

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة محمد الصديق بن يحيى - جيجل -

Université Mohamed Sadik Benyahia - Jijel -

Faculté des Sciences Exactes et Informatique

Département d'informatique



MEMOIRE

Présenté pour l'obtention du diplôme de MASTER

En : Informatique

Spécialité : Systèmes d'Information et Aide à la Décision

Thème

Implémentation d'un système de recommandation de livres

Réalisé par :

Belguet Abdennour
Hebbache Housseem

Encadré par :

Dr. Saloua Chettibi

Année universitaire 2018/2019

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

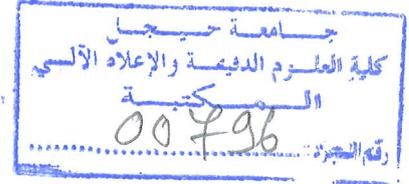
جامعة محمد الصديق بن يحيى - جيجل -

Université Mohamed Sadik Benyahia - Jijel -

Faculté des Sciences Exactes et Informatique

Département d'informatique

inf. SIAD. 06/19



MEMOIRE

Présenté pour l'obtention du diplôme de MASTER

En : Informatique

Spécialité : Systèmes d'Information et Aide à la Décision

Thème

Implémentation d'un système de recommandation de livres

Réalisé par :

Belguet Abdenour
Hebbache Housseem

Encadré par :

Dr. Saloua Chettibi

Année universitaire 2018/2019



Remerciement

Nous tenons tout d'abord à rendre grâce à Dieu le tout puissant qui nous a donné, la force, le courage, et la patience d'accomplir ce travail.

Nous adressons nos sincères remerciements à notre encadreur **Dr. Chettibi Saloua**
Pour son soutien, ses précieux conseils, sa compréhension, sa détermination et aussi Pour ses encouragements tout au long de ce travail.

Nous tenons à exprimer notre gratitude aux membres de jury qui ont accepté de juger notre travail.

Sans oublier également les enseignants qu'ils nous accompagnés durant nos études supérieures à l'université de Jijel.

A notre promotion SIAD informatique 2018-2019.
A toutes les personnes qui ont participés de près ou de loin à la réalisation de ce travail.

« Merci à tous »

Dédicaces

À

Ma chère mère

À

Mon cher père

Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

À

Mes frères

À

Mes sœurs

À

À mes chers amis

Ahmed, Fares, Mounir, Amel, Bilal

Abderahim, Soufiane, Farouk

Pour leur aide et support dans les moments difficiles.

À

Tous ceux que j'aime et ceux qui m'aiment

À

Toute ma famille

Et enfin à

Toute la promotion D'informatique 2018-2019.

Houssem

Dédicaces

À

Ma chère mère

À

Mon cher père

Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir et de m'épauler pour que je puisse atteindre mes objectifs.

À

Mes frères

À

Mes sœurs

À

À mes chers amis

Pour leur aide et support dans les moments difficiles.

À

Tous ceux que j'aime et ceux qui m'aiment

À

Toute ma famille

Et enfin à

Toute la promotion D'informatique 2018-2019.

Abdennour

RESUME

Les systèmes de recommandation sont devenus dominants dans la plupart des applications Web, y compris les grands sites commerciaux tels qu'Amazon, offrant aux consommateurs les éléments et les produits correspondant à leurs intérêts. Une bibliothèque universitaire donne accès à un grand nombre de livres. Les étudiants sont confrontés au problème de choix des livres les plus appropriés à emprunter. Afin de soutenir les étudiants dans cette tâche, ce mémoire propose l'utilisation d'un système de recommandation de livres. Le système proposé met en œuvre deux techniques de recommandation, à savoir : la recommandation basée sur le contenu et celle basée sur le filtrage collaboratif. Le système de recommandation est implémenté au sein de l'application web UBook (Université Book).

Mots-clés : Système de Recommandation, livre, recommandation basée sur le contenu, recommandation basée sur le filtrage collaboratif, application web

Abstract

Recommendation systems have become dominant in most web applications, including large commercial sites such as Amazon, offering consumers the elements and products that best suit their interests. A university library gives access to a large number of books. Students are faced with the problem of choosing the most appropriate books to borrow. In order to support students in this task, this thesis proposes the use of a book recommendation system. The proposed system implements two recommendation techniques, namely: the content-based recommendation and the collaborative filtering recommendation. The recommendation system is implemented within the UBook web application (Book University).

Keywords: Recommendation system, book, content-based recommendation, collaborative filtering recommendation, web application

المخلص

تحتوي المكتبات الجامعية على عدد كبير من الكتب العلمية التي يستفيد منها الأساتذة و الطلاب حيث يواجهان مشكلة اختيار أنسب الكتب من أجل اقتراضها و لمساعدتهم تقترح هذه الذاكرة استخدام نظام خاص لتوصية هذه الكتب ، حيث يعتمد هذا النظام على أسلوبين للتوصية هما: التوصية القائمة على المحتوى و التوصية القائمة على التصفية التعاونية ، والتي يتم تنفيذها ضمن تطبيق ويب.

الكلمات الرئيسية : نظام توصية ، الكتب ، التوصية القائمة على المحتوى ، التوصية القائمة على التصفية التعاونية ، تطبيق ويب

Table de matières

Résumé	I
Liste des figures	V
Liste des tableaux	VII
Introduction générale	01
Chapitre 01: Les applications web	
1.1. Introduction	04
1.2. Les applications Web	04
1.2.1. L'histoire de Web	04
1.2.2. Comparaison de l'application web avec l'application mobile et le site web....	05
1.2.3. L'architecture des applications web	08
1.2.4. L'application Client/serveur	09
1.2.5. Le fonctionnement d'une application web	09
1.3. Les langages utilisés pour la création des applications web	10
1.3.1. Les langages des bases de données.....	10
1.3.2. Les langages de programmations.....	11
1.3.3. Les langages et les outils d'interface	13
1.4. Application web adaptatif	14
1.4.1. La grille d'affichage flexible	15
1.4.2. Le média flexible	15
1.4.3. Le média Queries	15
1.5. Les avantages de développement des applications web.....	17
1.6. Conclusion.....	18
Chapitre 2 : Les systèmes de recommandation	
2.1. Introduction	20
2.2. Historique et définition des systèmes de recommandation	20
2.3. Objectifs des systèmes de recommandation et exemples existants.....	21
2.4. Approches de recommandation	22
2.4.1. Recommandation basé sur le contenu.....	22
2.4.1.1. Le modèle de l'espace vectoriel	23
2.4.1.2. La classification bayésienne	25

Table de matières

2.4.1.3. Avantages	25
2.4.1.4. Limites.....	25
2.4.2. Recommandation basé sur le Filtrage collaboratif.....	26
2.4.2.1. La méthode basée de mémoire	27
2.4.2.2. La méthode basée de modèle	28
2.4.2.3. Avantages	29
2.4.2.4. Limites.....	29
2.4.3. Le Filtrage démographique	29
2.4.4. Le Filtrage basé utilité.....	30
2.4.5. Le Filtrage communautaire	30
2.4.6. Recommandation basé sur le Filtrage hybride	30
2.5. Conclusion	35
Chapitre 03: Conception	
3.1. Introduction.....	37
3.2. Modélisations UML	37
3.3. Etude préliminaire.....	37
3.3.1. Identifications des acteurs	37
3.3.2. Identifications des messages	38
3.4. Capture des besoins.....	39
3.4.1. Capture des besoins fonctionnels	39
3.4.2. Capture des besoins techniques.....	40
3.5. Les cas d'utilisation	40
3.5.1. Identification des cas d'utilisation	40
3.5.2. Relations entre cas d'utilisation.....	42
3.5.3. Représentation de diagramme de cas d'utilisation.....	43
3.5.4. Description textuelle des cas d'utilisation	44
3.6. Les diagrammes	51
3.6.1. Diagramme de séquence principal de l'application	51
3.6.2. Le diagramme de classe de l'application	53
3.7. Méthodes de recommandation utilisées	55
3.7.1. La recommandation basée sur la recherche des livres.....	55
3.7.2. La recommandation basée sur les emprunts des livres	57
3.7.3. La recommandation basée les notes des livres	57
3.8. Classement des livres	59

Table de matières

3.9. Conclusion..... 59

Chapitre 04 : Implémentation et réalisation

4.1. Introduction..... 61

4.2. Langages de programmation et les outils utilisés 61

4.3. Description de l'interface et composantes 61

4.4. L'adaptatif de l'application 68

4.5. Conclusion..... 71

Conclusion générale 73

Bibliographie..... 74

Liste des Figures

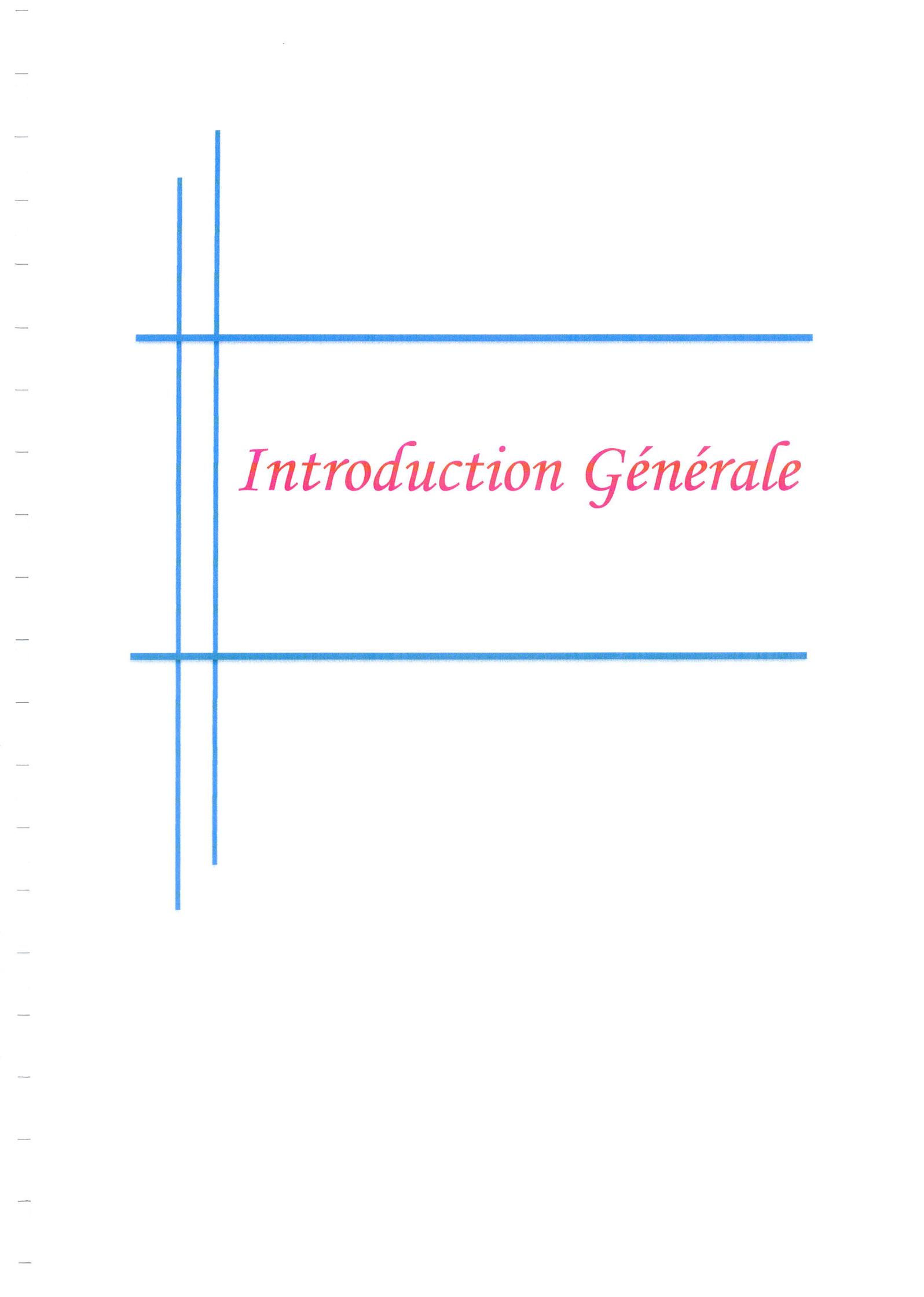
Figure 1.1. L'application web de facebook	06
Figure 1.2. L'application mobile facebook	07
Figure 1.3. Le site web boutique de prêt à porter	08
Figure 1.4. Architecture 3-tiers d'une application web	09
Figure 1.5. Fonctionnement d'une application web.....	10
Figure 1.6. Application adaptative	15
Figure 1.7. Exemple de norme HTML5.....	16
Figure 1.8. Exemple de style CSS	16
Figure 2.1. Un système de recommandation basé sur le contenu.....	23
Figure 2.2. Un système de recommandation collaboratif.....	27
Figure 2.3. Le système de recommandation hybride.....	31
Figure 2.4. Hybridation pondérée.....	32
Figure 2.5. Hybridation à bascule.....	32
Figure 2.6. Hybridation mixte.....	33
Figure 2.7. Hybridation par combinaison de caractéristiques	33
Figure 2.8. Hybridation en cascade	34
Figure 2.9. Hybridation en augmentation de caractéristiques	34
Figure 2.10. Hybridation méta-niveau.....	35
Figure 3.1. Diagramme de contexte dynamique de système	38
Figure 3.2. Diagramme de cas d'utilisation de l'application	44
Figure 3.3. Diagramme de séquence de l'authentification.....	52
Figure 3.4. Diagramme de séquence d'ajouter un livre	52
Figure 3.5. Diagramme de séquence de réservation un livre	53

