

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la technologie Houari Boumediene
U.S.T.H.B

Faculté D'Electronique et Informatique
Département d'informatique_



Projet GL2

Thème : CABINET DE PSYCHOLOGUE

Réalisé par :

1) AIT AMARA Mohamed L2 ISILB G1 181831072170

Email : mohm2733@gmail.com

2) SOUIDI Mohamed Amine L2 ISILB G3 181831044438

Email: mouhfc1970@gmail.com

Sous la direction du:

Dr.ABERBOUR Rima

Email : rima.aberbour@gmail.com

Année universitaire : 2020/2021

Résumé

L'objectif principal de ce projet consiste à concevoir et développer une application java de bureau pour la gestion d'un cabinet de psychologue qui lui permettra de diagnostiquer et traiter les patients atteints de dépression et de détresse principalement à cause du confinement et du covid.

Notre application contiendra 2 parties :

- Un espace pour la secrétaire du psychologue qui lui permet de gérer les rendez-vous et la liste des patients au cabinet et à distance via zoom, gmail et google meet.
- Un espace pour le psychologue qui lui permet de gérer les rendez-vous et la liste des patients en cas d'absence du secrétaire. Il lui permet aussi de faire ses consultations. Il lui permet aussi d'accéder à sa boîte email professionnelle ou bien à zoom pour faire des consultations à distance en cas de confinement totale.

L'application est codée en utilisant le langage Java avec une base de données SQLite.

Tandis que la partie graphique est développée avec le langage de script XML, la librairie JavaFX et son outil graphique SceneBuilder. On a utilisé notamment la technologie CSS et la librairie Swing pour nous permettre de décorer les interfaces d'une façon efficace et rapide.

À travers ce document, nous allons décrire en détail les différentes étapes de réalisation de ce projet.

Mots clés : Java, JavaFX, XML, SceneBuilder, Swing, CSS, SQLite.

Abstract

The main objective of this project is to design and develop a desktop java application for the management of a psychologist's practice that will allow him to diagnose and treat patients suffering from depression and distress mainly due to confinement and covid.

Our application will contain 2 parts:

- A space for the psychologist's secretary who allows her to manage appointments and the list of patients.
- A space for the psychologist who allows him to manage appointments and patient lists in the absence of the secretary. It also allows him to do his consultations. It also allows him to access his professional email box or zoom to make remote consultations in the event of total confinement.

The application is encoded using the Java language with an SQLite database.

While the graphic part is developed with the Xml scripting language, the JavaFX library and its SceneBuilder graphic tool. In particular, we used CSS technology and the Swing library to allow us to decorate the interfaces in an efficient and fast way.

Through this document, we will describe in detail the different stages of the realization of this project.

Keywords: Java, JavaFX, Xml, SceneBuilder, Swing, Css, SQLite.

Introduction général

L'informatique, cette science de travail rationnel de l'information est considérée comme le support des connaissances dans les domaines scientifiques, économiques et sociaux notamment à l'aide des machines automatique. Le monde connaît une avance technologique considérable dans tous les secteurs qui étudient les techniques du traitement automatique de l'information de l'entreprise et d'autres établissements. L'informatisation est donc le phénomène le plus important de notre époque. Elle s'immisce maintenant dans la plupart des objets de la vie courantes et ce, que ce soit dans l'objet proprement dit, ou bien dans le processus de conception ou de fabrication de cet objet. Et dans ce cas nous avons utilisé cette technologie de l'informatisation pour développer une application de gestion, la mémoire des cabinards était autrefois suffisante pour enregistrer les données relatives aux patients et servir l'exercice médical. Les données étaient rassemblées sous forme d'articles médicaux, de registres à visée épidémiologique, psychologique, nosologique et administrative, avec la multiplication des effets de l'environnement, de nos jours la bonne tenue d'un dossier exige des moyens informatiques.

Notre projet est réalisé au sein de l'université (USTHB) sous la supervision du :
Dr.ABERBOUR Rima.

Ce logiciel devrait mettre l'organisation et l'automatisation de la gestion d'un cabinet de psychologue, afin d'augmenter la fiabilité, l'efficacité de l'effort humain et faciliter les tâches pénibles au sein du cabinet.

Le présent document est organisé comme suit :

Dans le premier chapitre, nous présentons notre projet en étudiant son cadre général et la problématique avant de critiquer les solutions existantes pour dégager leurs insuffisances et proposer les orientations de notre solution future. La présentation de la solution souhaitée nous mène à bien analyser les différentes méthodologies de travail existantes afin de dégager celle la plus adéquate à notre projet.

Le deuxième chapitre consiste à formaliser les étapes préliminaires du développement

d'un système afin de rendre ce développement plus fidèle aux besoins du client qui sont divisés en besoins fonctionnels, besoins non fonctionnels et besoins techniques ainsi qu'une description du contexte.

Le troisième chapitre aborde la conception générique.

Le quatrième chapitre présente la conception détaillée et explique de manière approfondie les fonctionnalités de l'application.

Le cinquième chapitre présente l'environnement de développement matériel et logiciel que nous allons adopter et nous allons présenter quelques captures d'écrans de l'application.

Table of Contents

Résumé.....	2
Abstract.....	3
Introduction général.....	4
Chapitre 1 : Contexte et objectif.....	7
1 Introduction.....	7
2 Pourquoi l'informatisation ?.....	7
3 Problématiques générales dans un cabinet sanitaire ou médical sans (SI).....	7
4 À quoi sert un système information d'un cabinet médical /psychologique.....	8
5 Etudes de l'existant.....	8
6 Solution proposée.....	9
7 Repérage du domaine.....	9
8 Méthodologie de travail et modélisation.....	9
8.1 Le Processus Unifié (UP).....	10
8.2 Mise en pratique du processus 2TUP.....	10
9 Conclusion.....	12
Chapitre 2 : Etude préliminaire.....	12
1 introduction.....	12
2 Description du contexte.....	12
2.1 Définition des acteurs :.....	12
2.2 Diagramme de cas d'utilisation préliminaire (général) :.....	13
2.3 Diagramme de contexte statique.....	14
3 Besoins fonctionnels.....	14
4 Besoins non fonctionnels.....	15
4.1 Exigences de qualité.....	15
4.2 Exigences de performance.....	15

5 Besoins techniques.....	15
5.1 Pourquoi Java ?.....	15
6 Conclusion.....	16
Chapitre 3 : Conception générique.....	16
1 Introduction.....	16
2 Le langage UML.....	16
3 Les diagrammes d'UML.....	16
3.1 Six diagrammes structurels :.....	16
3.2 Sept diagrammes comportementaux :.....	17
4 Modèles d'UML utilisés.....	17
Chapitre 4 : Conception détaillée.....	18
1 Capture des besoins fonctionnels.....	18
1.1 Diagramme de cas d'utilisation.....	18
1.2 Diagrammes de séquence système.....	25
2 Analyse.....	29
2.1 Diagramme de classe.....	29
2.2 Diagramme de package.....	32
2.3 Diagramme d'activité.....	33
3 Capture des besoins techniques.....	33
3.1 Diagramme de déploiement.....	33
Chapitre 5 : Implémentation et réalisation.....	34
1 Introduction.....	34
2 Environnement de développement.....	34
2.1 Environnement matériel.....	34
2.2 Environnement logiciel.....	35
3 Présentation de la solution.....	36
3.1 Interface d'Authentification.....	36
3.2 Le Menu principale :.....	37
3.3 Interface « Gérer les Consultations » :.....	38
3.4 Interface « Gérer les Patients » :.....	39
3.5 Interface « Gérer les rendez-vous » :.....	39
3.6 Interface « Consulter les statistiques » :.....	40
3.7 Interface « Gérer les Comptes » :.....	41

4 Conclusion.....	41
Conclusion générale.....	41

Chapitre 1 : Contexte et objectif

1 Introduction

Au cours de ce chapitre, nous allons introduire notre projet en étudiant son cadre général et la problématique qui nous a poussé à réaliser cette application. Nous allons critiquer les solutions existantes pour dégager leurs insuffisances et proposer les orientations de notre solution future.

La présentation de la solution souhaitée nous mène à bien analyser les différentes méthodologies de travail existantes afin de dégager celle la plus adéquate à notre projet.

2 Pourquoi l'informatisation ?

Dès les premiers temps, les médecins s'orientaient à un objectif bien déterminé, celui-ci se résumait à comment travailler avec les patients à l'aide de moyens faciles et rapides. Mais malgré tout, il reste des difficultés, parmi lesquelles :

L'archivage des dossiers des patients sur des feuilles peut engendrer leur perte et perdre beaucoup de temps pour trouver un dossier ; la modification d'un élément dans un dossier peut obliger le médecin à refaire un nouveau dossier. A cause de ces difficultés, les médecins préfèrent s'approprier des bénéfices liés à l'informatisation de la gestion de leurs cabinets médicaux dans le but de simplifier, sécuriser et minimiser leur travail.

3 Problématiques générales dans un cabinet sanitaire ou médical sans (SI)

 Recherche difficile sur les registres.

 Perte de temps.



Détérioration des archives à cause de leur utilisation trop fréquente.

Difficulté de stockage.

Risque de perte des archives.

Mauvaise codification

Problèmes dans le calcul des statistiques.

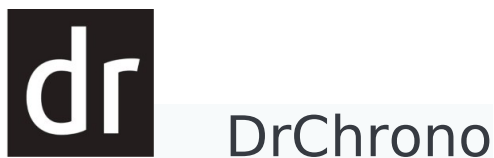
4 À quoi sert un système information d'un cabinet médical /psychologique

Il assure la gestion administrative et médicale, et pour cela on a fait introduire l'outil informatique qui gère l'ensemble des enregistrements sans redondance de l'information et un traitement automatique tout en assurant la sécurité, la fiabilité et la solution qu'apporte notre logiciel.

Cela consiste à se débarrasser de plusieurs problèmes (classement des dossiers, grande masse d'information circulaire, perte de temps dans les traitements manuels ,...etc.)

5 Etudes de l'existant

Il existe tellement d'application de gestion qui gère les cabinets de divers domaines à savoir :



Points forts en général :

- Ces applications sont très accessibles sur le net notamment des les grandes plateformes de vente sur le web.
- Maintenabilité à long terme.

Points faibles en général :

- Ces applications sont payantes.
- Coût de Maintenableté très élevé.
- Application non personnalisable.
- Ne sont pas mise à jour à la situation sanitaire actuel.
- Ne propose pas des consultations à distances.

6 Solution proposée

Après avoir détaillé notre problématique et bien analysé les différentes solutions existantes en Algérie et à l'étranger, la question suivante doit se poser naturellement:

Est ce que les applications de gestion de cabinet existantes que ça se soit médicales ou psychiques peuvent représenter une solution qui répond à notre problématique?

Les applications disponibles sur le marché sont globales et parfois payantes ce qui est a créé un grand problème en Algérie vu que l'achat en ligne n'est guère facile dans notre pays. Cependant, nous avons besoin d'une solution spécifique, moins couteuse et qui répond parfaitement aux besoins spécifiés par les clients tout en tenant compte de la situation sanitaire actuelle. Par conséquent, nous avons pensé à analyser et développer une application de bureau open source gratuite, personnalisable à grand échèle et qui facilite l'interaction Psychologue/client même à distance.

Cette application peut être modifié rapidement pour être destinée aux différents cabinets médicaux.

