



**République Algérienne Démocratique et Populaire**

**Université 20 août 1955 – Skikda**

**Faculté des Sciences**

**Département d'Informatique**



**Mémoire de fin d'études**

**Pour l'obtention du diplôme de Master en Informatique**

**Option: Génie logiciel avancée et application (GLAA)**

Thème

**Systeme de recommandation : Application dans le  
domaine médical**

**Réalisé par**

**BOUKHETALA Rawnak**

**Encadré par**

**Dr. NABET Aicha**

**Session : Juin-2024**

## Remerciement

*Tout d'abord je tiens à remercier ALLAH le tout puissant de m'avoir donné la santé, la volonté, le courage et la patience pour mener à terme ma formation et pour pouvoir réaliser ce travail.*

*Je tiens à remercier particulièrement mon encadreur, Mme NABET aïcha, professeur à l'université de SKIKDA, en tant qu'encadreur de mon projet de fin d'étude, pour son encadrement de qualité, sa motivation professionnelle, ses conseils et critiques constructives, ses corrections, sa gentillesse et sa patience ainsi que pour le temps qu'il a consacré à la réalisation de ce travail.*

*Je tiens à remercier les membres du jury pour leur présence, pour leur lecture attentive de ce mémoire, ainsi que pour les remarques qu'ils m'adressent lors de cette soutenance afin d'améliorer mon travail. Ainsi, mes enseignants, j'espère que vous allez voir, dans ce manuscrit, les fruits du dévouement avec lequel vous avez fait preuve durant les enseignements que vous m'avez prodigué.*

*Je remercierai tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.*

## Dédicace

*A ma chère belle-mère et mon beau père Vous m'avez accueilli à bras ouverts dans votre famille. En témoignage de l'attachement, de l'amour et de l'affection que je porte pour vous. Je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite.*

*A mon cher petit frère*

*A travers cette dédicace, j'exprime à mon frère Abderrahim la richesse que ta présence a apportée à ma vie et comment tu m'as enrichi dans mon voyage vers la connaissance.*

*A mes chères sœurs*

*Ordjowan et Selma Maria*

*En témoignage de l'attachement, de l'amour et de l'affection que je porte pour vous, je vous dédie ce travail avec tous mes vœux de bonheur, de santé et de réussite.*

*A tous les membres de ma famille petits et grands.*

*A mes amies*

*Ce travail n'aurait pu être mené à bien sans le soutien inconditionnel à mes très chers amies Sofya ,Wisseem et Maneel, pour leurs conseils et leurs soutiens sans faille.*

*A mes enseignants de l'université de Skikda;*

*A tous ceux qui me sont chères ;*

*A tous ceux qui m'aiment ;*

*A tous ceux que j'aime.*

**Rawnak.**

## Résumé

Ce mémoire explore l'importance des systèmes de recommandation dans le domaine médical. Ces systèmes aident les médecins et les patients à accéder à des informations pertinentes et à prendre des décisions éclairées, améliorant ainsi la qualité des soins et réduisant les coûts. Le mémoire présente le développement de Medics, une application médicale dotée d'un système de recommandation sensible au contexte. Ce système fournit des recommandations d'articles médicaux spécifiques en fonction des spécialités des médecins, qu'ils soient généralistes ou spécialistes. En offrant des recommandations personnalisées, Medics aide les médecins à rester à jour avec les dernières avancées dans leur domaine, améliore la qualité des soins qu'ils peuvent offrir à leurs patients et contribue à une prise de décision plus rapide et plus éclairée. L'accès rapide et pertinent aux informations médicales permet aux médecins de prendre des décisions basées sur les preuves les plus récentes, ce qui réduit les erreurs médicales et améliore les résultats des patients. De plus, des décisions de traitement mieux informées peuvent réduire les coûts liés à des essais et des erreurs, des traitements inefficaces ou des hospitalisations prolongées. En combinant l'accès à des informations pertinentes avec une personnalisation selon la spécialité du médecin, Medics représente une avancée significative dans l'utilisation des systèmes de recommandation pour améliorer les soins de santé.

**Mots clés :** Systèmes de recommandation, recommandation sensible au contexte, articles médicaux.

## Abstract

This thesis explores the importance and applications of recommendation systems in the medical field. These systems help doctors and patients access relevant information and make informed decisions, thereby improving the quality of care and reducing costs. The thesis presents the development of Medics, a medical application equipped with a context-sensitive recommendation system. This system provides recommendations for specific medical articles based on the specialties of doctors, whether they are general practitioners or specialists. By offering personalized recommendations, Medics helps doctors stay up-to-date with the latest advancements in their field, improves the quality of care they can provide to their patients, and contributes to faster and more informed decision-making. Quick and relevant access to medical information allows doctors to make decisions based on the most recent evidence, reducing medical errors and improving patient outcomes. Furthermore, better-informed treatment decisions can reduce costs associated with trial and error, ineffective treatments, or prolonged hospitalizations. By combining access to relevant information with personalization according to the doctor's specialty, Medics represents a significant advancement in the use of recommendation systems to improve healthcare.

**Key words :** Recommendation systems, context-aware recommendation, medical articles.

## الملخص

تستكشف هذه الرسالة أهمية وتطبيقات أنظمة التوصية في المجال الطبي. تساعد هذه الأنظمة الأطباء والمرضى على الوصول إلى المعلومات ذات الصلة واتخاذ قرارات، مما يحسن جودة الرعاية ويقلل التكاليف. تقدم الرسالة تطوير تطبيق طبي مزود بنظام توصية حساس. يوفر هذا النظام توصيات لمقالات طبية محددة بناء على تخصصات الأطباء. سواء كانوا ممارسين عامين أو متخصصين. من خلال تقديم توصيات مخصصة. يساعد الأطباء على البقاء على اطلاع بأحدث التطورات في مجالهم، ويحسن جودة الرعاية التي يمكنهم تقديمها لمرضاهم، ويساهم في اتخاذ قرارات أسرع وأكثر استنارة. يتيح الوصول السريع وذو الصلة إلى المعلومات الطبية للأطباء اتخاذ قرارات بناءً على أحدث الأدلة، مما يقلل من الأخطاء الطبية ويحسن نتائج المرضى. علاوة على ذلك، يمكن أن تقلل القرارات العلاجية الأكثر استنارة من التكاليف المرتبطة بالتجربة والخطأ، والعلاجات غير الفعالة، أو الإقامة المطولة في المستشفيات. من خلال الجمع بين الوصول إلى المعلومات ذات الصلة والتخصيص وفقاً لتخصص الطبيب، يمثل تقدماً كبيراً في استخدام أنظمة التوصية لتحسين الرعاية الصحية.

كلمات مفتاحية: أنظمة التوصية، التوصية الحساسة للسياق، المقالات الطبية

## Table de matières

Remerciement .....	II
Dédicace .....	III
Résumé .....	IV
Table de matières .....	VII
Liste des figures .....	XI
Liste des algorithmes.....	Error! Bookmark not defined.
Listes des tableaux .....	XIII
Introduction générale.....	1
<b>Chapitre 1: Systèmes de recommandation</b>	
<b>1. Introduction.....</b>	<b>4</b>
<b>2. Systèmes de recommandation.....</b>	<b>4</b>
2.1. Histoire .....	4
2.2. Définition des systèmes de recommandations .....	5
2.3 Entités constituant un système de recommandation .....	5
2.4 Objectifs des systèmes de recommandations .....	6
<b>3. Différentes approches de recommandation .....</b>	<b>6</b>
3.1 Approches basées sur le contenu (Content-based Filtering).....	6
3.1.1 Profils de ressources .....	7
3.1.2 Profils d'utilisateur .....	8
3.1.3. Arbres de décision .....	9
3.1.4. Classificateur naïf de Bayes .....	9
3.1.5. Réseaux de neurones.....	10
3.2. Approches basées sur le filtrage collaboratif (collaborative filtering) .....	10
3.2.1. Discussion .....	12
3.3. Systèmes hybrides .....	12
3.3.1. Hybridation pondérée .....	13
3.3.2 Hybridation à bascule .....	13
3.3.3. Hybridation mixée .....	13
3.3.4. Hybridation par combinaison de caractéristiques .....	14

# Tables des matières

---

3.3.5. Hybridation en cascade.....	14
3.3.6. Hybridation par ajout de caractéristiques.....	14
3.3.7. Hybridation méta-niveau .....	14
3.4. Autres approches de recommandation.....	15
3.4.1. Filtrage Démographique .....	15
3.4.2. Filtrage Basé sur l'Utilité .....	15
3.4.3. Filtrage Basé sur la Connaissance .....	15
3.5. Avantages et les inconvénients des systèmes de recommandation .....	16
3.5.1. Avantages.....	16
3.5.2. Inconvénients .....	16
<b>4. Systèmes de recommandation dans le domaine médical .....</b>	<b>16</b>
4.1. Histoire .....	17
4.2. Fonctionnement des systèmes de recommandation médicale .....	17
4.3. Applications des systèmes de recommandation dans le domaine médical .....	19
4.4. Défis spécifiques au domaine médical .....	21
4.5. Exemples de systèmes de recommandation médicaux existants .....	22
<b>5. Conclusion .....</b>	<b>22</b>
<b>Chapitre 2 : Systèmes de recommandation sensibles au contexte (CARS)</b>	
<b>1. Introduction.....</b>	<b>24</b>
<b>2. Contexte .....</b>	<b>24</b>
2.1. Définition du contexte .....	24
2.2. Définition de sensibilité au contexte (context-awareness).....	25
2.3. Types de contexte.....	26
2.3.1. Contexte de l'utilisateur .....	26
2.3.2. Contexte de l'item .....	26
2.3.3. Contexte de décision.....	26
2.4. Caractéristiques du contexte .....	27
<b>3. Information contextuelle .....</b>	<b>28</b>
3.1. Définition.....	28
3.2. Sources d'informations contextuelles.....	28
<b>4. Utilisation du contexte dans les systèmes de recommandation .....</b>	<b>29</b>
<b>5. Formalisation du contexte.....</b>	<b>30</b>
<b>6. Approches d'incorporation du contexte.....</b>	<b>31</b>
6.1. Approche de Pré-filtrage Contextuel (contextual pre-filtering) .....	32

# Tables des matières

---

6.2. Approche de Post-filtrage Contextuel (contextual post-filtering) .....	32
6.3. Approche de la Modélisation Contextuelle (contextual modeling) .....	33
<b>7.Exemples sur des systèmes de recommandation sensibles au contexte ...</b>	<b>34</b>
<b>8. Etude de l'existant .....</b>	<b>36</b>
<b>9. Conclusion .....</b>	<b>36</b>
<b>Chapitre 3:Analyse et conception du système Medics</b>	
<b>1. Introduction .....</b>	<b>49</b>
<b>2. Langage de modélisation UML.....</b>	<b>49</b>
2.1. Définition.....	49
2.2. Concepts fondamentaux de l'UML.....	50
2.3. Différents types de diagrammes UML .....	51
2.3.1. Diagrammes structurels .....	51
2.3.2. Diagrammes comportementaux :.....	52
<b>3. Description générale du système Medics.....</b>	<b>53</b>
3.1. Fonctionnalités et Objectifs .....	53
3.2. Noyau de Recommandation Medics.....	53
3.2.1. Description des scénarios .....	<b>Error! Bookmark not defined.</b>
3.2.1.1. Algorithme du 1 <sup>er</sup> scénario.....	54
3.2.1.2. Algorithme du 2 <sup>ème</sup> scénario.....	54
3.2.1.3. Algorithme du 3 <sup>ème</sup> scénario.....	54
3.2.1.4. Algorithme du 4 <sup>ème</sup> scénario.....	54
3.3. Acteurs cibles du système.....	59
<b>4. Conception du système Medics .....</b>	<b>60</b>
4.1. Diagrammes de cas d'utilisation.....	60
4.2. Diagrammes de séquence .....	65
4.3. Diagramme de classe .....	67
<b>5. Conclusion .....</b>	<b>67</b>
<b>Chapitre 4 : Implémentation du système Medics</b>	
<b>1. Introduction.....</b>	<b>69</b>
<b>2. Architecture générale du système.....</b>	<b>69</b>
<b>3. Environnement du travail.....</b>	<b>69</b>
3.1. Langage de développement .....	69
3.1.1. Javascript .....	69

# Tables des matières

---

3.1.2. NODE JS.....	70
3.2. Logiciels utilisés .....	70
3.2.1. Visuel Code Studio.....	70
3.2.2. Firebase.....	71
3.2.3. Git.....	71
3.3. Librairie .....	72
3.3.1. React .....	72
<b>4. Structure d'une application web .....</b>	<b>72</b>
4.1. Architecture MVC.....	72
4.1.1. Modèle (Model).....	73
4.1.2. Vue (View).....	73
4.1.3. Contrôleur (Controller).....	73
<b>5. Implémentation du module de recommandation.....</b>	<b>74</b>
5.1. Algorithme de recommandation.....	75
5.2. Filtrage des articles.....	75
5.3. Tri et classement des articles recommandés .....	76
<b>6. Interfaces de l'application.....</b>	<b>78</b>
<b>7. Conclusion .....</b>	<b>88</b>
<b>Conclusion générale .....</b>	<b>90</b>
<b>Bibliographie .....</b>	<b>92</b>

## Liste des figures

Figure 1.1 : Un système de recommandation basé sur le contenu .....	7
Figure 1.2 : Filtrage collaboratif utilisateur et article .....	10
Figure 1.3 : Un système de recommandation collaboratif.....	11
Figure 1.4 : Système de recommandation hybride .....	13
Figure 1.5 : Fonctionnement des systèmes de recommandation médicales .....	17
Figure 2.1 : Utilisation du contexte dans les systèmes de recommandation.....	30
Figure 2.2 : Les approches des systèmes de recommandation sensible au contexte.....	31
Figure 2.3: Pré-filtrage Contextuel.....	32
Figure 2.4: Post-filtrage Contextuel. ....	33
Figure 2.5: La modélisation contextuelle .....	33
Figure 2.6 : Barre d'outils du site Amazon.....	35
Figure 2.7 : Fonctionnement système de recommandation ( <b>Medics</b> ) .....	37
Figure 3.1 : Diagrammes UML .....	51
Figure 3.2 : Fonctionnement du système de recommandation ( <b>Medics</b> ).....	53
Figure 3.3 : Diagramme de cas d'utilisation du système concernant l'acteur Patient.....	55
Figure 3.4 : Diagramme de cas d'utilisation pour un médecin généraliste.....	56
Figure 3.5 : Diagramme de cas d'utilisation pour un médecin spécialiste. ....	56
Figure 3.6 : Diagramme de séquence pour la création d'un profil médecin.....	60
Figure 3.7 : Diagramme de séquence pour la recommandation des articles médicaux par un médecin généraliste.....	60
Figure 3.8: Diagramme de séquence pour la recommandation des articles Médicaux par un médecin spécialiste.....	61
Figure 3.9: Diagramme de classe pour le système Medics.....	62
Figure 4.1: Architecture générale du système Medics.....	64
Figure 4.2: Interface Visual Studio Code .....	65
Figure 4.3 : Git GUI de GitKraken.....	66
Figure 4.4 : Architecture MVC .....	67
Figure 4.5: Page d'accueil de la plateforme Medics .....	73

Figure 4.6 : Interface contact Admin.....	74
Figure 4.7 : interface d'inscription coté patient .....	74
Figure 4.8: Interface de connexion coté patient .....	75
Figure 4.9 : Recherche médecin à travers le nom .....	75
Figure 4.10 : Recherche médecin à travers la spécialité.....	76
Figure 4.11 : Interface des médecins généralistes ont le même champ d'intérêt (Neurology)...	77
Figure 4.12 : Interface des médecins spécialistes ont la même spécialité (Neurology) mais Différentes micro-spécialités .....	77
Figure 4.13 : vu du profil médecin par le patient .....	78
Figure 4.14: Prendre rendez-vous .....	78
Figure 4.15: profil patient .....	79
Figure 4.16 : Interface d'inscription coté médecin .....	79
Figure 4.17 : Interface de connexion coté médecin.....	80
Figure 4.18 : Gestion des patients .....	80
Figure 4.19 : Recommandation des articles du système Medics pour un médecin spécialiste (pulmonology-Asthma).....	81
Figure 4.20 : Recommandation des articles du système Medics pour un médecin généraliste (pulmonologie) .....	81
Figure 4.21 : Classification des articles avant l'évaluation et la mise a jour.....	82
Figure 4.22 : Evaluation d'articles (mettre dislike).....	83
Figure 4.23 : Mise à jour des articles.....	84

## Liste des algorithmes

Algorithme 4.1 : Filtrage d'Articles par Spécialisation Générale .....	70
Algorithme 4.2 : Filtrage d'Articles par Spécialisation Spécialiste.....	70
Algorithme 4.3 : Tri des Articles par Appréciations et Désappréciations .....	71
Algorithme 4.4 : Gestion de la Lecture et Mise à Jour des Appréciations des Articles .....	71
Algorithme 4.5 : Rôle du Code handleLikeOrDislike.....	72

## Listes des tableaux

Tableau 3.1 : Acteurs cible du système Medics .....	55
---	----

# Introduction générale

---

## Introduction générale

Les systèmes de recommandation dans le domaine médical jouent un rôle crucial en aidant les professionnels de la santé et les patients à accéder à des informations pertinentes et à prendre des décisions éclairées. Ces systèmes s'appuient sur des algorithmes sophistiqués pour proposer des recommandations personnalisées, contribuant ainsi à l'amélioration de la qualité des soins et à la réduction des coûts. Dans le contexte médical, les recommandations peuvent prendre la forme de directives cliniques, de traitements ou de thérapies adaptés à des situations spécifiques, des articles médicaux, etc. Cependant, l'efficacité et l'acceptation de ces recommandations dépendent de leur pertinence, de leur diversité et de leur capacité à s'intégrer dans les pratiques professionnelles. Par conséquent, l'étude des systèmes de recommandation dans le domaine médical revêt une importance capitale pour comprendre comment ces outils peuvent être optimisés pour répondre aux besoins uniques du secteur de la santé.

L'une des principales difficultés réside dans la capacité à fournir des recommandations de santé personnalisées et adaptées au contexte spécifique de chaque individu. Les recommandations génériques, bien que précieuses, ne tiennent souvent pas compte des facteurs environnementaux, des préférences personnelles et des conditions de santé uniques des patients... Cette lacune peut entraîner une adhésion sous-optimale aux plans de soins et, par conséquent, des résultats insatisfaisants.

Les articles médicaux sont essentiels pour les médecins, généralistes comme spécialistes, car ils permettent une mise à jour continue des connaissances et l'amélioration des soins aux patients. Ils fournissent des informations basées sur des preuves, soutenant des décisions cliniques éclairées et favorisant la formation continue. De plus, ils encouragent la collaboration mondiale, le partage des connaissances, le développement professionnel et l'innovation, garantissant ainsi que les médecins restent compétents et à la pointe des avancées médicales.

Pour relever ce défi, ce mémoire propose le développement d'une application médicale innovante testant divers scénarios afin d'améliorer la qualité des recommandations. Notre système nommé **Medics**, doté d'un noyau de recommandation sensible au contexte proposant

## Introduction générale

---

d'articles médicaux spécifiques en fonction de catégorie de médecin, qu'elle soit générale ou spécialisée ainsi d'autres paramètres liés aux articles.

Ce mémoire se divise en quatre chapitres, à l'exception de l'introduction générale et de la conclusion générale.

**Premier chapitre** ; présente les systèmes de recommandation, leurs objectifs, avantages, inconvénients, et diverses approches, etc.

**Deuxième chapitre** ; traite les systèmes de recommandation sensibles au contexte en explorant la notion du contexte, ses applications, types, modèles de représentation, techniques de recommandation avec une étude de l'existant ;

**Troisième chapitre** ; est consacré à la phase d'analyse et de conception du système Medics ;

**Quatrième chapitre** ; s'appuie sur la présentation de l'implémentation du système, incluant les algorithmes utilisés, leurs implémentations, interfaces de l'application, etc.

# Chapitre 1

---

## Systemes de recommandation

## 1. Introduction

La connectivité offerte par Internet a permis de relier des millions de services en ligne, offrant une immense quantité d'informations, de produits et de services (items). Un simple site web de commerce électronique peut proposer jusqu'à des millions d'articles de différentes catégories. Or, lorsqu'un utilisateur cherche un livre à lire ou une destination touristique à visiter, il risque d'être submergé par la multitude d'options disponibles. Confrontés à cette situation, les utilisateurs peuvent avoir du mal à sélectionner des informations pertinentes et à identifier des produits adaptés à leurs besoins spécifiques.

Des systèmes de recommandation (SR) ont été mis en avant pour résoudre ces défis. Appliqués avec succès dans de nombreux domaines, notamment les livres, les films, les services de voyage et bien d'autres, les SR aident les utilisateurs en ligne à maîtriser la surcharge d'informations en fournissant des recommandations personnalisées.

Dans ce chapitre, nous aborderons de manière détaillée le domaine des systèmes de recommandation, en explorant les approches utilisées, tandis que nous décrirons les différentes avantages et inconvénients de ces systèmes.

## 2. Systèmes de recommandation

### 2.1. Histoire

Les systèmes de recommandation ont émergé comme un domaine de recherche distinct au milieu des années 1990, tirant leurs origines de divers domaines tels que les sciences cognitives, la théorie d'approximation, la recherche documentaire, la théorie de la prévoyance, ainsi que des liens avec la science de la gestion et le marketing pour modéliser les choix des consommateurs. Ces systèmes ont évolué à partir du concept de filtrage d'informations, initialement développé dans le domaine de la Recherche d'Information (RI), où l'accent s'est déplacé vers la personnalisation des résultats présentés à l'utilisateur afin de réduire la surcharge informationnelle.

L'intérêt croissant pour ce concept de filtrage a conduit à l'émergence des systèmes de recommandation en tant qu'entités distinctes mais connexes aux systèmes de RI, se distinguant par leur spécialisation dans le traitement des profils à long terme (par opposition aux requêtes à court terme dans la RI). Jusqu'en 1997, le terme "Filtrage Collaboratif" était

utilisé au lieu de "Système de Recommandation". Cependant, cette dernière appellation a été officialisée cette année-là par Resnick et Varian, qui ont souligné que le filtrage collaboratif n'était pas l'unique approche pour formuler des recommandations. De nos jours, de nombreux sites internet populaires intègrent des systèmes de recommandation pour leur fonctionnement, parmi lesquels on retrouve YouTube, Yahoo, Trip Advisor, Google, Amazon, Netflix, etc. [1]

### 2.2. Définition des systèmes de recommandations

Il existe plusieurs définitions d'un système de recommandation à cause de la diversité des classifications proposées pour ces systèmes, mais la plus générale est celle de Robin Burke, dans son ouvrage de 2002, définit les systèmes de recommandation comme *"des systèmes capables de fournir des recommandations personnalisées permettant de guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un espace de données important"*. [2]

Cette définition met en évidence la capacité des systèmes de recommandation à fournir des suggestions personnalisées pour aider les utilisateurs à trouver des ressources pertinentes au sein d'un large ensemble de données.

Les systèmes de recommandations utilisent diverses techniques telles que le filtrage collaboratif, le filtrage basé sur le contenu, le filtrage démographique, le filtrage basé sur les connaissances, et le filtrage communautaire pour atteindre cet objectif.

### 2.3 Entités constituant un système de recommandation

Un système de recommandation utilise des entités afin de pouvoir calculer la similarité entre par exemple utilisateur / élément puis générer des recommandations personnalisées.

Un système de recommandation est généralement représenté par les entités suivantes :

- **Utilisateurs:** ce sont les individus pour lesquels des recommandations sont générées. les systèmes de recommandation utilisent les préférences et le comportement passés des utilisateurs pour générer des recommandations personnalisées.
- **Éléments (articles, produits, services):** ce sont les éléments pour lesquels des recommandations sont fournies. Les systèmes de recommandation utilisent les caractéristiques de ces éléments pour évaluer leur pertinence par rapport aux préférences des utilisateurs.

- **Score d'intérêt (pour chaque élément pour chaque utilisateur):** ces scores représentent l'utilité d'un élément pour un utilisateur particulier. Les systèmes de recommandations utilisent ces scores pour évaluer et classer les éléments en fonction des préférences des utilisateurs.[3]

### 2.4 Objectifs des systèmes de recommandations

Les objectifs principaux des systèmes de recommandation incluent :

- Proposer des éléments (items) pertinents et intéressants pour chaque utilisateur afin d'améliorer son expérience ;
- Améliorer continuellement les performances clés, telles que la durée de visionnement, le temps de lecture, et réduire les délais de recherche de contenus/produits ;
- Gérer de manière automatisée un volume croissant de données impossible à traiter manuellement ;
- Réaliser une analyse pointue des données pour offrir des recommandations personnalisées pertinentes ;
- Automatiser le filtrage des données pour faciliter le processus de recommandation.
- Accroître la confiance de l'utilisateur dans le système en rendant les recommandations explicables ;
- Éviter la monotonie et offrir une variété d'options aux utilisateurs en diversifiant les recommandations.[4]

### 3. Différentes approches de recommandation

Les systèmes de recommandation utilisent différentes approches pour générer des recommandations personnalisées. Les principales approches incluent l'approche basées sur le contenu, le filtrage collaboratif et les systèmes hybrides.