Liste des Figures

Figure 3.6. Diagramme de séquence récupération des livres	53
Figure 3.7. Diagramme de classes de l'application UBook	54
Figure 4.1. La page d'accueil de l'application UBook.....	61
Figure 4.2. Les livres les plus empruntés	62
Figure 4.3. Les livres de meilleure notation.....	62
Figure 4.4. La page d'authentification	63
Figure 4.5. La page des informations des livres.....	64
Figure 4.6. La liste des livres recommandés à base de notation	64
Figure 4.7. Le code PHP des fonctions distance et moyenne.....	65
Figure 4.8. Le code PHP des fonctions corrélation et facteur de normalisation	65
Figure 4.9. Le code PHP de la similarité (Pearson)	66
Figure 4.10. La liste des livres recommandés par les emprunts.....	66
Figure 4.11. Le code PHP de la fonction Cosinus.....	67
Figure 4.12. La liste des livres recommandés par les informations des livres.....	67
Figure 4.13. Le code PHP des fonctions similarité de Dice.....	68
Figure 4.14. L'affichage de l'application en large écran	68
Figure 4.15. L'affichage de l'application en PC	69
Figure 4.16. L'affichage de l'application en tablette	70
Figure 4.17. L'affichage de l'application en mobile	71

Liste des Tableaux

Tableau 3.1. Spécifications des messages	39
Tableau 3.2. Identification des cas d'utilisation de système de recommandation	41
Tableau 3.3. Identification des cas d'utilisation de l'utilisateur	41
Tableau 3.4. Identification des cas d'utilisation de l'administrateur	42
Tableau 3.5. Identification des cas d'utilisation de l'anonyme.....	42
Tableau 3.6. Description textuelle du cas d'utilisation d'authentification.....	45
Tableau 3.7. Description textuelle du cas d'utilisation d'ajout un utilisateur.....	46
Tableau 3.8. Description textuelle du cas d'utilisation d'ajout un livre	47
Tableau 3.9. Description textuelle du cas d'utilisation de consulter un livre	47
Tableau 3.10. Description textuelle du cas d'utilisation de rechercher un livre	48
Tableau 3.11. Description textuelle du cas d'utilisation de noter un livre.....	48
Tableau 3.12. Description textuelle du cas d'utilisation de recommander des livres	49
Tableau 3.13. Description textuelle du cas d'utilisation de consulter les recommandations.....	49
Tableau 3.14. Description textuelle du cas d'utilisation réserver un livre	50
Tableau 3.15. Description textuelle du cas d'utilisation de récupération d'un livre.....	51
Tableau 3.16. Comparaison d'un livre avec les autres livres de même spécialité.....	55
Tableau 3.17. Résultat de similarité de Dice sur l'auteur éditeur	56
Tableau 3.18. L'intersection des mots entre livre2 et les autres livres	56
Tableau 3.19. La somme des similarités	56
Tableau 3.20. L'emprunt des livres par utilisateur en binaire.....	57
Tableau 3.21. Notation des livres par les utilisateurs.....	58

A decorative graphic on the left side of the page consists of a vertical scale with horizontal tick marks, two vertical blue lines, and two horizontal blue lines. The title is centered between the two horizontal lines.

Introduction Générale

Introduction Générale

Bien avant l'arrivée d'internet, les individus se recommandaient des objets par l'intermédiaire des opinions orales. Ces dernières années, et principalement sur les applications Web qui sont maintenant les plus communes dans le monde de l'internet, le volume d'information disponible augmente continuellement. Les opinions orales sont inadéquates pour gérer cette énorme quantité d'informations. Il est nécessaire de disposer de techniques plus avancées afin de fournir des informations pertinentes pour les utilisateurs. Les systèmes de recommandation aident à résoudre ce problème.

Les systèmes de recommandation sont spécifiquement adaptés à l'objet à recommander (films, musique, livres, nouvelles, images, pages Web, etc.). Grâce à une forme spécifique de filtrage de l'information, un système de recommandation permet de présenter aux utilisateurs des éléments qui sont susceptibles de les intéresser, et ce, en se basant sur ses préférences et son comportement.

Il existe plusieurs approches de recommandation :

- le filtrage basé sur le contenu consiste à recommander à un utilisateur les documents similaires à ceux appréciés par cet utilisateur dans le passé.
- le filtrage collaboratif recommande les objets appréciés par les utilisateurs qui ont auparavant fait des choix similaires à ceux de l'utilisateur courant.
- le filtrage démographique recommande les objets à partir des données d'utilisateur (le sexe, l'âge, la ville, le pays, etc.).
- le filtrage basé utilité il base sa recommandation sur une évaluation de la correspondance entre les besoins des utilisateurs et l'ensemble des items disponibles.
- le filtrage communautaire fait des recommandations à partir des relations de l'utilisateur avec ces amis dans les réseaux sociaux (Facebook, Twitter, etc.)
- le filtrage basé connaissance se basant sur des inférences des besoins et des préférences d'un utilisateur.
- recommandation basé sur le filtrage hybride il combine deux ou plusieurs techniques de recommandation différentes.

Problématique

Une bibliothèque universitaire contient une énorme quantité de livres, de mémoires et de thèses. Même si un effort considérable est consacré à la classification de ces différents documents, l'étudiant trouve toujours des difficultés à choisir les livres qui lui sont les plus adaptés. Un système de recommandation pourrait être d'une grande utilité dans ce contexte.

L'objectif de ce mémoire est la conception et l'implémentation d'un système de recommandation de livres dans une bibliothèque universitaire. Ce système doit être accessible aux étudiants via une application Web adaptative. L'approche de recommandation proposée repose sur le filtrage basé sur le contenu et le filtrage collaboratif.

Organisation du mémoire

Notre mémoire est organisé en 4 chapitres:

- **Chapitre 01** : Dans ce chapitre, nous allons parler sur les applications Web avec des caractéristiques, des exemples, et des avantages de développement des applications Web.
 - **Chapitre 02** : Ce chapitre porte sur les systèmes de recommandation: leur avantages et leur limites.
 - **Chapitre 03** : Dans ce chapitre est présentée la conception de notre système de recommandation ainsi que de notre application web.
 - **Chapitre 04** : Ce chapitre est centré sur la partie implémentation de notre travail.
- Enfin, nous terminons avec une conclusion générale et quelques perspectives.

Chapitre 1

Les applications Web

- 1. Introduction**
- 2. Les applications Web**
- 3. Les langages utilisés pour la création des applications web**
- 4. Application web adaptatif**
- 5. Les avantages de développement des applications web**
- 6. Conclusion**

1.1. Introduction

Une application Web est un ensemble des pages manipulables directement en ligne grâce à un navigateur web (comme Internet Explorer, Mozilla Firefox, Google Chrome, Safari, etc.) et qui ne nécessite donc pas d'installation sur les machines clientes, contrairement aux applications mobiles, de la même manière que les sites web.

L'utilisateur interagit avec l'application Web à l'aide des pages HTML générées dynamiquement par un serveur WEB. Les pages HTML contiennent des formulaires qui envoient des données et des requêtes au serveur WEB.

Une application web est développée par un ou plusieurs personnes, en utilisant des langages des bases de données SGBD, des langages de codage comme java, PHP, C#, des langages d'interface comme HTML et CSS. Ce chapitre met l'accent sur les caractéristiques des applications WEB.

1.2. Les applications Web

1.2.1. L'histoire de WEB

Au début avec le Web 1.0, l'internet était un espace où les rares personnes qui y avaient accès mettaient en ligne des documents statiques (du HTML pur). À l'époque, il était principalement utilisé par des scientifiques. Ils utilisaient ce support pour facilement partager leurs rapports avec la communauté. Petit à petit, le Net s'est démocratisé mais il ne servait toujours qu'à héberger des sites statiques [1].

Maintenant l'utilisateur n'est plus un simple consommateur passif mais prend part à la production d'informations et à l'évaluation de leur valeur. Tim O'Reilly baptise cette mutation progressive du web statique vers un web participatif par l'appellation Web 2.0.

Ces dernières années, le web a encore beaucoup évolué. Il est devenu encore plus collaboratif et participatif avec le développement de nombreuses applications allant dans ce sens. On a également assisté à une nouvelle forme de web avec le phénomène de web en temps réel ou encore de réalité augmentée, ce que Tim O'Reilly et John Battelle nomme le Web² dans leur article « Web squared: web 2.0 five years on », publié en 2009. Il s'agit d'une évolution du web 2.0 mais pas encore le web 3.0. L'idée est d'étendre la portée du web 2.0 dans le monde réel, directement sur le terrain. Les nouvelles applications sur les smartphones en sont d'ailleurs la démonstration [2].

Au vu des avancées technologiques, nombreux sont ceux à se demander ce que sera le web de demain. Plusieurs visions du web 3.0 apparaissent, dont la plus importante et celle de dire que le web 3.0 sera le web sémantique.

L'idée est de parvenir à un web intelligent, où les machines comprendraient le langage naturel et la signification de l'information sur le web. Les pages web étant actuellement lisibles uniquement par l'Homme, le web sémantique a comme objectif de les rendre lisibles par les ordinateurs. Ils parviendraient à associer les données pour réaliser certaines tâches comme la recherche ou l'association d'informations dans le but de répondre précisément à une question d'un utilisateur et lui apporter ce qu'il cherche vraiment [2].

1.2.2. Comparaison de l'application web avec l'application mobile et le site web

C'est quoi une application web ?

Une application web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide de son navigateur favori. La tendance actuelle est d'offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur les ordinateurs.

Les technologies utilisées pour développer les applications web sont les mêmes que celles employées dans la création des sites internet [3].

Par exemple une gestion de réservation pour un hôtel, une application de facturation en ligne pour une entreprise ou un commerçant, un outil de gestion de dossiers patients pour un médecin, l'application web de Facebook (Cf. Figure 1.1), une gestion de bibliothèque des livres PDF, etc.

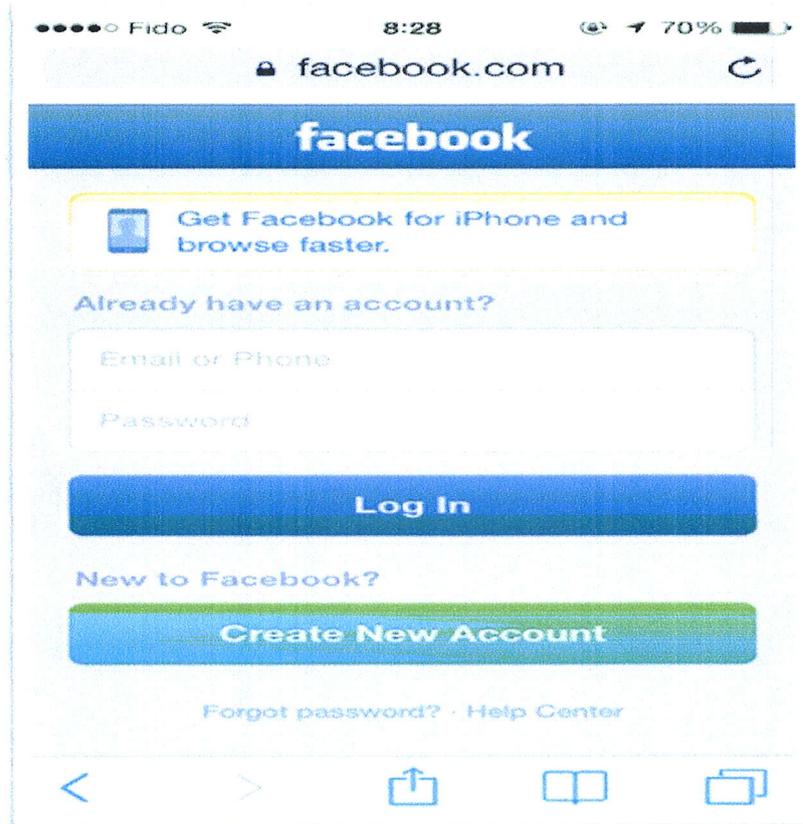


Figure 1.1 : L'application web de facebook. [4]

C'est quoi une application mobile ?