7 Repérage du domaine

Dans un cabinet de psychologue on trouve un psychologue et une secrétaire dont chacun a des activités bien précises.

Activités de la secrétaire :

La secrétaire s'occupe de la gestion des rendez-vous au cabinet et à distance.

Activités de médecin :

Le psychologue possède le droit d'accéder aux différentes rubriques, Il peut gérer les patients , les consultation , les rendez-vous, les comptes et il peut aussi consulter les statistiques liées à son cabinet et à cause de la situation sanitaire actuelle, on s'est appuyer aussi sur des des applications existantes comme : gmail, google meet, zoom...pour permettre au psychologue de professer à distance.

8 Méthodologie de travail et modélisation

Une méthodologie de développement est un cadre utilisé pour structurer, planifier et contrôler le développement d'une application. C'est le fait de modéliser un système avant sa réalisation pour bien comprendre son fonctionnement et assurer sa cohérence. Un modèle est ainsi un facteur de réduction des coûts et des délais. Il est donc indispensable pour assurer un bon niveau de qualité de produit dont la maintenance est efficace.

Nous présentons la principale méthodologie de développement dédiée aux démarches de conception orientées objet pour monter le processus suivi dans ce travail.

8.1 Le Processus Unifié (UP)

Le Processus Unifié est un processus de développement logiciel organisée suivant les quatre phases suivantes:

1. La phase d'initialisation permet de décider la poursuite du projet ou son arrêt.
2. La phase d'élaboration poursuit trois objectifs principaux en parallèle : (identifier et décrire, construire, lever les risques)
3. La phase de construction consiste à concevoir et implémenter l'ensemble des éléments opérationnels.
4. En fin, la phase de transition permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux.

8.1.1 Conclusion et choix de la Méthodologie à suivre :

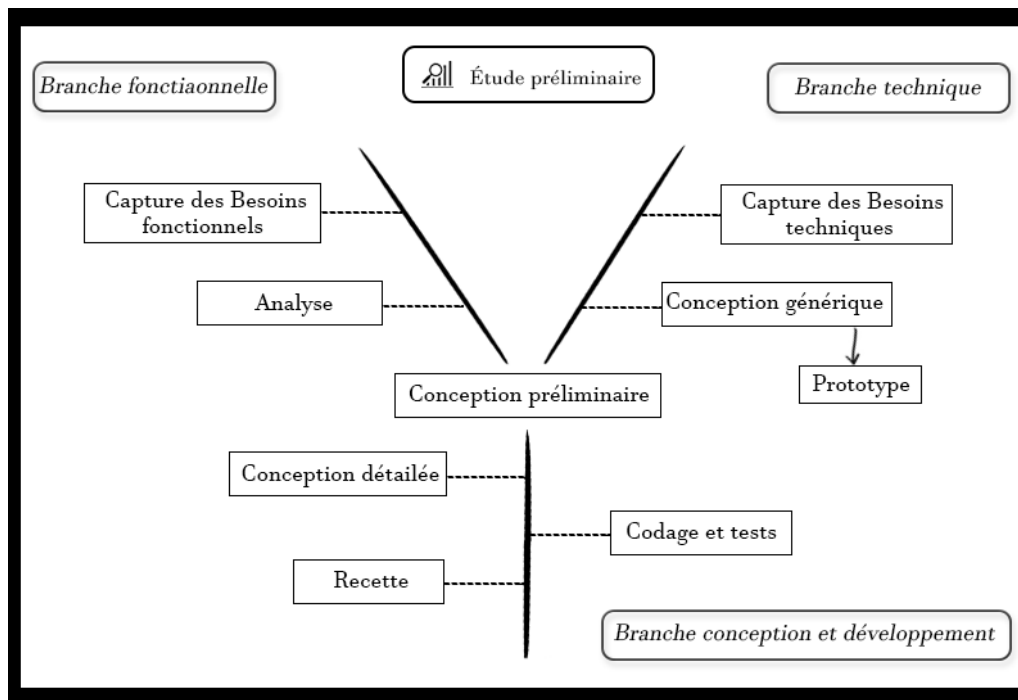
Notre projet est basé sur un processus de développement bien défini qui va de la détermination des besoins fonctionnels attendus du système jusqu'à la conception et le codage final. C'est pour cela qu'on est besoin d'un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels tout en commençant par une étude préliminaire. Notre choix s'est alors porté vers la méthode 2TUP vu qu'elle est caractérisée par une approche nouvelle et originale et qu'elle respecte le cadre de notre projet.

Nous adaptons alors le processus 2TUP pour la suite de notre travail.

8.2 Mise en pratique du processus 2TUP

2 Tracks Unified Process est un processus unifié (c'est-à-dire construit sur UML, itératif, centré sur l'architecture et conduit par les cas d'utilisation). Ce processus de développement logiciel est caractérisé par le suivant :

- Itératif et incrémental : Le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global.
- Centré sur l'architecture : Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées.
- Piloté par les risques : Les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible.
- Conduit par les cas d'utilisation : Les cas d'utilisation du futur système sont identifiés et décrits avec précision.



Les étapes du processus 2TUP

Le principe fondateur du 2TUP est que toute évolution imposée à un logiciel peut se décomposer

et se traiter parallèlement, suivant un axe fonctionnel et un axe technique. La réalisation du logiciel consiste à fusionner les résultats de ces deux branches du processus.

Selon le processus 2TUP que nous suivons, nous passons obligatoirement par les phases suivantes à la suite de notre travail:

1. L'étude préliminaire qui contient une description du service et les cas d'utilisation principaux, c'est une première version de la spécification générale.
2. La capture des besoins fonctionnels, qui définit le quoi faire à travers une spécification générale qui décrit le service à développer d'un point de vue fonctionnel et une spécification détaillée qui précise les traitements qui concernent chaque scénario des cas d'utilisation présent en spécification générale tout en respectant les contraintes fonctionnelles et non fonctionnelles.
3. L'Analyse où on effectue simultanément l'étude des données et l'étude des traitements à effectuer.
4. La capture des besoins techniques qui permettent de satisfaire les contraintes techniques présentes dans le cahier des charges et donc répondre aux attentes de client.
5. La conception générique qui définit le comment faire.
6. La conception préliminaire qui liste les exigences techniques à partir de l'étude préliminaire et définit l'architecture technique et les choix d'implémentation technique.
7. La conception détaillée qui précise l'implémentation technique de l'application. Elle consiste en la fusion de la spécification détaillée et de la conception générique, pour déterminer comment faire le quoi faire dans le détail. On y trouve le schéma de base de données, les diagrammes de classes et les diagrammes de séquence supplémentaires qui détaillent les interactions entre les composants du logiciel.
8. Le Codage et les Tests qui décrivent comment est l'application est réalisée et installée

sur un environnement d'exécution et comment la stratégie de validation est effectuée.

9 Conclusion

Dans ce premier chapitre, nous avons fait une étude de l'existant nous permettant de bien entamer notre projet et nous avons clôturé par une présentation de la méthodologie 2TUP que nous allons personnaliser un petit peu pour l'adopter à nos besoins et notre stratégie de développement de notre projet. Le chapitre suivant sera consacré à l'étude préliminaire de notre projet.

Chapitre 2 : Etude préliminaire

1 introduction

Comme les bonnes questions représentent la moitié de la réponse dans la plupart des domaines, en informatique une bonne spécification des besoins est primordiale. En effet, elle représente le travail le plus délicat et le plus significatif, mais elle-même repose sur une bonne spécification des besoins. Dans cette section nous allons comprendre le contexte du système, il s'agit de déterminer les fonctionnalités et les acteurs et d'identifier les cas d'utilisation initiaux. Ainsi que la spécification des besoins.

2 Description du contexte

2.1 Définition des acteurs :

Le Rôle d'un acteur représente une entité externe qui interagit directement avec le système étudié, en échangeant de l'information (en entrée et en sortie). On trouve les acteurs en observant les utilisateurs directs du système, les responsables de la maintenance, ainsi que les autres systèmes qui interagissent avec lui.

Dans notre projet, nous pouvons distinguer deux acteurs principaux :

- ❖ Secrétariat
- ❖ Psychologue

Utilisateur	Leur Rôle

Secrétariat	Gestion des rendez-vous Utiliser l'espace internet
Psychologue	Gestion des Comptes Gestion des patients Gestion des rendez-vous Gestion les Consultation Consulter les statistiques Utiliser l'espace internet

Tableau 1:tableau des rôles de psy et secrétaire

2.2 Diagramme de cas d'utilisation préliminaire (général) :

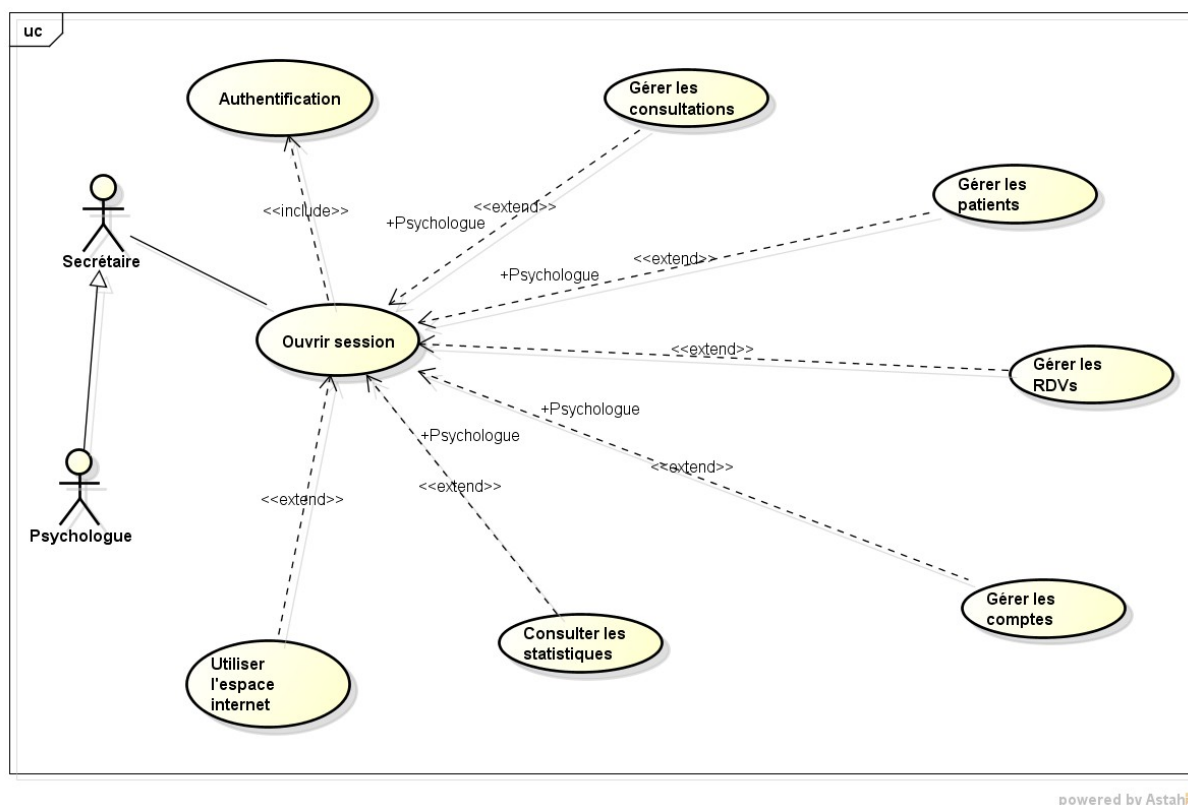


Figure : Diagramme de cas d'utilisation général

Le déroulement de l'application est représenté par les cas d'utilisation suivants :

Authentification : Assure la sécurité d'accès à l'application et spécifie le type de l'utilisateur.