#### 3.1 Approches basées sur le contenu (Content-based Filtering)

La recommandation basée sur le contenu consiste à analyser le contenu des ressources ou des descriptions de ces ressources afin de déterminer quelles ressources sont susceptibles d'être utiles ou intéressantes pour un utilisateur donné. Ce sous-domaine est fortement similaire au domaine de la recherche d'information.

En effet, les mêmes techniques sont utilisées, la différence se trouve essentiellement dans l'absence de requêtes explicites formulées par l'utilisateur. Par conséquent, beaucoup de concepts généraux de la recommandation basée sur le contenu proviennent de la recherche d'information. [5]

La plupart des systèmes de recommandation basée sur le contenu identifient les ressources similaires aux ressources qu'un utilisateur donné a appréciées. Ainsi, quand de nouvelles ressources sont introduites dans le système, elles peuvent être recommandées directement sans que cela ne nécessite un temps d'intégration.

Ce Type de filtrage doit remplir deux fonctions principales :

- Identifier les documents pertinents pour l'utilisateur en fonction de son profil.
- Mettre à jour le profil utilisateur en fonction des évaluations, implicites ou explicites de l'utilisateur des documents reçus.

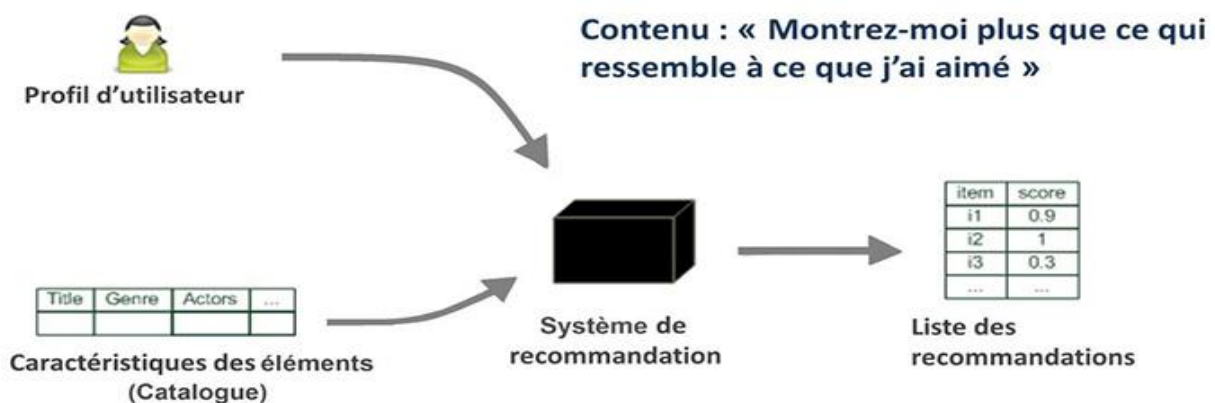


Figure 1.1: Un système de recommandation basé sur le contenu

L'approche de recommandation basée sur le contenu nécessite de construire deux éléments fondamentaux : les profils de ressources et les profils d'utilisateurs.

### 3.1.1 Profils de ressources

Les profils de ressource consistent en un ensemble d'attributs décrivant les ressources, de façon analogue à l'index utilisé dans le domaine de la recherche d'information. Comme dans le domaine de la recherche d'information, la précision de cette approche est donc hautement dépendante de la nature des ressources : elle est beaucoup plus élevée pour des

ressources de recommandation basée sur le contenu textuel que pour des ressources telles que les images, les vidéos ou les ressources audio, dont il est difficile d'extraire des attributs.

En général, quand cette approche est employée pour des ressources non textuelles, des métadonnées sont utilisées. Par conséquent, la plupart des recherches sur la recommandation basée sur le contenu porte sur des données textuelles.

Une étape clé de cette approche est de transformer les données textuelles non structurées en une représentation structurée. Le stemming réduit les mots relatifs à un même concept en un terme unique. Chaque terme reçoit ensuite un poids basé sur son importance, souvent calculé par la méthode (Term Frequency- Inverse Document Frequency) TF-IDF qui est une mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document. Cette technique ne prend toutefois pas en compte le contexte, ce qui peut entraîner des erreurs, notamment avec des tournures négatives ou ironiques. Finalement, le système obtient les mots les plus importants ou des vecteurs de termes avec des poids associés.

### 3.1.2 Profils d'utilisateur

Le profil d'un utilisateur définit ses centres d'intérêt. De tels profils consistent en un ensemble d'informations qui peuvent être entrées manuellement par l'utilisateur, ou extraites automatiquement à partir du contenu des ressources qu'il a consultées.

**Centres d'intérêt déduit directement** ; cette approche consiste de demander à l'utilisateur de fournir directement ses centres d'intérêts, à l'aide de formulaires, en lui demandant d'entrer manuellement une liste de termes, etc. Si un nombre restreint d'informations est demandé, cette approche rendra le système opérationnel rapidement, mais ne pourra pas fournir de recommandations précises. À l'inverse, en demandant un grand nombre d'informations, les recommandations seront plus précises mais le système sera trop contraignant pour l'utilisateur. De plus, les centres d'intérêts des utilisateurs peuvent évoluer au cours du temps, et une telle approche impose une actualisation manuelle régulière, ce qui est également contraignant. Un dernier inconvénient, est que l'utilisateur peut ne pas remplir le formulaire honnêtement, auquel cas, les recommandations qui lui seront fournies ne pourront pas être pertinentes.

**Centres d'intérêt extrait automatiquement** ; consiste à extraire automatiquement les préférences des utilisateurs à partir du contenu des ressources consultées. Une des méthodes les plus simples est de représenter les centres d'intérêt des utilisateurs par des vecteurs de

termes à partir des vecteurs de termes représentant les ressources que l'utilisateur a appréciées.

Les appréciations peuvent être obtenues de façon explicite en demandant directement aux utilisateurs de les fournir par exemple le pouce en haut pour indiquer la satisfaction ou au contraire le pouce en bas, ou implicitement en utilisant des algorithmes basés sur les usagés pour calculer les recommandations, il suffit alors de calculer la similarité entre les profils de ressource et les profils d'utilisateur. Cela peut être effectué selon diverses méthodes, comme la mesure de similarité cosinus. C'est dans ce cadre que cette approche est la plus similaire aux approches de la recherche d'information.

Beaucoup d'autres méthodes d'extraction automatique de profils qui se démarquent davantage de la recherche d'information ont été proposées. Dans ce cadre, les recommandations sont calculées selon la probabilité qu'un utilisateur donné appréciera une ressource. Cela peut être considéré comme un problème de classification où chaque classe représente un niveau d'appréciation (« aime » et « n'aime pas »). Trois des algorithmes de classification célèbres souvent utilisés dans ce contexte sont présentés dans cette section : les arbres de décision, le classificateur naïf de Bayes et les réseaux de neurones.[6]

### 3.1.3. Arbres de décision

Un arbre de décision est obtenu en séparant de façon récursive les ressources en sous-groupes homogènes relativement à des variables déterminées au préalable. Dans le cas de la recommandation de ressources textuelles, ces variables sont en principe des variables booléennes sur la présence ou l'absence de termes. Ensuite, pour chaque sous-groupe, la probabilité que l'utilisateur appréciera une ressource de ce sous-groupe est conservée. Le problème principal de l'application de cette approche à la recommandation basée sur le contenu est que la précision obtenue est dépendante du nombre de variables manipulées. Cette approche est simple et performante dans le cadre de recommandations portant sur des ressources ayant un nombre d'attributs limité, mais n'est pas appropriée dès que ces attributs sont en nombre élevé, ce qui est le cas des ressources textuelles sans restriction.[7]

### 3.1.4. Classificateur naïf de Bayes

Le classificateur naïf de Bayes est un algorithme de classification basé sur le théorème de Bayes, avec une hypothèse simplificatrice d'indépendance entre les caractéristiques (attributs) des données. Malgré cette hypothèse d'indépendance (qui est souvent irréaliste), il

fonctionne bien dans de nombreux contextes pratiques et est particulièrement utile pour les tâches de classification textuelle comme la catégorisation de courriels (spam vs non-spam) et l'analyse de sentiment.[8]

### 3.1.5. Réseaux de neurones

Les réseaux de neurones sont des modèles d'apprentissage automatique inspirés du fonctionnement des neurones biologiques. Ils sont composés de couches de neurones artificiels (unités de calcul) qui se connectent entre elles via des poids ajustables. Ces réseaux sont capables d'apprendre des représentations complexes des données en ajustant ces poids lors de la phase d'entraînement.[9]

## 3.2. Approches basées sur le filtrage collaboratif (collaborative filtering)

L'approche de filtrage collaboratif se base sur le partage d'opinions entre les utilisateurs afin de trouver un groupe d'utilisateurs ayant des préférences semblables à celles de l'utilisateur ciblé. Ainsi, elle utilise ce groupe pour recommander des éléments (par exemple des produits). Bien qu'elle repose sur l'hypothèse selon laquelle les utilisateurs avec des goûts similaires auront les mêmes préférences.

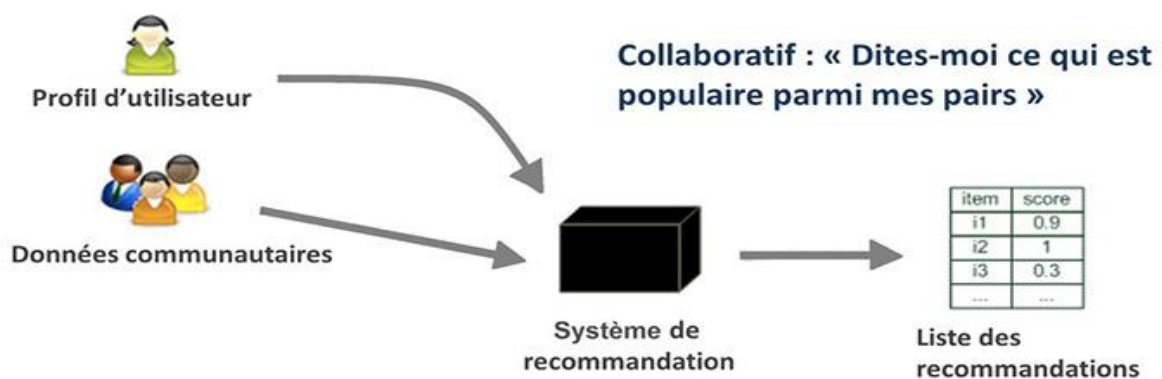


Figure 1.2 : Système de recommandation collaboratif

Cette approche peut être divisée en deux sous-catégories :

- **Filtrage collaboratif utilisateur (User-based collaborative filtering) :**

Cette approche utilise les évaluations ou les comportements passés des utilisateurs pour identifier des utilisateurs similaires à celui pour lequel une recommandation est recherchée. Par exemple, si deux utilisateurs A et B ont tous deux adoré les mêmes films X, Y et Z,

alors lorsqu'un nouveau film est introduit, si l'utilisateur A adore ce film, il est probable que B l'apprécie également.

**Exemple :** Sur une plateforme de streaming vidéo, si un utilisateur a regardé et apprécié une série de science-fiction, le système de recommandation peut lui suggérer d'autres séries ou films du même genre qui ont été bien notés par d'autres utilisateurs ayant regardé la même série.

- **Filtrage collaboratif article (Item-based collaborative filtering)**

À la différence de l'approche utilisateur, cette approche se concentre sur les similarités entre les articles plutôt que sur les utilisateurs. Elle recommande des articles similaires à ceux que l'utilisateur a appréciés dans le passé.

**Exemple :** Dans un site de commerce électronique, si un utilisateur a acheté un téléphone portable, le système peut recommander des accessoires compatibles tels que des étuis de protection des téléphones portables ou des chargeurs, car ces articles sont souvent achetés ensemble par d'autres utilisateurs qui ont acheté le même téléphone.

Ces deux approches peuvent être implémentées de différentes manières, notamment par des techniques telles que la similarité cosinus, les méthodes de voisinage (k-nearestneighbors), ou les modèles de factorisation matricielle. Chacune présente ses propres avantages et inconvénients en termes de précision, de scalabilité et de performances.[10]

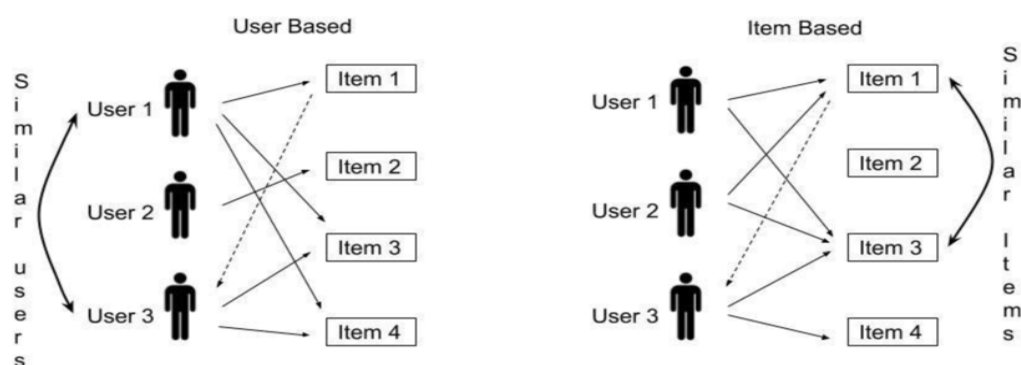


Figure 1.3 : Filtrage collaboratif utilisateur et article

En résumé, le filtrage collaboratif est une technique de recommandation qui s'appuie sur les opinions partagées entre les utilisateurs pour identifier des groupes homogènes et recommander des éléments en fonction de leurs préférences communes.

### 3.2.1. Discussion

La recommandation basée sur le contenu présente plusieurs limitations. Elle nécessite un nombre suffisant d'attributs décrivant les ressources, ce qui la rend plus adaptée aux ressources textuelles ou lorsque des descriptions textuelles sont saisies manuellement. Cependant, les méthodes de classification de texte peuvent échouer à différencier des ressources similaires en termes d'attributs mais de qualité ou pertinence différentes. De plus, ces modèles ne recommandent que des ressources similaires à celles qu'un utilisateur a déjà appréciées, limitant ainsi la diversité des recommandations. Ce problème peut être atténué par des recommandations aléatoires. Un autre défi est le démarrage à froid : un nouvel utilisateur doit d'abord interagir avec plusieurs ressources pour recevoir des recommandations pertinentes. Pour réduire cet impact, on peut demander des informations initiales à l'utilisateur et utiliser un profil type correspondant.

Une des principales limitations du filtrage collaboratif est le problème du manque de données ou data sparsity problem en anglais.

En effet, dans le cadre de notes explicites, le pourcentage moyen de ressources pour lesquelles les utilisateurs ont fourni une appréciation est très basse. Par exemple, une des bases de données fournies par MovieLens 6 consiste en 100 000 notes pour 1 642 films par 943 utilisateurs, soit 6,3% de notes fournies. Dans un tel cadre, la similarité entre deux utilisateurs ne peut être calculée que s'ils ont noté un minimum de ressources communes.

Tout comme les approches basées sur le contenu, le filtrage collaboratif souffre du problème du démarrage à froid : avant que le système puisse fournir des recommandations pertinentes à un utilisateur, il faut que ce dernier ait fourni, implicitement ou explicitement, des appréciations pour un nombre suffisant de ressources. Un problème supplémentaire par rapport aux recommandations basées sur le contenu est que cette limitation s'applique également aux nouvelles ressources introduites dans le système. Des solutions à ces problèmes se trouvent dans les approches hybrides.

### 3.3. Systèmes hybrides

Les approches hybrides sont des approches qui combinent deux ou plusieurs approches de recommandation. Ces combinaisons permettent de bénéficier des avantages des approches utilisées, de pallier leurs inconvénients et de proposer des recommandations plus pertinentes. Par exemple, le démarrage à froid au niveau des items des approches collaboratives, peut

bénéficier des avantages des approches basées sur le contenu. En effet, dans une approche basée sur le contenu, seule la description de l'item est utilisée dans la recommandation, il n'y a donc pas besoin que l'item soit noté par un certain nombre d'utilisateurs avant de pouvoir être recommandé, comme c'est le cas dans les approches collaboratives.[11]

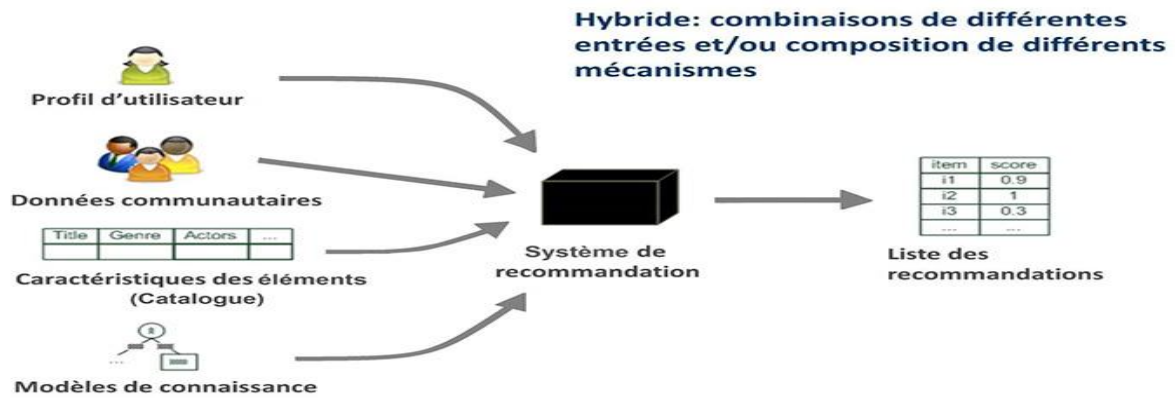


Figure 1.4 : Système de recommandation hybride

### 3.3.1. Hybridation pondérée

L'hybridation pondérée consiste à calculer le score d'un item candidat à la recommandation (un item potentiellement pertinent pour l'utilisateur et donc recommandable) grâce à une fonction définie effectuant la somme pondérée du score de l'item dans chaque approche de recommandation présente dans le système. Une fois les scores de tous les items candidats à la recommandation calculés, ceux-ci sont classés par ordre décroissant avant d'être présentés à l'utilisateur. Ce type d'hybridation nécessite que les systèmes de recommandations utilisés effectuent une tâche de prédiction de note.

### 3.3.2 Hybridation à bascule

Dans ce type d'hybridation, le système sélectionne une approche de recommandation plutôt qu'une autre en fonction d'un certain critère. Par exemple, si la confiance du système pour les résultats obtenus est insuffisante, une autre approche sera sélectionnée.[12]

### 3.3.3. Hybridation mixée

L'hybridation mixée consiste à présenter à l'utilisateur des recommandations issues de plusieurs approches de recommandation.[13]

### 3.3.4. Hybridation par combinaison de caractéristiques

Dans cette hybridation, des caractéristiques d'une approche de recommandation sont injectées dans une autre approche de recommandation. Par exemple, lors de l'utilisation d'une approche collaborative U2U, les items peuvent être remplacés par une caractéristique des approches basées sur le contenu, qui est la description des attributs des items. De ce fait, au lieu de traiter l'information « l'utilisateur apprécie l'item  $i$  », on traite l'information « l'utilisateur a apprécié l'attribut  $a$  ». Le calcul des similarités entre les utilisateurs est ensuite effectué en se basant sur les attributs au lieu de se baser sur les items.[14]

### 3.3.5. Hybridation en cascade

L'hybridation en cascade implique l'application successive des approches de recommandation du système. La première approche a pour but de générer un ensemble d'items candidats à la recommandation. Chaque approche de recommandation du système est ensuite appliquée à l'ensemble d'items candidats à la recommandation. Les items de cet ensemble sont donc petit à petit filtrés et l'ensemble final est recommandé à l'utilisateur.[15]

### 3.3.6. Hybridation par ajout de caractéristiques

Cette hybridation nécessite également une application successive des approches de recommandation disponibles. Cependant, chaque approche prend en paramètre les résultats de l'approche précédente et l'utilise comme information supplémentaire durant son exécution. Par exemple, une première approche de recommandation est utilisée afin de générer un ensemble d'items. Ces items sont intégrés au profil de l'utilisateur afin de l'enrichir. La deuxième approche de recommandation est ensuite appliquée sur ce profil enrichi afin de générer des items à proposer à l'utilisateur.[16]

### 3.3.7. Hybridation méta-niveau

Dans ce type d'hybridation, la première approche de recommandation génère un modèle qui est ensuite utilisé par la deuxième approche de recommandation. La deuxième approche de recommandation remplace complètement son entrée (la source des données) par le modèle généré par la première approche. Cette hybridation n'est pas applicable à toutes les approches de recommandation car elle nécessite l'utilisation d'approches basées sur des modèles.

### **3.4. Autres approches de recommandation**

En plus des trois catégories traditionnelles de systèmes de recommandation largement acceptées par la communauté de recherche, Burke (2002) a identifié trois autres types de systèmes de recommandation :

#### **3.4.1. Filtrage Démographique**

Ce type de filtrage est justifié par le fait que les individus ayant des caractéristiques démographiques similaires (sexe, âge, pays, etc.) auront également des préférences communes. Les recommandations démographiques classifient les utilisateurs en fonction de leurs attributs personnels pour proposer des suggestions basées sur des catégories démographiques spécifiques.

#### **3.4.2. Filtrage Basé sur l'Utilité**

Ce système recommande des éléments en évaluant l'utilité de chaque objet pour l'utilisateur. Il implique la création d'une fonction d'utilité personnalisée pour évaluer les objets recommandés en fonction des préférences individuelles de chaque utilisateur.

#### **3.4.3. Filtrage Basé sur la Connaissance**

Ce système suggère des objets en se basant sur des inférences concernant les besoins et les préférences des utilisateurs. Il nécessite une représentation des connaissances, comme des règles, sur la manière dont un élément répond spécifiquement aux besoins d'un utilisateur donné. L'utilisation de métadonnées, telles que des informations sur le contenu des éléments, est également courante pour améliorer la qualité des recommandations.

Ces approches complémentaires offrent une personnalisation plus avancée dans la recommandation de contenu en se basant sur l'utilité perçue par l'utilisateur et sur des connaissances spécifiques à ses besoins et préférences individuelles.

### 3.5. Avantages et les inconvénients des systèmes de recommandation

#### 3.5.1. Avantages

Les avantages des systèmes de recommandation incluent :

- Adapter les recommandations aux préférences et aux comportements individuels des utilisateurs.
- Réduire le temps nécessaire aux utilisateurs pour trouver les produits ou les informations qu'ils recherchent.
- Augmenter les ventes et les conversions grâce à des recommandations pertinentes.
- Aider les utilisateurs à découvrir de nouveaux produits qu'ils n'auraient peut-être pas trouvés autrement.
- Contribuer à la fidélisation des clients en créant une expérience d'achat personnalisée.

#### 3.5.2. Inconvénients

Les inconvénients des systèmes de recommandation incluent :

- Limiter la découverte de nouveaux produits.
- Comprendre la difficulté de profilage des utilisateurs.
- Rencontrer le problème du démarrage à froid pour les nouveaux utilisateurs.
- Faire face à un calcul croissant dans les systèmes complexes.
- Gérer la faible densité de la matrice Utilisateurs  $\times$  Éléments.

Ces inconvénients nécessitent une attention particulière lors de la conception et de la mise en œuvre de systèmes de recommandation, afin d'atténuer leur impact sur l'expérience utilisateur et d'améliorer l'efficacité des recommandations.

## 4. Systèmes de recommandation dans le domaine médical

Les avancées technologiques dans le domaine de la santé ont conduit à l'émergence de systèmes de recommandation dans le domaine médical. Ces systèmes, basés sur des algorithmes sophistiqués et des techniques d'intelligence artificielle, sont conçus pour aider les professionnels de la santé à prendre des décisions éclairées et à fournir des soins personnalisés aux patients. Ce rapport examine le fonctionnement, les applications, les avantages et les défis associés aux systèmes de recommandation dans le domaine médical.

### 4.1. Histoire

L'histoire des systèmes de recommandation dans le domaine médical remonte aux années 1970-1980, avec la production de référentiels dans les pays anglo-saxons et scandinaves par des sociétés savantes. Ces recommandations médicales sont des documents destinés à guider les praticiens dans la fourniture de soins basés sur des données scientifiques. En 1990, la création de l'Agence nationale pour le développement de l'évaluation médicale (ANDEM) a marqué le début d'une série d'évolutions aboutissant à la Haute Autorité de Santé (HAS) en France.

Avec l'avènement du Big Data et des algorithmes d'apprentissage statistique, les systèmes de recommandation médicale ont évolué vers des stratégies hybrides combinant exploration de données et prédiction par machine learning pour contourner les limitations des approches collaboratives.