Une application mobile est un logiciel applicatif développé pour être installé sur un appareil électronique mobile, comme un Smartphone, une tablette ou un baladeur numérique.

Une application mobile peut être soit installée directement sur l'appareil dès sa fabrication en usine soit téléchargée depuis un magasin d'applications dit « application store » telle que Google Play, l'App Store ou encore le Windows Phone Store. Une partie des applications disponibles sont gratuites tandis que d'autres sont payantes [5].

Il existe plusieurs systèmes d'exploitation mobiles: iOS (Apple) utilisé sur iPhone et iPad, Android (Google) qui anime un grand nombre de smartphones (Samsung, HTC, LG, Motorola), Blackberry OS, Windows Phone (Microsoft), Symbian (Nokia), Bada (Samsung).

Par exemple : les applications mobile comme Instagram, Youtube, whatsApp, Facebook (Cf. Figure 1.2), etc.

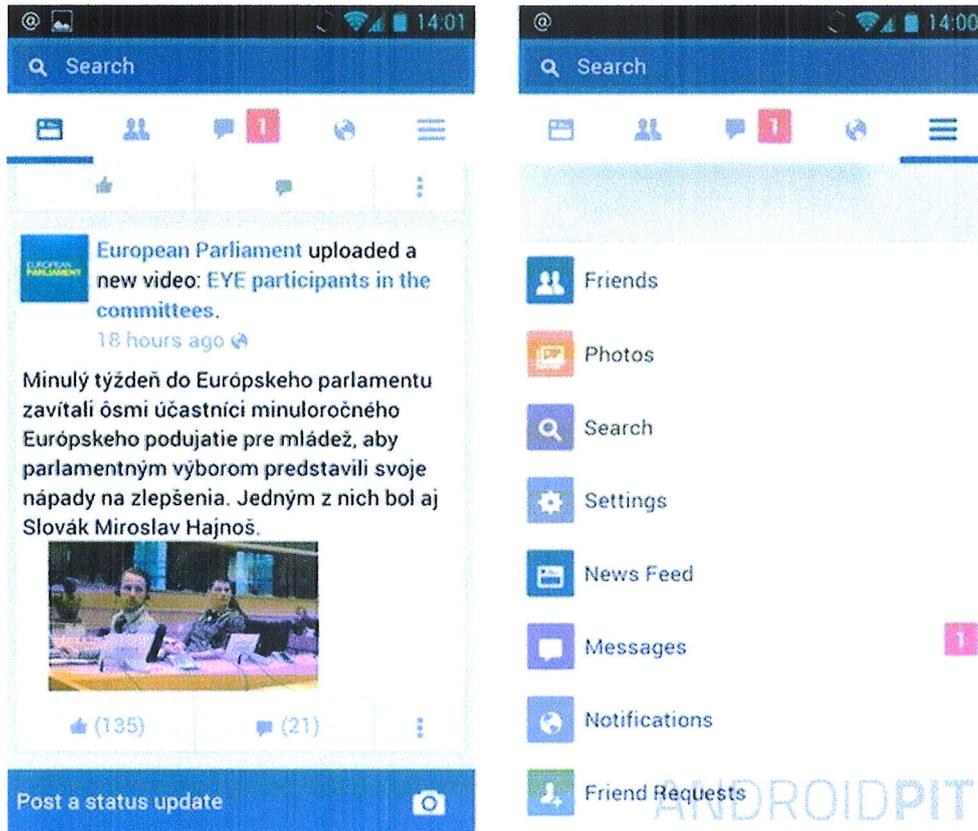


Figure 1.2 : L'application mobile facebook. [6]

C'est quoi un site web ?

Le site web (Cf. Figure 1.3 pour un exemple) quant à lui est constitué de l'ensemble des pages créées par une structure ou un individu reliées entre elles par des liens hypertextes, stockées sur un serveur et accessibles via l'Internet. Chaque site web possède un identificateur appelé adresse URL1, permettant de la retrouver sur le Web et d'identifier le type de protocole d'accès qui lui est associé. Le protocole étant l'ensemble des règles (codes) à respecter pour établir un échange d'informations entre ordinateurs [7].

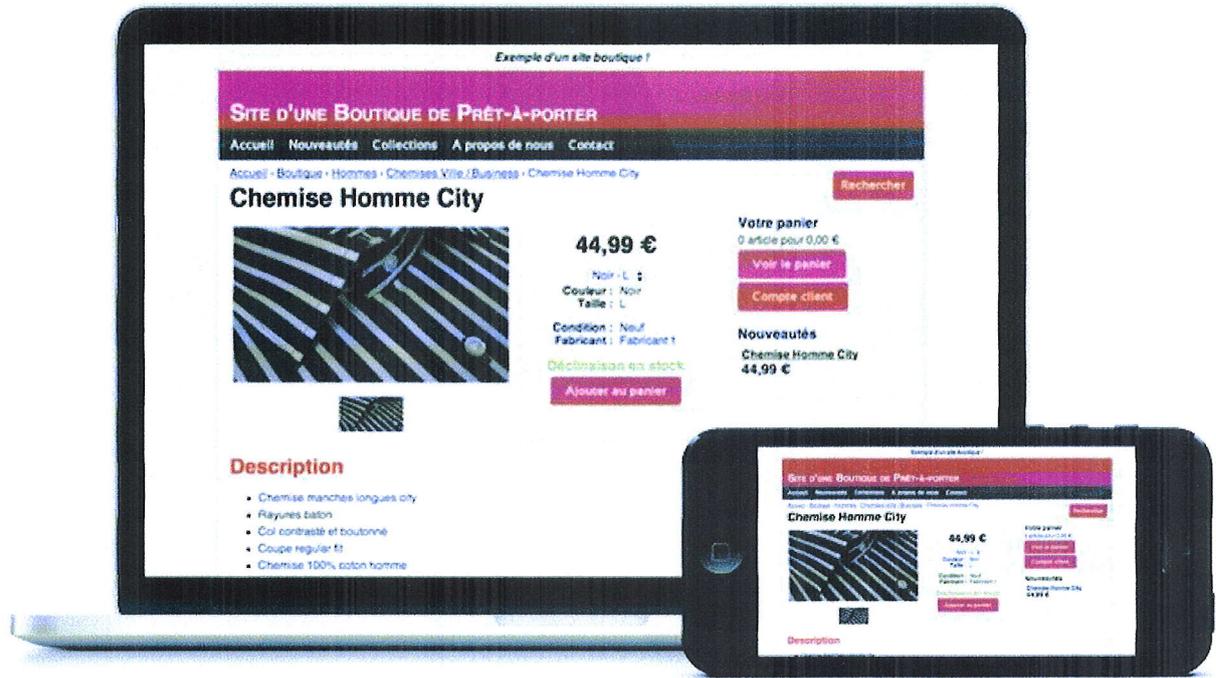


Figure 1.3 : Le site web boutique de prêt à porter. [8]

1.2.3. L'architecture des applications web

Une architecture est un modèle générique et conceptuel qui se rapporte à un sujet et qui représente la fonctionnalité, la structure, le positionnement, l'interrelation des différents types d'éléments (matériel, logiciels, infrastructure) qui la composent [9].

En règle générale, une application est créée sur 3 niveaux (couches) d'abstraction [10] :

La couche présentation : c'est la partie de l'application visible par les utilisateurs (nous parlerons d'interface utilisateur). Dans notre cas, cette couche est un navigateur web, qui se présente sous forme de pages HTML, composé de formulaire et de bouton.

La couche métier: correspond à la partie fonctionnelle de l'application, celle qui est implémentée la logique, et qui a écrit les opérations qui ont été effectuées, en fonction des demandes d'un utilisateur effectuées au travers de la couche présentation.

La couche accès aux données: elle consiste en la partie gérante de l'accès à la base de données du système.

Il existe différentes architectures pour une application web [10] :

- Architecture 1-tiers
- Architecture à 2 niveaux
- Architecture à 3 niveaux
- Architecture n-tiers

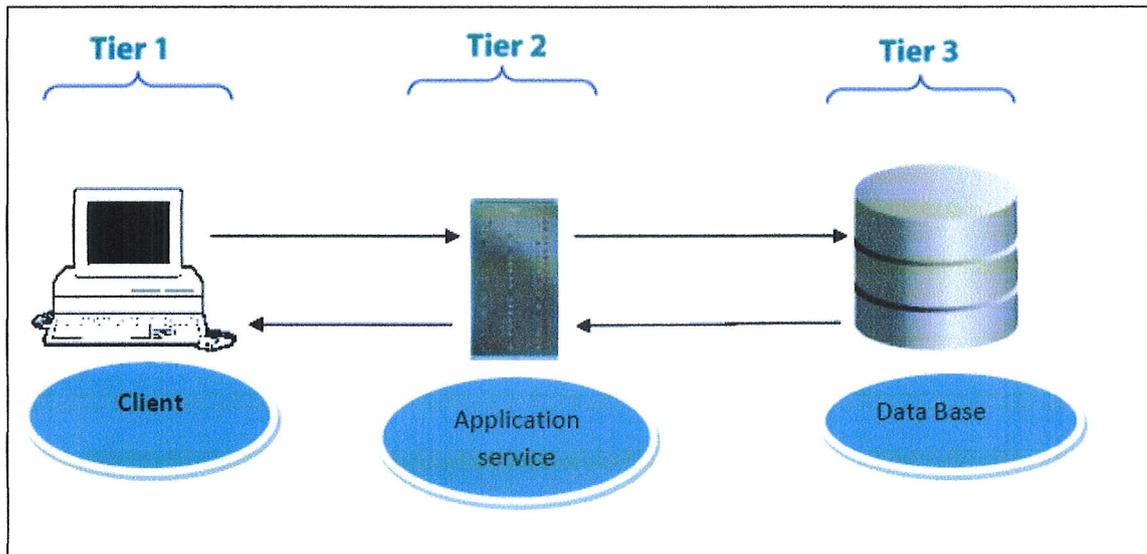


Figure 1.4 : Architecture 3-tiers d'une application web. [11]

1.2.4. L'application Client/serveur

Une application web est dite basée sur un modèle Client - Serveur si la base de données se trouve sur un serveur qui ne sert qu'à ça, et pour interagir avec cette base de données, il faut utiliser un logiciel "client" qui va interroger le serveur et transmettre la réponse que le serveur lui aura donnée. Le serveur peut être installé sur une machine différente du client. Par conséquent, lorsque nous installons un SGBD (système de gestion de base des données), nous installons en réalité deux choses :

- Le serveur
- Le client

Chaque requête (insertion/modification/lecture de données) est faite par l'intermédiaire du client. Nous pouvons donc avoir besoin d'un langage pour discuter avec le client, pour lui donner les requêtes que nous souhaitons effectuer.

1.2.5. Le fonctionnement d'une application web

Lorsqu'un serveur web reçoit une requête de page web statique, il transmet simplement cette page au navigateur requérant. En revanche, lorsque le serveur web reçoit une requête de page dynamique, il transmet cette page à une extension logicielle spéciale chargée d'achever la page. Ce logiciel spécial est appelée serveur d'application. Le serveur d'application lit le code de la page, termine cette page en fonction des instructions figurantes dans le code, puis en retire le code. Il en résulte une page statique que le serveur d'application renvoie au serveur web, lequel transmet alors cette page au navigateur requérant. Le navigateur reçoit uniquement du code HTML pur lorsque la page lui est transmise.

Un serveur d'application nous permet de travailler avec des ressources côté serveur telles que les bases de données. Une page dynamique peut, par exemple, ordonner au serveur d'application d'extraire des données de la base de données et de les insérer dans le code HTML de la page. L'instruction d'extraction des données de la base est nommée requête de base de données. Une requête est composée de critères de recherches rédigés dans un langage de base de données appelé SQL. La requête SQL est rédigée dans les scripts ou les balises côté serveur de la page [11]. Toutes ces actions sont résumées dans le schéma présenté dans la figure 1.5 sise ci-dessous.