Gestion des Comptes : modifier les données de l'utilisateur.

Gestion des patients : Ajouter, modifier, supprimer et rechercher une Patient.

Gestion des rendez-vous : Ajouter, modifier, supprimer, rechercher et ajouter un patient à la base de données à partir d'un rendez-vous.

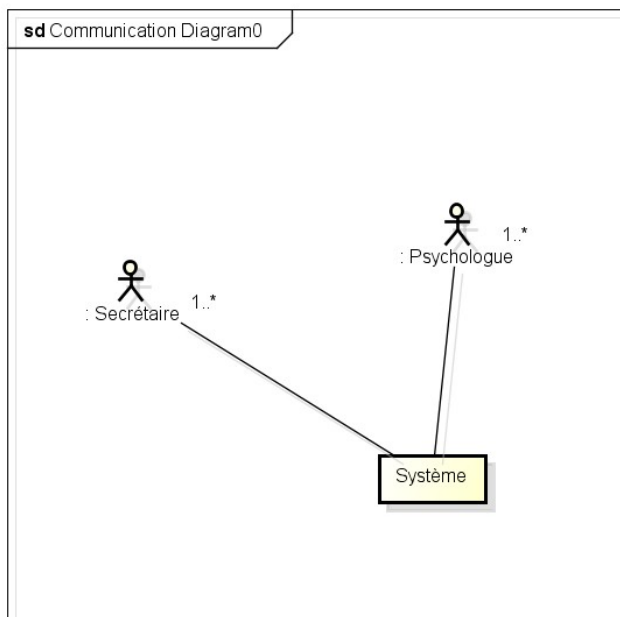
Gestion les Consultation : Ajouter, modifier, supprimer et rechercher une Consultation.

Consulter les statistiques: Nombre de patients/mois & revenu des consultations/mois.

Utiliser l'espace internet : Utiliser ZOOM, Google meet et Gmail.

2.3 Diagramme de contexte statique

Nous présentons ci-dessous le diagramme de contexte statique. Nous n'avons pas représenté le diagramme de contexte dynamique vu qu'il y a pas beaucoup d'acteur et que les messages entre le système et les acteurs seront représenter avec détails dans le reste des chapitres.



3 Besoins fonctionnels

Les besoins fonctionnels représentent les principales fonctionnalités du système. Ces besoins proviennent généralement des utilisateurs du système. Cette application va permettre

Au psychologue :

La gestion des Comptes

La gestion des patients

La gestion des rendez-vous

La gestion les Consultation

La consultation des statistiques

L'utilisation de l'espace internet

Au Secrétaire :

Gestion des rendez-vous

Utiliser l'espace internet

4 Besoins non fonctionnels

4.1 Exigences de qualité

Ergonomie sobre et efficace : Utiliser une application de gestion ne doit pas prendre beaucoup de temps ou demander une maîtrise de mathématiques !

L'ergonomie de notre application facilitera au maximum la démarche à l'aide d'une présentation claire et intuitive.

Les applications trop riches et trop complexes n'incitent pas à l'achat et l'utilisation, car ils demandent un effort intellectuel important et non souhaité.

4.2 Exigences de performance

Besoin de sécurité : L'application doit garantir à l'utilisateur connecté l'intégrité et la confidentialité de ses données. La sécurité du système est assurée par l'authentification des clients par un login et un mot de passe crypté.

Performance : L'application doit fournir tous les statuts et informations en temps réel et d'une manière optimale.

La flexibilité : Une ouverture sur d'autres domaines serait un atout majeur pour notre application. Ce point ne doit pas être mis à côté lors de la conception de l'architecture du projet.

L'intégrité : Le système doit savoir comment traiter les échecs des interfaces, faire la capture des différentes erreurs d'entrées-sorties, effectuer le traitement des mauvaises données sans oublier l'intégrité des informations dans les tables de la base de données et des interfaces

La portabilité : L'application doit s'exécuter sans problème sur tous les navigateurs. Il est ainsi primordial qu'elle soit compatible avec les différents système d'exploitation.

5 Besoins techniques

Dans cette section on va juste dire pourquoi on a utilisé le langage java car on va détaillé les besoins techniques ultérieurement.

5.1 Pourquoi Java ?

Pour plusieurs raisons, nous allons adopter le langage Java pour développer notre application de gestion. Tout d'abord, Java est gratuit et ne nécessite aucune licence d'utilisation. Java est le langage de programmation le plus utilisé dans les applications

bureau de gestion. Il existe une communauté de développeurs très active qui rend disponibles des dizaines de milliers de librairies Java de grande qualité ainsi qu'une vaste quantité de documentation et tutoriels pour le bénéfice de chacun. Ces ressources facilitent notre travail et réduisent notre temps d'exécution.

6 Conclusion

Durant ce chapitre, Nous avons non seulement exposé la mise en pratique du processus 2TUP, mais nous avons également capturé les différents besoins de notre projet. Nous avons classé ces besoins en besoins fonctionnelles, non fonctionnelles et techniques. Après l'analyse des exigences et le choix des techniques a utilisé, Il est temps de passer à la conception de notre application dans le chapitre suivant.

Chapitre 3 : Conception générique

1 Introduction

Dans cette partie nous traitons l'aspect conceptuel de notre application. Pour la conception et la réalisation de cette dernière, nous nous appuyons sur le formalisme UML basé sur les diagrammes et offrant une flexibilité marquante.

La phase de conception permet de décrire de manière non ambiguë, le plus souvent en utilisant un langage de modélisation, le fonctionnement futur du système, afin d'en faciliter la réalisation.

2 Le langage UML

Pour faciliter notre tâche nous avons recours au langage de modélisation unifié (UML : Unified Modelling Language) c'est une notation qui permet de modéliser un problème de façon standard. Ce langage qui est né de la fusion de plusieurs méthodes existantes auparavant est devenu une référence en terme de modélisation objet, UML est caractérisé par:

- C'est un langage formel et normalisé.
 - Il permet le gain de précision, encourage l'utilisation d'outils et constitue à cet effet un gage de stabilité.
 - UML est un support de communication performant.
 - Il cadre l'analyse et facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

3 Les diagrammes d'UML

UML 2 s'articule autour de treize types de diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces types de diagrammes sont répartis en deux grands groupes :

3.1 Six diagrammes structurels :

Diagramme de classes : Il montre les briques de base statiques : classes, associations, interfaces, attributs, opérations, généralisations, etc.

Diagramme d'objets : Il montre les instances des éléments structurels et leurs liens à l'exécution.

Diagramme de packages : Il montre l'organisation logique du modèle et les relations

entre packages.

Diagramme de structure composite : Il montre l'organisation interne d'un élément statique complexe.

Diagramme de composants : Il montre des structures complexes, avec leurs interfaces fournies et requises.

Diagramme de déploiement : Il montre le déploiement physique des 'artefacts' sur les ressources matérielles.

3.2 Sept diagrammes comportementaux :

. *Diagramme de cas d'utilisation* : Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.

. *Diagramme de vue d'ensemble des interactions* : Il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.

. *Diagramme de séquence* : Il montre la séquence verticale des messages passés entre objets au sein d'une interaction.

. *Diagramme de communication* : Il montre la communication entre objets dans le plan au sein d'une interaction.

. *Diagramme de temps* : fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.

. *Diagramme d'activité* : Il montre l'enchaînement des actions et décisions au sein d'une activité.

. *Diagramme d'états* : Il montre les différents états et transitions possibles des objets d'une classe.

4 Modèles d'UML utilisés

Les diagrammes qu'on a montré dans la partie précédente sont des moyens de description des objets ainsi que des liens qui les relient. Nous n'utilisons pas les treize types de diagrammes proposés par UML 2, mais seulement un tiers parmi eux en insistant particulièrement sur les diagrammes suivants :

.Diagramme de cas d'utilisation

.Diagramme de séquence

.Diagramme de package

.Diagramme de classes

.Diagramme d'activité

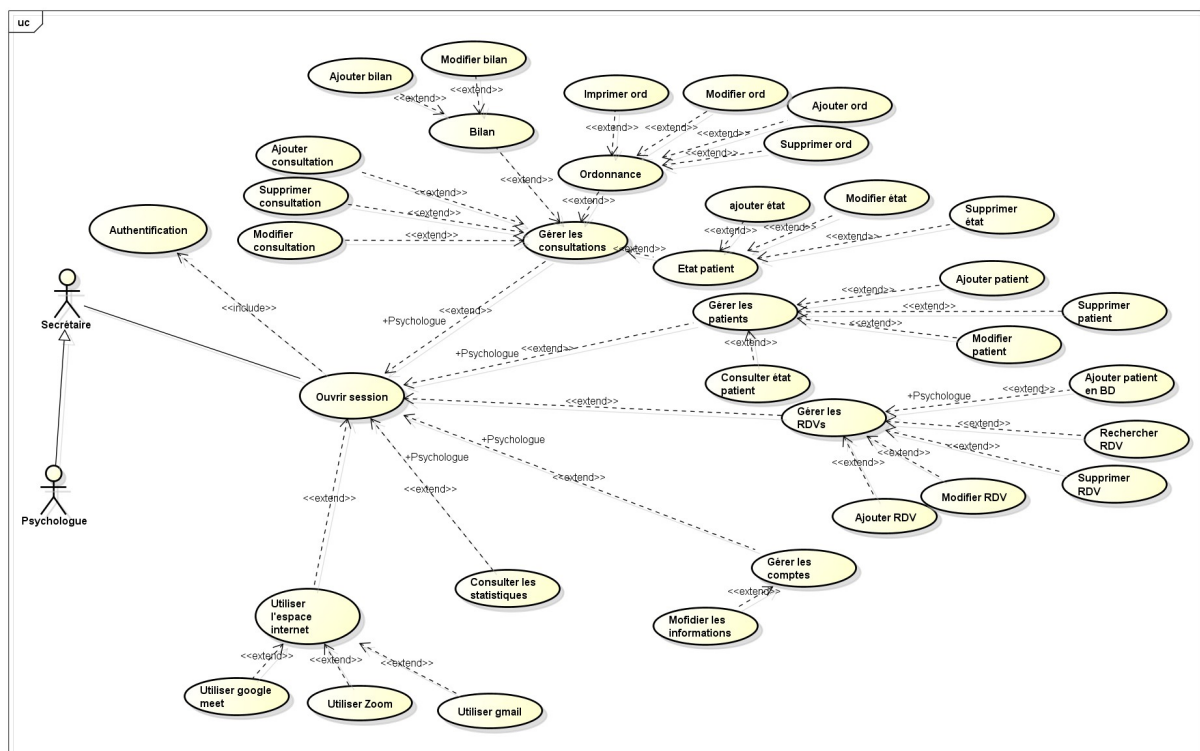
.Diagramme de déploiement

Cette limitation est largement suffisante pour la plupart des projets et particulièrement pour notre travail.