Aujourd'hui, ces systèmes personnalisés contribuent à améliorer la qualité des soins et des décisions médicales en se basant sur les préférences et historiques des utilisateurs. [17]

### 4.2. Fonctionnement des systèmes de recommandation médicale

Les systèmes de recommandation médicaux ont pour objectif de fournir des recommandations personnalisées aux professionnels de santé et aux patients, en exploitant diverses techniques d'intelligence artificielle et d'analyse de données. Leur fonctionnement repose généralement sur un ensemble d'étapes. La figure 1.5 présente les étapes du fonctionnement d'un système de recommandation médical.

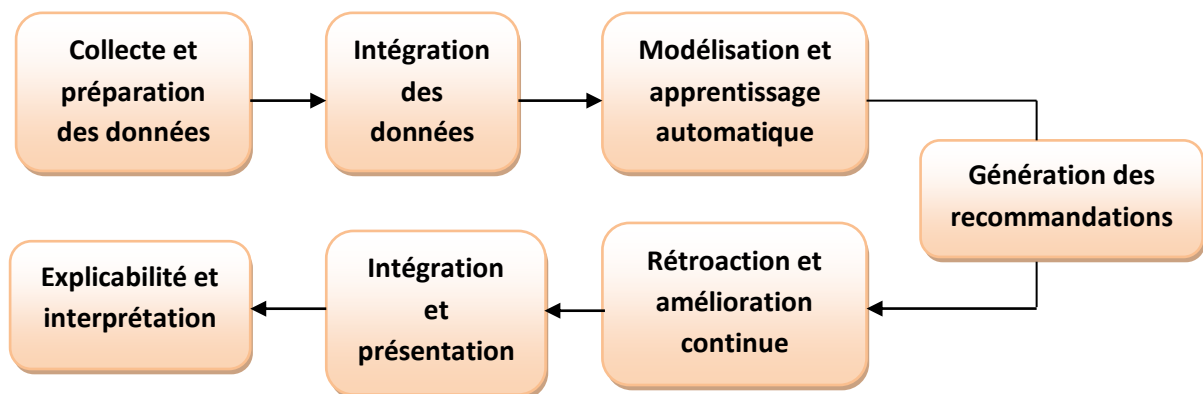


Figure 1.5 : Fonctionnement des systèmes de recommandation médicale

### Collecte et préparation des données

- Rassembler les données pertinentes à partir de sources diverses, telles que les dossiers médicaux électroniques, les bases de données cliniques, les capteurs de suivi de santé, etc.
- Nettoyer, formater et structurer les données pour les rendre exploitables par les algorithmes d'apprentissage automatique.

### Intégration des données

- Combiner différents types de données, notamment les données des patients (historique médical, symptômes, résultats de tests, etc.), les données cliniques (lignes directrices, protocoles de traitement, littérature médicale, etc.) et les données contextuelles (informations démographiques, géographiques, environnementales, etc.).
- Assurer l'interopérabilité et la normalisation des données provenant de sources hétérogènes.

### Modélisation et apprentissage automatique

- Appliquer des techniques d'apprentissage automatique, telles que le filtrage collaboratif, le filtrage basé sur le contenu, les réseaux de neurones, les arbres de décision, etc., pour construire des modèles de recommandation.
- Entraîner les modèles sur les données collectées pour qu'ils apprennent à reconnaître les schémas et les relations pertinentes.

### Génération des recommandations

- Utiliser les modèles entraînés pour générer des recommandations personnalisées en fonction des données spécifiques d'un patient ou d'un cas clinique.
- Les recommandations peuvent porter sur divers aspects, comme le diagnostic, le traitement, la prévention, le suivi, la gestion des maladies chroniques, etc.

## Explicabilité et interprétation

- Fournir des explications compréhensibles sur la logique et les raisons sous-jacentes aux recommandations générées, afin de gagner la confiance des professionnels de santé et des patients.
- Utiliser des techniques d'explicabilité de l'IA, telles que les modèles interprétables, les visualisations ou les descriptions textuelles, pour rendre les recommandations plus transparentes.

## Intégration et présentation

- Intégrer les recommandations dans les flux de travail cliniques existants, tels que les systèmes d'information hospitaliers, les plateformes de dossiers médicaux électroniques ou les applications mobiles de santé.
- Présenter les recommandations de manière conviviale et compréhensible pour les utilisateurs finaux, en utilisant des interfaces utilisateur adaptées et des visualisations claires.

## Rétroaction et amélioration continue

- Collecter les retours des professionnels de santé et des patients sur la pertinence et l'utilité des recommandations.
- Utiliser ces retours pour ajuster et améliorer continuellement les modèles de recommandation, en adoptant une approche d'apprentissage continu.

Tout au long de ce processus, il est essentiel de respecter les normes de confidentialité et de sécurité des données de santé, ainsi que de garantir la précision, la fiabilité et l'explicabilité des recommandations. De plus, une collaboration étroite entre les experts médicaux, les développeurs de systèmes de recommandation et les utilisateurs finaux est cruciale pour assurer l'adoption et l'efficacité de ces systèmes dans le domaine médical.

## 4.3. Applications des systèmes de recommandation dans le domaine médical

L'application des systèmes de recommandation dans le domaine médical présente un grand potentiel pour améliorer les soins de santé et les résultats pour les patients. Voici quelques exemples d'utilisation de ces systèmes:

### **Aide au diagnostic**

Les systèmes de recommandation peuvent analyser les symptômes, les antécédents médicaux et les données de tests d'un patient, puis recommander les diagnostics les plus probables. Cela peut aider les médecins à prendre des décisions plus précises et rapides.

### **Prescription de médicaments**

En prenant en compte les caractéristiques du patient, son historique médical, les interactions médicamenteuses possibles et les directives de traitement, ces systèmes peuvent recommander les meilleurs médicaments et posologies.

### **Plans de traitement personnalisés**

Les systèmes de recommandation peuvent suggérer des plans de traitement complets en combinant différentes options thérapeutiques (médicaments, procédures, régimes alimentaires, etc.) sur la base du profil du patient.

### **Gestion des maladies chroniques**

Pour les patients souffrant de maladies chroniques, ces systèmes peuvent recommander des changements de mode de vie, des suivis spécifiques et des interventions préventives pour éviter les complications.

### **Recherche clinique**

Dans le cadre d'essais cliniques, les systèmes de recommandation peuvent aider à identifier les patients éligibles, à les assigner aux bons groupes d'étude et à suivre leur progression.

### **Recherche des articles médicaux**

Les systèmes de recommandation peuvent aussi être appliqués pour aider les professionnels de la santé à trouver des articles médicaux pertinents. En analysant les besoins spécifiques et les domaines d'intérêt des médecins, ces systèmes peuvent suggérer des publications et des études pertinentes. Cela permet aux professionnels de rester à jour avec les dernières recherches et avancées dans leur domaine, améliorant ainsi leur pratique et la qualité des soins qu'ils fournissent.

Cependant, il est important de noter que ces systèmes doivent être considérés comme des outils d'aide à la décision et non comme des substituts aux médecins. Une supervision humaine et une évaluation clinique restent essentielles pour garantir des soins sûrs et efficaces.

### 4.4. Défis spécifiques au domaine médical

#### Confidentialité et sécurité des données

- Protection des données de santé sensibles et respect des réglementations strictes en matière de confidentialité.
- Mise en place de mesures de sécurité robustes pour prévenir les fuites de données.

#### Précision et fiabilité des recommandations

- Nécessité d'une précision élevée pour éviter les erreurs potentiellement graves.
- Importance de la validation clinique et de l'utilisation de données probantes.

#### Explicabilité et transparence

- Capacité à expliquer la logique derrière les recommandations pour gagner la confiance des professionnels de santé.
- Éviter les "boîtes noires" et favoriser la transparence des algorithmes.

#### Intégration dans les flux de travail cliniques

- Intégration harmonieuse des systèmes de recommandation dans les processus et les systèmes existants.
- Minimisation de la charge de travail supplémentaire pour les professionnels de santé.

#### Acceptation par les patients et les professionnels

- Garantir que les recommandations sont compréhensibles et acceptables pour les patients.
- Surmonter la résistance potentielle des professionnels de santé envers les systèmes automatisés.

### 4.5. Exemples de systèmes de recommandation médicaux existants

**Watson for Oncology (IBM)** : Système d'aide à la décision pour les traitements contre le cancer, basé sur des données cliniques et des cas réels.

**Gère Santé (Québec)** : Application mobile offrant des recommandations personnalisées pour la gestion des maladies chroniques et la promotion d'un mode de vie sain.

**Recommandation de médicaments (Lareb, Pays-Bas)** : Système de recommandation pour la prescription de médicaments, en tenant compte des interactions médicamenteuses et des contre-indications.

**MedWatcher (Stanford)** : Outil d'apprentissage automatique pour détecter les effets secondaires potentiels des médicaments et recommander des ajustements de traitement.

**Médiale (France)** : Plateforme de recommandation de parcours de soins personnalisés pour les patients atteints de maladies chroniques.

Ces exemples illustrent la diversité des applications possibles des systèmes de recommandation dans le domaine médical, allant du diagnostic au suivi en passant par la prévention et la gestion des maladies chroniques. Cependant, il est crucial de relever les défis spécifiques à ce domaine pour assurer une adoption réussie et bénéfique de ces systèmes.

## 5. Conclusion

Dans ce chapitre, nous avons défini la notion des systèmes de recommandation, et on a présentés les trois approches les plus utilisées par ces systèmes qui sont le filtrage collaboratif et filtrage basé contenu ainsi que leur hybridation et leurs limitations,

Dans le chapitre suivant, nous présenterons les systèmes de recommandation sensible au contexte, nous verrons les différentes techniques de recommandations, puis nous présenterons les différents domaines d'application ainsi qu'une étude de cas dans le domaine de la santé.

# **Chapitre 2**

---

**Système de recommandation  
sensible au contexte (CARS)**

### **1. Introduction**

Les systèmes de recommandation jouent un rôle essentiel pour améliorer les interactions entre les utilisateurs et les ressources informatiques. Cependant, les systèmes traditionnels de recommandation se concentrent principalement sur deux dimensions (utilisateur et contenu), cela ne couvre pas tous les aspects importants de la recommandation, car il existe d'autres facteurs influençant les choix d'information pertinente, tels que quand, où, pourquoi, etc.

Conscients de cet aspect, une branche croissante de systèmes de recommandation sensible au contexte (CRC, Context-Aware Recommendation Systems) se développe dans la littérature récente. Ces systèmes se distinguent par leur utilisation du contexte de l'utilisateur pour modéliser des données telles que sa localisation, l'heure et les personnes proches. Une fois ces données mises en place, le filtrage se fait en examinant le contexte courant de l'utilisateur et en sélectionnant les contenus disponibles qui correspondent le mieux à la situation décrite.

En résumé, les systèmes de recommandation sensibles au contexte constituent une avancée importante dans le domaine de l'informatique, permettant de fournir des suggestions pertinentes et adaptées aux différents contextes et situations rencontrés par les utilisateurs.

### **2. Contexte**

#### **2.1. Définition du contexte**

La notion de contexte est complexe et difficile à définir de manière générale et opérationnelle. En informatique, plusieurs définitions ont été proposées par différents chercheurs au fil des années.

Au départ, les chercheurs ont tenté de définir le contexte en énumérant tous les éléments qui peuvent être considérés comme des informations contextuelles. Ces éléments sont des informations non fonctionnelles de l'application, mais qui peuvent avoir un impact sur celle-ci. Par exemple, Schilit et Theimer, ont défini le contexte comme étant "la localisation et l'identité des personnes et objets à proximité ainsi que les modifications pouvant intervenir sur ces objets".

## **Chapitre II : Système de recommandation sensible au contexte (CARS)**

---

Dans le domaine des systèmes de recommandation, Anind K. Dey et Gregory, ont proposé une définition plus précise du contexte. Selon eux, "le contexte est l'ensemble de toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité". En d'autres termes, une information fait partie du contexte si elle a une influence sur l'interaction entre deux entités, une entité pouvant être un acteur, un lieu ou un objet de l'environnement considéré comme utile à l'interaction entre la personne et le système.

En résumé, la définition du contexte en informatique est une notion complexe et multidimensionnelle qui peut varier en fonction du domaine d'application. Les définitions proposées par les chercheurs ont pour objectif de fournir une compréhension plus précise et opérationnelle de cette notion pour faciliter la conception et le développement de systèmes informatiques sensibles au contexte.

### **2.2. Définition de sensibilité au contexte (context-awareness)**

Le concept de sensibilité au contexte a été introduit pour la première fois dans le domaine des Interactions Homme-Machine (IHM) par Weiser en 1991, dans le but de réconcilier le monde virtuel et le monde physique. Selon lui, les technologies les plus performantes sont celles qui sont associées à la vie quotidienne. La première définition de la sensibilité au contexte a été proposée par en 1993, Schilit et al., qui l'ont définie comme la capacité d'un système à s'adapter au contexte d'exécution en fonction de son emplacement, des personnes à proximité, des équipements accessibles, etc.[18]

Hull et al. Ont défini la sensibilité au contexte comme la capacité des appareils à détecter, ressentir, interpréter et répondre aux aspects de l'utilisateur et de l'environnement de l'appareil. Ryan et al. Ont défini les applications sensibles au contexte comme des applications qui surveillent les entrées de capteurs environnementaux et permettent aux utilisateurs de sélectionner les contextes en fonction de leurs intérêts et activités.[19]

Brown a défini les applications sensibles au contexte comme celles qui fournissent automatiquement des informations ou proposent des actions en fonction des contextes des utilisateurs détectés par les capteurs. Les actions peuvent présenter des informations à l'utilisateur pour exécuter un programme ou configurer l'interaction graphique en fonction du contexte.

Dey a proposé une définition plus générale de la sensibilité au contexte, en la définissant comme la capacité d'un système à utiliser le contexte pour fournir des

## **Chapitre II : Système de recommandation sensible au contexte (CARS)**

---

informations et/ou des services pertinents à l'utilisateur, où la pertinence dépend de la tâche de l'utilisateur.

Enfin, Byun et Cheverst ont donné une définition plus largement acceptée, en définissant un système contextuel comme celui qui peut extraire, interpréter et utiliser des informations contextuelles et adapter les fonctionnalités au contexte de l'usage courant. Toutes ces définitions concernent la capacité d'un appareil à adapter son comportement en fonction du contexte utilisateur.[20]

### **2.3. Types de contexte**

Cette section présente une brève description des différents types de contexte généralement utilisés dans les systèmes de recommandation. Le contexte peut être catégorisé en contexte d'utilisateur, contexte d'élément et contexte de décision. Les prochaines sous-sections présentent succinctement ces différents contextes pour une meilleure compréhension.

#### **2.3.1. Contexte de l'utilisateur**

Le contexte utilisateur représente les caractéristiques contextuelles, telles que l'âge, le sexe, les amis, etc. Le contexte d'un utilisateur  $u$  est noté  $U_c$  et représenté comme l'ensemble des paires caractéristique-valeur. Par exemple,  $U_c = \langle \text{Sexe} : \text{Homme} \rangle ; \langle \text{Age} : 18 \rangle$  fournit des informations contextuelles sur l'utilisateur  $u$  dont le sexe est homme et qui a 18 ans.

#### **2.3.2. Contexte de l'item**

Le contexte de l'item représente les caractéristiques contextuelles des items pour leur caractérisation. Par exemple, le genre, le sous-genre et acteurs peuvent être considérés comme des caractéristiques contextuelles des items de film. Le contexte d'un item  $I$  est noté  $I_c$  et représenté comme l'ensemble des paires caractéristique-valeur. Par exemple,  $I_c = \langle \text{Genre} : \text{Sci Fi} \rangle ; \langle \text{Réalisateur} : \text{James Cameron} \rangle$  fournit des informations contextuelles sur l'item  $I$  qui a Science-Fiction comme genre et James Cameron en tant que réalisateur.

#### **2.3.3. Contexte de décision**

Le contexte de décision représente les caractéristiques contextuelles, telles que l'emplacement, le compagnon et l'heure, qui sont utilisées par les utilisateurs pour l'utilisation d'un item. En fonction des valeurs de contexte d'utilisateur et d'élément, les contextes de décision changent également. Le contexte d'une décision  $D$  est noté  $D_c$  et

## **Chapitre II : Système de recommandation sensible au contexte (CARS)**

---

représenté comme l'ensemble des paires de valeurs de caractéristiques. Par exemple,  $D_c = \langle \text{Emplacement : Domicile} \rangle ; \langle \text{Compagnon : Enfants} \rangle$  représente un utilisateur qui a regardé un film à la maison avec ses enfants.

### **2.4. Caractéristiques du contexte**

Dans ce qui suit, nous allons présenter les caractéristiques techniques de l'information contextuelle :

**Continuité :** Lorsque l'utilisateur se déplace d'une localisation à une autre, entraînant un changement dans la valeur du contexte, ou lorsqu'il acquiert de nouvelles ressources, le contexte évolue. Il est essentiel que les systèmes fonctionnent de manière continue malgré ces changements incessants du contexte.

**Hétérogénéité:** Le contexte est capturé à partir de diverses sources, ce qui entraîne une grande diversité en termes de modélisation, de traitement et de qualité. Par exemple, les ordinateurs portables, les smartphones et les tablettes présentent des différences significatives à tous les niveaux (matériel, logiciel, communications, etc.).

**Imprévisibilité :** Les systèmes doivent être capables de gérer l'imprévisibilité des changements environnementaux et des interactions utilisateur. Certains éléments du contexte peuvent ne pas être connus à l'avance par le système, posant ainsi un défi en termes d'adaptabilité.

**Interdépendance:** Les informations contextuelles peuvent être interdépendantes, ce qui signifie que la modification d'une information peut influencer une autre information du contexte.

**Contexte imparfait :** La qualité du contexte peut varier en fonction de sa source. Le contexte peut être imprécis, erroné ou même inconnu, ce qui souligne l'importance de gérer efficacement cette incertitude.

Ces caractéristiques techniques soulignent la complexité et la diversité des informations contextuelles et mettent en lumière les défis auxquels sont confrontés les systèmes sensibles au contexte pour garantir une prise en compte adéquate et efficace du contexte dans leurs fonctionnements.

### 3. Information contextuelle

#### 3.1. Définition

Une information contextuelle est une information qui fournit un contexte ou des détails supplémentaires permettant de mieux comprendre une situation, un événement ou une donnée.

Quelques exemples d'informations contextuelles :

**Contexte spatial/géographique :** L'emplacement, la localisation d'un événement ou d'une entité.

**Contexte temporel :** Le moment, la période, la date/heure auxquels quelque chose s'est produit.

**Contexte environnemental :** Les conditions météorologiques, le niveau de luminosité, le niveau sonore, etc.

**Contexte utilisateur :** Les préférences, l'historique, le profil d'un utilisateur.

**Contexte situationnel :** Les circonstances, l'activité en cours, l'état d'une personne ou d'un système.

**Contexte relationnel :** Les relations, les liens entre différentes entités ou personnes impliquées.

Les informations contextuelles aident à donner un sens plus complet et plus riche aux données ou événements. Elles fournissent des détails qui permettent de mieux interpréter et comprendre une situation particulière.

Dans de nombreux domaines comme l'informatique, le marketing, la prise de décision, etc. le contexte est essentiel pour analyser correctement les données et en tirer les bonnes conclusions ou recommandations.

#### 3.2. Sources d'informations contextuelles

On dénote trois grands types de sources d'informations contextuelles : explicite, implicite ou inférée.

**Explicite:** l'information sur le contexte est déjà incluse dans les données ou directement demandée à l'utilisateur. Par exemple, sur des plateformes d'achats en ligne, on peut

## **Chapitre II : Système de recommandation sensible au contexte (CARS)**

---

demander à l'utilisateur s'il effectue un achat pour des raisons personnelles ou professionnelles ou encore pour offrir un cadeau au moment de l'achat. On peut aussi, lors de l'inscription de l'utilisateur sur le système, lui demander de remplir un formulaire contenant des informations personnelles le concernant, telles que son âge, sa profession, etc.

**Implicite:** l'information est obtenue à partir des données ou de l'environnement dans lequel se trouve un utilisateur sans la lui demander explicitement. Pour des plateformes de recommandation de sites touristiques à partir des Smartphones, on peut par exemple connaître la situation géographique exacte d'un individu au moment d'effectuer une recommandation.

**Inférée:** l'information est obtenue à l'aide de méthodes d'exploitation et d'exploration des données. Par exemple, l'identité d'une personne parcourant les chaînes de télévision peut ne pas être explicitement connue pour une société de télévision par câble. Cependant, le système peut arriver à apprendre le moment de la journée, la chaîne et le type de programme regardés par les différents utilisateurs d'un même foyer (père, mère, enfants...) et ceci avec une précision acceptable en utilisant des techniques de data mining.

### **4. Utilisation du contexte dans les systèmes de recommandation**

Les systèmes de recommandation traditionnels reposent sur deux concepts clés, à savoir l'utilisateur et l'item, pour prédire les évaluations. En d'autres termes, un système de recommandation peut être vu comme une fonction qui prend des utilisateurs et des items en entrée pour prédire des évaluations :  $R : U \times I \rightarrow R$ . Lorsque le concept de contexte est introduit, il devient un paramètre supplémentaire dans la fonction d'évaluation, transformant ainsi la fonction en :  $R : U \times I \times C_{sit} \rightarrow R$ .

Le contexte, représenté sous forme d'un vecteur contenant diverses informations contextuelles, devient un élément clé dans cette approche. Pour mettre en œuvre cette méthode multidimensionnelle et évaluer sa performance, il est nécessaire de collecter des données à la fois sur les items (évaluations) et sur le contexte. Les résultats empiriques démontrent une amélioration significative dans la prédiction des systèmes sensibles au contexte par rapport à ceux qui n'intègrent pas cette dimension. Cette amélioration est due à la prise en compte en temps réel de la situation contextuelle de l'utilisateur cible lors du calcul des prédictions.

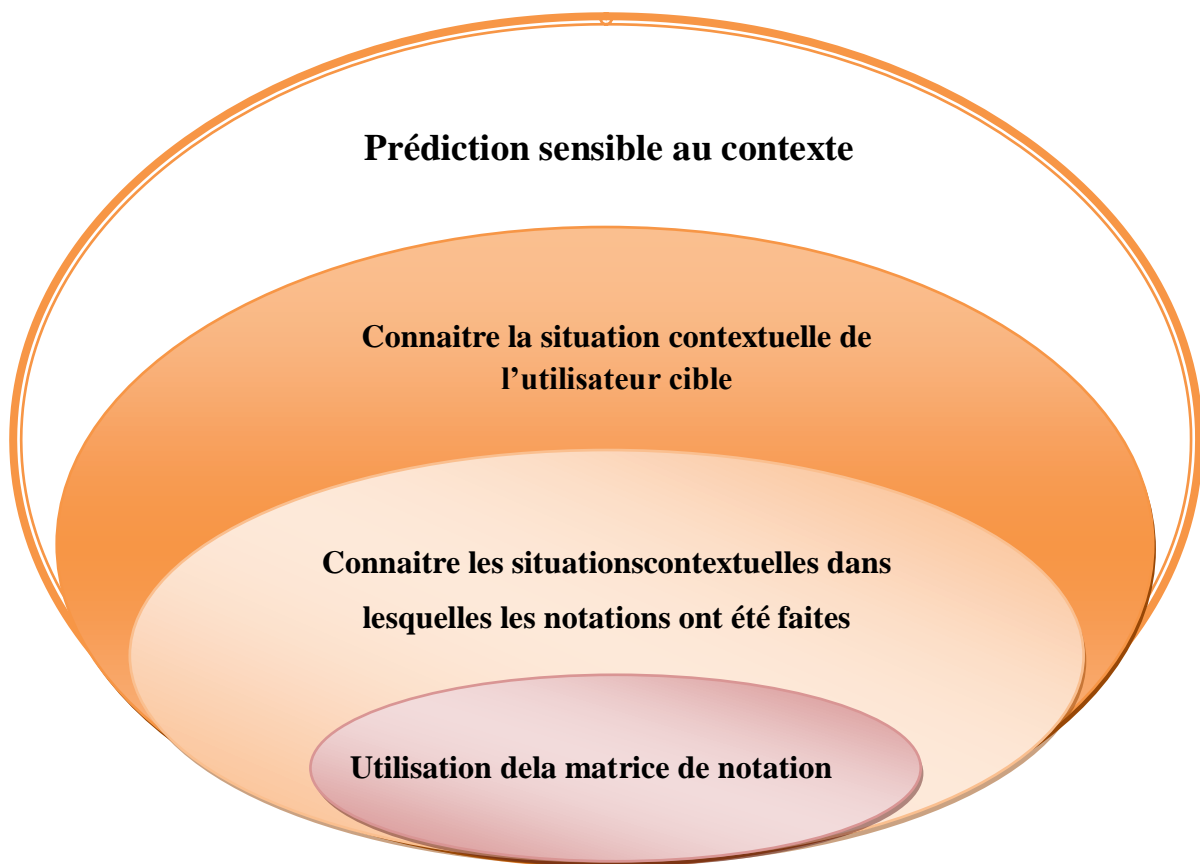


Figure 2.1 : Utilisation du contexte dans les systèmes de recommandation

### 5. Formalisation du contexte

La source de données de base utilisée dans le système de recommandation est la matrice  $\text{user} \times \text{item} \rightarrow \mathbb{R}$ , qui contient toutes les notations données par les utilisateurs pour  $n$  items. L'ensemble des utilisateurs est noté  $U = \{u_1, u_2, \dots, u_m\}$ , et l'ensemble des items est noté  $I = \{i_1, i_2, \dots, i_n\}$ . Si un utilisateur  $u$  évalue un article  $i$ , cela génère une note notée  $r_{ui}$ .