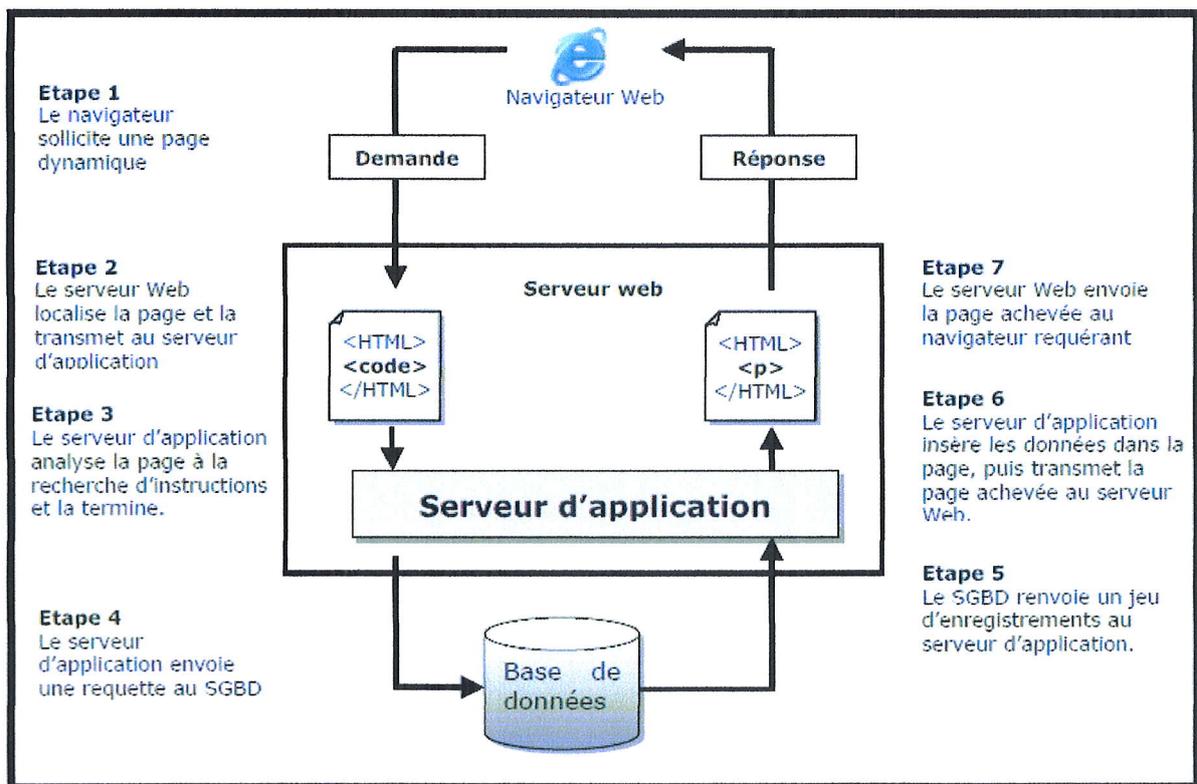


Figure 1.5 : Fonctionnement d'une application web. [11]

1.3. Les langages utilisés pour la création des applications web

Les applications web sont souvent créées par des équipes composées de développeurs qui ont utilisé une grande quantité de langages pour développer ces applications. Tous sont adaptés à des situations particulières, privilégiant des aspects particuliers comme la simplicité, la robustesse, la légèreté, le dynamisme, etc.

1.3.1. Les langages des bases de données

SQLite : SQLite est un moteur de bases de données libre qui implémente la plus part des fonctionnalités du SQL standard. Il utilise donc la plus part des fonctionnalités de SGBDR C'est le SGBDR embarqué le plus utilisé au monde, il ne nécessite aucune configuration, ni serveur

pour fonctionner. Entièrement écrit en C, ce qui le rend très performant. SQLite est également transactionnel, c'est-à-dire qu'il respecte les principes ACID (opération atomique, Consistance des données, indépendant des processus et durable longévité des données). Évidemment, il est public d'où sa popularité [12].

MySQL : MySQL est un Système de Gestion de Bases de Données (SGBD) qui gère pour vous les fichiers constituant une base, prend en charge les fonctionnalités de protection et de sécurité et fournit un ensemble d'interfaces de programmation facilitant l'accès aux données.

La complexité de logiciels comme MySQL est due à la diversité des techniques mises en œuvre, à la multiplicité des composants intervenant dans leur architecture, et également aux différents types d'utilisateurs (administrateurs, programmeurs, non informaticiens, ...) confrontés, à différents niveaux, au système. Au cours de ce livre nous aborderons ces différents aspects, tous ne vous étant d'ailleurs pas utiles, en particulier si votre objectif n'est pas d'administrer une base MySQL. Pour l'instant, nous nous contenterons de décrire l'essentiel, à savoir son architecture et ses composants [13].

Oracle : Oracle est un SGBDRO permettant de gérer des bases de données jusqu'à 65 536 fichiers de 128 Tio (téraoctets c.-à-d. 1012 octets) chacun soit jusqu'à quelques trillions d'octets. Oracle est essentiellement écrits en langage C et certains codes sont actuellement écrits ou réécrits en Java. Oracle est ainsi portable sur près d'une centaine de plates-formes matérielles (des très gros ordinateurs aux micro-ordinateurs) et sur une soixantaine de systèmes d'exploitation (HP-UX, Linux, Mac OS X, Microsoft Windows, Sun Solaris, IBM AIX, IBM z/OS, etc.) [14].

PostgreSQL™ : est un système de gestion de bases de données relationnelles objet (ORDBMS) fondé sur POSTGRES, Version 4.2™. Ce dernier a été développé à l'université de Californie au département des sciences informatiques de Berkeley. POSTGRES est à l'origine de nombreux concepts qui ne seront rendus disponibles au sein de systèmes de gestion de bases de données commerciaux que bien plus tard. PostgreSQL™ est un descendant libre du code original de Berkeley. Il supporte une grande partie du standard SQL tout en offrant de nombreuses fonctionnalités modernes (par exemple les requêtes complexes, les clés étrangères, les triggers, vues modifiables, etc.), grâce à sa licence libérale, PostgreSQL™ peut être utilisé, modifié et distribué librement, quel que soit le but visé, qu'il soit privé, commercial ou académique [15].

1.3.2. Les langages de programmation

PHP : Le langage PHP a été créé par Rasmus Lerdorf en 1994, pour ses besoins personnels.

Comme dans beaucoup d'autres cas la mise à disposition du langage sur l'internet est à l'origine de son développement par d'autres utilisateurs qui y ont vu un outil propre à satisfaire leurs

besoins. Après plusieurs évolutions importantes, PHP en est à sa version 5.2, celle que nous utilisons. La version 6 est annoncée à l'heure où ces lignes sont écrites. PHP le plus souvent associé à MySQL est à l'heure actuelle le plus répandu des langages de programmations pour sites web.

PHP est un langage de programmation, très proche syntaxiquement du langage C, destiné à être intégré dans des pages HTML. Contrairement à d'autres langages, PHP est principalement dédié à la production de pages HTML générées dynamiquement [13].

Java EE : Java Enterprise Edition est apparue à la fin des années 90. Cette évolution apporte au langage Java une plateforme logicielle robuste et complète pour le développement. La plateforme Java EE a souvent été remise en cause, mal utilisée et mal comprise. Des outils OpenSource sont venus la concurrencer. Ces remarques et la concurrence ont permis à Sun d'améliorer son produit et d'éditer des versions de plus en plus abouties. Java EE ne remplace en aucun cas J2SE.

Au contraire, J2SE est la base de Java EE qui est plus complet et qui est axé sur le Web. La plateforme J2SE offre des outils de développement d'applications client/serveur, applications graphiques fenêtrées et Applets. La plateforme J2SE est composée des éléments suivants :

- **La machine virtuelle Java (JVM) :** permet d'exécuter des applications Java. Elle constitue une passerelle et permet une portabilité entre les architectures (Windows, Linux, Mac...).
- **La bibliothèque de classes Java :** un ensemble de composants logiciels prêt à l'emploi.
- **Les outils de développement :** le compilateur javac, un interpréteur Java nommé java, le générateur de documentation javadoc, la console de supervision Java EE est une extension de la plateforme J2SE. Elle permet un développement d'applications qui vont s'exécuter sur un serveur d'applications. Les applications seront utilisées par des clients légers (comme des navigateurs Web) ou bien des applications lourdes (IHM). La dernière version stable de Java EE est la version Java EE 5.0 et fonctionne avec le JDK 5.0 et 6.0 [16].

C#: est un langage de programmation orientée objet, commercialisé par Microsoft depuis 2002 [12], Il est dérivé du C et C++, très proche du Java dont il reprend la syntaxe générale ainsi que les concepts, Il est utilisé pour développer des applications web, ainsi que des applications de bureau, des services web, des commandes, des widgets ou des bibliothèques de classes. En C#, une application est un lot de classes où une des classes comporte une méthode Main, comme cela se fait en Java [17].

Les exécutables en C# sont subdivisés en assemblies, en namespaces, en classes et en membres de classe [18]. Un assembly est la forme compilée, qui peut être un programme (un

exécutable) ou une bibliothèque de classes (une dll). Un assembly contient le code exécutable en MSIL, ainsi que les symboles. Le code MSIL est traduit en langage machine au moment de l'exécution par la fonction *just-in-time* de la plateforme .NET [18].

1.3.3. Les langages et les outils d'interface

JQUERY : est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript.

HTML & CSS :

- **HTML** : C'est un langage de description permettant de structurer et d'afficher différents objets sur un écran (qu'on appellera abusivement "page"). Ces objets peuvent être du texte, des tableaux, des images voire de la vidéo et des sons. Ils peuvent être aussi les éléments devenus classiques des environnements graphiques, à savoir les boutons, pop-listes, listes à ascenseurs, boîtes de dialogue... Sur la plupart de ces objets, il est possible de placer des liens qui vont permettre de se connecter à d'autres pages. Ce langage est donc à l'origine essentiellement statique. Sur le serveur, le fichier HTML décrit une "scène" qui sera envoyée à la demande vers un client (le browser, logiciel de navigation présent sur le poste de consultation). Ce client va décoder les instructions HTML pour afficher la scène qui restera ensuite la même seule une nouvelle requête vers le serveur sera à même d'en modifier l'apparence [19].
- **CSS**: feuille de style c'est un langage qui permet de gérer l'apparence de la page web (agencement, positionnement, décoration, couleur, taille du texte...).

Le langage JavaScript : Créé à l'origine par Netscape, ce langage de programmation est conçu pour traiter localement des événements provoqués par le lecteur (par exemple, lorsque le lecteur fait glisser la souris sur une zone de texte, cette dernière change de couleur). C'est un langage interprété, c'est-à-dire que le texte contenant le programme est analysé au fur et à mesure par l'interprète, partie intégrante du browser, qui va exécuter les instructions. Ce langage a fait l'objet d'une normalisation sous le nom d'ECMAScript [19].

L'AJAX: ou Asynchronous JavaScript And XML n'est pas une technologie en elle-même, mais plutôt un ensemble des technologies: HTML, CSS, JavaScript et XML (parfois remplacé par JSON). L'AJAX consiste à chercher des informations sur le serveur (au format XML ou JSON) puis à les afficher sous forme de HTML/CSS sur la page suite à une action de l'utilisateur observée par du JavaScript ou de manière périodique. Cela permet d'obtenir des applications plus

réactives tout en économisant de la bande passante car seules les données utiles sont échangées, elles sont traitées après réception par le JavaScript [1].

Bootstrap : est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.

Le Bootstrap a été créé au départ par Mark Otto et Jacob Thornton pour répondre à des besoins internes de développement de Twitter au niveau de l'uniformisation de l'aspect des pages web. Il s'agissait juste de stylisation CSS, mais le framework s'est ensuite enrichi de composants Javascript. Il a ensuite été publié en 2011 en devenant rapidement populaire parce qu'il est venu se positionner dans un espace vacant du développement. Son système de grille de 12 colonnes est devenu une référence. Le framework en est actuellement à la version 3. La version 4 est actuellement en phase de finalisation [20].

1.4. Application web adaptatif

Créé par Ethan Marcotte. Il a réuni trois techniques existantes (positionnement en grille fluide, images fluides et media queries) en une seule approche qu'il a appelée responsive web design. Les termes suivants sont souvent utilisés et désignent peu ou pour la même chose : design fluide, mise en page élastique, design liquide, mise en page adaptative, design multi-appareils ou design flexible [21].

Le Responsive Web Design (RWD), ou conception web adaptative, regroupe une série de techniques de conception graphique et de développement permettant de créer un site qui pourra s'auto-adapter en fonction de la taille d'un écran (Cf. Figure 1.6). Ordinateur de bureau, tablette, Smartphone, télévision connectée. Son objectif consiste à prévoir tous les formats de moniteur, et proposer des modes de lecture et de navigation pour chacun d'eux [22].