Chapitre 4 : Conception détaillée

1 Capture des besoins fonctionnels

1.1 Diagramme de cas d'utilisation



powered by Astah

1.1.1 Description détaillée des cas d'utilisation

Dans ce qui suit, on a représenté les descriptions des cas d'utilisation les plus importants et on n'a pas représenté tous car il y a plusieurs cas d'utilisation qui se traitent de la même façon comme : ajouter patient et ajouter état.

1.1.1.1 Description textuelle de cas d'utilisation « authentification »

Titre	Authentification
Résumé	chaque utilisateur doit taper son propre nom d'utilisateur et son mot de passe pour accéder à l'interface qui le concerne
Acteur	psychologue, secrétaire.
Précondition	l'utilisateur n'est pas authentifié donc il ne peut pas accéder à l'application.
Postcondition	l'utilisateur est authentifié, il accède à l'interface qui le concerne.
Description des scénarios	<p>Scénario nominal :</p> <ol style="list-style-type: none">1. L'utilisateur tape son propre nom d'utilisateur et son mot de passe.2. L'utilisateur demande l'accès.3. Le système vérifie la validité des coordonnées de l'utilisateur.4. L'utilisateur va accéder à la session qui le concerne. <p>Scénario alternatif :</p> <p>A1 : l'enchaînement démarre à l'étape 3 du scénario nominal. Le système indique la non validité des coordonnées de l'utilisateur, alors il ne peut pas accéder à sa propre session.</p> <p>L'enchaînement reprend à l'étape 1 du scénario nominal.</p>

1.1.1.2 Description textuelle de cas d'utilisation « Ajouter patient »

Titre	Ajouter patient
Résumé	Ajouter un patient en BD
Acteur	psychologue.
Précondition	L'utilisateur est connecté, le patient n'est

	pas créée
Postcondition	patient créée
Description des scénarios	<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1. L'utilisateur demande le formulaire d'ajout d'un patient. 2. le système affiche le formulaire. 3. L'utilisateur remplit le formulaire. 4. Le système enregistre la demande d'ajout. <p>Scénario alternatifs :</p> <p>A : Les champs du formulaire ne sont pas remplis L'enchaînement démarre au point 4 du scénario nominal</p> <ol style="list-style-type: none"> 5. Le système indique que le formulaire n'est pas bien rempli et le scénario nominal reprend au point 3

1.1.1.3 Description textuelle de cas d'utilisation « Modifier patient »

Titre	Modifier patient
Résumé	Mise à jour des infos du patient
Acteur	psychologue.
Précondition	L'utilisateur est connecté, le patient est créée
Postcondition	patient mise à jour
Description des scénarios	<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1 L'utilisateur demande le formulaire de fiche patient 2 Le système affiche le formulaire 3 L'utilisateur demande de rechercher l'identifiant du patient 4 Le système effectue la recherche 5 Le système envoie les informations du patient concerné 6 L'utilisateur effectue les modifications nécessaires 7 L'utilisateur valide les modifications 8 Le système enregistre la mise à jour <p>Enchaînement d'exception :</p>

	<p>E1 : L'utilisateur annule la demande de modification</p> <p>L'enchaînement E1 démarre au point 6 du scénario nominal</p> <p>L'utilisateur annule la modification</p> <p>Le système enregistre la page sans modification</p>
--	--

1.1.1.4 Description textuelle de cas d'utilisation « Rechercher patient »

Titre	Rechercher patient
Résumé	ce cas d'utilisation permet de trouver rapidement les informations nécessaires
Acteur	psychologue.
Précondition	Patient non trouvée
Postcondition	Patient recherché
Description des scénarios	<p>Scénario nominal :</p> <p>1 L'utilisateur lance une recherche par identifiant.</p> <p>2 Le système vérifie l'identifiant.</p> <p>3 Le système affiche les résultats de la recherche.</p> <p>Enchaînement alternatifs:</p> <p>A : Le système n'a pas trouvé l'identifiant recherché.</p> <p>L'enchaînement démarre au point 2 du scénario nominal.</p>

1.1.1.5 Description textuelle de cas d'utilisation « Gérer les consultations »

Titre	Gérer les consultations
Résumé	C'est le médecin qui se charge de la gestion des consultations qui se base sur l'enregistrement des ordonnances, des bilans de liaison, des maladies et des informations sur la consultation.
Acteur	psychologue.
Précondition	consultation non enregistrée
Postcondition	consultation enregistrée

1.1.1.6 Description textuelle de cas d'utilisation « Ordonnance »

Titre	Ordonnance
Résumé	Etablir l'ordonnance
Acteur	psychologue.
Précondition	ordonnance non existante
Postcondition	ordonnance ajoutée
Description des scénarios	Scénario nominal : 1 Le système injecte les infos(patient, consultation) dans le modèle. 2 Le médecin prescrit les traitements 3 Le médecin confirme le traitement 4 le système valide l'ajout 5 Le système effectue l'enregistrement

1.1.1.7 Description textuelle de cas d'utilisation « Ajouter RDV »

Titre	Ajouter RDV
Résumé	C'est la secrétaire qui se charge de la gestion des RDV, elle consulte le planning des visites et selon la disponibilité du médecin elle ajoute un nouveau RDV. En outre, elle peut modifier ou annuler un RDV selon le choix du patient.
Acteur	Secrétaire
Précondition	date de RDV non renseignée
Postcondition	date RDV fixée
Description des scénarios	Scénario nominal : 1 L'utilisateur choisit le patient 2 L'utilisateur demande d'ajouter un nouveau RDV. 3 Le système affiche la formule 4 L'utilisateur choisit une date et une heure selon la disponibilité 5 Le système vérifie la disponibilité 6 Le système confirme l'ajout 7 Le système enregistre l'ajout Scénario d'exception :

	<p>E1 : si la date des RDV est non disponible, le système va afficher un message d'erreur car l'opération est impossible.</p> <p>L'enchaînement reprend à l'étape 3 du scénario nominal.</p>
--	--

1.1.1.8 Description textuelle de cas d'utilisation « Supprimer RDV »

Titre	Supprimer RDV
Résumé	Supprimer le RDV de la BD
Acteur	Secrétaire
Précondition	RDV existant.
Postcondition	RDV non existant.
Description des scénarios	<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1 La secrétaire demande la liste des RDVs. 2 Le système affiche la liste. 3 La secrétaire cherche le RDV demandé(par n° fiche patient). 4 Le système effectue la recherche. 5 Le système affiche le RDV demandé. 6 Le secrétaire supprime le RDV. 7 Le système enregistre la suppression. 8 Le système confirme la suppression. <p>Scénario d'exception :</p> <p>E1 : Si les informations saisies par l'utilisateur sont erronées, le système affiche un message d'erreur.</p> <p>L'enchaînement démarre à l'étape 4 du scénario nominal.</p> <p>L'enchaînement reprend à l'étape 3 du scénario nominal.</p>

1.1.1.9 Description textuelle de cas d'utilisation « Modifier RDV »

Titre	Modifier RDV
Résumé	Mettre à jour le RDV dans la BD
Acteur	Secrétaire
Précondition	RDV avec la date initiale.
Postcondition	RDV modifié.
Description des scénarios	<p>Scénario nominal :</p> <ol style="list-style-type: none"> 1 La secrétaire demande la liste des RDVs. 2 Le système affiche la liste. 3 La secrétaire cherche le RDV demandé(par n° fiche patient). 4 Le système effectue la recherche. 5 Le système affiche le RDV demandé. 6 Le secrétaire modifie le RDV. 7 Le système enregistre la modification. 8 Le système confirme la modification. <p>Scénario d'exception :</p> <p>E1 : Si les informations saisies par l'utilisateur sont erronées, le système affiche un message d'erreur.</p> <p>L'enchaînement démarre à l'étape 4 du scénario nominal.</p> <p>L'enchaînement reprend à l'étape 3 du scénario nominal.</p>

1.1.1.10 Description textuelle de cas d'utilisation « Modifier information »

Titre	Modifier information
Résumé	Mettre à jour le RDV dans la BD
Acteur	Psychologue
Précondition	utilisateur avec les informations initiaux.
Postcondition	utilisateur avec les informations modifiées.
Description des scénarios	<p>Scénario nominal :</p> <p>Le médecin tape le nom de l'utilisateur et le mot de passe .</p> <p>Le système enregistre la modification.</p>

1.2 Diagrammes de séquence système

1.2.1 Diagramme de séquence système « authentication »

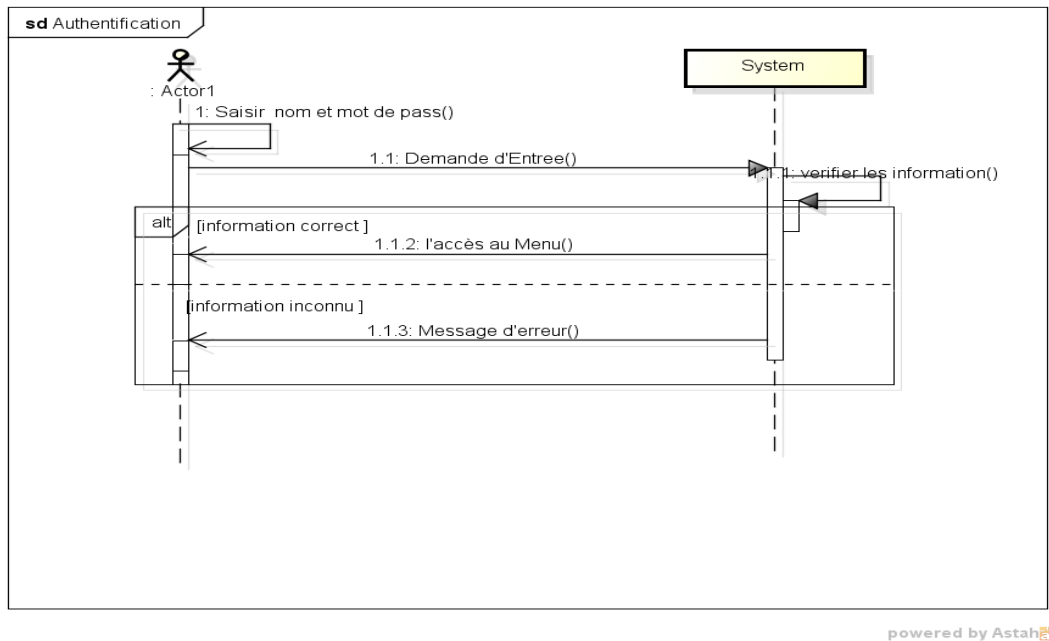


Figure : Diagramme de séquence système « authentication »