Les systèmes de recommandation contextuels étendent cette technologie en incorporant les informations contextuelles de l'utilisateur. La matrice utilisée dans ces systèmes est  $U \times I \times C_{\text{sit}} \rightarrow \mathbb{R}$ , où  $C_{\text{sit}}$  représente une situation contextuelle décrivant le contexte dans lequel l'utilisateur a évalué l'item.

Formellement, le contexte est représenté comme un ensemble de dimensions contextuelles, comme suit :  $C = \{C_1, C_2, \dots, C_k\}$ , où  $C_k$  présente une dimension du contexte,

## Chapitre II : Système de recommandation sensible au contexte (CARS)

comme le temps. Une valeur spécifique dans une dimension contextuelle, représente une condition contextuelle et est définie comme suit :  $Cd = \{cd1, cd2, \dots, cdk, \dots, cds\}$ , où  $s$  est le nombre de valeurs de variante de la dimension  $Cd$  et  $cdk$  est l'une des valeurs qu'une dimension peut prendre.

Par exemple, dans la dimension Temps, il existe plusieurs valeurs (telles que matin, midi, après-midi et soir) qui se présentent comme suit :  $Ctime = \{\text{matin, midi, après-midi, soir}\}$ . Désormais, les données du jeu de données sont représentées dans un espace de notation multidimensionnel. Chaque note est représentée par les tuples  $(u, i, Csit)$ . Par exemple :  $(u1, i1, 5, \text{weekend, home, friend})$  signifie  $u1$  notés  $i1$  à 5 dans la situation contextuelle : le temps est le week-end, l'emplacement est la maison et le compagnon est son ami.

### 6. Approches d'incorporation du contexte

Les approches d'incorporation du contexte utilisateur sont essentielles pour améliorer la navigation et les recommandations personnalisées sur les sites web et les applications. Le contexte utilisateur, définit comme une délimitation précise de l'activité des utilisateurs dans un environnement donné, inclut des éléments tels que l'identité de l'utilisateur, l'emplacement et le périphérique utilisé. Il guide l'utilisateur en lui fournissant des informations pertinentes pour faciliter son expérience de navigation.

Diverses techniques sont utilisées pour intégrer le contexte dans les systèmes de recommandation, telles que le pré-filtrage où seules les données correspondant au contexte sont sélectionnées avant d'appliquer une méthode de recommandation traditionnelle, ou le post-filtrage qui ajuste les recommandations en fonction du contexte utilisateur. L'évaluation des scores dans les systèmes de recommandation contextuels repose sur une fonction spécifique prenant en compte les utilisateurs, les éléments et les contextes pour générer des scores pertinents. La figure 2.2 présente ces approches : approche pré-filtrage contextuel, post-filtrage contextuel et modélisation contextuelle.

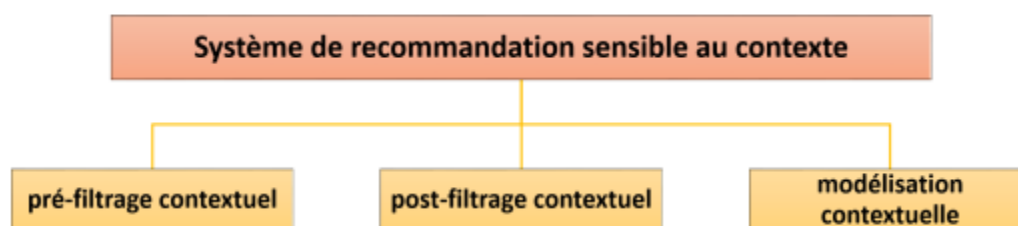


Figure 2.2 : Les approches des systèmes de recommandation sensible au contexte

### 6.1. Approche de Pré-filtrage Contextuel (contextual pre-filtering)

Le système de pré-filtrage contextuel contient un filtre de prétraitement, qui est utilisé pour contextualiser les données d'entrée des systèmes de recommandation traditionnels. Ce filtre est utilisé pour sélectionner ou construire l'ensemble de données pertinent.

Ensuite, les préférences peuvent être prédites à l'aide d'un système de recommandation traditionnel sur les données sélectionnées. De ce fait, les méthodes de recommandation traditionnelles sont utilisées pour prédire les préférences des utilisateurs en fonction de données contextualisées.

Prenons l'exemple d'un utilisateur qui souhaite trouver des hôtels dans une saison particulière le système de recommandation utilise uniquement les données de préférence de l'utilisateur et des autres utilisateurs pour cette saison particulière.

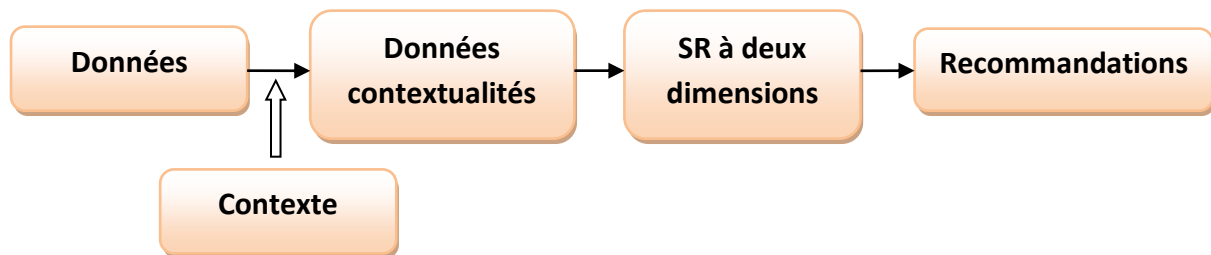


Figure2.3 : Pré-filtrage Contextuel.

Il existe différents types de pré-filtrage, tels que le pré-filtrage exact (Exact Pre-Filtering) et le pré-filtrage généralisé (Generalized Pre-Filtering). Exact Pre-Filtering (EPF) sélectionne toutes les opérations relatives exactement au contexte spécifique, tandis que GeneralizedPre-Filtering sélectionne toutes les opérations visées à un contexte spécifique basée sur la généralisation de l'information contextuelle (par exemple, un "Cadeau pour ses parents" peut être généralisé à "cadeau pour ses relations").

### 6.2. Approche de Post-filtrage Contextuel (contextual post-filtering)

Dans cette approche, les informations contextuelles ne sont prises en compte qu'à la dernière étape de la recommandation. C'est seulement à la fin que l'ensemble de recommandations résultant de ce processus est ajusté (contextualisé) pour chaque utilisateur à l'aide des informations de contexte cible. Tout comme les approches de pré-filtrage

contextuel, le post-filtrage contextuel est compatible avec les algorithmes de recommandation traditionnels, avant qu'un filtre ne soit appliqué pour sélectionner les items pertinents.

Les méthodes de post-filtrage peuvent fournir une réponse rapide à l'utilisateur. Les tâches de recommandation traditionnelles peuvent être effectuées à l'avance. Lorsque des recommandations contextuelles sont nécessaires, le système peut contextualiser (filtrer ou ajuster) les résultats des recommandations.

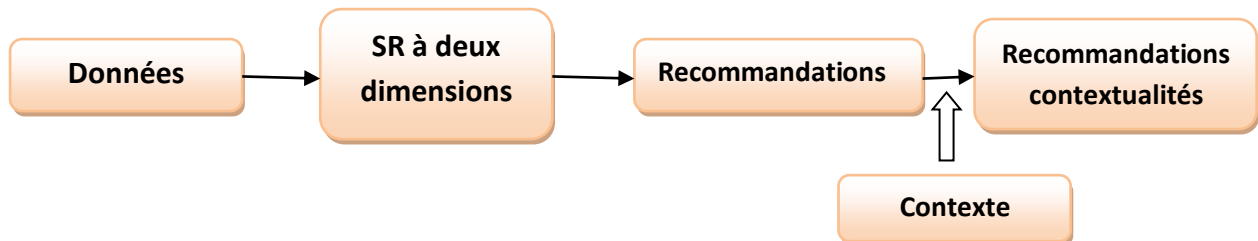


Figure 2.4: Post-filtrage Contextuel.

### 6.3. Approche de la Modélisation Contextuelle (contextual modeling)

L'approche de modélisation contextuelle dans les systèmes de recommandation implique l'intégration directe du contexte dans le processus de prédiction des notations inconnues. Contrairement aux méthodes de pré-filtrage et de post-filtrage, cette approche considère le contexte comme une dimension supplémentaire essentielle dans le modèle de recommandation. Selon Liu (2014), cette méthode utilise une fonction de recommandation tridimensionnelle (3D) qui prend en compte simultanément l'utilisateur, l'élément à recommander, et le contexte spécifique de la recommandation. Par exemple, pour recommander des films à un utilisateur, le contexte pourrait inclure des informations sur l'heure de la journée, l'emplacement géographique, ou l'humeur de l'utilisateur. En intégrant ces dimensions contextuelles, cette approche permet de personnaliser les recommandations de manière plus précise et adaptative, capturant ainsi les variations dans les préférences et les besoins des utilisateurs en fonction de différents scénarios. Cependant, cela nécessite souvent des données contextuelles riches et des techniques de modélisation avancées pour traiter efficacement les interactions complexes entre l'utilisateur, l'élément et le contexte.



Figure 2.5: Modélisation contextuelle

### 7.Exemples sur des systèmes de recommandation sensibles au contexte

Les systèmes de recommandation sensibles au contexte se déploient dans divers secteurs, exploitant leur capacité à personnaliser les suggestions en fonction du contexte spécifique de l'utilisateur. Voici quelques domaines clés d'application :

#### Micro-profiling

L'approche micro-profiling a pour but de recommander des chansons/artistes inconnus à un utilisateur. Le type du système de recommandation est le pre-filtering car les données sont filtrées selon le contexte (le temps : moment de la journée, jour, mois ou année). Dans cette approche, supposant que les goûts des utilisateurs changent selon une période et peuvent être similaires dans une même période. Par exemple, un utilisateur préfère écouter un certain genre de chanson quand il travaille, et un autre genre de chanson avant de dormir. On dispose d'un profil utilisateur dans lequel on affectera les chansons/artistes écoutés par cet utilisateur. Ce profil sera décomposé de façon à ce qu'il s'adapte à chaque contexte : dans cette approche, le temps représente le moment de la journée, il sera divisé en plusieurs segments en fonction des chansons que l'utilisateur écoute le plus durant chaque moment (segment de temps) de la journée.

L'utilisation de ces micro-profiles améliore l'exactitude de la recommandation. L'un des problèmes posés par cette méthode, réside dans le fait que le partitionnement du temps diffère d'un utilisateur à un autre (au lieu d'être fixe pour tous les utilisateurs), ce qui risque de diminuer la précision du système.

#### Amazon

Un système de recommandation sensible au contexte similaire à celui d'Amazon pourrait fonctionner de la manière suivante:

Amazon collecte et exploite de nombreuses informations contextuelles sur ses utilisateurs pour personnaliser ses recommandations de produits. Voici quelques éléments contextuels clés utilisés:

**Profil utilisateur:** Amazon rassemble des données démographiques de base comme l'âge, le sexe, la localisation, etc. pour mieux cerner les intérêts généraux.

## Chapitre II : Système de recommandation sensible au contexte (CARS)

**Historique d'achat:** Le système analyse les achats précédents de l'utilisateur pour détecter ses préférences et centres d'intérêt.

**Activité de navigation:** Amazon suit les pages web consultées, les produits visionnés, les temps de consultation pour comprendre ce qui intéresse l'utilisateur.

**Évaluations et notations:** Les notes et avis laissés par l'utilisateur sur des produits donnent des indices sur ses goûts.

**Contexte social:** Amazon peut exploiter les données des réseaux sociaux et des contacts pour identifier les influenceurs.

**Contexte temporel:** Le système adapte ses recommandations en fonction des périodes comme les vacances, la météo saisonnière, etc.

**Contexte géographique:** La localisation de l'utilisateur permet des recommandations localement pertinentes.

En combinant ces multiples signaux contextuels via des algorithmes d'apprentissage machine, Amazon peut dresser un profil de préférences très riche pour chaque client. Le moteur de recommandation suggère alors des produits susceptibles de l'intéresser en se basant sur ceux achetés par d'autres utilisateurs similaires.

La grande force est la prise en compte de ce contexte multidimensionnel pour affiner constamment la pertinence des recommandations, au-delà des simples critères de produit. C'est ce qui permet à Amazon de personnaliser efficacement l'expérience utilisateur.[21]



Figure 2.6 : Barre d'outils du site Amazon.

### **8. Etude de l'existant**

Un système de recommandation d'articles médicaux a été proposé par des chercheurs de l'Université de Pittsburgh en 2015. Ce système vise à aider les médecins à rester à jour avec les dernières recherches et découvertes pertinentes pour leur spécialité médicale. L'algorithme combine le filtrage collaboratif basé sur les similarités entre les médecins et leurs collègues, avec des données contextuelles telles que la spécialité médicale, les sujets d'intérêt, les publications récentes du médecin, ainsi que le contenu sémantique des articles. Le système a été développé pour résoudre le problème de la surcharge d'informations médicales et aider les médecins à identifier efficacement les articles les plus pertinents parmi l'immense quantité de littérature scientifique publiée chaque année. Les résultats expérimentaux ont montré que l'incorporation de données contextuelles, en particulier la spécialité médicale et les sujets d'intérêt, améliore considérablement la précision des recommandations par rapport aux techniques de filtrage collaboratif standard. Le système a été déployé dans un réseau hospitalier universitaire, aidant les médecins à rester informés des dernières avancées dans leur domaine d'expertise.

Dans cet exemple, l'algorithme de recommandation exploite les données contextuelles telles que la spécialité médicale du médecin et le contenu sémantique des articles, en plus des similitudes entre collègues, afin de recommander des articles médicaux pertinents et à jour pour chaque médecin.

Cependant, elle ne s'adresse pas spécifiquement aux médecins généralistes et ne détaille pas la gestion des nouveaux utilisateurs ni l'importance des évaluations d'articles. Notre solution répond à ces lacunes en distinguant clairement les scénarios pour les généralistes et les spécialistes, en tenant compte de leurs évaluations respectives et en gérant les recommandations initiales en fonction de leurs domaines d'intérêt. De plus, elle met l'accent sur l'expérience utilisateur d'une application destinée aux médecins. En combinant les forces des deux approches, on pourrait obtenir un système de recommandation complet et adapté aux différents profils médicaux.

### **9. Conclusion**

Nous avons discuté dans ce chapitre sur l'importance du contexte dans le domaine des systèmes de recommandation. Nous avons d'abord présenté dans la première partie la notion du contexte dans les systèmes de recommandation. Puis, nous avons étalé les terminologies

## **Chapitre II : Système de recommandation sensible au contexte (CARS)**

---

utilisées dans le domaine des SRs. Puis nous avons détaillé les différentes approches utilisées pour intégrer l'information contextuelle. Nous avons clôturé ce chapitre par l'ensemble des problèmes et limites rencontrés par les systèmes de recommandation sensible au contexte, d'où la nécessité d'intégrer d'autres informations.

# Chapitre 3

---

**Analyse et conception du système  
Medics**

## 1. Introduction

L'utilisation de l'Unified Modeling Language (UML) dans le développement d'applications web joue un rôle crucial pour planifier, concevoir et comprendre la complexité des systèmes informatiques sur les différents appareils.

L'UML fournit un ensemble de diagrammes visuels qui permettent aux développeurs de représenter graphiquement la structure, le comportement et les interactions des différents composants. Ces diagrammes offrent une vue d'ensemble claire et organisée de l'application, ce qui facilite la communication et la collaboration au sein de l'équipe de développement.

Dans le contexte des applications web, l'UML est utilisé pour modéliser les différents aspects de l'application, y compris l'interface utilisateur, les fonctionnalités principales, les flux de données et les interactions avec les utilisateurs. Par exemple, les diagrammes de cas d'utilisation aident à définir les fonctionnalités de l'application et les scénarios d'utilisation, tandis que les diagrammes de séquence décrivent la manière dont les différents composants de l'application interagissent entre eux.

En utilisant l'UML, les développeurs peuvent mieux comprendre la structure et le comportement de l'application, ce qui facilite le processus de développement et permet de créer des applications plus robustes, évolutives et conviviales pour les utilisateurs finaux. En résumé, l'UML est un outil essentiel pour concevoir et développer des applications de qualité supérieure.

## 2. Langage de modélisation UML

### 2.1. Définition

L'Unified Modeling Language (UML), traduit en français par Langage de Modélisation Unifié, est un langage de modélisation visuel utilisé dans le domaine de l'ingénierie logicielle pour spécifier, visualiser, concevoir et documenter les systèmes logiciels. Créé à la fin des années 1990, l'UML est devenu un standard de facto dans l'industrie du développement logiciel, offrant un ensemble de notations et de conventions pour représenter graphiquement

la structure, le comportement, les interactions et les fonctionnalités des différents composants d'un système logiciel.

### **2.2. Concepts fondamentaux de l'UML**

Les concepts fondamentaux de l'UML sont les piliers sur lesquels repose la modélisation des systèmes logiciels. Voici un aperçu de chacun de ces concepts :

#### **Objets**

Ce sont des entités représentant des instances individuelles d'une classe, avec des caractéristiques et des comportements. Ils sont souvent visualisés dans les diagrammes de séquence ou d'états.

#### **Classes**

Elles définissent la structure et le comportement des objets en spécifiant leurs attributs et leurs méthodes. Les classes organisent et catégorisent les objets dans un système, représentées sous forme de boîtes dans les diagrammes de classes.

#### **Relations**

Elles décrivent les associations et interactions entre les classes et objets, comme l'association, l'héritage ou la dépendance, permettant de modéliser la structure et le comportement des systèmes logiciels.

#### **Comportements**

Ils décrivent le fonctionnement dynamique d'un système, représentant les interactions entre ses composants. Modélisés à l'aide de diagrammes de séquence, d'activité ou d'états, ils capturent les flux d'exécution et les interactions utilisateur-système.

#### **Cas d'utilisation**

Ils décrivent les interactions entre les utilisateurs et le système, représentant les fonctionnalités ou actions possibles. Modélisés dans les diagrammes de cas d'utilisation, ils mettent en lumière les acteurs et les scénarios d'utilisation.

États

Ils décrivent les conditions internes d'un objet ou système, ainsi que les transitions entre ces états. Utilisés pour modéliser le comportement des objets, ils sont représentés dans les diagrammes d'états.

En combinant ces concepts, les développeurs peuvent créer des modèles UML complets qui capturent tous les aspects d'un système logiciel, de sa structure à son comportement, facilitant ainsi sa conception, sa compréhension et sa communication.

2.3. Différents types de diagrammes UML

L'Unified Modeling Language (UML) propose une gamme diversifiée de diagrammes qui offrent des perspectives uniques sur les systèmes logiciels. Voici une analyse détaillée des deux principaux types de diagrammes UML :

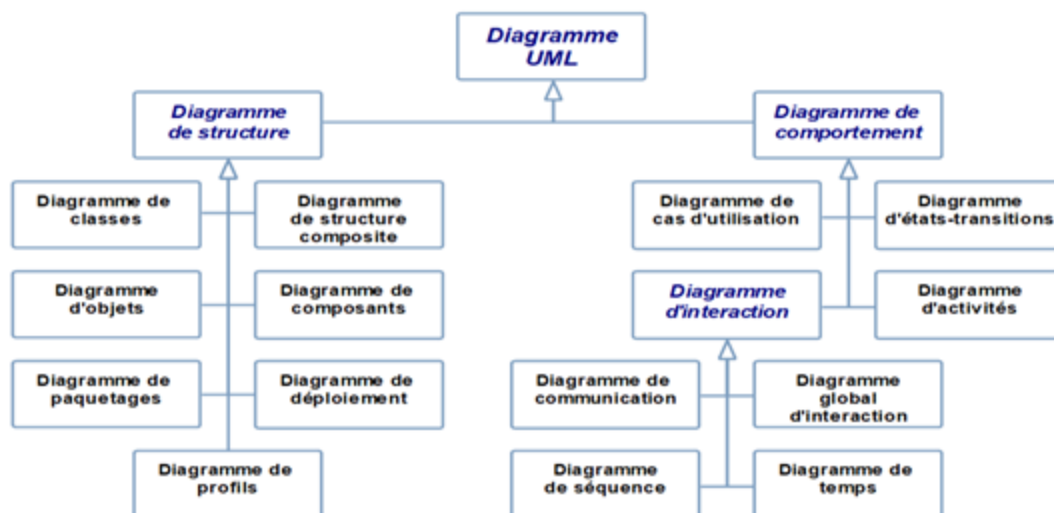


Figure 3.1 : Diagrammes UML

2.3.1. Diagrammes structurels

**Diagramme de classes** : Ce diagramme met en évidence les éléments fondamentaux statiques, notamment les classes, les associations, les interfaces, les attributs, les opérations et les héritages.

**Diagramme d'objets** : Ce diagramme illustre les instances spécifiques des éléments structurels et leurs interactions pendant l'exécution du système.

**Diagramme de packages** : Ce diagramme représente l'organisation logique du modèle et les relations entre les différents packages.

**Diagramme de structure composite** : Ce diagramme expose l'organisation interne d'un élément statique complexe, permettant de visualiser sa structure interne.

**Diagramme de composants** : Ce diagramme présente des structures complexes, avec leurs interfaces fournies et requises, offrant ainsi une vue détaillée de l'architecture du système.

**Diagramme de déploiement** : Ce diagramme montre comment les différents « artefacts » logiciels sont déployés physiquement sur les ressources matérielles, facilitant ainsi la planification et la compréhension du déploiement du système.

### 2.3.2. Diagrammes comportementaux

**Diagramme de cas d'utilisation** : Ce diagramme illustre les interactions fonctionnelles entre les acteurs et le système étudié, mettant en évidence les différentes fonctionnalités offertes par le système.

**Diagramme de vue d'ensemble des interactions** : Ce diagramme fusionne les éléments des diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flux, fournissant ainsi une vue globale des interactions du système.

**Diagramme de séquence** : Ce diagramme représente la séquence verticale des messages échangés entre les objets au sein d'une interaction, permettant de visualiser le déroulement temporel des opérations.

**Diagramme de communication** : Ce diagramme illustre la communication entre les objets dans le plan au sein d'une interaction, offrant une vue alternative aux diagrammes de séquence.

**Diagramme de temps** : Ce diagramme fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps, permettant de visualiser les changements d'état et les transitions dans le temps.

**Diagramme d'activité** : Ce diagramme représente l'enchaînement des actions et des décisions au sein d'une activité, offrant une vue détaillée du déroulement des processus ou des scénarios.

**Diagramme d'états:** Ce diagramme illustre les différents états et les transitions possibles des objets d'une classe, permettant de modéliser le comportement dynamique des objets dans un système.

En combinant ces différents types de diagrammes comportementaux, les développeurs peuvent mieux comprendre et analyser le comportement d'un système logiciel, ce qui facilite la conception, la communication et la documentation des systèmes logiciels.

### 3. Description générale du système Medics

#### 3.1. Fonctionnalités et Objectifs

Medics est un système permettant aux médecins d'accéder à des articles médicaux pertinents et de les consulter sur leurs appareils. Elle est conçue pour offrir des informations précises et actuelles sur les dernières études, les découvertes médicales et les meilleures pratiques en médecine. L'application vise à faciliter l'accès aux connaissances médicales pour les médecins, en leur offrant un outil de référence rapide et facile à utiliser.

Medics est un système complexe doté d'un noyau de recommandation sensible au contexte pour les articles et d'autres fonctionnalités par exemple la gestion de la liste des patients de la part des médecins, le patient peut demander un rendez-vous à distance...

#### 3.2. Noyau de recommandation Medics

Le noyau du système Medics met en place plusieurs scénarios avec plusieurs paramètres. Pour chaque scénario explore un algorithme spécifique. La section ci-dessous explique en détail les scénarios ainsi les algorithmes.

##### 3.2.1. Description des Scénarios

Pour la recommandation des articles pour les médecins, nous pouvons spécifier deux catégories de médecin : médecin généraliste, médecin spécialiste. En effet, nous mettons en place imaginer quatre scénarios possibles selon ces catégories :

**Scénario1** : un médecin généraliste utilise le système pour la première fois. Le noyau détecte la similarité entre catégories d'articles et les champs d'intérêt du médecin demandés à l'avance lors de création de son profil. L'intérêt ici est d'éviter le problème de démarrage à froid.

**Scénario2 :** médecin généraliste qui a déjà utilisé le système. Le noyau va adapter la recommandation en tenant compte de deux paramètres : évaluation des articles par la communauté et les interactions de médecin avec le système.