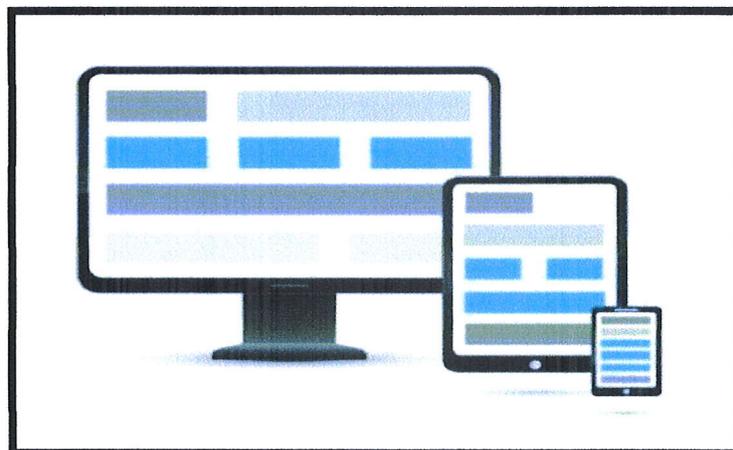


Figure 1.6 : Application Adaptative. [23]

Nous expliquons dans ce qui suit quelques notions relatives au RWD [24].

1.4.1. La grille d'affichage flexible

Pour mettre en œuvre une grille d'affichage flexible il faut éviter les mesures fixes pour décrire la largeur et la hauteur de la zone d'action pour un élément HTML. Si on exprime les mesures, soit en pourcentage (%), soit en mesure "em" (où 1em= 100%) la flexibilité est assurée à la fois pour la taille des polices et pour la taille des éléments HTML. Les valeurs attribuées de cette façon permettent d'avoir une valeur proportionnelle à la taille des caractères utilisés dans la page en fonction du type média.

1.4.2. Le média flexible

Pour un site Web responsive les images, les vidéos et les autres objets médias ne doivent pas déborder le cadre défini. En effet, il est désagréable de voir une image déborder l'écran ou trop petite, écraser et pratiquement invisible.

1.4.3. Les média Queries

Pour rendre un site responsive il est important d'adapter l'affichage en fonction de la résolution et du type de média (*screen, print, tv...*), mais aussi en fonction des caractéristiques du support en prenant en compte le *ratio* et l'*orientation*. Le plus simple est d'appliquer différents styles (CSS) suivant le profil d'utilisation. Le profil peut être détecté avec la fonctionnalité *Media Queries* offerte par CSS3.

HTML5 : est une évolution du langage avec une multitude de nouvelles fonctionnalités et un ensemble de nouvelles balises afin de donner plus de sémantique (de sens) aux pages Web. Cette évolution contribue à la structuration du contenu et avec l'utilisation des feuilles de style (CSS3) on s'approche considérablement à la séparation du fond et de la forme [16].

Exemple :

```

<!doctype html>      <!-- Défini Le type de document -->
<html Lang="fr">     <!-- Début avec Langue par défaut -->
<head>               <!-- Début entête de fichier -->
<meta charset="utf-8"> <!-- Codage par défaut -->
<title>Titre de La page</title> <!-- titre navigateur -->
                    <!-- Lien vers feuille de style -->
<link rel="stylesheet" href="style.css">
<script src="script.js"> </script> <!-- Script à importer -->
</head>              <!-- Fin entête de fichier
<body>               <!-- Début corps de page -->
<header>             <!-- début entête de page -->
<nav></nav>          <!-- Liens de navigation -->
</header>            <!-- Fin entête de page -->
<section>            <!-- Début section -->
<article></article> <!-- Contenu de page -->
<article></article> <!-- Contenu de page -->
<article></article> <!-- Contenu de page -->
</section>           <!-- Fin section -->
<aside></aside>      <!-- Contenu en complément
<footer></footer>    <!-- Pied de page -->
</body>              <!-- Fin corps de page -->
</html>              <!-- Fin document HTML -->

```

Figure 1.7 : Exemple de norme HTML5.

CSS : (*Cascading Style Sheets*) ou feuilles de style contrôlent la mise en page des éléments composant une page Web. Ainsi, on peut définir la couleur et la taille d'une police, le positionnement d'un objet, l'espacement entre les paragraphes, ou gérer des effets de transition et définir des points de rupture, afin de séparer le contenu et la forme. Une feuille de style est applicable à une infinité de documents HTML, ce qui facilite la maintenance et réduit les temps de chargement

Exemple :

```

p {
text-align: justify;
font-style: italic;
color: blue;
font-family: Times;
font-size: 1em;
}

```

Figure 1.8 : Exemple de style CSS.

1.5. Avantages de développement des applications web

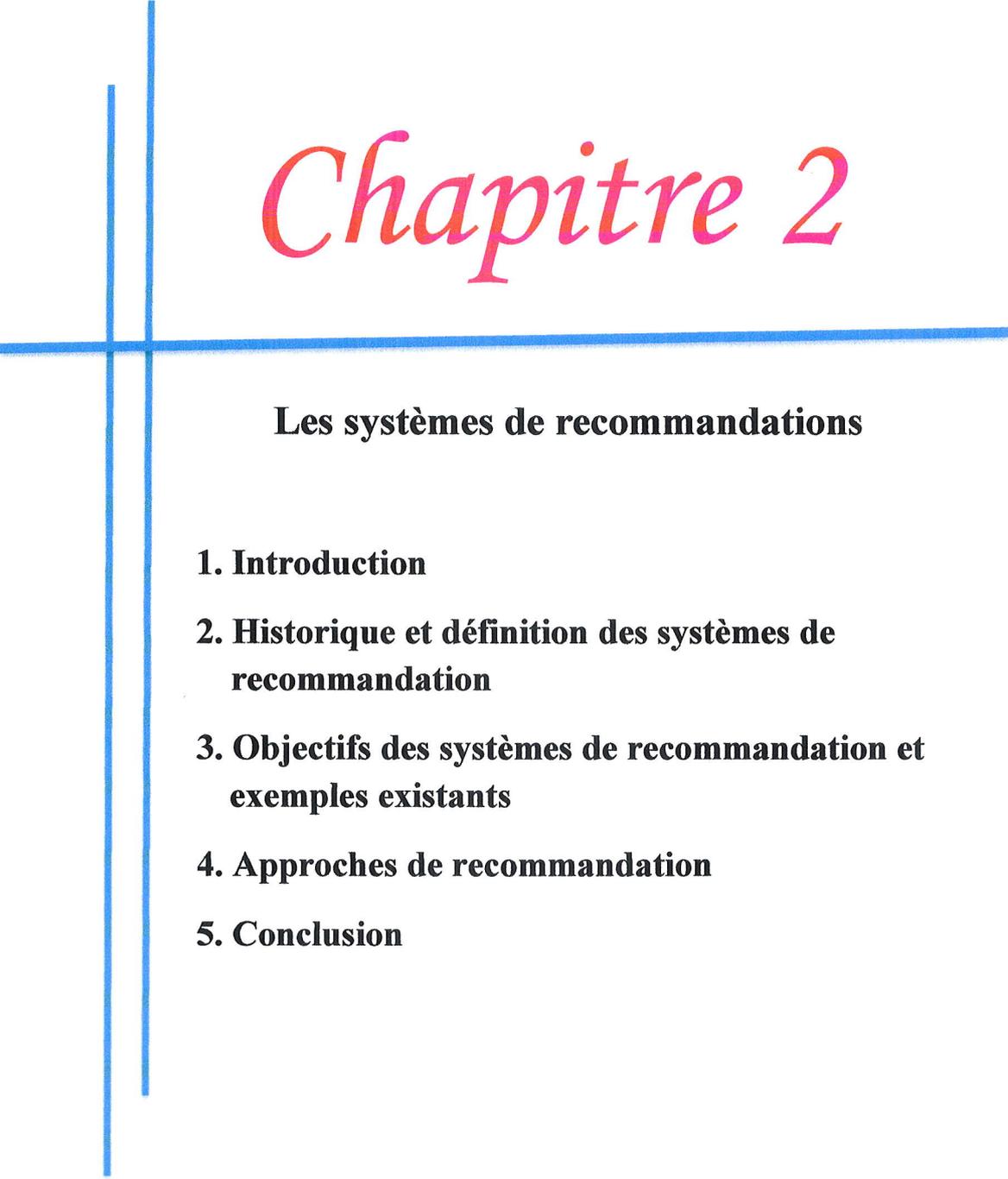
Développer une application web présente les avantages suivants [25] :

- **C'est moins cher** : Un des points les plus intéressants pour les pages en ligne qui choisissent d'opter pour le développement d'application Web est le budget. Cette option est la plus intéressante pour startups, les PME et les sites qui ont tout juste commencé car elle vous permet d'être présent sur les appareils mobiles sans avoir à investir dans une application native.
- **Gain de temps** : Si nous pensons à la façon d'utiliser le PC et les appareils mobiles, nous réalisons quelque chose de très important. La plupart des utilisateurs choisissent d'utiliser une application web à partir de leurs appareils mobiles pour plus de rapidité, et avec la même qualité que depuis le PC.
- **Fonctionne sur tous les systèmes d'exploitation** : Comme ce n'est pas une application native, l'un des avantages de développement d'application Web est que, peu importe le système d'exploitation, elle est conçue de façon s'adapter à l'écran et le « langage » de l'appareil. C'est précisément cette qualité des applications Web qui fait qu'elles vont être exécutées à travers les navigateurs eux-mêmes.
- **Accessibilité optimisée** : La principale raison pour laquelle vous devriez choisir avec un développement d'application Web est l'accessibilité. Elle offre une grande mobilité et un accès facile à partir de n'importe quel appareil mobile, n'importe où et à tout moment avec une simple connexion Internet. La possibilité de ne pas compter sur un seul PC pour accéder aux informations que vous avez stockées dans l'application est également un plus. Un exemple est l'application Evernote (considérée dans une liste comme l'une des meilleures applications mobiles de 2014), avec laquelle vous pouvez prendre des notes et créer des documents à partir de votre PC, à partir d'un Smartphone ou tablette et d'accéder à ces fichiers n'importe où en entrant simplement vos informations d'identification.
- **Meilleure gestion de la sécurité** : Quelque chose à garder à l'esprit est la sécurité dans le développement d'application web. Vu que l'accès au contenu de l'application se fait via Internet, il y a toujours la possibilité que nos contenus soient exposés. Par conséquent, nous devons inclure les protocoles nécessaires pour sécuriser nos informations, les développeurs et les utilisateurs lors du choix de leur mot de passe.

1.6. Conclusion

Dans ce chapitre, nous avons présenté plusieurs notions relatives au développement des applications web mais aussi au RWD (Responsive Web Design). Par ses multiples avantages, le langage PHP est choisi pour l'implémentation de notre application web adaptative.

Dans le chapitre suivant, nous décrirons les systèmes de recommandation qui sont actuellement très utilisés au sein des applications web.



Chapitre 2

Les systèmes de recommandations

- 1. Introduction**
- 2. Historique et définition des systèmes de recommandation**
- 3. Objectifs des systèmes de recommandation et exemples existants**
- 4. Approches de recommandation**
- 5. Conclusion**

2.1. Introduction

Les systèmes de recommandations sont devenus une méthodologie dominante dans la plupart des applications Web. Avec la très grande masse d'information aujourd'hui disponible sur l'Internet, il est devenu primordial de concevoir des mécanismes qui permettent aux utilisateurs d'accéder à ce qui les intéresse le plus rapidement possible. Ce mécanisme est dit système de recommandation.

Un système de recommandation (SR) doit mettre en relation deux entités : des utilisateurs et des items. Les items peuvent être de diverses natures (livre, films, vidéos, restaurants, lieux d'activité...). Les informations qui permettent de relier ces deux entités sont elles-mêmes de natures différentes : notes, achats, clics, historiques, etc. Les systèmes de recommandation se focalisent majoritairement sur l'utilisation des "notes". La note est généralement située sur une échelle graduée, et elle permet à l'utilisateur d'exprimer un avis positif ou négatif sur l'item qu'il considère.

Dans ce chapitre nous allons voir : les SRs , leurs objectifs, les différentes approches de recommandation avec des exemples de SR.

2.2. Historique et définition des systèmes de recommandation

Les premiers systèmes de recommandation se réduisent aux systèmes de filtrage collaboratif. Ils remontent au début des années 1990s, cette période à laquelle ils sont reconnus comme étant un domaine de recherche indépendant.