1.1.2 Diagramme de séquence système « ajouter un RDV »

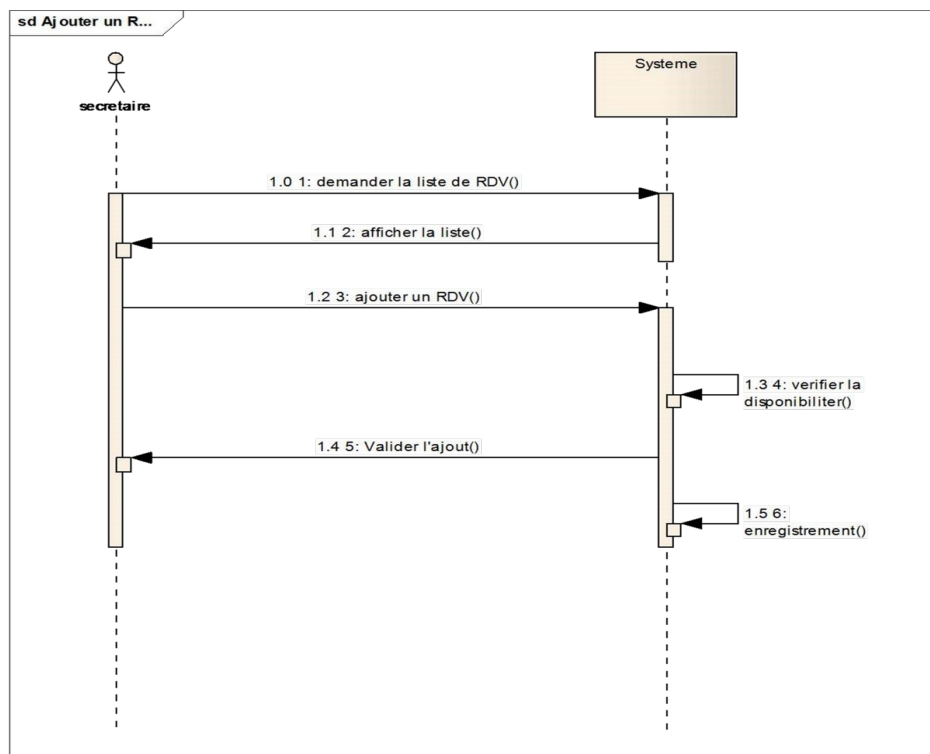


Figure : Diagramme de séquence système « ajouter un RDV »

1.1.3 Diagramme de séquence de conception :

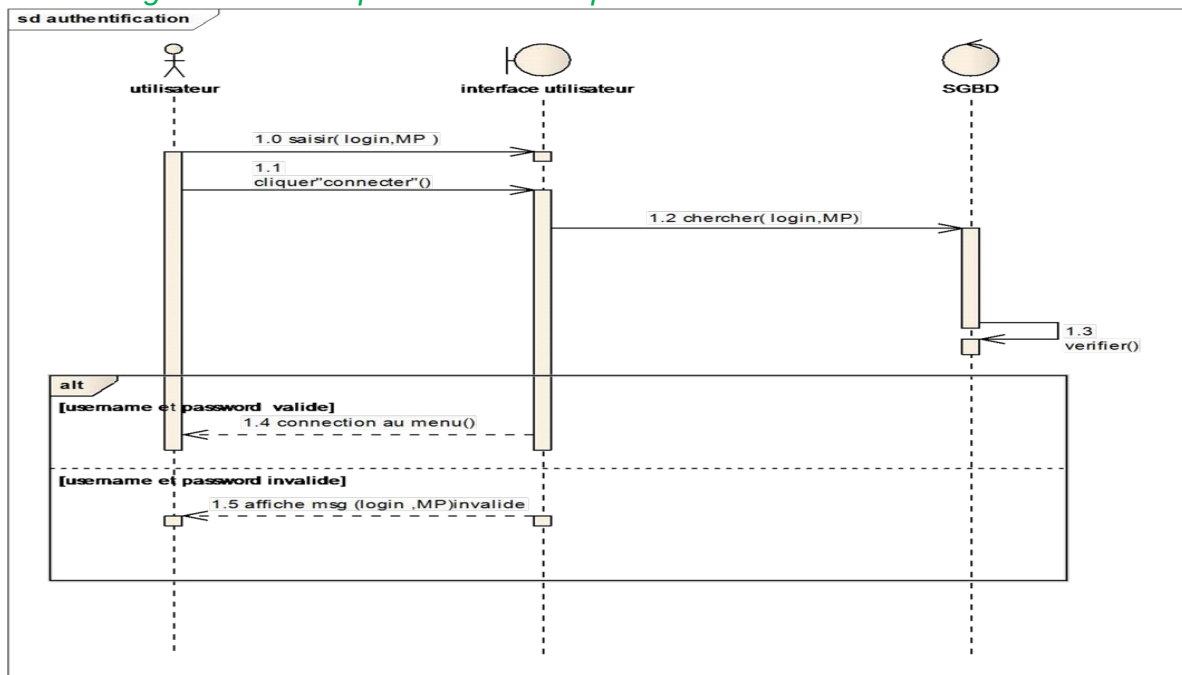


Figure : Diagramme de séquence « Authentification »

1.1.4 Diagramme de séquence système « ajouter patient »

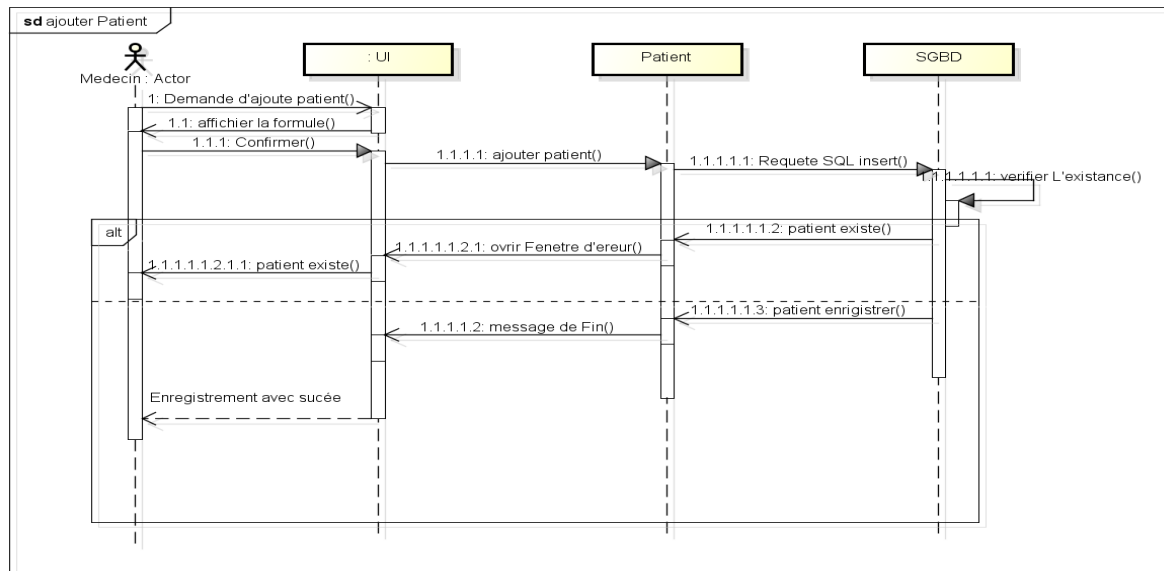
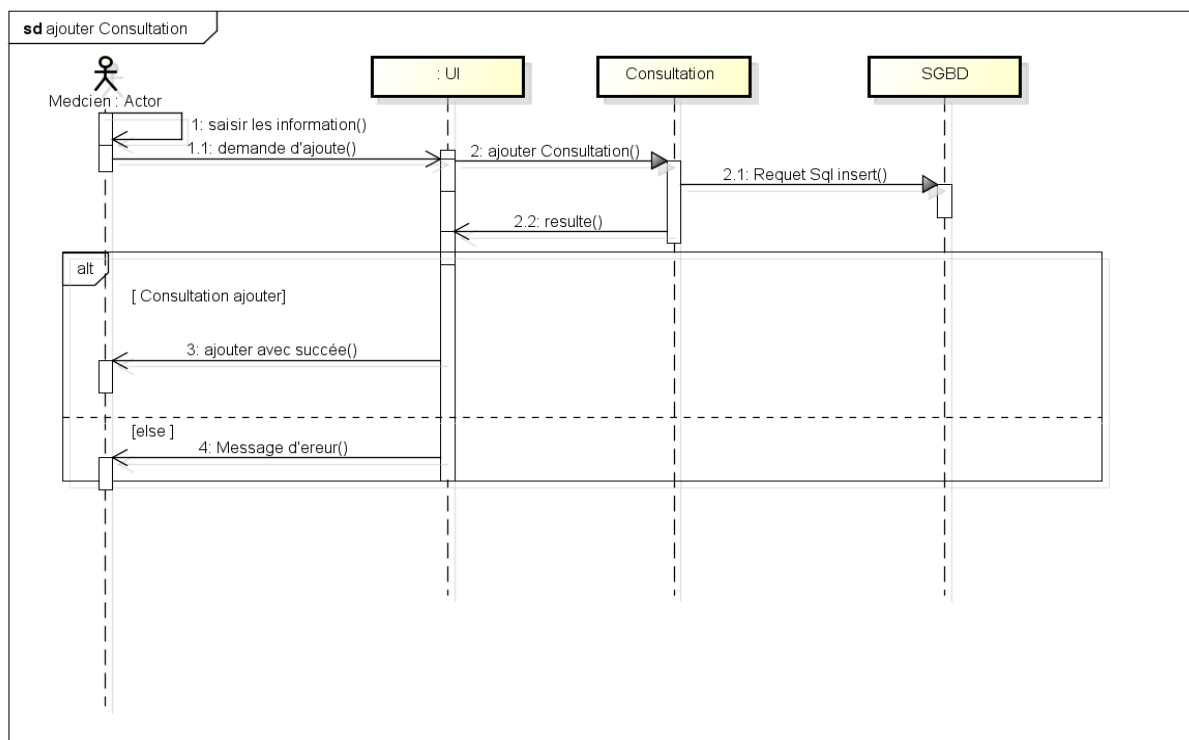


Figure : Diagramme de séquence système « ajouter patient »

1.1.5 Diagramme de séquence système « ajouter Consultation »

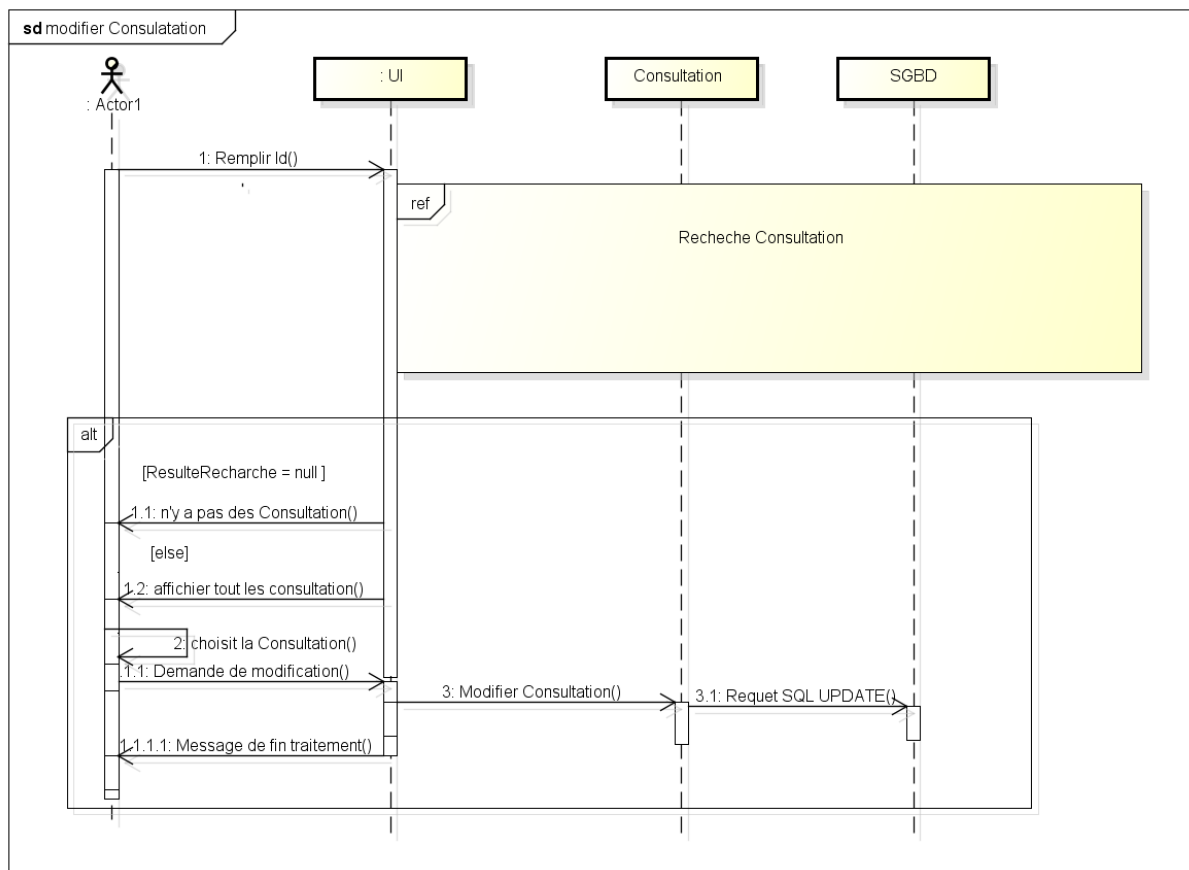


powered by Astah

Figure : Diagramme de séquence système « ajouter Consultation »