**Scénario3 :** médecin spécialiste utilise le système pour la première fois. Le noyau de recommandation calcul le vecteur de similarité entre article en se basant sur la spécialité du médecin spécifiée lors de l’inscription.

**Scénario4 :** médecin spécialiste a déjà utilisé le système. Le noyau va adapter la recommandation par rapport : aux évaluations des articles, micro-spécialité du médecin et son interaction avec le système. La figure 3.2 résume les quatre scénarios.

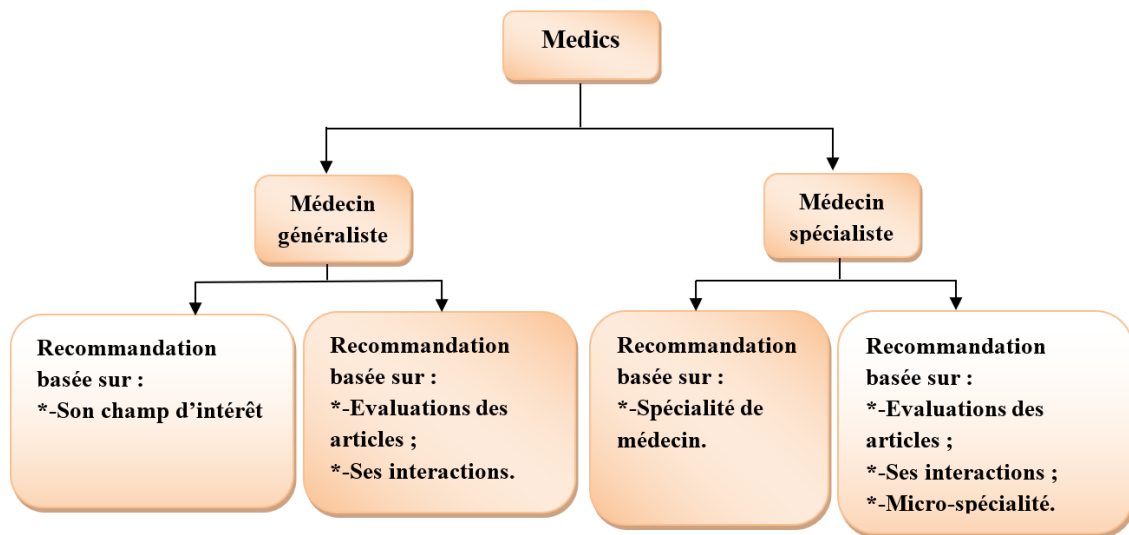


Figure 3.2 : Fonctionnement système de recommandation (**Medics**)

**3.2.1.1. Algorithme du 1<sup>er</sup> scénario**

Cet algorithme exploite la représentation vectorielle des articles et du champ d'intérêt pour identifier les articles les plus pertinents à recommander au médecin généraliste lors de sa première utilisation du système.

**Entrées**

- Profil du médecin généraliste (nom, spécialité, etc.)
- Champ d'intérêt spécifié par le médecin
- Corpus d'articles médicaux avec leurs vecteurs de représentation

**Sortie**

- Une liste des top N articles recommandés, triés par ordre décroissant de similitude avec le champ d'intérêt spécifié par le médecin généraliste.

**Algorithme**

1. Demander au médecin généraliste de spécifier son champ d'intérêt lors de la création de son profil
2. Représenter le champ d'intérêt comme un vecteur de mots-clés (par exemple, en utilisant TF-IDF)
3. Pour {chaque article dans le corpus} :
  - a. Calculer la similitude entre le vecteur de représentation de l'article et le vecteur du champ d'intérêt (par exemple, avec la similarité cosinus)
  - b. Enregistrer cette similitude dans une liste triée
4. Générer les top N articles avec les plus hautes similitudes par rapport au champ d'intérêt
5. Afficher et enregistrer les top N articles au médecin généraliste.

**3.2.1.2. Algorithme du 2<sup>ème</sup> scénario**

Voici un algorithme de recommandation pour le deuxième scénario, où un médecin généraliste a déjà utilisé le système et a évalué (rating) certains articles précédemment recommandés :

**Entrées**

- Profil du médecin généraliste
- Historique des évaluations (ratings) du médecin sur les articles précédents
- Corpus d'articles médicaux avec leurs vecteurs de représentation

**Sortie**

- Une liste de N articles recommandés, triés par ordre décroissant de similitude avec les préférences déduites du médecin généraliste par rapport aux évaluations d'autres médecins

**Algorithme**

1. Extraire les articles ayant reçu une évaluation positive du médecin (par exemple, un rating supérieur à un certain seuil)
2. Calculer un vecteur de préférences pour le médecin en faisant la moyenne pondérée des vecteurs de représentation des articles appréciés, en utilisant les ratings comme poids
3. Pour chaque { article non évalué dans le corpus } :
  - a. Calculer la similitude entre le vecteur de représentation de l'article et le vecteur de préférences du médecin (par exemple, avec la similarité cosinus)
  - b. Enregistrer cette similitude dans une liste triée
4. Sélectionner les N articles avec les plus hautes similitudes par rapport au vecteur de préférences
5. Recommander ces N articles au médecin généraliste.
6. Mise à jour des articles recommandés.

**3.2.1.3. Algorithme du 3<sup>ème</sup> scénario**

Cet algorithme de recommandation pour le scénario où un médecin spécialiste utilise le système pour la première fois où les recommandations sont basées sur la similarité avec d'autres médecins spécialistes :

**Entrées**

- Profil du nouveau médecin spécialiste (spécialité)
- Historique des évaluations (ratings) des autres médecins spécialistes
- Corpus d'articles médicaux avec leurs vecteurs de représentation

**Sortie**

- Une liste de N articles recommandés au nouveau médecin spécialiste

**Algorithme**

1. Identifier un groupe de médecins spécialistes similaires au nouveau médecin (même spécialité.)
2. Pour chaque médecin du groupe similaire :
  - a. Extraire les articles évalués positivement (avec un rating supérieur à un certain seuil)
  - b. Calculer le vecteur de préférences comme la moyenne pondérée des vecteurs des articles appréciés (en utilisant les ratings comme poids)
3. Calculer un vecteur de préférences global pour le groupe en faisant la moyenne des vecteurs de préférences individuels
4. Pour chaque { article non évalué dans le corpus } :
  - a. Calculer la similitude entre le vecteur de représentation de l'article et le vecteur de préférences global (par exemple, avec la similarité cosinus)
  - b. Enregistrer cette similitude dans une liste triée
5. Sélectionner les N articles avec les plus hautes similitudes par rapport au vecteur de préférences global
6. Recommander ces N articles au nouveau médecin spécialiste
7. Mise à jour des articles recommandés.

**3.2.1.4. Algorithme du 4<sup>ème</sup> scénario**

Cet algorithme pour le scénario 4, où un médecin spécialiste a déjà utilisé le système et les recommandations sont basées sur ses évaluations (ratings), sa micro-spécialité, etc.

**Entrées**

- Profil du médecin spécialiste (spécialité, micro-spécialité, interactions.)
- Historique des évaluations (ratings) du médecin sur les articles précédents
- Corpus d'articles médicaux avec leurs vecteurs de représentation et leurs métadonnées (micro-spécialité, mots-clés, etc.)

**Sorties**

- Une liste de N articles recommandés, triés par ordre décroissant de pertinence pour le médecin

**Algorithme**

1. Extraire les articles ayant reçu une évaluation positive du médecin (rating > seuil)
2. Calculer un vecteur de préférences pour le médecin en faisant la moyenne pondérée des vecteurs de représentation des articles appréciés (en utilisant les ratings comme poids)
3. Pour chaque { article non évalué dans le corpus } :
  - a. Calculer la similitude entre le vecteur de représentation de l'article et le vecteur de préférences du médecin (par exemple, avec la similarité cosinus)
  - b. Calculer un score de pertinence micro-spécialité en fonction de la correspondance entre la micro-spécialité de l'article et celle du médecin
  - c. Combiner la similitude et le score de pertinence micro-spécialité en un score global
  - d. Enregistrer ce score global dans une liste triée
4. Sélectionner les N articles avec les scores globaux les plus élevés
5. Recommander ces N articles au médecin spécialiste
6. Mise à jour des articles recommandés.

Medics est un système complexe contenant d'autres fonctionnalités. Afin de couvrir la phase conceptuelle du système nous choisissons UML qui est un langage de modélisation permet une implémentation orientée objet. Dans ce qui suit, nous présentons en détail l'approche conceptuelle en montrant les différents diagrammes.

### 3.3. Acteurs cibles du système

Le système Medics est découpé en trois parties, chaque partie est réservée à un acteur particulier du système. Le tableau 3.1 représente les acteurs cibles du système Medics.

Acteur	Action	Description
Utilisateur	S'inscrire	Permet l'inscription de l'utilisateur sur la plateforme
Patient	S'authentifier	Permet d'accéder à l'espace personnel de l'utilisateur
	Gérer médecins	Permet de gérer des médecins au niveau de son espace. Où il peut faire une : <ul style="list-style-type: none"> <li>✓ Recherche d'un médecin</li> <li>✓ Evaluation à un médecin</li> </ul>
	Demander une réservation.	Permet de prendre ou d'annuler une réservation.
Médecin généraliste	S'authentifier	Permet d'accéder à l'espace personnel de l'utilisateur
	Gérer sa liste des patients	Permet de gérer les patients au niveau de l'espace utilisateur, où il peut les : <ul style="list-style-type: none"> <li>✓ Accepter une réservation.</li> <li>✓ Refuser une réservation.</li> </ul>
	Avoir articles	Permet de consulter des articles recommandés et de les évaluer (champ d'intérêt).
Médecin spécialiste	S'authentifier	Permet d'accéder à l'espace personnel de l'utilisateur

	Gérer sa liste des patients	Permet de gérer les patients au niveau de l'espace utilisateur, ou il peut les : <ul style="list-style-type: none"> <li>✓ Accepter une réservation.</li> <li>✓ Refuser une réservation.</li> </ul>
	Avoir articles	Permet de consulter des articles recommandés et de les évaluer (micro-spécialité).

Tableau 3.1 : Acteurs cible du système Medics.

Afin de décrire la conception du système Medics, nous utiliserons les trois diagrammes UML suivants :

- ✓ Diagramme de cas d'utilisation
- ✓ Diagramme de séquence
- ✓ Diagramme de classe

## 4. Conception du système Medics

### 4.1. Diagrammes de cas d'utilisation

La figure 3.3 représente le diagramme de cas d'utilisation du système concernant l'acteur Patient.

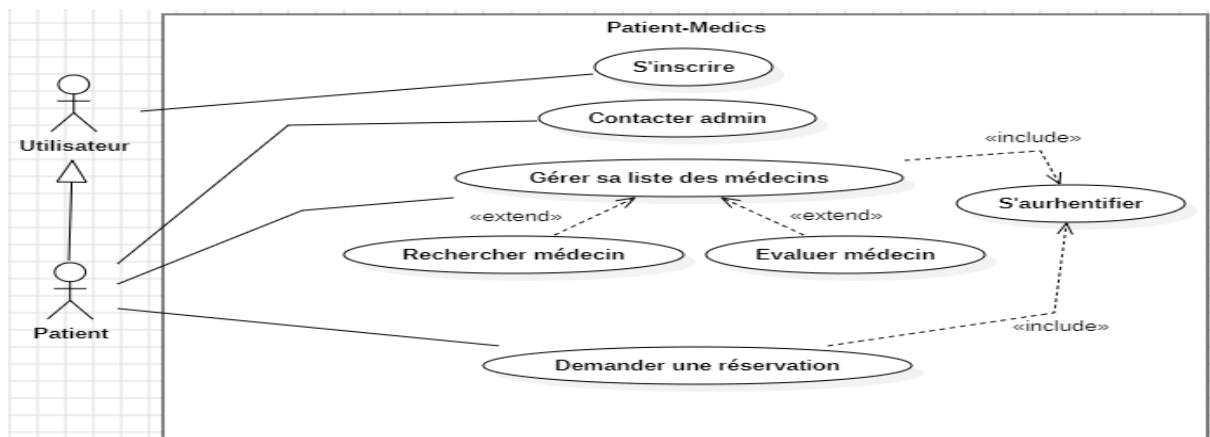


Figure 3.3 : Diagramme de cas d'utilisation du système concernant l'acteur Patient

La figure 3.4 représente le diagramme de cas d'utilisation pour un médecin généraliste.

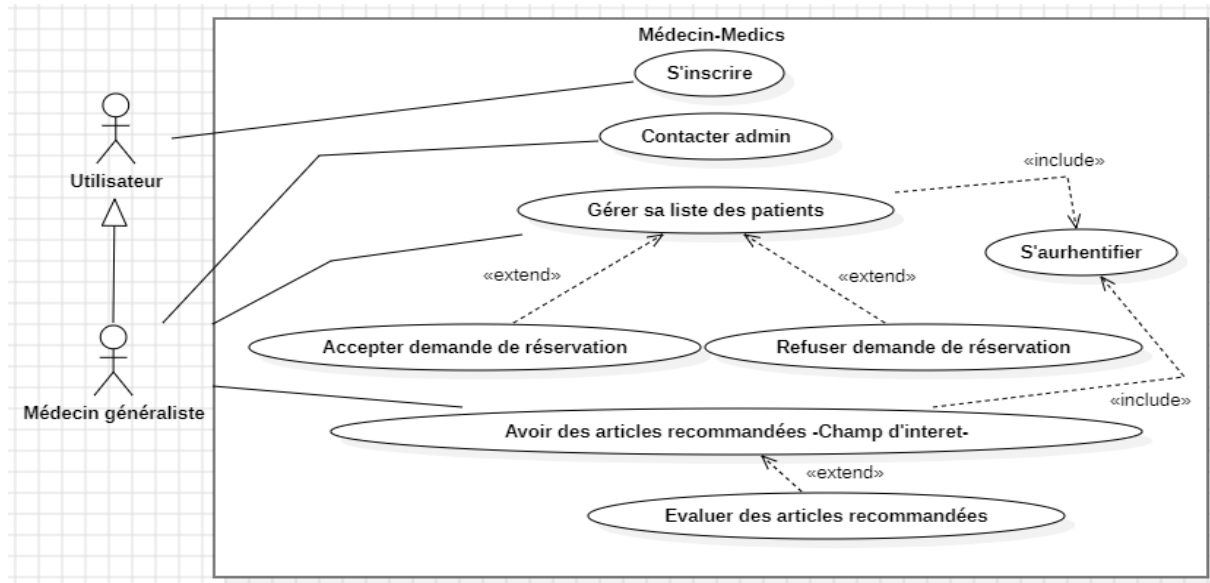


Figure 3.4 : Diagramme de cas d'utilisation pour un médecin généraliste.

La figure 3.5 représente le diagramme de cas d'utilisation pour un médecin spécialiste.

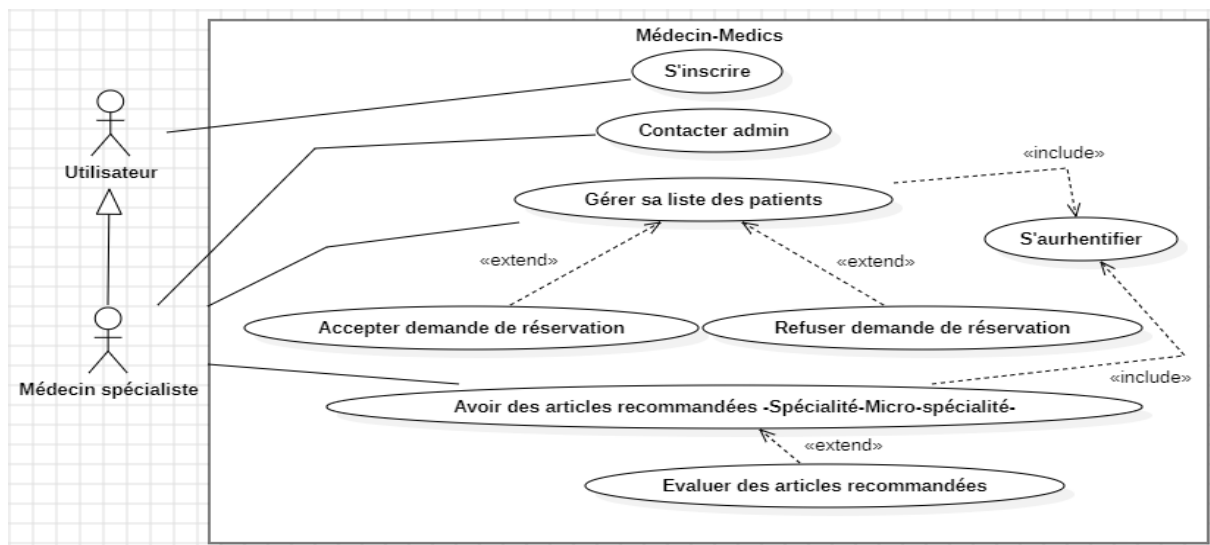


Figure 3.5 : Diagramme de cas d'utilisation pour un médecin spécialiste.

### Description textuelle du système

- Créer un profil (S'inscrire)

#### Pour un médecin généraliste

Nom du cas d'utilisation : Créer un profil médecin généraliste.

## **Chapitre III : Analyse et conception du système Medics**

---

**Résumé:** ce cas d'utilisation permet à un médecin généraliste de créer son profil dans le système lors de sa première utilisation.

**Acteur :** Médecin généraliste.

### **Précondition**

- Le médecin n'a pas encore de compte dans le système.

### **Postcondition**

- Le profil du médecin est créé avec succès dans le système.

### **Scénario nominal**

1. Le médecin accède à l'interface de création de compte.
2. Le système affiche le formulaire de création de compte (nom, adresse e-mail, mot de passe, etc.).
3. Le médecin remplit le formulaire et indique qu'il est un médecin généraliste, et sélectionne son champ d'intérêt.
4. Le système crée le profil du médecin avec les informations fournies.
5. Le système confirme la création réussie du profil.

### **Pour un médecin spécialiste**

**Nom du cas d'utilisation :** Créer un profil médecin spécialiste.

**Résumé:** ce cas d'utilisation permet à un médecin spécialiste de créer son profil dans le système lors de sa première utilisation.

**Acteur :** Médecin spécialiste.

### **Précondition**

- Le médecin n'a pas encore de compte dans le système.

### **Postcondition**

- Le profil du médecin est créé avec succès dans le système.

### **Scénario nominal**

1. Le médecin accède à l'interface de création de compte.
2. Le système affiche le formulaire de création de compte (nom, adresse e-mail, mot de passe, etc.).
3. Le médecin remplit le formulaire et indique qu'il est un médecin spécialiste, et sélectionne sa spécialité et sa micro-spécialité.
4. Le système crée le profil du médecin avec les informations fournies.
5. Le système confirme la création réussie du profil.

### Enchaînement alternatif

A3 : Si le médecin spécialiste entre des informations incomplètes ou incorrectes.

- Le système affiche un message d'erreur/avertissement indiquant les champs manquants ou er
- Le scénario reprend au point 3.

- **Avoir des articles**

### Pour un médecin spécialiste

**Nom du cas d'utilisation :** Avoir des articles recommandés.

**Résumé :** Ce cas d'utilisation permet à un médecin spécialiste d'obtenir des recommandations d'articles pertinents pour son domaine d'expertise lors de sa première utilisation du système.

**Acteur:** Médecin spécialiste

### Préconditions

- Le médecin spécialiste a créé son profil et indiqué son domaine de spécialité.
- Le système dispose d'une base de données d'articles médicaux et des profils d'autres médecins spécialistes.

### Postconditions

- Le médecin spécialiste a accès à une liste d'articles recommandés pertinents pour son domaine d'expertise.
- Le système a enregistré les interactions du médecin avec les articles recommandés pour améliorer les recommandations futures.

## **Chapitre III : Analyse et conception du système Medics**

---

### **Scénario nominal**

1. Le médecin spécialiste se connecte au système pour la première fois.
2. Le système récupère le domaine de spécialité du médecin à partir de son profil.
3. Le système analyse les profils d'autres médecins spécialistes dans le même domaine pour identifier les articles les plus populaires ou les plus pertinents pour ce domaine.
4. Le système utilise un algorithme de recommandation basé sur la similarité avec les autres médecins spécialistes pour générer une liste d'articles recommandés.
5. Le système affiche la liste d'articles recommandés au médecin spécialiste.
6. Le médecin spécialiste peut consulter les articles recommandés et interagir avec eux (lire , évaluer , etc.).

### **Scénario alternatif**

- Aucun

### **Pour un médecin généraliste**

**Nom du cas d'utilisation :** Obtenir des recommandations d'articles pour un médecin généraliste

**Résumé :** ce cas d'utilisation permet à un médecin généraliste d'obtenir des recommandations d'articles pertinents pour ses champs d'intérêt lors de sa première utilisation du système.

**Acteur :** Médecin généraliste

### **Préconditions**

- Le médecin généraliste a créé son profil et spécifié ses champs d'intérêt médicaux.
- Le système dispose d'une base de données d'articles médicaux.

### **Postconditions**

- Le médecin généraliste a accès à une liste d'articles recommandés pertinents pour ses champs d'intérêt.
- Le système a enregistré les interactions du médecin avec les articles recommandés pour améliorer les recommandations futures.

## Scénario nominal

1. Le médecin généraliste se connecte au système pour la première fois.
2. Le système récupère les champs d'intérêt du médecin à partir de son profil.
3. Le système identifie les articles existants dans la base de données correspondant aux champs d'intérêt du médecin.
7. Le système affiche la liste des N articles recommandés au médecin généraliste.
8. Le médecin généraliste peut consulter les articles recommandés et interagir avec eux (lire, évaluer, etc.).
9. Le système suit les interactions du médecin avec les articles recommandés (lectures, évaluations, etc.).

## 4.2. Diagrammes de séquence

La figure 3.6 représente le diagramme de séquence pour la création d'un profil médecin.

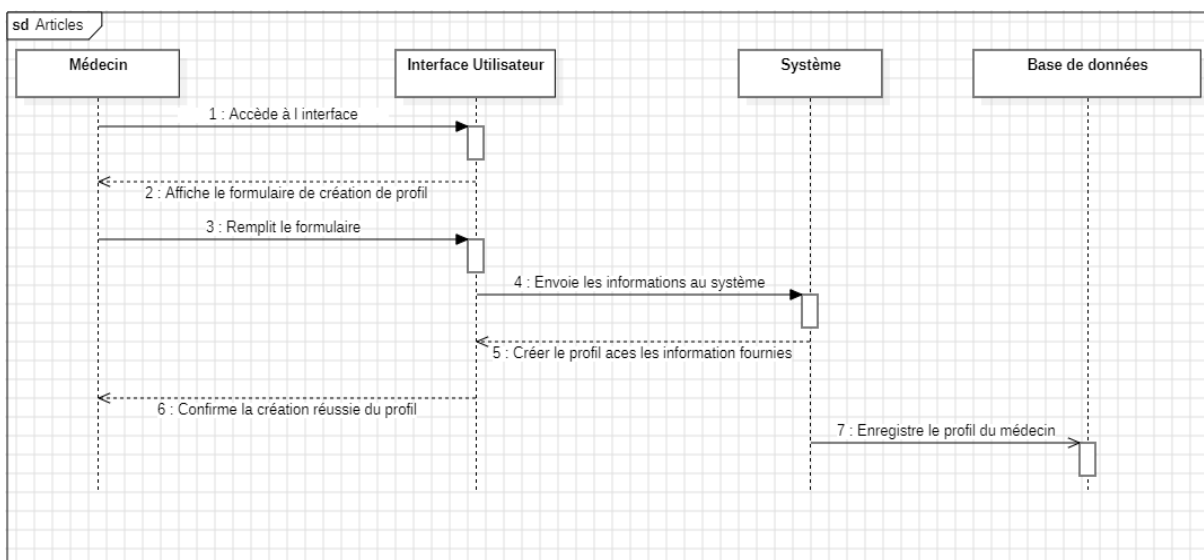


Figure 3.6 : Diagramme de séquence pour la création d'un profil médecin.

La figure 3.7 représente le diagramme de séquence pour le médecin spécialiste.