Les racines des systèmes de recommandation remontent aux travaux étendus dans les sciences cognitives, la théorie d'approximation, la recherche documentaire, la théorie de la prévoyance et ont également des liens avec la science de la gestion et le marketing, dans la modélisation des choix du consommateur [26].

Intuitivement, un système de recommandation, par exemple de livres, peut être comparé à un agent de bibliothèque à qui on demande de proposer des ouvrages après lui avoir exprimé nos besoins. En effet, le bibliothécaire avec sa vaste connaissance des ouvrages de cette bibliothèque comparativement à un simple usager et compte tenu de sa longue expérience avec les usagers ainsi qu'un feedback en provenance de ces derniers peut prodiguer de précieuses recommandations qui épargneront un dur effort de recherche et beaucoup de temps en conséquence. Bien sûr, l'usager doit lui fournir suffisamment de données. Cet exemple illustre un recommander humain, afin de rapprocher l'image à l'échelle de la machine, le bibliothécaire virtuel doit avoir trois sortes d'informations [26] :

- La représentation de la bibliothèque (profil des items)

- La représentation des usagers (profil des usagers)
- Les algorithmes qui interfèrent entre les deux pour produire les recommandations En d'autres termes, un système de recommandations aide les utilisateurs à faire leurs choix dans un domaine où ils disposent de peu d'informations pour trier et évaluer les alternatives possibles L'importance de tels systèmes apparaît clairement dans les environnements où de gigantesques quantités d'informations en ligne surpassent de loin les capacités de recherche de n'importe quel être humain. Actuellement, les systèmes de recommandation font partie intégrante de certains sites de commerce électronique tels qu'Amazon1 et CDNow2.

2.3. Objectifs des systèmes de recommandation et exemples existants

Un système de recommandation a pour objectif de fournir à un utilisateur des ressources pertinentes en fonction de ses préférences. Ce dernier voit ainsi réduit son temps de recherche mais reçoit également des suggestions de la part du système auxquelles il n'aurait pas spontanément prêtées attention. L'essor du Web et sa popularité ont notamment contribué à la mise en place de tels systèmes comme dans le domaine du e-commerce. Citons par exemple les sites Web populaires Amazon dans le e-commerce mais également CiteSeer2, outil référençant des articles de recherche. Les systèmes de recommandation peuvent être vus initialement comme une réponse donnée aux utilisateurs ayant des difficultés à prendre une décision dans le cadre d'utilisation d'un système de recherche d'information classique [27].

Voici quelques exemples de recommandation [28] :

- **Recommandation de produits :**

- **Amazon** est une entreprise de commerce en ligne américaine. Elle compte plus de 183 millions produits référencés. En France, Amazon compte 16,8 millions d'utilisateurs uniques par mois.
- **Cdiscount** est l'entreprise leader de commerce en ligne française. Actuellement, Cdiscount est le deuxième site de commerce en ligne le plus visité en France, talonnant le géant américain Amazon.

- **Recommandation d'actualités :**

- **Google actualités** recense des milliers de sources d'actualités. Il est considéré comme un agrégateur d'informations externes, et doit donc être capable de faire le lien entre plusieurs actualités abordant le même sujet mais dont les sources sont distinctes.
- **Yahoo ! News** est un service gratuit présentant aux utilisateurs des actualités issues d'autres sources d'actualités, telles qu'ABC News ou Fox News, il s'agit également d'un agrégateur d'informations externes. Début 2016, Yahoo ! met à disposition de la communauté

scientifique un jeu de données de 13,5 Terabytes, ce qui représente 110 billions d'interactions entre les utilisateurs et les actualités. Il s'agit du jeu de données d'apprentissage le plus volumineux rendu public à ce jour.

- **Recommandation de titres musicaux :**

- **Spotify** est un logiciel suédois de musique en ligne (streaming). En juin 2015, il compte plus de 30 millions de titres pour 75 millions d'utilisateurs actifs.
- **Deezer** est un site web français d'écoute de musique à la demande, disposant de plus de 40 millions de titres pour 16 millions d'utilisateurs actifs.

- **Recommandation de films ou séries :**

- **Netflix** est un service américain de vidéos à la demande comptant plus de 75 millions d'utilisateurs. En 2006, Netflix lance un challenge avec un million de dollars de récompense pour l'équipe qui obtiendra une amélioration de 10% des prédictions de notes effectuées pour un film par l'algorithme utilisé par l'entreprise. Pour cela Netflix met à disposition 100 millions de notes, représentant 480 000 utilisateurs pour 17 000 films environ. Ce n'est qu'en 2009 qu'une équipe nommée "BellKor's Pragmatic Chaos" arrive à atteindre cet objectif.
- **MovieLens** est un SR de films non-commercial créé en 1997 par l'université du Minnesota. Plusieurs jeux de données de référence en recommandation ont été fournis par MovieLens.

2.4. Approches de recommandation

Il existe une large variété d'approches de recommandation. A travers les travaux de recherche, différentes tentatives de classification des approches ou des techniques ont été réalisées. La classification de ces approches dépend notamment du type de données exploitées et de la méthode d'apprentissage utilisée par le système de recommandation.

Dans cette section, nous présentons les principales techniques de recommandation avec leurs apports et leurs limites.

2.4.1. Recommandation basé sur le contenu

Pour la recommandation basée sur le contenu, le system essaie de recommander des éléments qui correspondent au profil de l'utilisateur. De manière générale, le profil est basé sur les éléments que l'utilisateur a aimés dans le passé ou sur le contenu fait coïncider les caractéristiques de l'élément avec le profil de l'utilisateur pour décider de sa pertinence pour l'utilisateur. La tâche de recommandation consiste donc à déterminer quels éléments coïncident le mieux avec les préférences de l'utilisateur. Par exemple des films a un utilisateur, le système analyse les corrélations entre ces films et les films consultés antérieurement par cet utilisateur.

Ces corrélations sont évaluées en considérant des attributs comme le titre et le genre. De ce fait, parmi ces films, ceux qui seront recommandés à l'utilisateur, sont les plus similaires (en termes d'attribut) aux films consultés par cet utilisateur [29].

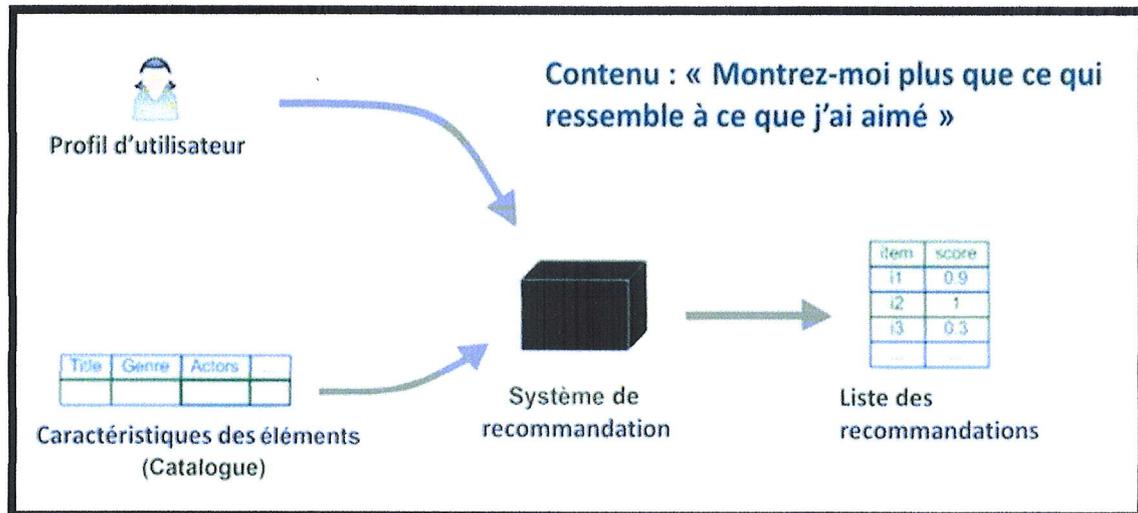


Figure 2.1 : Un système de recommandation basé sur le contenu. [30]

Dans ce qui suit, nous discutons la représentation des objets dans un système de recommandation basé sur contenu et nous présenterons deux techniques : le modèle de l'espace vectoriel et la classification bayésienne.

2.4.1.1. Le modèle de l'espace vectoriel

La plupart des systèmes de recommandation basés sur le contenu utilisent le modèle de représentation vectoriel VSM (Vector Space Model) avec la pondération classique TF-IDF (Term Frequency-Inverse Document Frequency). Dans ce modèle, chaque document (qui représente un item) est représenté par un vecteur de dimension n , tout document est représenté par un vecteur poids sur des termes, où chaque poids indique le degré d'association entre le document et le terme [31].

Soit $D = \{d_1, d_2, \dots, d_n\}$ dénotant un ensemble de documents ou corpus ; Soit $T = \{t_1, t_2, \dots, t_n\}$ le dictionnaire, c'est-à-dire l'ensemble des mots du corpus. T est généralement obtenu en appliquant des opérations de traitement du langage naturel. Chaque document d_j est représenté par un vecteur dans un espace vectoriel à n dimensions, tel que $d_j = \{w_{1j}, w_{2j}, \dots, w_{nj}\}$, où w_{kj} est le poids du terme t_k dans le document d_j .

La représentation de documents en utilisant le modèle d'espace vectoriel fait apparaître deux difficultés : la pondération des termes et la mesure de similarité des vecteurs représentant les documents. La méthode de pondération de termes la plus couramment utilisée est la pondération *TF-IDF*, qui est basée sur des observations empiriques sur le texte [32].

- La quantité TF représente le nombre d'occurrences d'un terme dans un document.

$$TF(t_k, d_j) = \frac{f_{k,j}}{\max_z f_{z,j}} \in [0,1] \quad (2.1)$$

Où $f_{k,j}$ représente le nombre d'occurrences du terme t_k dans le document d_j , et $\max_z f_{z,j}$ est le maximum des fréquences $f_{z,j}$ des termes t_z apparaissant dans le document d_j .

- La quantité IDF exprime l'importance d'un terme au sein du corpus considéré, telle que :

$$IDF(t) = \log\left(\frac{N}{n_k}\right) \quad (2.2)$$

Où :

N Dénote le nombre de documents dans le corpus, et n_k représente le nombre de documents de la collection dans lesquels le terme t_k apparaît au moins une fois.

La pondération TF.IDF d'un terme t_k pour un objet d_j est alors exprimée comme le produit des deux quantités précédentes :

$$TFIDF(t_k, d_j) = TF(t_k, d_j) \times \log\left(\frac{N}{n_k}\right) \quad (2.3)$$

Afin que tous les poids appartiennent à l'intervalle $[0;1]$, et que tous les documents soient représentés par des vecteurs de même longueur, les poids obtenus par la fonction TFIDF sont généralement normalisés en utilisant la normalisation cosinus :

$$w_{k,j} = \frac{TFIDF(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} TFIDF(t_s, d_j)^2}} \quad (2.4)$$

Une fois que les poids sont calculés et normalisés. Le contenu d'un item d_j est défini par :

$$content(d_j) = (w_{1j}, w_{2j}, \dots, w_{kj}) \quad (2.5)$$

Après cette étape de pondération des termes et de normalisation, il faut définir une mesure de similarité des vecteurs caractéristiques. Cette mesure de similarité est requise pour déterminer la proximité entre deux documents. Il existe de nombreuses mesures de similarité, mais la mesure la plus largement utilisée dans la littérature est la similarité cosinus :

$$\text{sim}(d_i, d_j) = \frac{\sum_k w_{ki} \cdot w_{kj}}{\sqrt{\sum_k w_{ki}^2} \times \sqrt{\sum_k w_{kj}^2}} \quad (2.5)$$

2.4.1.2. La classification bayésienne

Le filtrage est traité comme un problème de classification, dont l'objectif est de calculer la probabilité qu'un objet appartienne à une des deux catégories : les objets pertinents ou les objets non pertinents [33]. Cette approche emploie le théorème de Bayes qui stipule: étant donné un élément d_j , la probabilité qu'il appartienne à une classe e peut être calculée comme suit:

$$p(e|d_j) = \frac{p(e)p(d_j|e)}{p(d_j)} \quad (2.6)$$

Où

$P(d_j | e)$ = la probabilité que l'objet d_j , fasse partie de la classe e , $P(e)$ la probabilité qu'un objet appartienne à la classe e , $P(d)$ = la probabilité que le document soit choisi.