La figure suivante illustre le diagramme de séquence relatif au cas d'utilisation « modifier Consultation » :

1.1.6 Diagramme de séquence système «Modifier Consultation»



powered by Astah

Figure : Diagramme de séquence système «Modifier Consultation»

1.1.7 Diagramme de séquence système « Rechercher Consultation »

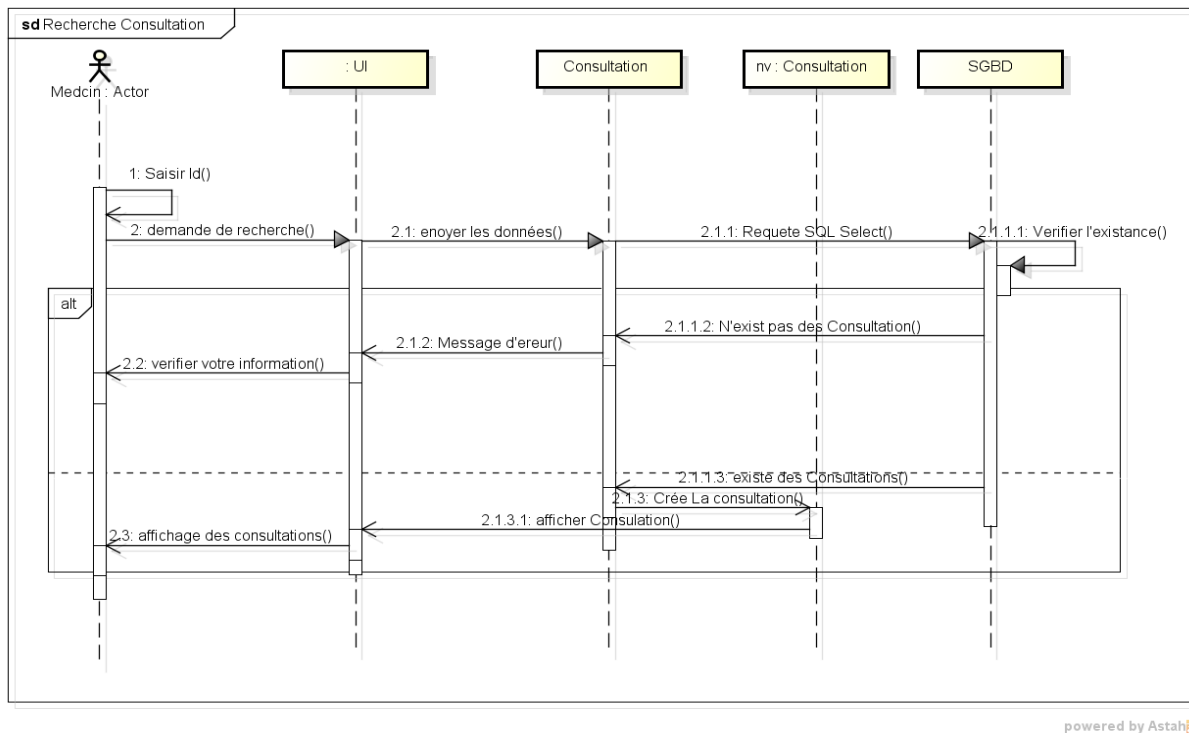


Figure : Diagramme de séquence système « Rechercher Consultation »

2 Analyse

2.1 Diagramme de classe

Les diagrammes de classes de conception représentent bien la structure statique du code, par le biais des attributs et des relations entre classes, mais ils contiennent également les opérations (aussi appelées méthodes) qui décrivent les responsabilités dynamiques des classes logicielles. L'attribution des bonnes responsabilités aux bonnes classes est l'un des problèmes les plus délicats de la conception orientée objet. Pour chaque service ou fonction, il faut décider quelle est la classe qui va le contenir.

Ci-dessous le Diagramme des classes

Le premier diagramme représente la structure générale de notre application avant de commencer le code.

Le deuxième diagramme représente la structure générale de notre application après la fin du code.

Explication de la différence :

Notre première vue de l'application était basée sur ce qu'on a étudié en module GL et qu'on devait lier la base de données avec les classes et ces dernières avec l'interface graphique. Mais en cherchant, on a découvert la technologie des « java beans » qui nous a permis de relier la base de données avec l'interface graphique directement sans se baser sur les classes et aussi utiliser que des attributs de type « StringProperty » pour faciliter le transfert de données entre l'interface graphique et la base de données tout en étant capable de les transformer à n'importe quel type selon l'utilisation dans le code et vous pouvez constater dans le deuxième diagramme qu'on a pas besoin des agrégations entre les classes.

Les beans sont des composants réutilisables ou des objets autonomes qui doivent pouvoir être facilement assemblés entre eux pour créer un programme.

Microsoft propose la technologie ActiveX pour définir des composants mais celle-ci est spécifiquement destinée aux plates-formes Windows.

Les Java beans proposés par Sun reposent bien sûr sur Java et de fait en possèdent toutes les caractéristiques : indépendance de la plate-forme, taille réduite du composant, ...

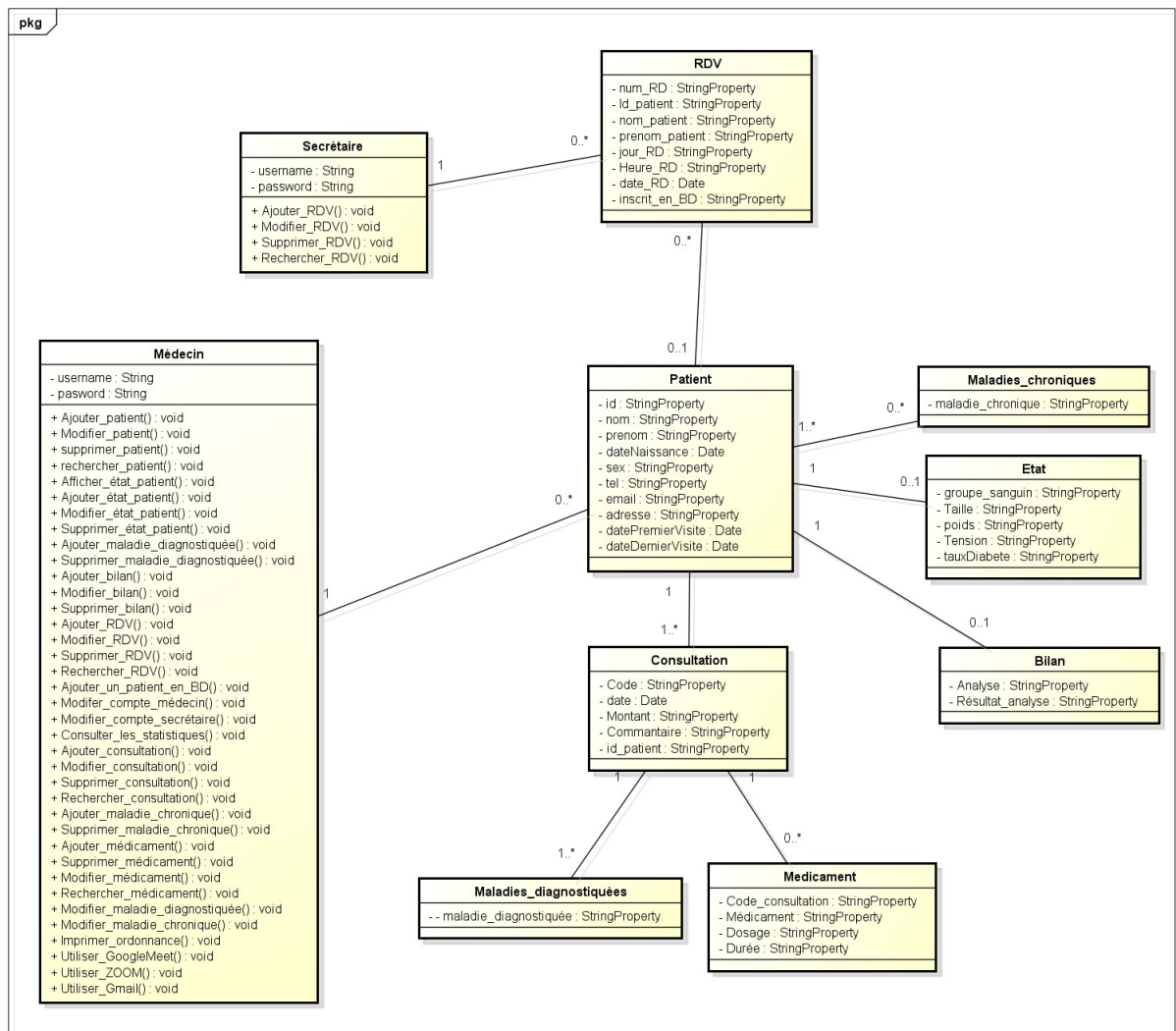
La technologie JavaBeans propose de simplifier et faciliter la création et l'utilisation de composants.