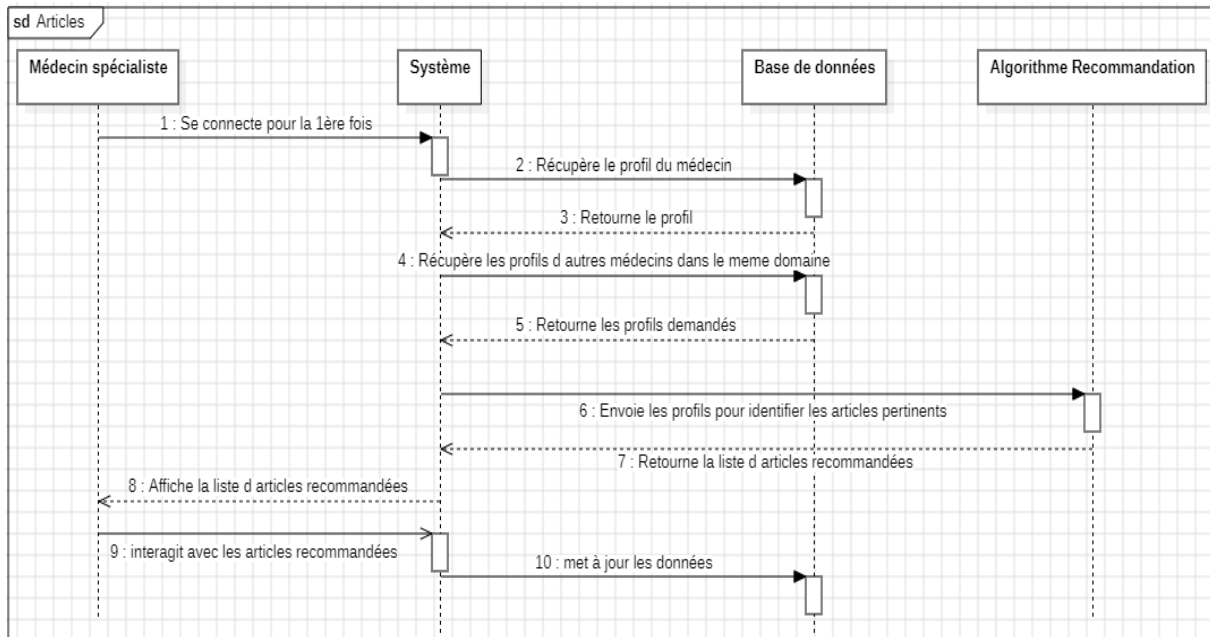


Figure 3.7 : Diagramme de séquence pour la recommandation des articles médicaux par un médecin spécialiste.

- La figure 3.8 représente le diagramme de séquence pour le médecin généraliste qui utilise le système pour la 1<sup>ère</sup> fois.

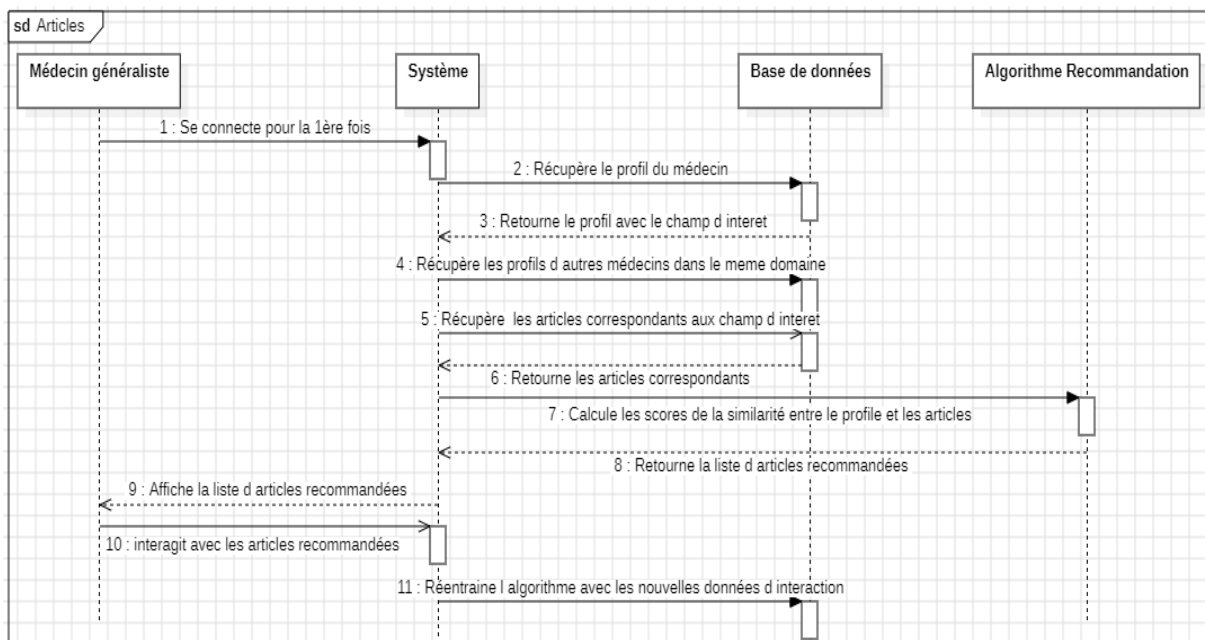


Figure 3.8: Diagramme de séquence pour la recommandation des articles médicaux par un médecin généraliste

### 4.3. Diagramme de classe

La figure 3.9 représente le diagramme de classe du système Medics.

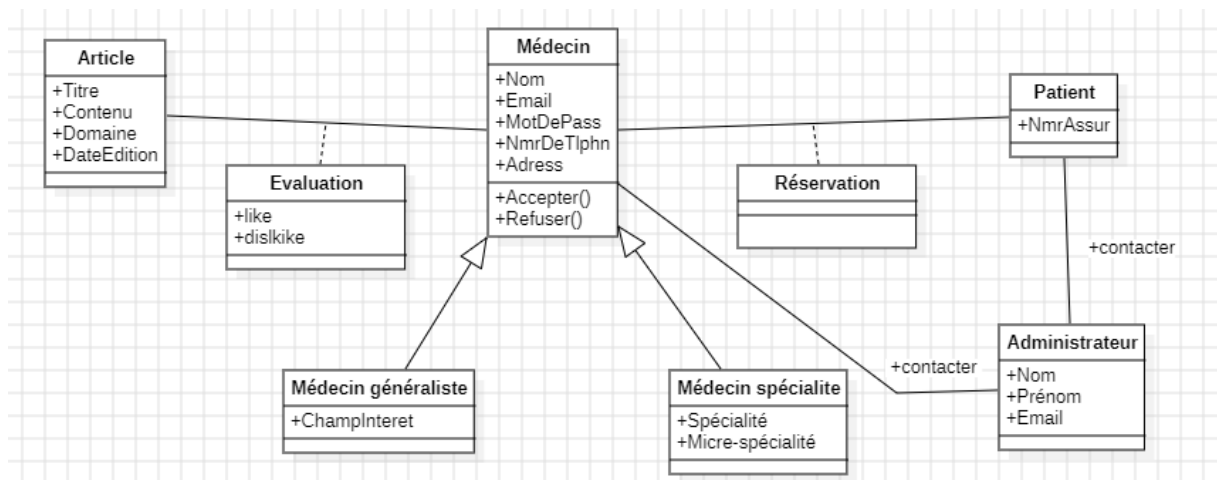


Figure 3.9 : Diagramme de classe pour le système Medics

## 5. Conclusion

Dans cette section, nous avons examiné les exigences du projet et identifié les différentes fonctionnalités requises à l'aide du diagramme de cas d'utilisation. Nous avons également exploré les différents scénarios d'utilisation du système ainsi que sa structure objet. Cette analyse nous a offert une vision claire et détaillée de notre système, ce qui facilitera son implémentation, comme détaillé dans le prochain chapitre.

# **Chapitre 4**

---

## **Implémentation du système Medics**

## 1. Introduction

Dans ce chapitre, nous présenterons les outils et technologies utilisés pour développer le système Medics. Nous aborderons également la structure de l'application, l'algorithme de recommandation du système et son implémentation. Enfin, nous concluons ce chapitre par une présentation des interfaces de l'application.

## 2. Architecture générale du système

Medics est un système implémenté en respectant une architecture client/ serveur. La figure 4.1 relate cette architecture en détail.

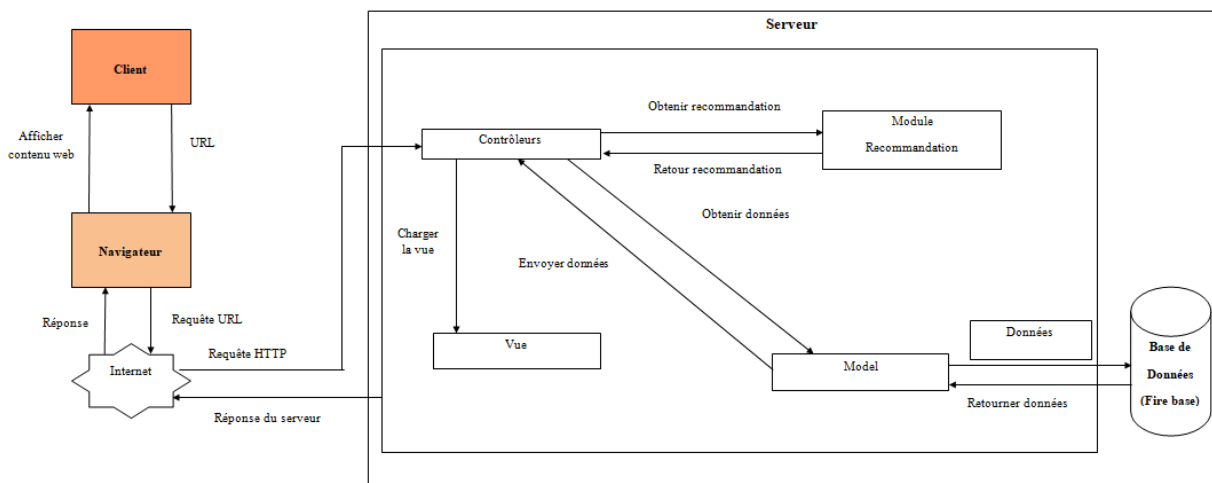


Figure 4.1 : Architecture générale du système Medics.

## 3. Environnement du travail

### 3.1. Langage de développement

#### 3.1.1. Javascript

JavaScript est un langage de programmation polyvalent et essentiellement utilisé pour le développement web. Contrairement à HTML et CSS, qui définissent la structure et le style d'une page web, JavaScript permet d'ajouter des fonctionnalités interactives et dynamiques. Il est largement utilisé pour créer des applications web, des jeux, des applications mobiles et même des logiciels serveur grâce à Node.js. JavaScript est un langage côté client, ce qui signifie qu'il est exécuté sur le navigateur web de l'utilisateur. Sa syntaxe est similaire à celle de nombreux autres langages de programmation, ce qui le rend relativement facile à

## Chapitre IV : Implémentation du système Medics

---

apprendre pour les développeurs débutants. JavaScript est également un langage interprété, ce qui signifie que les instructions sont exécutées ligne par ligne au fur et à mesure de leur rencontre, ce qui permet une grande flexibilité et une facilité de développement.[22]

### 3.1.2. NODE JS

Node.js est un environnement d'exécution JavaScript libre et open-source qui permet aux développeurs de créer des applications serveur, des outils de ligne de commande et des scripts côté serveur, sans avoir à utiliser un navigateur. Il est conçu pour les applications réseau évènementielles hautement concurrentes et peut être utilisé sur plusieurs plateformes, incluant Linux, Windows, BSD et macOS. Node.js utilise la machine virtuelle V8 et la bibliothèque libuv pour sa boucle d'évènements, et implémente les spécifications CommonJS sous licence MIT. Il est utilisé par de nombreuses entreprises telles que GoDaddy, IBM, Netflix, Amazon Web Services, Groupon, Vivaldi, SAP, LinkedIn, Microsoft, Yahoo!, Walmart, Rakuten, Sage et PayPal.[23]

## 3.2. Logiciels utilisés

### 3.2.1. Visual Code Studio

Visual Studio Code (VSCode) est un environnement de développement intégré (IDE) léger et polyvalent, disponible sur Windows, macOS et Linux. Bien qu'il soit axé sur JavaScript et Node.js, il offre un écosystème d'extensions vaste pour prendre en charge de nombreux autres langages, tels que Flutter, C++, C#, Java, Python, PHP et Go, ainsi que des environnements tels que .NET et Unity. VSCode combine les fonctionnalités d'un éditeur de code simple avec des outils avancés de développement, notamment la complétion IntelliSense, le débogage interactif, l'inspection des variables et des piles d'appels. Il intègre également la gestion du contrôle de version Git, permettant d'afficher les différences et de travailler sur les modifications directement depuis l'éditeur. Visual Studio Code.[24]

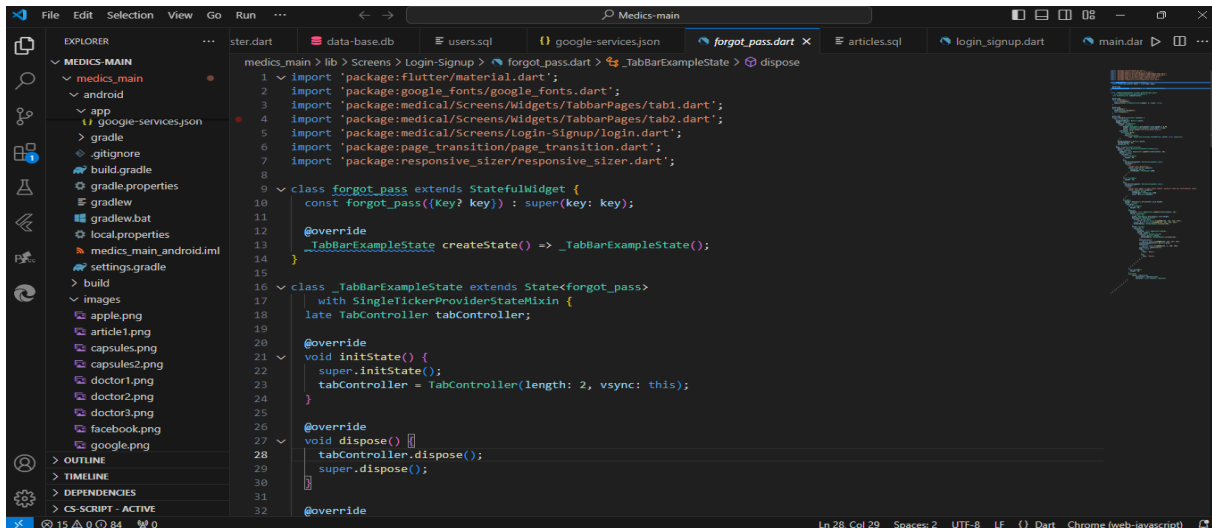


Figure 4.2: Interface Visual Studio Code

## 3.2.2. Firebase

Firebase est une plateforme de développement d'applications mobiles et web conçue par Google. Elle permet aux développeurs de créer des applications scalables, sécurisées et personnalisables en utilisant une variété de produits et services tels que Firebase Cloud Messaging, FirebaseAnalytics, FirebaseAuthentication, FirebaseRealtimeDatabase et bien plus encore. Firebase est disponible sur plusieurs plateformes, incluant Android, iOS, ReactNative, Node.js et le web, et offre des intégrations avec des outils tels que Google Cloud, Google Analytics et Google Maps. Firebase est utilisé par de nombreuses entreprises et applications populaires, telles que NPR, Halfbrick, Duolingo et Venmo.

## 3.2.3. Git

Git est un système de contrôle de version distribué, gratuit et open-source, conçu pour gérer efficacement des projets de toutes tailles, des plus petits aux plus grands, avec rapidité et efficacité. Il se distingue par sa facilité d'apprentissage et son faible encombrement, offrant des performances ultra-rapides. Comparé à d'autres outils SCM tels que Subversion, CVS, Perforce et ClearCase, Git offre des fonctionnalités avancées telles que le branchement local bon marché, les zones de transit pratiques et la prise en charge de plusieurs flux de travail. Avec Git, il est possible d'avoir plusieurs branches locales totalement indépendantes les unes des autres. Cela permet des fonctionnalités telles que le changement de contexte sans frictions, les codes basés sur les rôles, les flux de travail basés sur les fonctionnalités et l'expérimentation jetable. Bien que Git soit principalement utilisé en ligne de commande, il



## Chapitre IV : Implémentation du système Medics

Contrôleur, favorisant ainsi une meilleure organisation, une maintenabilité accrue et une réutilisabilité du code.

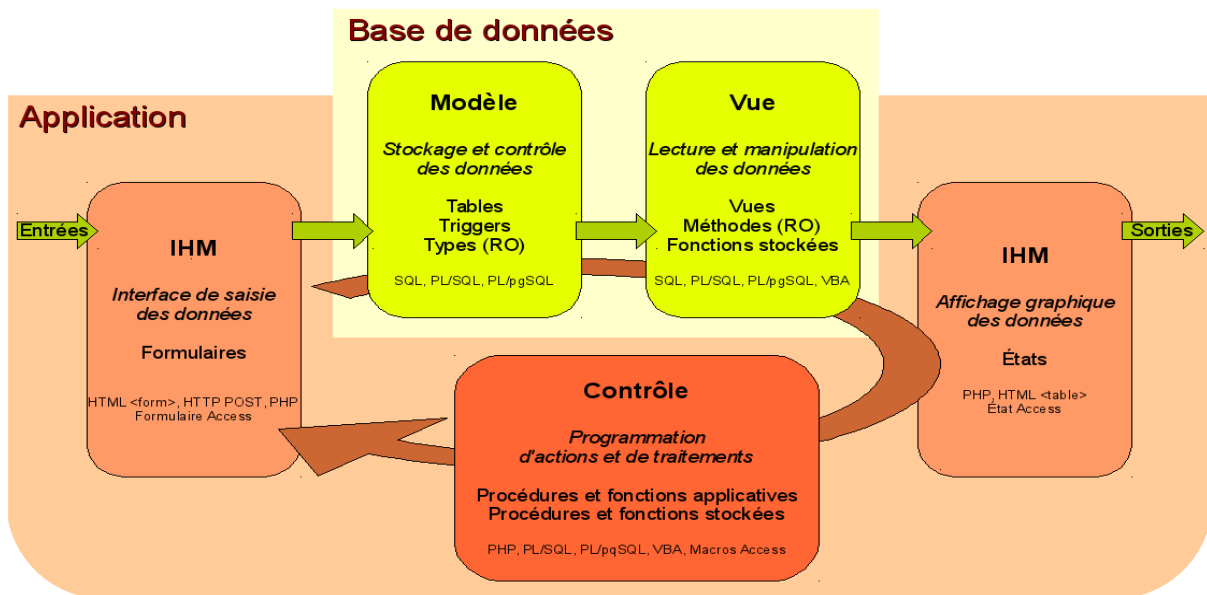


Figure 4.4 : Architecture MVC

### 4.1.1. Modèle (Model)

Le Modèle représente la couche de données de l'application web. Il gère les données, leur structure, leur validation et les règles métier associées. Le Modèle est indépendant de la Vue et du Contrôleur. Dans les applications web, les Modèles sont souvent implémentés sous forme de classes ou d'objets représentant les entités de données de l'application, telles que les utilisateurs, les produits, les commandes, etc.

### 4.1.2. Vue (View)

La Vue est responsable de la présentation des données à l'utilisateur. Elle définit l'interface utilisateur, les dispositions, les boutons, les menus et tous les autres éléments visuels. Dans les applications web, les Vues sont généralement construites à l'aide de technologies telles que HTML, CSS et JavaScript, souvent en utilisant des frameworks comme React, Angular ou Vue.js.

### 4.1.3. Contrôleur (Controller)

Le Contrôleur agit comme un intermédiaire entre le Modèle et la Vue. Il gère la logique métier de l'application, traite les entrées de l'utilisateur provenant de la Vue, les valide, et interagit avec le Modèle pour effectuer les opérations nécessaires (récupération, mise à jour,

## Chapitre IV : Implémentation du système Medics

---

suppression de données, etc.). Il récupère ensuite les données du Modèle et les transfère à la Vue pour un affichage approprié.

Le flux de fonctionnement typique de l'architecture MVC dans les applications web est le suivant :

- L'utilisateur interagit avec la Vue (interface utilisateur) en effectuant une action (clic sur un bouton, saisie de données, etc.).
- La Vue transmet l'événement ou les données d'entrée au Contrôleur correspondant.
- Le Contrôleur traite les données d'entrée, valide les règles métier et interagit avec le Modèle pour effectuer les opérations nécessaires (récupération, mise à jour, suppression de données, etc.).
- Le Modèle effectue les opérations demandées et renvoie les données mises à jour au Contrôleur.
- Le Contrôleur prépare les données pour l'affichage et les transmet à la Vue appropriée.
- La Vue met à jour son interface utilisateur avec les nouvelles données pour refléter les changements.

L'architecture MVC offre plusieurs avantages dans le développement d'applications, notamment une meilleure séparation des préoccupations, une amélioration de la maintenabilité et de la testabilité du code, ainsi qu'une réutilisabilité accrue des composants. Cependant, il existe également d'autres modèles de conception, comme MVP (Modèle-Vue-Présentateur) ou MVVM (Modèle-Vue-Vue Modèle), qui sont également utilisés dans le développement d'applications web, en fonction des préférences de l'équipe de développement et des spécificités du projet. Dans le cadre de notre travail, nous avons choisi l'architecture Model View Controller (MVC) pour l'implémentation de notre système Medics.

### 5. Implémentation du module de recommandation

Le système Medics offre divers services aux utilisateurs, qu'ils soient patients ou médecins, tels que la recherche de médecins, la gestion des rendez-vous, etc. Toutefois, le module central de cette plateforme reste le module de recommandation, qui joue un rôle crucial en proposant des articles médicaux susceptibles d'intéresser les utilisateurs (médecins).

### 5.1.Algorithme de recommandation

Cet algorithme a pour but de récupérer des articles médicaux qui correspondent aux intérêts de l'utilisateur (médecin). Afin d'améliorer la précision et la performance des recommandations, nous avons décidé de combiner deux méthodes : le filtrage collaboratif basé sur les utilisateurs voisins et la méthode contextuelle de post-filtrage.

### 5.2.Filtrage des articles

Le filtrage des articles en fonction du type de spécialisation de l'utilisateur. Il y a deux cas possibles :

- **Spécialisation générale**

Si le type de spécialisation de l'utilisateur est "général", le code filtre la liste des articles pour ne garder que ceux dont la spécialité correspond à la spécialisation de l'utilisateur (médecin)

```
const userData = JSON.parse(localStorage.getItem('userData'));

let filteredArticles = articleList;
if (userData.specializationType === 'general') {
  filteredArticles = articleList.filter(article => article.speciality === userData.specialization);
}
```

Algorithme 4.1 :Filtrage d'Articles par Spécialisation Générale

- **Spécialisation spécialiste**

Si le type de spécialisation de l'utilisateur est "spécialiste", le code filtre la liste des articles pour ne garder que ceux dont la spécialité correspond à la spécialisation de l'utilisateur et dont la sous-spécialité correspond à la sous-spécialisation de l'utilisateur.

```
} else if (userData.specializationType === 'specialist') {
  filteredArticles = articleList.filter(article =>
    article.speciality === userData.specialization &&
    article.subspeciality === userData.subSpecialization
  );
}
```

Algorithme 4.2 : Filtrage d'Articles par Spécialisation Spécialiste

### 5.3.Tri et classement des articles recommandés

Après avoir filtré les articles en fonction du profil de l'utilisateur, le code procède au tri et au classement des articles recommandés. Cela permet de présenter les articles les plus pertinents en premier.

Le critère de tri est basé sur les éléments suivants, par ordre de priorité :

- **Absence de notations** : Les articles sans aucune notation (raters) sont classés en premier.
- **Nombre de "dislikes"** : Parmi les articles notés, ceux ayant le moins de "dislikes" (notes à 0) sont classés en premier.
- **Nombre de "likes"** : Enfin, parmi les articles ayant le même nombre de "dislikes", ceux ayant le plus de "likes" (notes à 1) sont classés en premier.

```
filteredArticles.sort((a, b) => {
  const aLikes = Object.values(a.raters || {}).filter(value => value === 1).length;
  const bLikes = Object.values(b.raters || {}).filter(value => value === 1).length;
  const aDislikes = Object.values(a.raters || {}).filter(value => value === 0).length;
  const bDislikes = Object.values(b.raters || {}).filter(value => value === 0).length;

  // Sort priority: no raters > fewer dislikes > more likes
  if (!a.raters) return -1;
  if (!b.raters) return 1;
  if (aDislikes !== bDislikes) return aDislikes - bDislikes;
  return bLikes - aLikes;
});

setArticles(filteredArticles);
};
```

Algorithme 4.3 : Tri des Articles par Appréciations et Dés-appréciations

La fonction `handleReadClick` est déclenchée lorsqu'un utilisateur clique pour lire un article. Elle extrait les informations de l'utilisateur du stockage local pour récupérer son identifiant unique (UID), puis accède aux données de l'article depuis Firestore en utilisant cet identifiant. Si l'article existe, la fonction obtient les appréciations (likes/dislikes) et met à jour l'état de l'appréciation de l'utilisateur en conséquence. Enfin, elle enregistre l'article sélectionné et affiche une fenêtre modale avec son contenu. Cette fonction permet ainsi de gérer la lecture des articles et de suivre les appréciations des utilisateurs de manière dynamique. Cet algorithme représente le code de cette fonction :

```
const handleReadClick = async (article) => {
  const userData = JSON.parse(localStorage.getItem('userData'));
  const userID = userData.uid;
  const docRef = doc(db, 'articles', article.id);
  const docSnap = await getDoc(docRef);

  if (docSnap.exists()) {
    const articleData = docSnap.data();
    const raters = articleData.raters || {};
    setLikeStatus(raters[userID]);
  }

  setSelectedArticle(article);
  setIsModalVisible(true);
};
```

Algorithme 4.4 : Gestion de la Lecture et Mise à Jour des Appréciations des Articles

La fonction handleLikeOrDislike gère l'action d'aimer (like) ou de ne pas aimer (dislike) un article par un utilisateur(médecin).

```
const handleLikeOrDislike = async (isLike) => {
  const userData = JSON.parse(localStorage.getItem('userData'));
  const userID = userData.uid;

  const docRef = doc(db, 'articles', selectedArticle.id);
  const docSnap = await getDoc(docRef);

  if (docSnap.exists()) {
    const articleData = docSnap.data();
    const raters = articleData.raters || {};
    const currentStatus = raters[userID];

    if (currentStatus !== isLike) {
      raters[userID] = isLike ? 1 : 0;

      if (currentStatus === undefined) {
        if (isLike) {
          selectedArticle.likes += 1;
        } else {
          selectedArticle.dislikes += 1;
        }
      } else if (currentStatus === 1) {
        selectedArticle.likes -= 1;
        selectedArticle.dislikes += 1;
      } else {
        selectedArticle.dislikes -= 1;
        selectedArticle.likes += 1;
      }
    }
  }
};
```

Algorithme 4.5 : Rôle du Code handleLikeOrDislike

## 6. Interfaces de l'application

Dans cette section, nous présentons quelques interfaces de l'application Medics, la figure 4.16 représente la page d'accueil de la plateforme Medics où l'utilisateur peut soit s'inscrire ou se connecter.