Hypothèse d'indépendance

Supposons que chaque objet d_j peut être représenté par un ensemble de termes :

$$d_j = \{t_{ki} | k = 1, 2, \dots, r\} \quad (2.7)$$

L'hypothèse d'indépendance suppose que la présence d'un terme dans un document est indépendante par rapport aux autres termes [22]. En conséquence :

$$p(d_j|e) = \prod_{k=1}^r p(t_{kj}|e) \quad (2.8)$$

Finalement, nous calculons le rapport $\frac{p(\text{positif}|d_j)}{p(\text{négatif}|d_j)}$. Plus il est élevé pour un document, plus ce document est pertinent pour l'utilisateur.

2.4.1.3. Avantages [34]

- La connaissance du domaine n'est pas requise, le processus de recommandation se base uniquement sur les évaluations des items.
- La recommandation devient plus précise au fur et à mesure que le système est utilisé plus par l'utilisateur.
- Une liste de recommandations peut être générée même s'il n'y a qu'un seul utilisateur.

2.4.1.4. Limites [35]

- Nécessité du profil d'utilisateur

- Filtrage basé sur le critère thématique uniquement, absence d'autres facteurs comme la qualité scientifique, le public visé, l'intérêt porté par l'utilisateur.
- Les difficultés à recommander des documents multimédia (images, vidéos, etc.) et ceci à cause de la difficulté à indexer ce type de documents, c'est en fait la même problématique dont souffrent les systèmes de recherche.
- Problème de démarrage à froid, un nouvel utilisateur du système éprouve des difficultés à exprimer son profil en spécifiant des thèmes qui l'intéressent. Ceci malgré les techniques d'apprentissage ou l'utilisateur fournit des textes exemples.

2.4.2. Recommandation basé sur le Filtrage collaboratif

Le filtrage collaboratif est une approche basée sur le partage d'opinions entre les utilisateurs, il reprend le principe du "bouche à oreille" pratiqué depuis toujours par les humains pour se construire une opinion sur un produit ou un service qu'ils ne connaissent pas, l'hypothèse fondamentale de cette méthode est que les opinions des autres utilisateurs peuvent être utilisées pour fournir une prédiction raisonnable de la préférence de l'utilisateur actif sur un item qu'il n'a pas encore noté. Ces méthodes supposent que si des utilisateurs ont les mêmes préférences sur un ensemble d'items, alors ils auront probablement les mêmes préférences sur un autre ensemble d'items qu'ils n'ont pas encore notés [36].

Plus précisément, le filtrage collaboratif utilise une matrice dont les lignes correspondent aux utilisateurs et les colonnes aux ressources. Chaque cellule de la matrice correspond alors à une note fournie par l'utilisateur correspondant pour la ressource correspondante. Le but est alors de prédire les notes que les utilisateurs attribueraient aux ressources pour lesquelles ils n'ont pas encore fourni de note, pour ensuite recommander les ressources ayant les meilleures notes prédites [37].

Le filtrage collaboratif est en général classé en deux approches : l'approche mémoire et l'approche modèle.

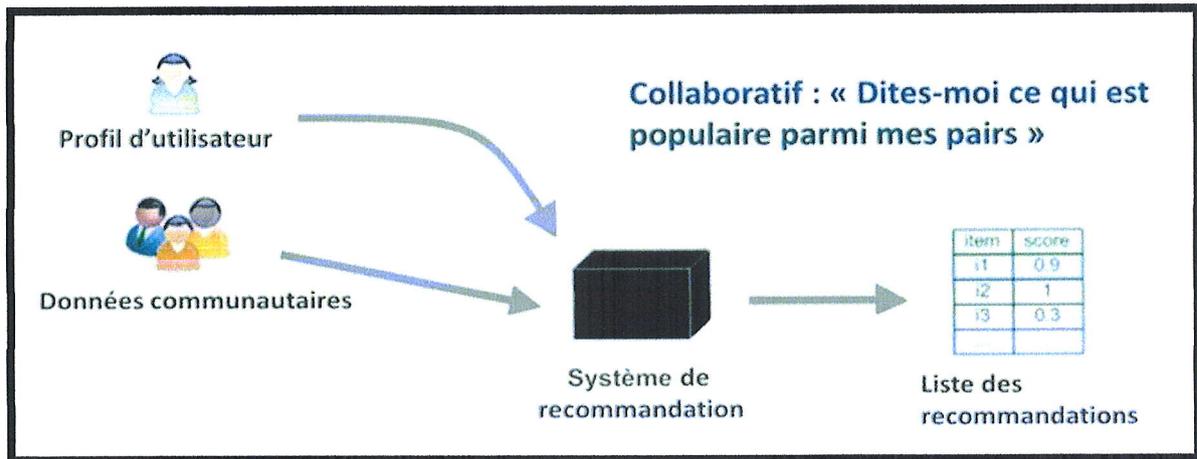


Figure 2.2 : Un système de recommandation collaboratif. [30]

2.4.2.1. La méthode basée de mémoire

Le filtrage collaboratif à base de mémoire ou heuristique considère la totalité des évaluations des usagers disponibles au moment du calcul de la recommandation. Le processus de calcul de la recommandation pour un usager U_i passe par deux étapes successives :

❖ Phase du calcul du voisinage:

En se basant sur le profil de cet usager U_i , le système recherche les usagers U_j (j diffère de i) qui lui sont les plus similaires. Deux mesures de similarité qui sont très utilisées sont : la similarité vectorielle et la corrélation de Pearson [38].

La similarité vectorielle :

Dans cette méthode les usagers A et B sont considérés comme deux vecteurs de même origine dans un espace de m dimensions, m est égale au nombre d'items évalués par les deux usagers. Plus deux usagers sont similaires, plus l'angle entre leur vecteur est plus petit. Empiriquement, la similarité entre ces deux usagers est calculée par la formule du *Cosinus* suivante :

$$\text{cosinus}(A, B) = \sum_{j=1}^{l1} \frac{v_{A,j}}{\sqrt{\sum_{j=1}^{l1} v_{A,j}^2}} \times \frac{v_{B,j}}{\sqrt{\sum_{j=1}^{l1} v_{B,j}^2}} \quad (2.9)$$

$l1$: Nombre d'items communs entre A et B votés par v.

$v_{A,j}$: Vote de A pour l'item j.

$v_{B,j}$: Vote de B pour l'item j.

La corrélation de Pearson :

La corrélation de Pearson telle qu'utilisée par [39] est une méthode issue des statistiques. Elle est aussi très utilisée dans le domaine des systèmes de recommandation pour mesurer la



similarité entre deux usagers. La formule ci-dessous, nous donne cette valeur pour deux usagers A et B :

$$w(A, B) = \frac{\sum_j (v_{A,j} - \bar{v}_A)(v_{B,j} - \bar{v}_B)}{\sqrt{\sum_j (v_{A,j} - \bar{v}_A)^2 \sum_j (v_{B,j} - \bar{v}_B)^2}} \quad (2.10)$$

j : Nombre d'objets ayant été voté à la fois par A et B .

$v_{A,j}$: Vote de A pour l'item j .

\bar{v}_A : Moyenne des votes de A .

❖ **Phase de prédiction :**

Une fois que toutes les similarités de l'utilisateur cible A par rapport aux autres usagers sont calculées et que les n usagers les plus similaires qui constituent le voisinage de cet usager cible sont définis, la prédiction de la valeur d'un item j évaluée par l'utilisateur A est calculée à l'aide de la formule suivante :

$$p_{A,j} = \bar{v}_A + \frac{\sum_{i=1}^n w(A,i)(v_{i,j} - \bar{v}_i)}{\sum_{i=1}^n |w(A,i)|} \quad (2.11)$$

n : nombre d'utilisateurs présents dans le voisinage de A , ayant déjà voté sur l'objet j .

$v_{i,j}$: vote de l'utilisateur i pour l'objet j .

\bar{v}_i : Moyenne des vote de l'utilisateur i .

2.4.2.2. La méthode basée de modèle

Les algorithmes basés sur le modèle se basent aussi sur les évaluations précédentes (Les profils) des usagers, sauf que cette fois-ci, on ne calcule pas directement les prédictions, mais on essaie de classer les usagers suivant des groupes ou d'apprendre les modèles à partir de leurs données. Une fois les groupes ou les modèles d'usagers sont trouvés, la prédiction pour un usager donné est générée automatiquement à partir de son profil. Pour la construction du modèle plusieurs méthodes sont utilisées, par exemple deux méthodes statistiques pour construire les modèles, la première est très utilisée dans le domaine de l'apprentissage machine : Les modèles de classification (en Anglais Cluster models). La deuxième méthode est celle des réseaux Bayésiens. En général, les méthodes basées sur le modèle utilisent les techniques d'apprentissage machine et les techniques statistiques pour apprendre le modèle à partir des profils des usagers [40].

2.4.2.3. Avantages [41]

- Tous les utilisateurs du système de filtrage collaboratif peuvent tirer profit des évaluations des autres en recevant des recommandations pour lesquelles les utilisateurs les plus proches ont émis un jugement de valeur favorable. et cela sans que le système dispose d'un processus d'extraction du contenu des documents. Grâce à son indépendance vis-à-vis de la représentation des données, cette technique peut s'appliquer dans les contextes où le contenu est soit indisponible 1 soit difficile à analyser, et en particulier elle peut s'utiliser pour tout type de données : texte, image, audio et vidéo.
- l'utilisateur est capable de découvrir divers domaines intéressants, car le principe du filtrage collaboratif ne se fonde absolument pas sur la dimension thématique des profils, et n'est pas soumis à l'effet « entonnoir ».
- les jugements de valeur des utilisateurs intègrent non seulement la dimension thématique mais aussi d'autres facteurs relatifs à la qualité des documents tels que la diversité, la nouveauté, etc.

2.4.2.4. Limites

Un problème du système collaboratif est que sa performance dépend beaucoup de la distribution des évaluations (notes) données par utilisateurs. Dans le cas où il y a plusieurs items qui ont été utilisés et évalués par très peu d'utilisateurs, ces items seraient recommandés très rarement, même si ces utilisateurs ont donné des notes très hautes pour ces items. Ce problème est connu comme le problème de parcimonie (sparsity problem). De la même façon, si dans le système il existe des utilisateurs qui ont des goûts très différents en comparaison avec les autres, le système ne peut pas trouver des similarités entre utilisateurs et donc ne peut pas donner des bonnes recommandations [42].

2.4.3. Le filtrage démographique

Ce type de filtrage utilise les données d'utilisateur démographiques (le sexe, l'âge, la profession, la localisation, la langue, le pays, etc.) pour lui générer des recommandations.

Le principe de cette approche est que deux utilisateurs ayant évolué dans un environnement similaire partagent des goûts communs que deux utilisateurs ayant évolué dans des environnements différents et ne partageant donc pas les mêmes codes [43].

Par exemple, un groupe des utilisateurs contient quatre femme (même sexe), trois femmes de cet groupe sont partagés un livre de cuisine, par le filtrage démographique il recommander ce livre a la quatrième femme de groupe.

2.4.4. Le filtrage basé utilité

Ce système ne construit pas une généralisation à long terme pour ses utilisateurs, mais il base sa recommandation sur une évaluation de la correspondance entre les besoins des utilisateurs et l'ensemble des items disponibles. Dans ce type de filtrage, l'utilité de chaque item pour l'utilisateur est calculée afin de recommander les items les plus utiles. Dans ce cas, le profil de l'utilisateur est la fonction d'utilité que le système dérive pour lui, et le système emploie des techniques de contraintes de satisfaction afin de localiser la meilleure correspondance [44].

Certaines limites sont associées à ce type de filtrage du fait que l'utilisateur doit introduire une fonction d'utilité au système. De plus, ces systèmes n'apprennent pas les préférences des utilisateurs ce qui conduit à une capacité de suggestion statique [44].

2.4.5. Le filtrage communautaire

La recommandation communautaire ou sociale, vu que la plupart des réseaux sociaux (Facebook, Twitter, etc.) s'appuient sur cette classification dans leurs recommandations.

L'idée de base est donc de dire que si des utilisateurs ont partagés des mêmes intérêts dans le passé, il y a de fortes chances qu'ils partagent aussi les mêmes goûts dans la future.

Le système propose des recommandations à partir des relations de l'usager avec ces amis dans le réseau social, et parfois cette recommandation dépend aussi de la valeur de confiance d'usager dans chacun de ses amis, l'exemple le plus connu de cette recommandation est la section des pages et des groupes qui apparait dans la partie droite d'une page Facebook. L'importance décisionnelle du bouton « I Like » de Facebook a donné un succès croissant dont 55% des utilisateurs sont influencés par leurs amis [32].

2.4.6. Recommandation basé sur le Filtrage hybride

Un système de recommandation est dit hybride quand il combine deux ou plusieurs approches de recommandation différentes. La recommandation basée sur le contenu et la recommandation collaborative ont souvent été considérées comme complémentaires [45].