Les JavaBeans possèdent plusieurs caractéristiques :

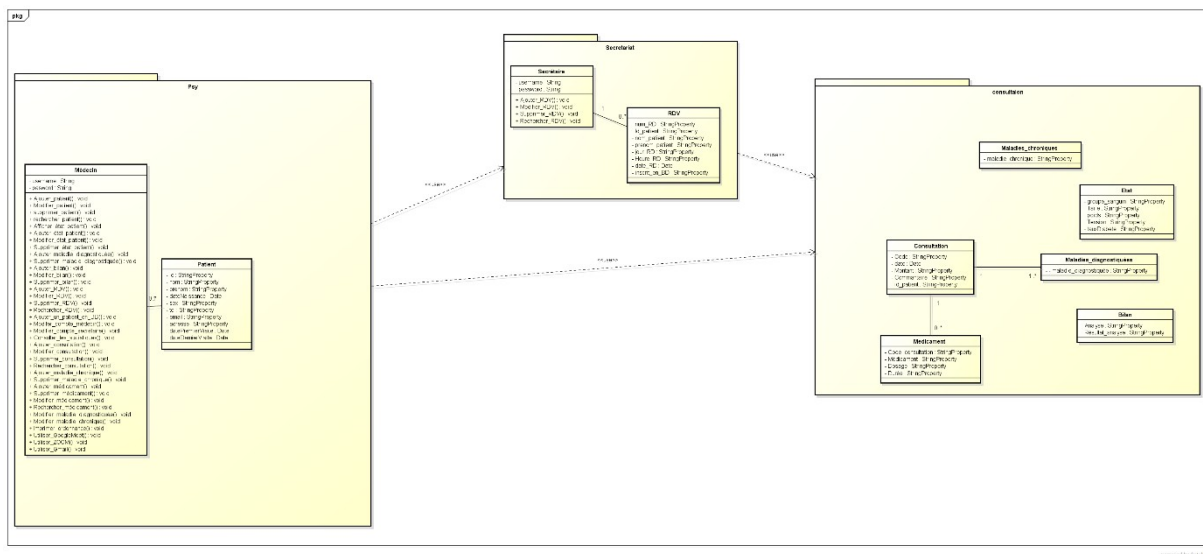
- La persistance : elle permet grâce au mécanisme de sérialisation de sauvegarder l'état d'un bean pour le restaurer. Ainsi si on assemble plusieurs beans pour former une application, on peut la sauvegarder.
- La communication, grâce à des événements, qui utilise le modèle des écouteurs introduit par Java 1.1.
- L'introspection : ce mécanisme permet de découvrir de façon dynamique l'ensemble des éléments qui composent le bean (attributs, méthodes et événements) sans avoir le code source.
- La possibilité de paramétrer le composant : les données du paramétrage sont conservées dans des propriétés.

Ainsi, les beans sont des classes Java qui doivent respecter un certain nombre de règles, c'est pour cela que notre code va être différent des applications développées avec swing et même celle avec javafx(qu'on a bel et bien utilisé).

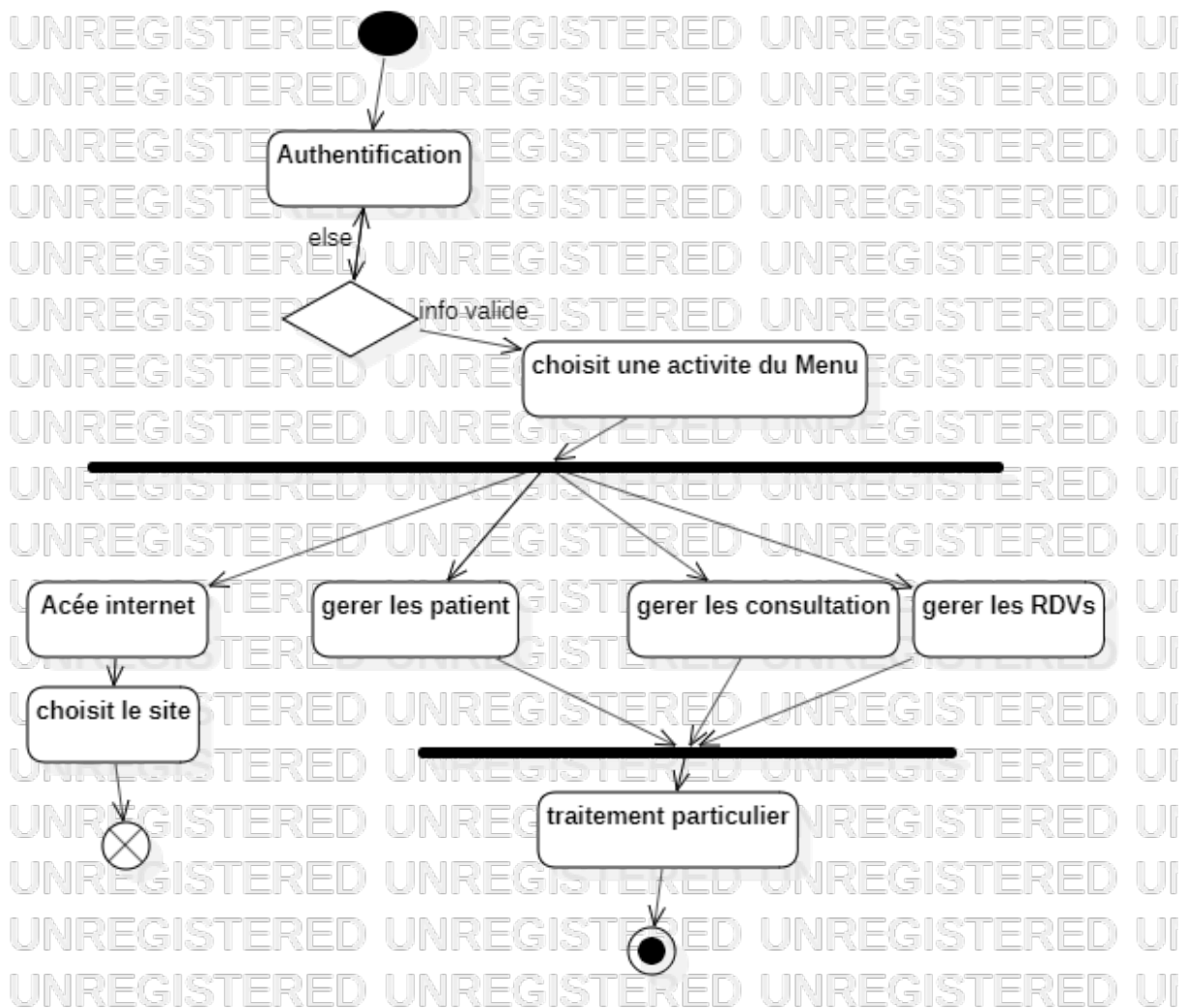




2.2 Diagramme de package



2.3 Diagramme d'activité



3 Capture des besoins techniques

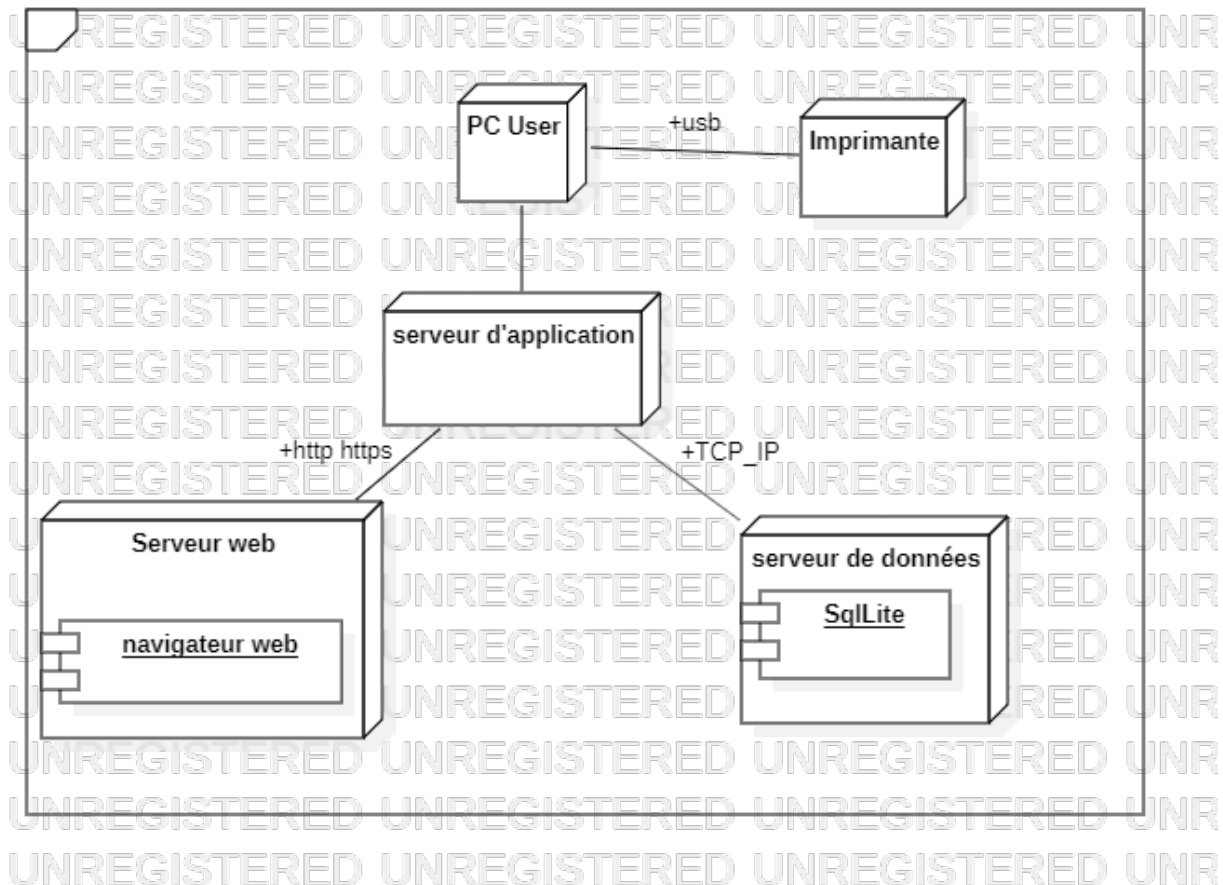
3.1 Diagramme de déploiement

Le diagramme de déploiement montre la disposition physique des matériels qui compose le système et la répartition de ses composants qui sont les suivants :

Le serveur d'application est responsable de la gestion des sessions utilisateurs, la gestion des patients, la gestion des consultations, gestion des RDV, etc.

Le serveur de données représente la partie centrale qui gère la base de données et ses accès.

Le serveur web transmet à l'utilisateur le résultat de la requête envoyé.



Chapitre 5 : Implémentation et réalisation

1 Introduction

La phase qui suit une conception est éventuellement l'implémentation. Après les 4 chapitres précédents, nous arrivons à rassembler nos informations et réaliser notre système, nous commençons ce chapitre par décrire l'environnement de développement matériel et logiciel que nous avons adoptés. Nous présentons ensuite quelques captures d'écrans de notre application.

2 Environnement de développement

2.1 Environnement matériel

Pour la réalisation de ce projet on a disposé d' :

. Un ordinateur :

- _ Marque : HP
- _ Processor : Intel(R) Core(TM) i3-3210M CPU 2.50GHz x 2
- _ RAM : 4Go
- _ Disque dur : 500Go
- _ Système d'exploitation : Windows 8.1

. Un ordinateur :

- _ Marque : Asus
- _ Processor : Intel(R) Core(TM) i5-8500U CPU 1.80GHz x 4
- _ RAM : 8Go
- _ Disque dur : 1To HDD + 128Go SSD
- _ Système d'exploitation : Windows 10

2.2 Environnement logiciel

Cette section décrit l'environnement logiciel avec lequel nous avons réalisé ce projet avec le rapport.



2.2.1 Eclipse IDE :

est un environnement de développement intégré (IDE) développé par la fondation Eclipse. Eclipse est un IDE complet misant sur la productivité avec des systèmes d'auto-complétion intelligente, d'analyse de code en temps réel, de refactoring avancé l'intégration d'outils de tests et de debugging et une pléthore de raccourcis clavier permettant de réaliser presque n'importe quelle tâche rapidement sans jamais lever les mains du clavier pour utiliser la souris.