The screenshot displays the homepage of the Medics platform. At the top, there is a navigation bar with the 'Medics' logo on the left and a 'Start Now' button on the right. The main content area is divided into several sections:

- Your Trusted Medical Partner:** A large banner featuring a group of four medical professionals (three men and one woman) in white lab coats, looking at a tablet. Below the image is a 'Start Now' button.
- Our Medical Services:** A section with four sub-sections:
  - Primary Care:** Focuses on preventive care, diagnosis, and treatment of common conditions.
  - Specialized Treatments:** Offers services for various conditions like cardiology, orthopedics, and neurology.
  - Health Education:** Empowers patients with knowledge and skills for managing health.
  - Telemedicine:** Enables remote access to medical care through virtual consultations.
- Patient Success Stories:** A section with a 'Start Now' button and a list of benefits:
  - Personalized treatment plans
  - Improved quality of lifeBelow the list is a 'GET STARTED' button. To the right is an image of a patient in a hospital bed being attended to by medical staff.
- Improving Patient Outcomes:** A section with a 'Start Now' button and a list of benefits:
  - Advanced medical technologies
  - Compassionate patient careBelow the list is a 'GET STARTED' button. To the left is an image of a doctor in a white coat talking to two seated patients in a clinical setting.

Figure 4.5: Page d'accueil de la plateforme Medics

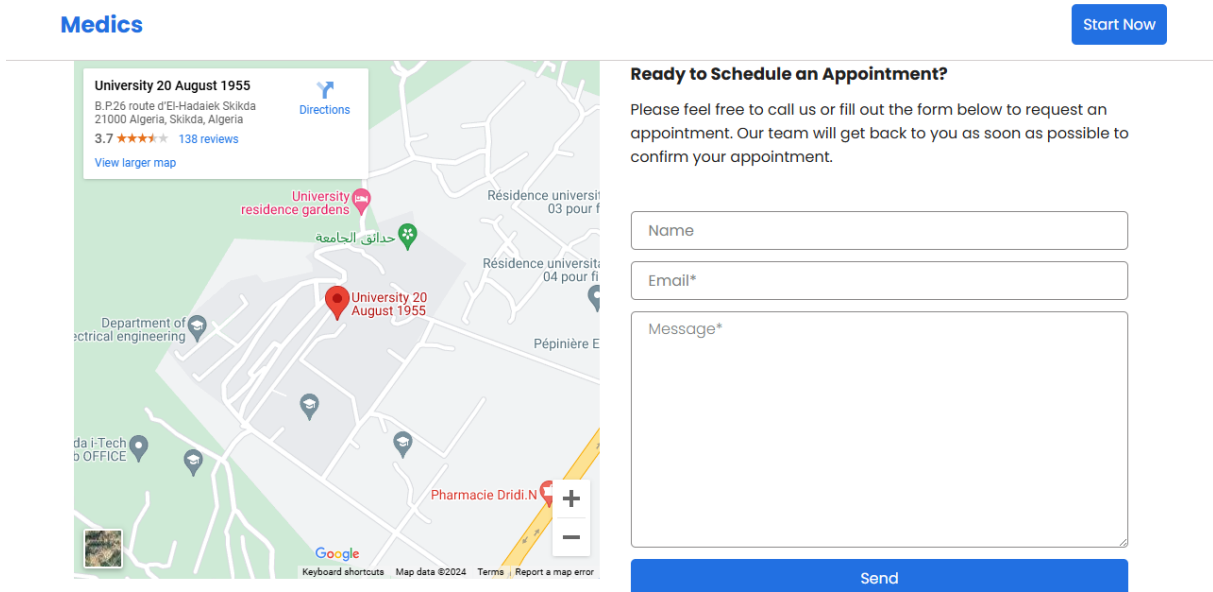


Figure 4.6 : Interface contact Admin

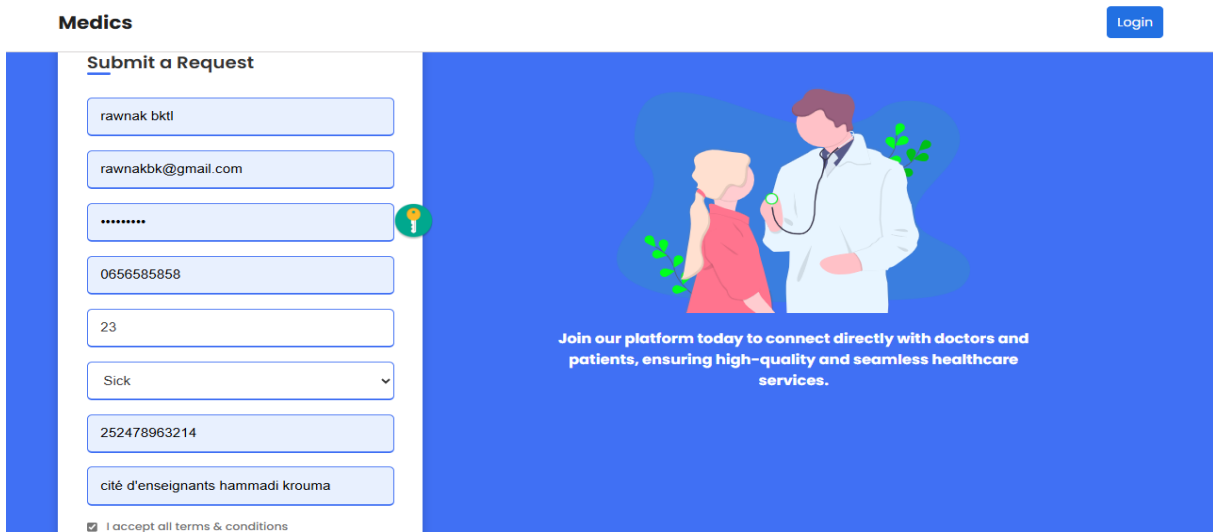


Figure 4.7 : interface d'inscription coté patient

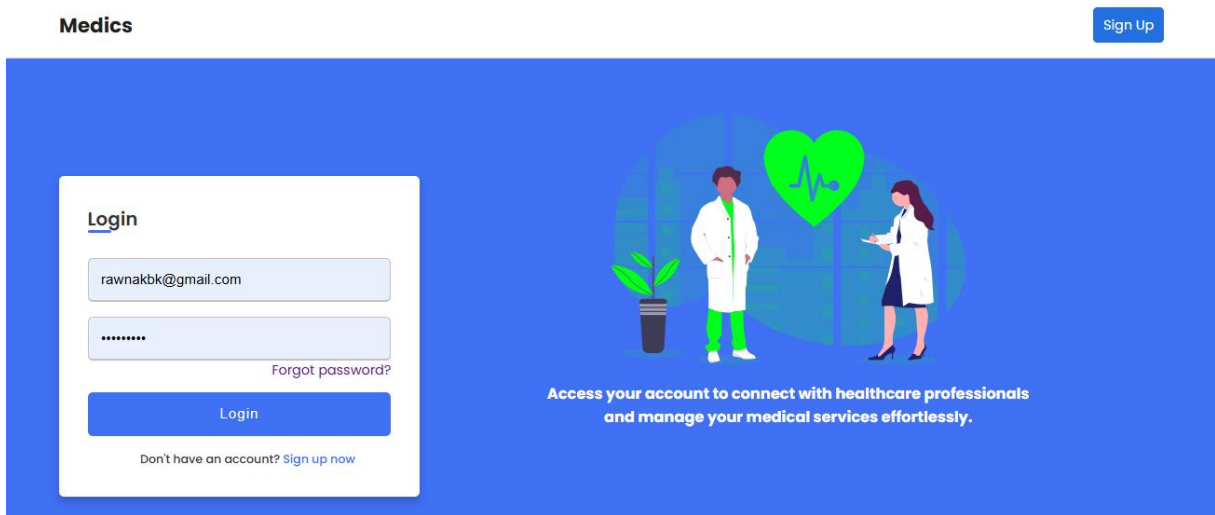
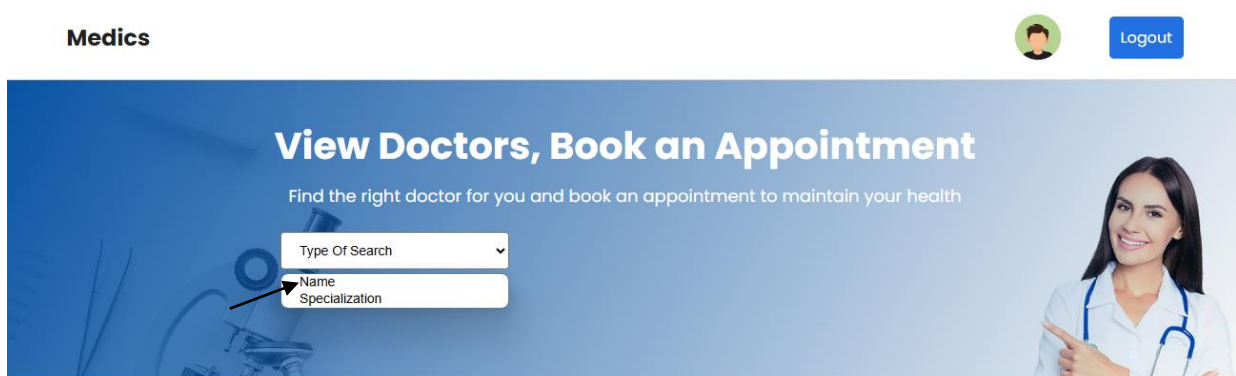


Figure 4.8: Interface de connexion cotépatient



**Largest Healthcare Network Across Algeria**

Find best doctors across specialities or names in your city.

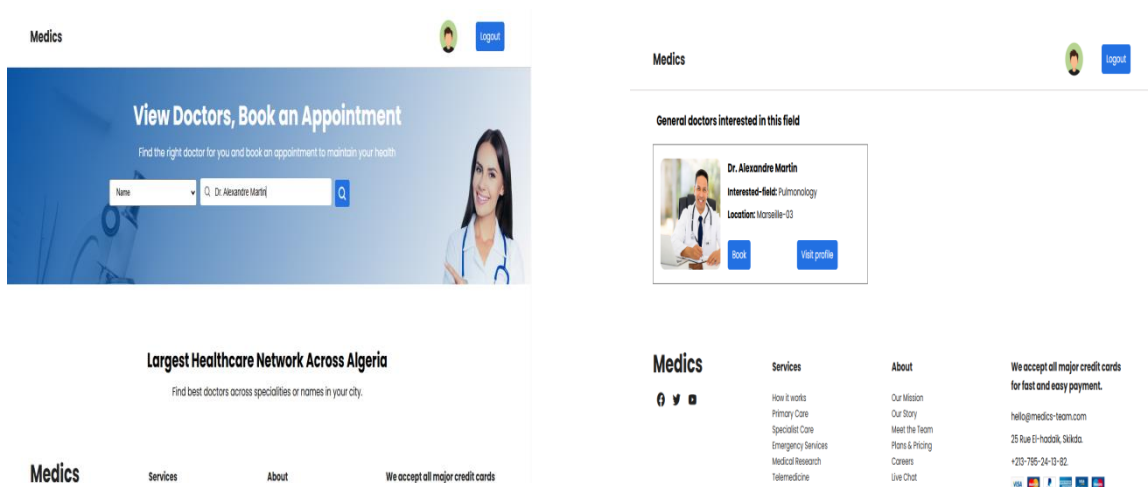


Figure 4.9: Recherche médecin à travers le nom

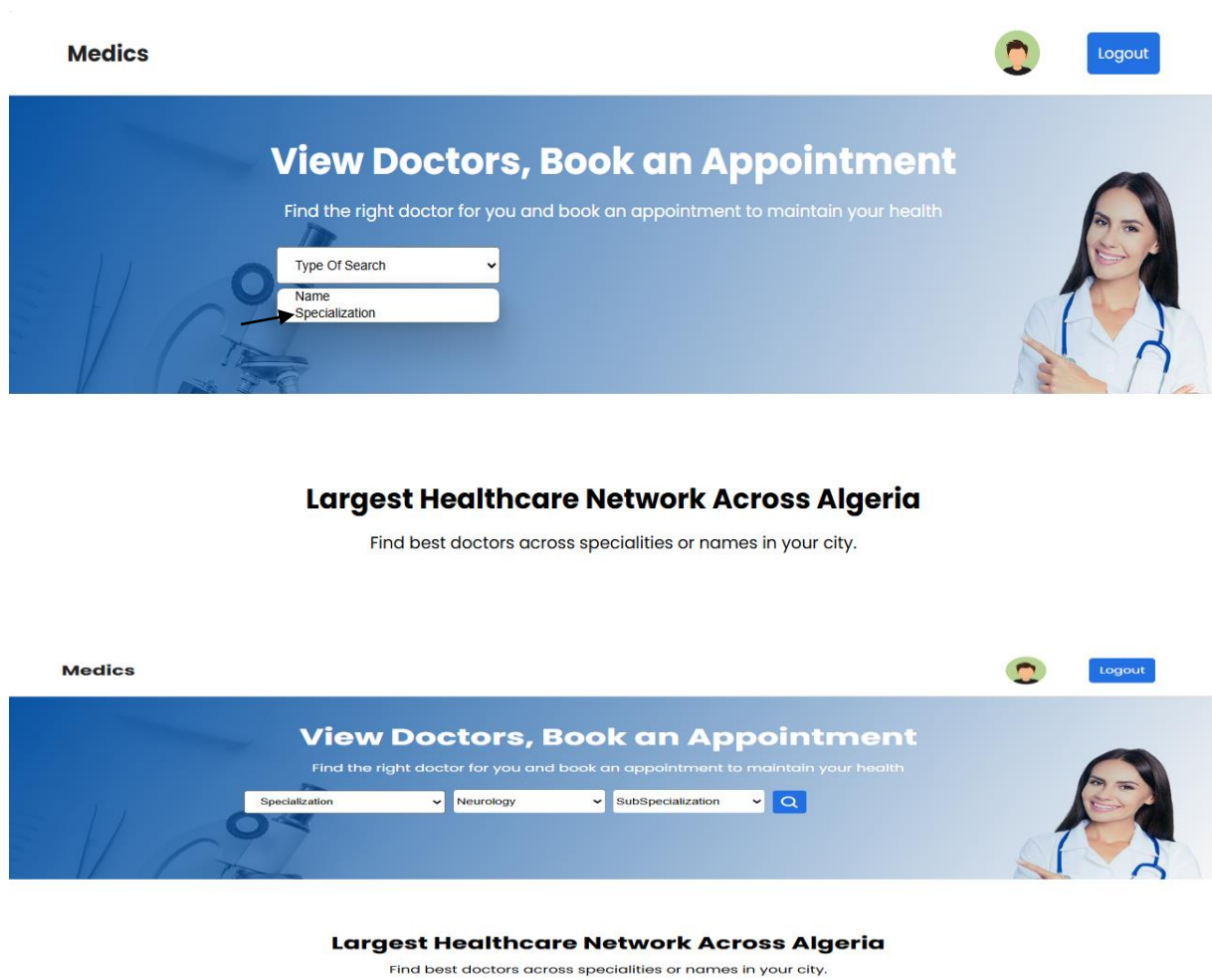




Figure 4.10: Recherche médecin à travers la spécialité

Medics  [Logout](#)


---

**General doctors interested in this field**




**Dr. Pauline Marchand**  
**Interested-field:** Neurology  
**Location:** 789 Rue Saint-Honoré, 75001 Paris

[Book](#) [Visit profile](#)



**Dr. Elodie Lambert**  
**Interested-field:** Neurology  
**Location:** 234 Boulevard de la Villette, 75019 Paris


[Book](#) [Visit profile](#)



**Dr. François Lemoine**  
**Interested-field:** Neurology  
**Location:** 12 Avenue du Maine, 75014 Paris


[Book](#) [Visit profile](#)

Figure 4.11 : Interface des médecins généralistes ont le meme champ d’interet (Neurology).

Medics  [Logout](#)


---

**Specialist doctors do not have the same SubSpecialization**




**Dr. Isabelle Brunet**  
**SubSpecialization:** Movement Disorders  
**Location:** 89 Rue de la Chapelle, Paris

[Book](#) [Visit profile](#)




**Dr. Yannick Leroy**  
**SubSpecialization:** Stroke  
**Location:** 678 Boulevard Voltaire, 75011 Paris

[Book](#) [Visit profile](#)




**Dr. Laurent Durand**  
**SubSpecialization:** Epilepsy  
**Location:** 123 Avenue de Clichy, 75018 Paris

[Book](#) [Visit profile](#)



**Dr. Patrice Fabre**  
**SubSpecialization:** Epilepsy  
**Location:** 67 Rue de Tolbiac, 75013



**Dr. Julie Carpentier**  
**SubSpecialization:** Stroke  
**Location:** 345 Rue de Belleville, 75020

Figure 4.12 : Interface des médecins spécialistes ont la meme spécialité (Neurology)mais différentes micro-spécialités.

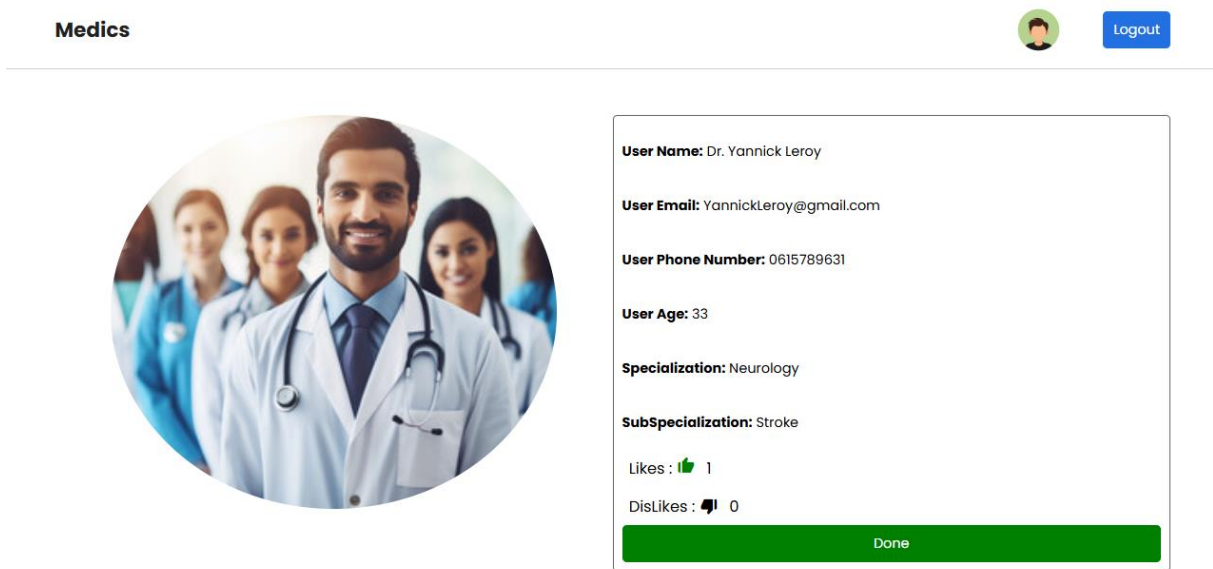


Figure 4.13: vu du profil médecin par le patient.

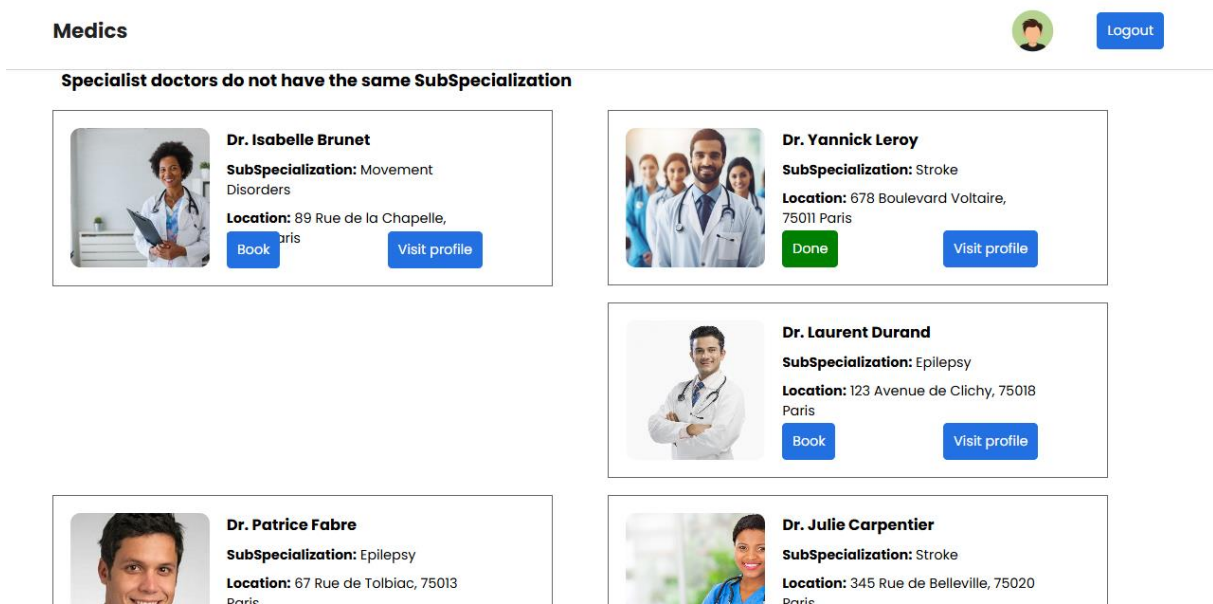


Figure 4.14: Prendre rendez-vous.

# Chapitre IV : Implémentation du système Medics

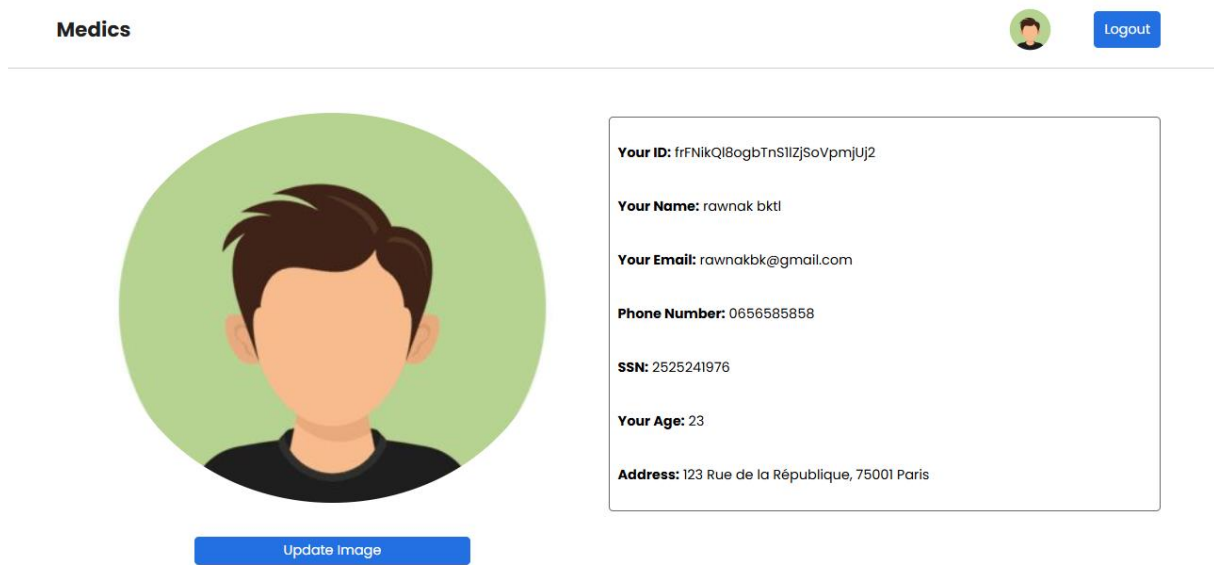


Figure 4.15: profil patient.