2.2.2 SceneBuilder :

est un outil interactif de conception d'interface graphique pour JavaFX. Créé par Oracle, il permet de construire rapidement des interfaces utilisateurs sans avoir besoin de (savoir) coder.



2.2.3 Microsoft word :

est un logiciel de traitement de texte publié par Microsoft.



2.2.4 Astah :

Anciennement appelé **Jude**, **Astah** est un outil de modélisation [UML](#) créé par la compagnie japonaise ChangeVision¹. Il fonctionne avec l'[environnement d'exécution Java](#). Le nom vient de l'acronyme *Java and UML developers' environment*.

3 Présentation de la solution

Nous allons présenter dans cette partie les interfaces utilisateurs les plus importantes de notre application.

3.1 Interface d'Authentification

L'authentification est un certificat de sécurité du système, et à partir d'elle, chaque utilisateur authentifié peut accéder à notre application. Pour se faire, l'utilisateur introduit son nom d'utilisateur, son mot de passe comme le montre la figure ci-dessous.



Figure : Interface « authentification »

3.2 Le Menu principale :

Suite à l'authentification « Psychologue » un menu qui s'affiche qui nous permet de choisir l'interface qu'on souhaite y accéder :

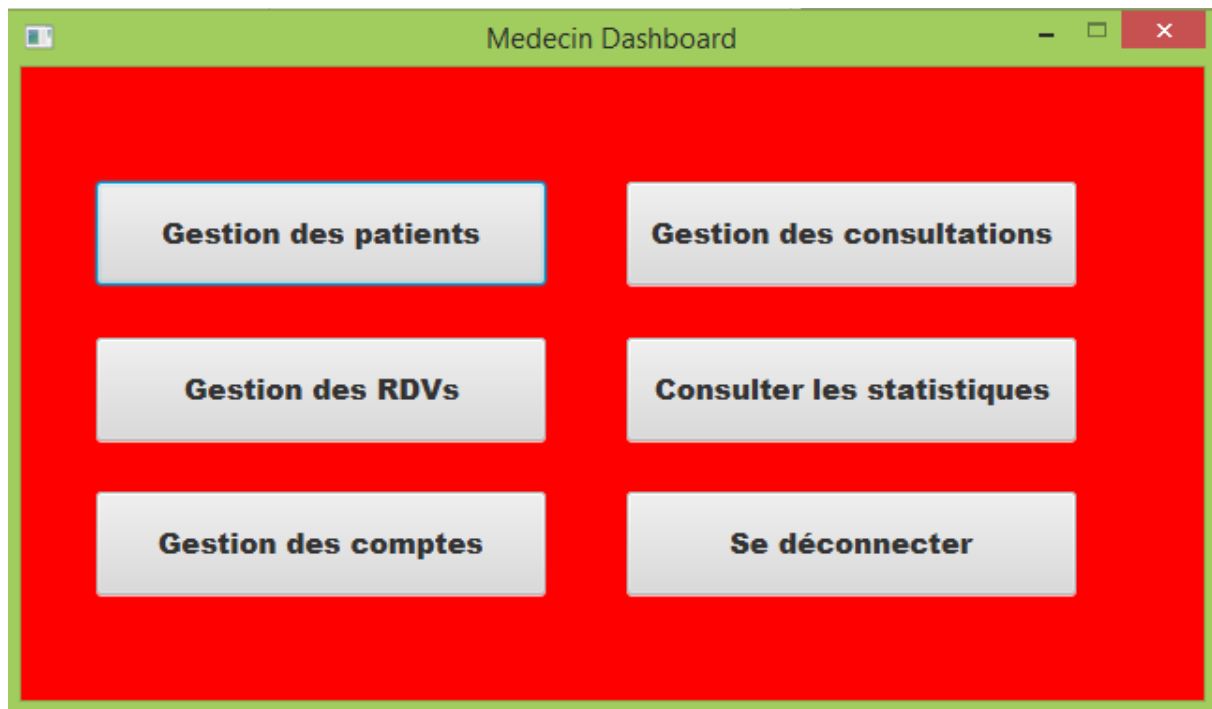


Figure : Interface « Menu »

3.3 Interface « Gérer les Consultations » :

Pour vérifier la consultation de chaque patient ou bien ajouter ou supprimer, il suffit de cliquer sur Gérer les Consultation :

Figure : «Gérer les Consultation»

3.4 Interface « Gérer les Patients » :

The screenshot shows a web application window titled "Gérer les patients". It features a table with the following columns: id, nom, prenom, date de naissance, sex, tel, email, adresse, date de la première visite, and date de la dernière visite. The table contains five rows of patient data. Below the table, there are several buttons: "Ajouter patient", "Supprimer patient", "afficher état patient", "Modifier patient", "rechercher", and a text input field labeled "nom". At the bottom right, there are two buttons: "Retour" and "Se déconnecter".

id	nom	prenom	date de naissance	sex	tel	email	adresse	date de la première visite	date de la dernière visite
12	mlkjhg...								08/07/2020
15	himane	khled	24/07/1992	h...	+213548...	khaleddjhk@gmail...	alger	01/07/2020	09/07/2020
17	himo	mohamed	30/06/1995	h...			paris	16/07/1999	09/07/2020
1	miHob	simo	12/04/1994	h...				03/07/2020	14/07/2020
2	kimo	amina	01/07/2020	f...	1452			30/06/2020	09/07/2020

Figure : Interface « Gérer les Patients »

3.5 Interface « Gérer les rendez-vous » :

The screenshot shows a web application window titled "Gérer les rendez-vous". It features a table with the following columns: Num_RD, id, nom, prénom, tel, email, jour_RD, heure_RD, date_RD, and inscrit_en_BD. The table is currently empty, displaying the message "Aucun contenu dans la table". Below the table, there are several buttons: "ajouter_RD", "supprimer_RD", "modifier_RD", "ajout_en_BD", "spécifier le jour", "jour_RD" (a dropdown menu), "rechercher_RD", and a text input field labeled "nom_patient". At the bottom right, there are two buttons: "retour" and "Se deconnecter".

Num_RD	id	nom	prénom	tel	email	jour_RD	heure_RD	date_RD	inscrit_en_BD
Aucun contenu dans la table									

Figure : Interface « Gérer les rendez-vous »

3.6 Interface « Consulter les statistiques » :

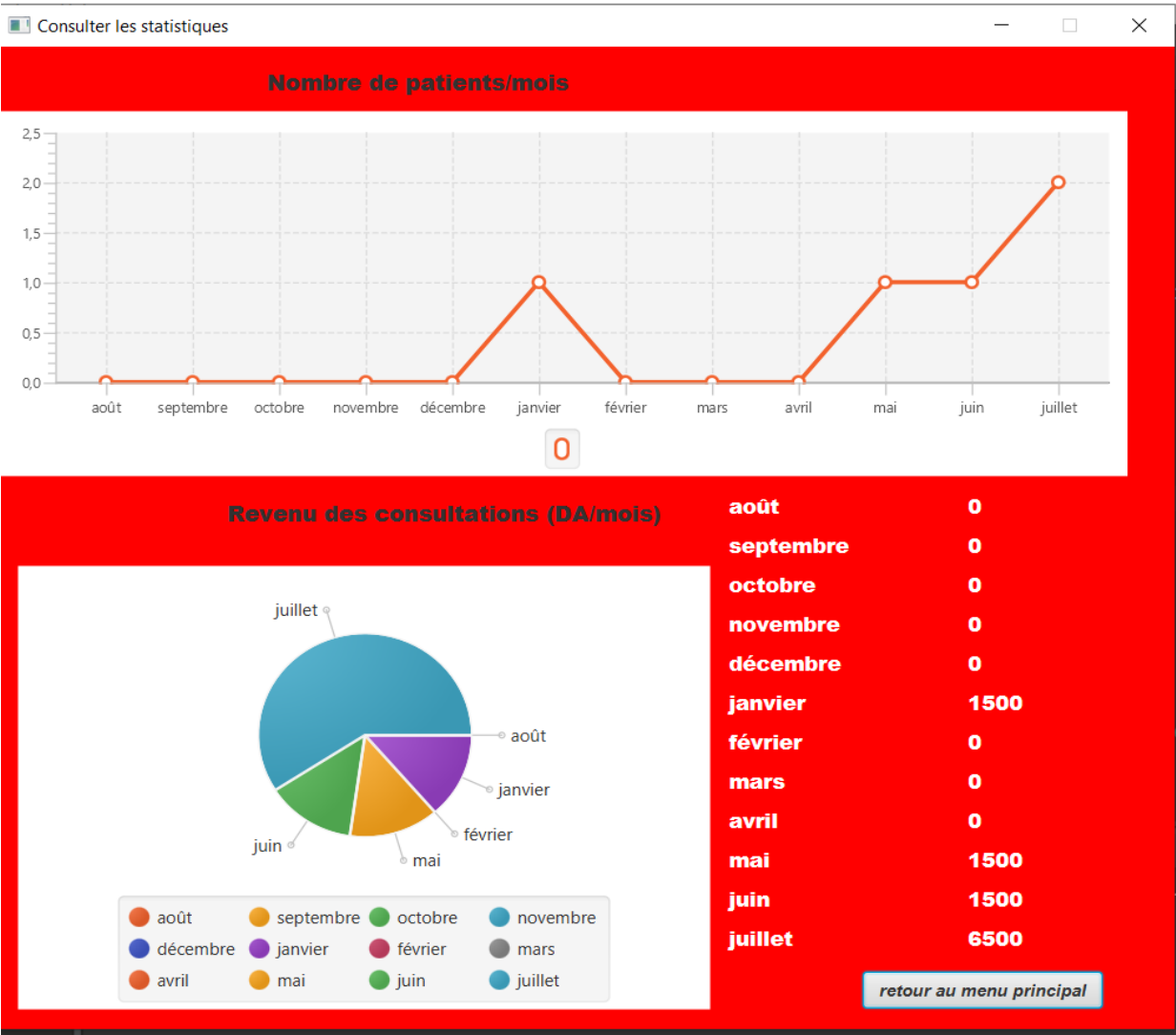


Figure : Interface « Consulter les Statistiques »

3.7 Interface « Gérer les Comptes » :

Figure : Interface « Gérer les Comptes »

4 Conclusion

Dans ce chapitre, nous avons présenté l'environnement matériel et logiciel utilisé lors du développement de l'application. Ensuite, on a présenté quelques captures d'écran montrant le bon fonctionnement de notre application.

Conclusion générale

Le présent rapport est réalisé dans le cadre de notre projet du module GL2.

Ce travail a pour objectif de concevoir et développer une application de gestion permet de gérer un cabinet de psychologue.

Pour pouvoir compléter notre mission, nous avons détaillé les différentes étapes d'analyse, de conception et de réalisation de ce système. Nous avons utilisé UML comme un langage de modélisation et 2TUP comme un processus de développement.

Ce projet était bénéfique pour nous dans plusieurs sens. Il nous ouvre sur le monde professionnel qui implique de la responsabilité et de la discipline ainsi que du travail collaboratif d'équipe, il a été une occasion pour améliorer nos connaissances dans les environnements de travail et de maîtriser les langages de programmation pour qui seront certes utiles au niveau professionnel.

Enfin, nous espérons que nous avons réussi à réaliser une application qui satisfera les besoins des futurs utilisateurs et qui répondra à leurs attentes et surtout contribuer à faciliter la vie des patients au confinement et aider les psychologues à atteindre leurs patients.