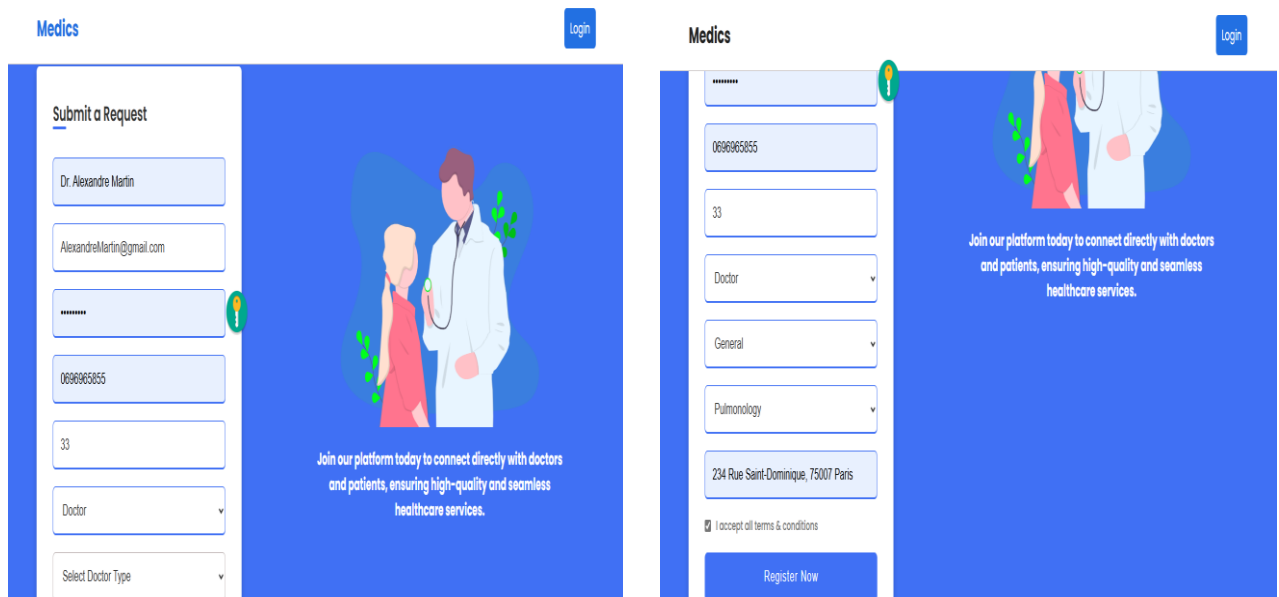


Figure 4.16 : Interface d'inscription coté médecin

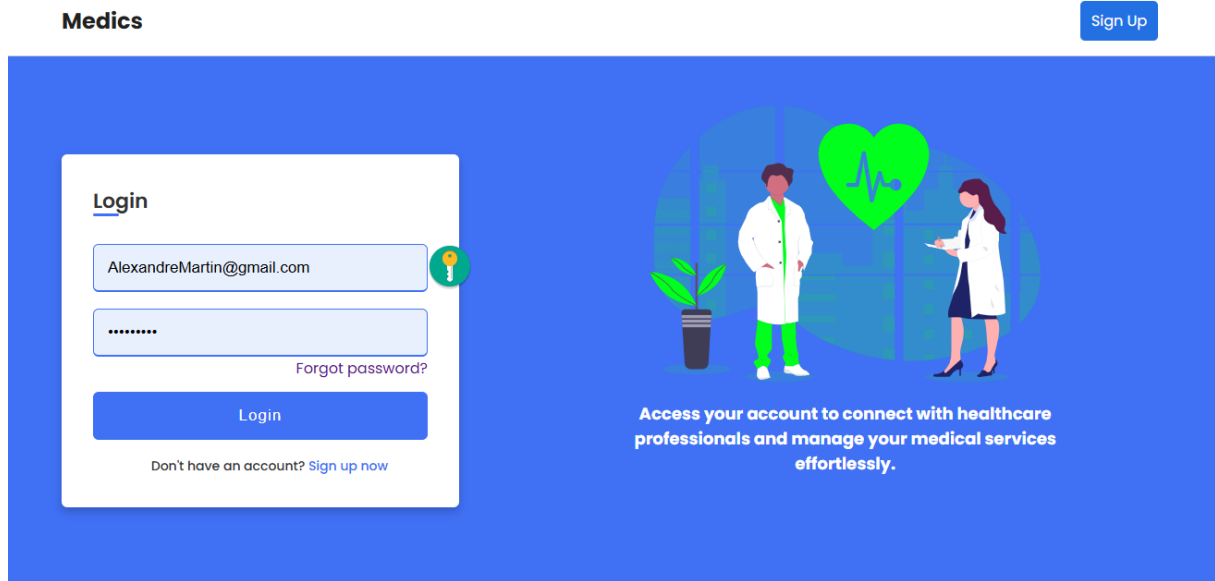


Figure 4.17 : Interface de connexion coté médecin.

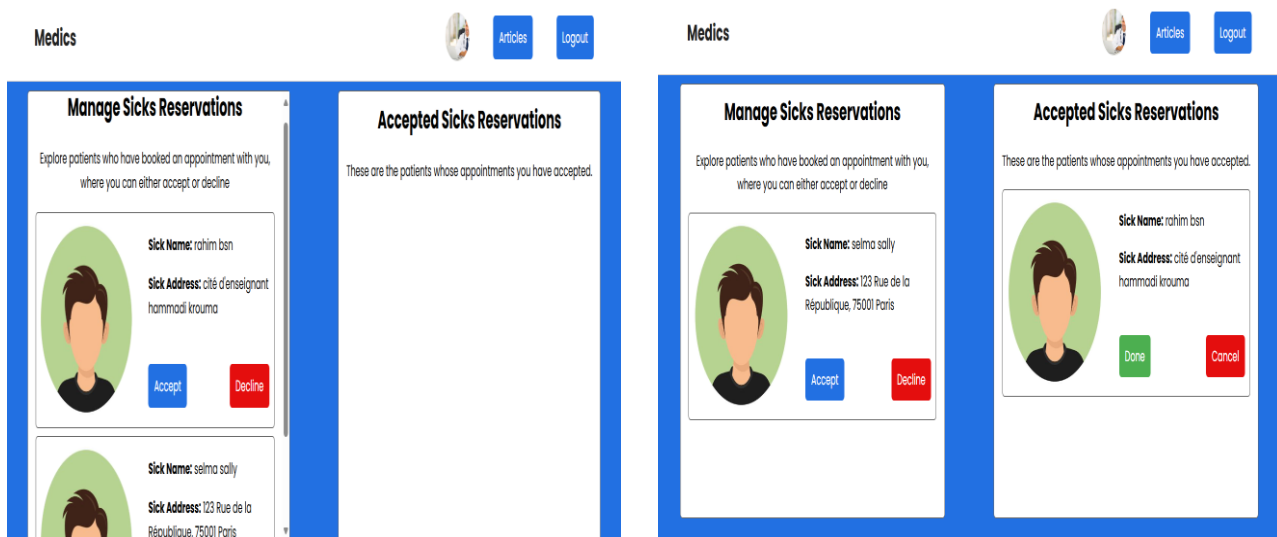


Figure 4.18 : Gestion des patients

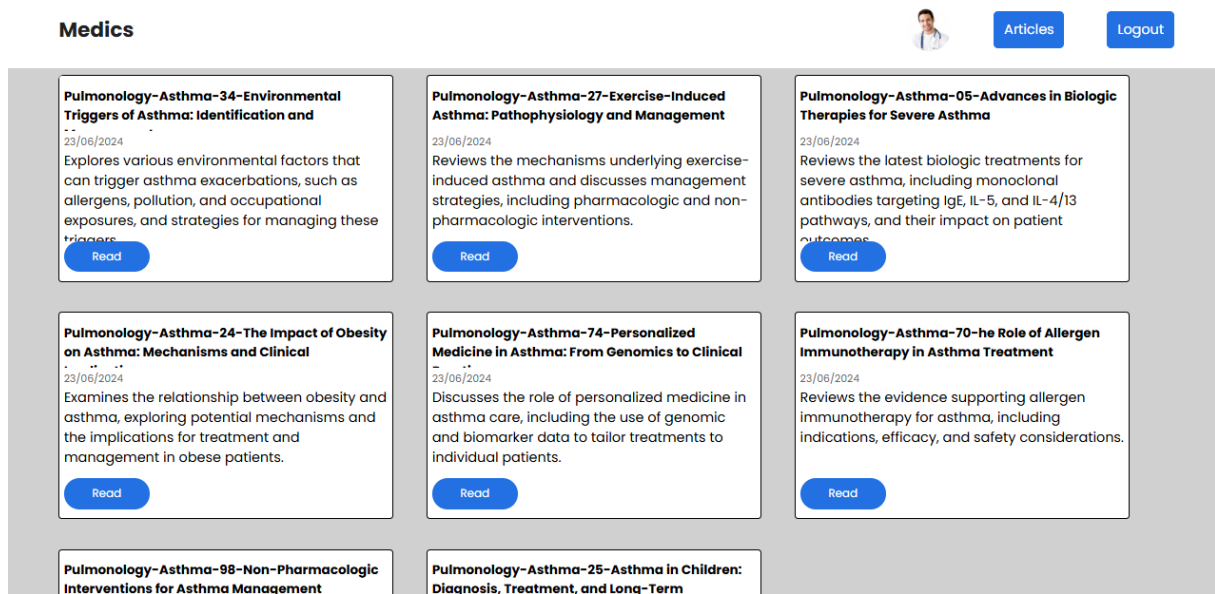


Figure 4.19 : Recommandation des articles du système Medics pour un médecin spécialiste (pulmonology-Asthma)

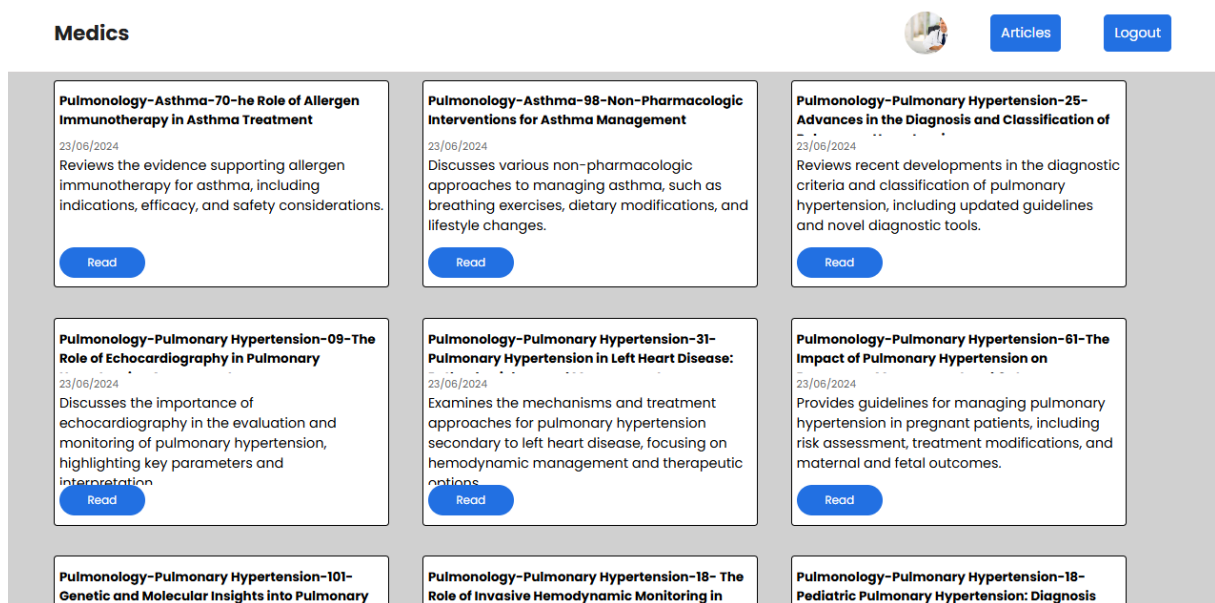


Figure 4.20 : Recommandation des articles du système Medics pour un médecin généraliste (pulmonologie)

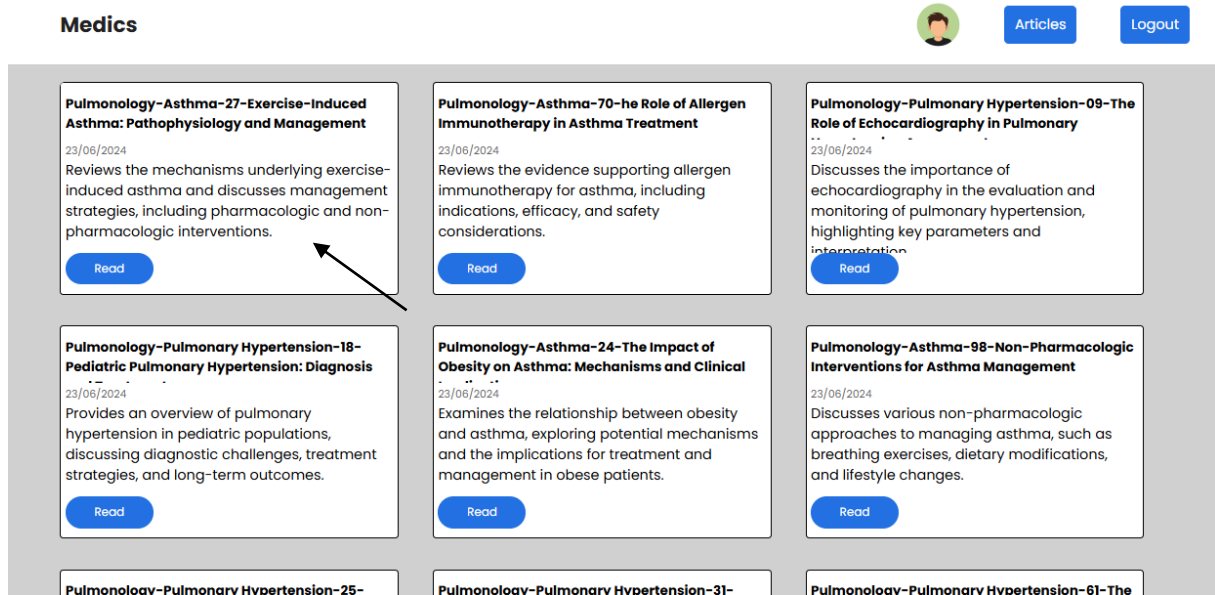


Figure 4.21 : Classification des articles avant l'évaluation et la mise à jour

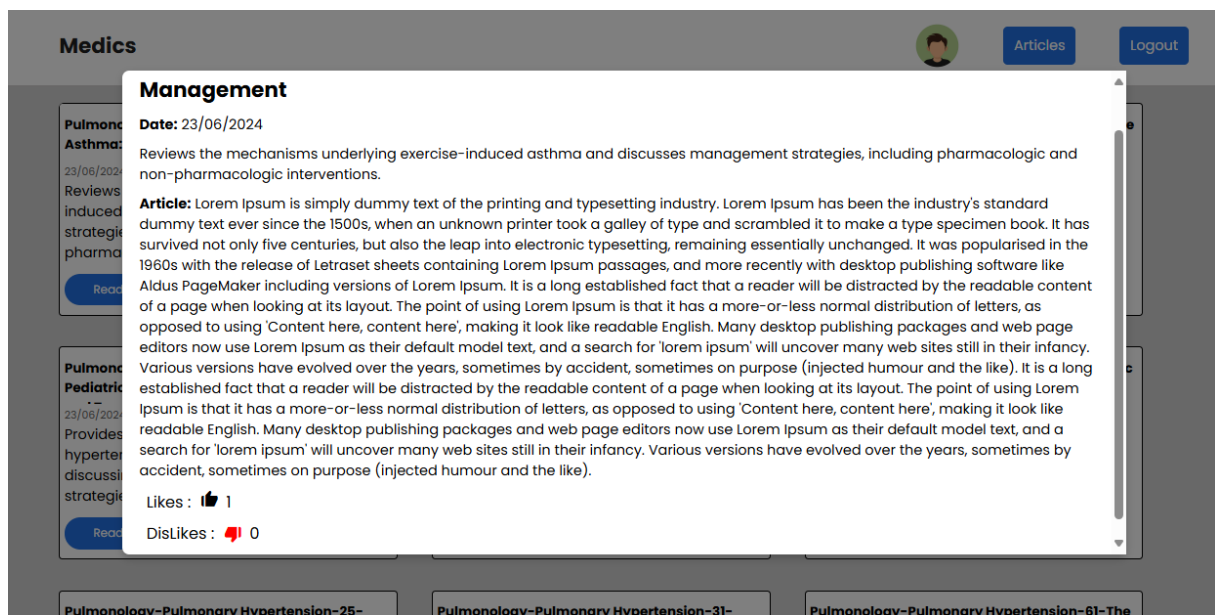


Figure 4.22 : Evaluation d'article (mettre dislike)

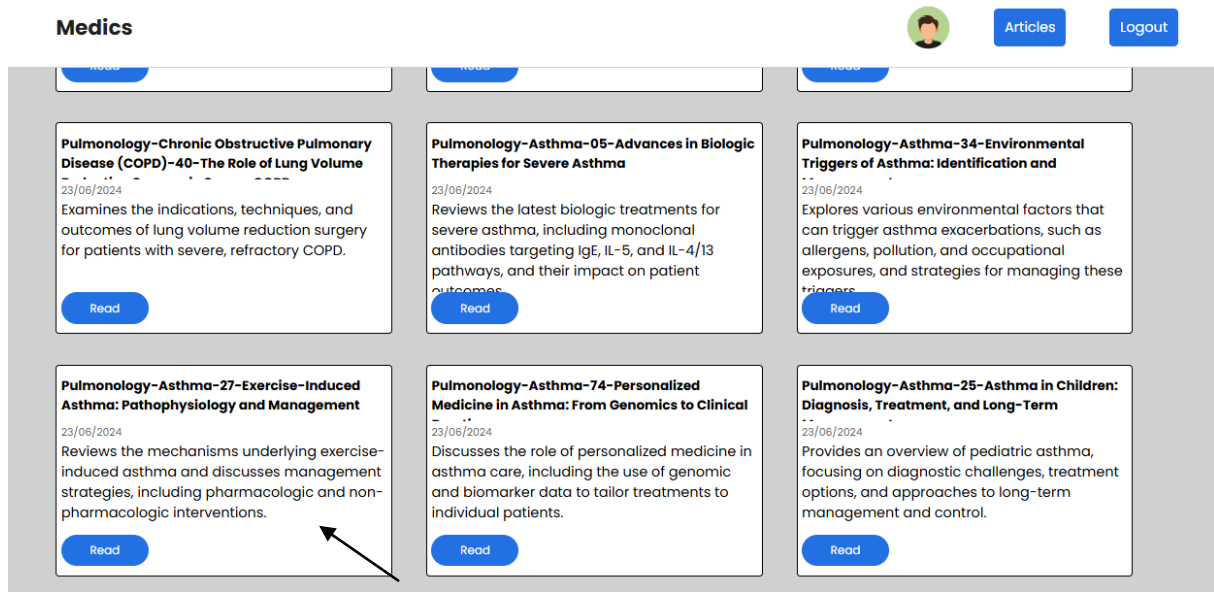


Figure 4.23 : mise à jour des articles

## 7. Conclusion

Dans ce chapitre, nous avons présenté les outils et langages nécessaire au développement du système Medics, nous avons détaillé l'implémentation du système de recommandation sensible au contexte et avons finis par la présentation de quelques interfaces de l'application.

# Conclusion générale

---

## Conclusion générale

---

### Conclusion générale

En conclusion, ce mémoire a mis en évidence la valeur des systèmes de recommandation dans le domaine médical, en particulier leur capacité à fournir des recommandations personnalisées et adaptées au contexte spécifique de chaque utilisateur.

Le développement de l'application Medics a démontré comment ces systèmes peuvent être intégrés dans les pratiques professionnelles pour améliorer la qualité des soins et faciliter l'accès à des informations médicales pertinentes.

Les défis associés à l'implémentation de tels systèmes, comme la gestion des préférences personnelles et des contextes spécifiques, ont été abordés et surmontés grâce à une conception soignée et à l'utilisation de technologies adaptées.

L'application Medics, avec son système de recommandation sensible au contexte, représente une avancée significative dans le domaine des technologies médicales, offrant aux professionnels de santé un outil puissant pour la prise de décisions éclairées et la mise à jour continue des connaissances médicales.

# **Bibliographie**

---

## Bibliographie

- [1] Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3), 56-58.
- [2] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4), 331-370.
- [3] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6), 734-749.
- [4] Negre, E. (2015). Les systèmes de recommandation : une catégorisation. *Interstices.info*.
- [5] Negre, E. (2015). Les systèmes de recommandation : une catégorisation. *Interstices.info*.
- [6] Towle, B., & Quinn, C. (2000). Knowledge based recommender systems using explicit user models. *Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04*.
- [7] DataScienceToday. (n.d.). Apprentissage avec les arbres de décision - DataScienceToday. [Online]. Available: <https://datasciencetoday.net/index.php/fr/machine-learning/109-ml-sup/188-apprentissage-avec-les-arbres-de-decision>
- [8] Guru99. (n.d.). Algorithm Bayes naïf [Online]. Available: <https://www.guru99.com/fr/naive> dans l'apprentissage automatique - Guru99. -bayes-classifiers.html
- [9] Ryax. (n.d.). Deep learning : comprendre les réseaux de neurones artificiels. [Online] Available: <https://ryax.tech/fr/deep-learning-comprendre-les-reseaux-de-neurones-artificiels-artificial-neural-networks/>
- [10] Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1-35). Springer, Boston, MA.
- [11] Kanaan, A., Benabdeslem, K., Bourhis, K., & Canitia, B. (2020). Approches hybrides pour la recommandation dans le domaine du pneumatique. In *EGC (Vol. RNTI-E-36, pp. 133-144)*.
- [12] Billsus, D., & Pazzani, M. J. (2000). User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, 10(2-3), 147-180.
- [13] Smyth, B., & Cotter, P. (2000). A personalized television listings service. *Communications of the ACM*, 43(8), 107-111.
- [14] Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence* (pp. 714-720).
- [15] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4), 331-370.

## Bibliographie

---

- [16] Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In Proceedings of the Eighteenth National Conference on Artificial Intelligence (pp. 187-192).
- [17] Boumediene, A. (2020). Une-application-medicale-de-recommandation-contextuelle-des-documents [PDF]. Université de Tlemcen. Available: <http://dspace.univ-tlemcen.dz/bitstream/112/5908/1/Une-application-medicale-de-recommandation-contextuelle-des-documents.pdf>
- [18] Schilit, B. N., Adams, N. I., & Want, R. (1993). Context-aware computing applications. Proceedings of the 1993 ACM SIGMOBILE International Conference on Mobile Computing and Networking, 85-90.
- [19] Hull, R., Neaves, P., & Bedford-Roberts, J. (1997). Towards situated computing. Proceedings of the 1st International Symposium on Wearable Computers, 146-153.
- [20] Byun, H. E., & Cheverst, K. (2004). Utilizing context history to provide dynamic adaptations. Applied Artificial Intelligence, 18(6), 533-548.
- [21] Site : [www.amazon.fr](http://www.amazon.fr)
- [22] Site: <https://developer.mozilla.org/fr/docs/Web/JavaScript>
- [23] Site : <https://nodejs.org/>
- [24] Site: <https://code.visualstudio.com/>
- [25] Site: <https://git-scm.com/>
- [26] Site : <https://reactjs.org/>



## بطاقة معلومات خاصة بمذكرة التخرج

رقم التسجيل :

36002891

اسم و لقب الطالب :

Rawnak Boukhetala

.....\*

.....\*

.....\*

.....\*

.....\*

.....\*

اسم و لقب المشرف على المذكرة : ..Nabet Aicha.....

عنوان المذكرة : ..Système de recommandation.....

Application dans le domaine médical.....

القسم : ..Informatique.....

المستوى : ..Master 2.....

التخصص : ..Génie logiciel avancée et application.....

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 20 Août 1955- skikda-

Faculté des Sciences

Département d'Informatique



جامعة 20 أوت 1955 - سكيكدة

كلية العلوم

قسم الاعلام الالى

الرقم : ..... / ق / 1 / أ / ل.م.ع / 2024

### Autorisation de Dépôt de Mémoire de Master

Je soussigné: ... *Nobet Aïcho* .....

Certifie que l'étudiant(e) : ... *Roumouk Boukhetala* .....

Spécialité : ... *Création logiciel Avancée et Application* .....

Ayant soutenu le projet intitulé : ... *Système de* .....

... *recommandation : Application dans le* .....

... *domaine Medical* .....

A apporté les corrections nécessaires sur son manuscrit de Master

Signature de l'encadreur

*Maher*