Les approches basées sur le contenu ont l'avantage de pouvoir recommander les nouveaux items non encore évalués par un utilisateur, alors que le filtrage collaboratif ne peut recommander un item que s'il a été noté par un certain nombre d'utilisateurs auparavant. Les approches basées sur le contenu nécessitent de disposer des attributs des items, en plus d'une étape d'analyse pour pouvoir les extraire et les représenter, alors que le filtrage collaboratif ne requiert pas d'accès au contenu des items pour pouvoir faire de la recommandation. Il s'appuie uniquement sur la matrice des notes d'utilisateurs pour les différents items. L'hybridation de ces deux techniques, afin de traiter les insuffisances de chaque technique utilisée seule et profiter de leurs points forts, a fait l'objet de plusieurs travaux de recherche.

Le système FAB [46] est un des premiers systèmes de recommandation hybrides. Il combine le filtrage collaboratif et une approche basée sur le contenu afin de traiter à la fois le problème du démarrage à froid pour les items et la sur-spécialisation. Dans ce système, deux critères doivent être satisfaits pour recommander un item : son contenu doit être similaire au profil de l'utilisateur, et il doit être apprécié par les voisins les plus proches.

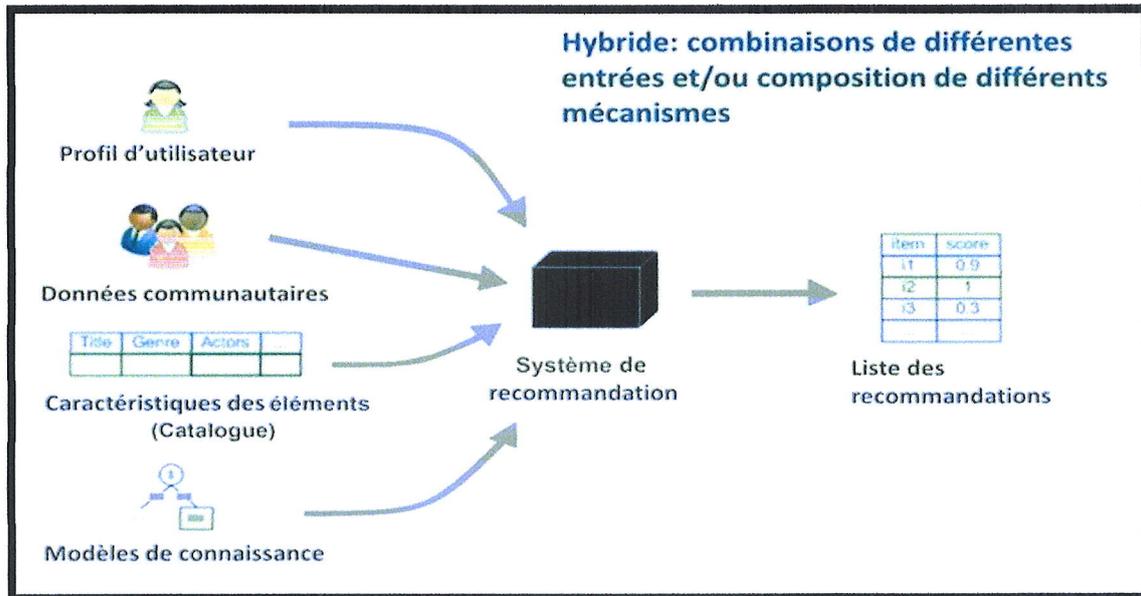


Figure 2.3 : Le système de recommandation hybride. [30]

Il existe plusieurs manières de faire de l'hybridation et aucun consensus n'a été défini par la communauté des chercheurs. Toutefois, Burke [47] a identifié sept manières différentes de faire l'hybridation :

- **Pondération**

Une méthode hybride qui combine la sortie d'approches distinctes, utilisant, par exemple, une combinaison linéaire des scores de chaque technique de recommandation. Cette hybridation est décrite dans la figure 2.4.

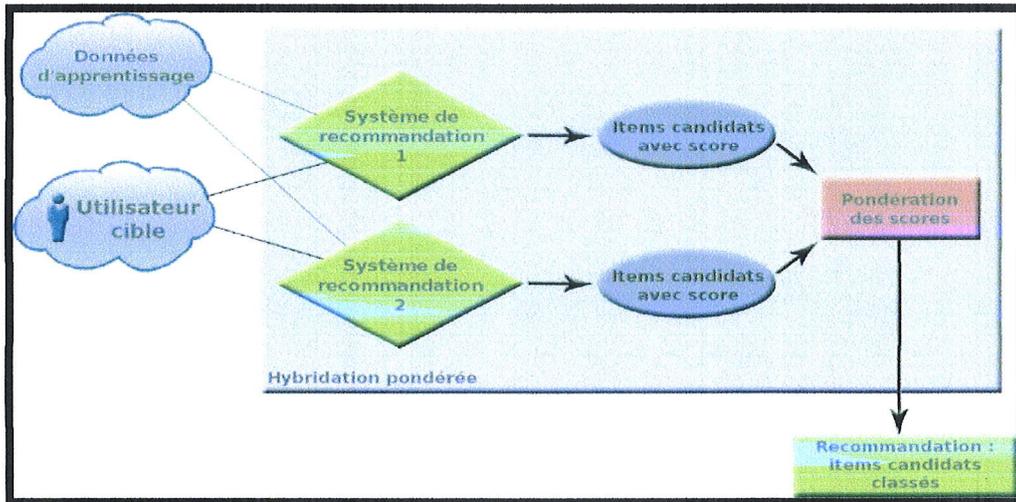


Figure 2.4 : Hybridation pondérée. [48]

- **Bascule**

Dans ce type d'hybridation, le système sélectionne une approche de recommandation plutôt qu'une autre en fonction d'un certain critère. Cette hybridation est décrite dans la figure 2.5.

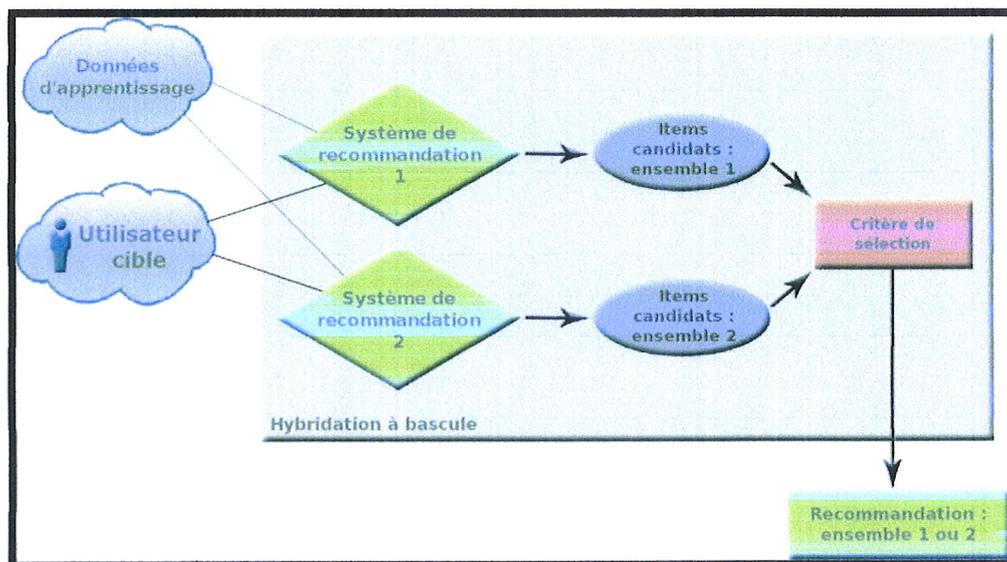


Figure 2.5 : Hybridation à bascule. [48]

- **Mixte**

Dans cette approche, le recommandeur ne combine pas, mais augmente la description des ensembles de données, en prenant en considération les estimations des utilisateurs et la description des items. La nouvelle fonction de prédiction doit faire face aux deux types de descriptions et permet d'éviter les problèmes posés par le filtrage collaboratif, à savoir, le démarrage à froid. Cette hybridation est décrite dans la figure 2.6.

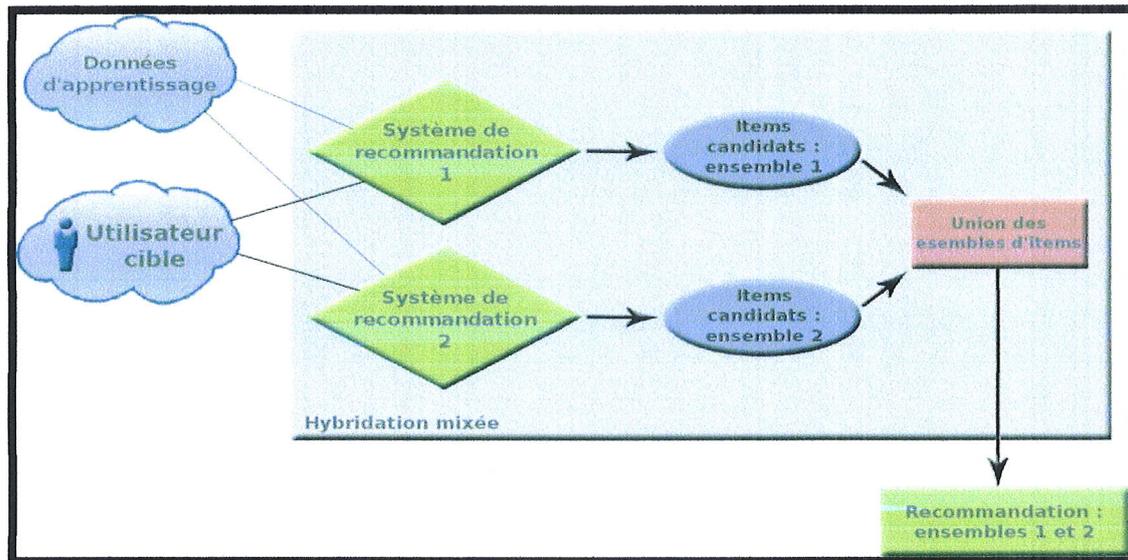


Figure 2.6 : Hybridation mixte. [48]

- **Combinaison de caractéristiques**

Dans une hybridation basée sur la combinaison de caractéristiques, les données provenant de techniques collaboratives sont traitées comme une caractéristique, et une approche basée sur le contenu est utilisée sur ces données. Cette hybridation est décrite dans la figure 2.7.

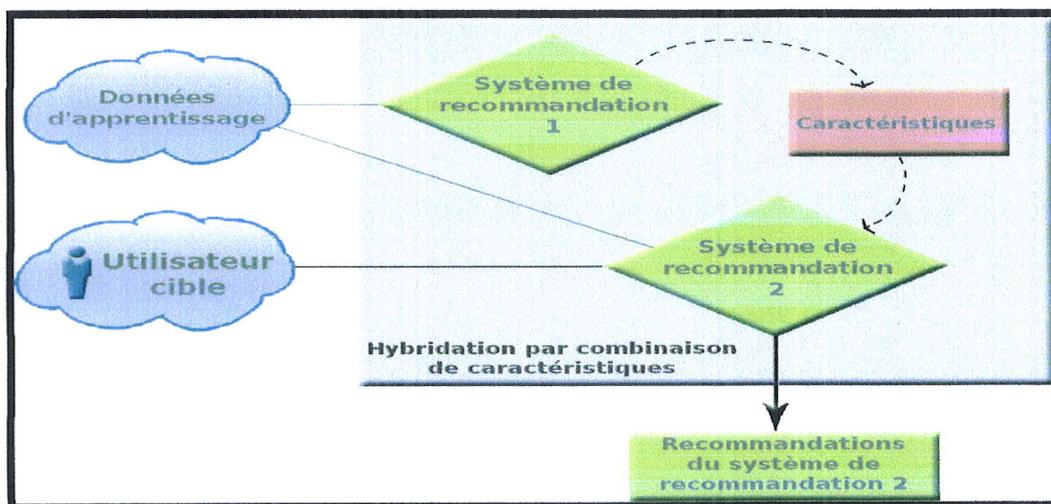


Figure 2.7 : Hybridation par combinaison de caractéristiques. [48]

- **Cascade**

La cascade implique un processus étape par étape. Dans ce cas, une technique de recommandation est appliquée en premier, produisant un ensemble de candidats potentiels. Puis, une deuxième technique affine les résultats obtenus dans la première étape. Cette méthode a pour avantage que si la première technique génère peu de recommandations, ou si ces

recommandations sont ordonnées afin de permettre une sélection rapide, la deuxième technique ne sera plus utilisée. Cette hybridation est décrite dans la figure 2.8.

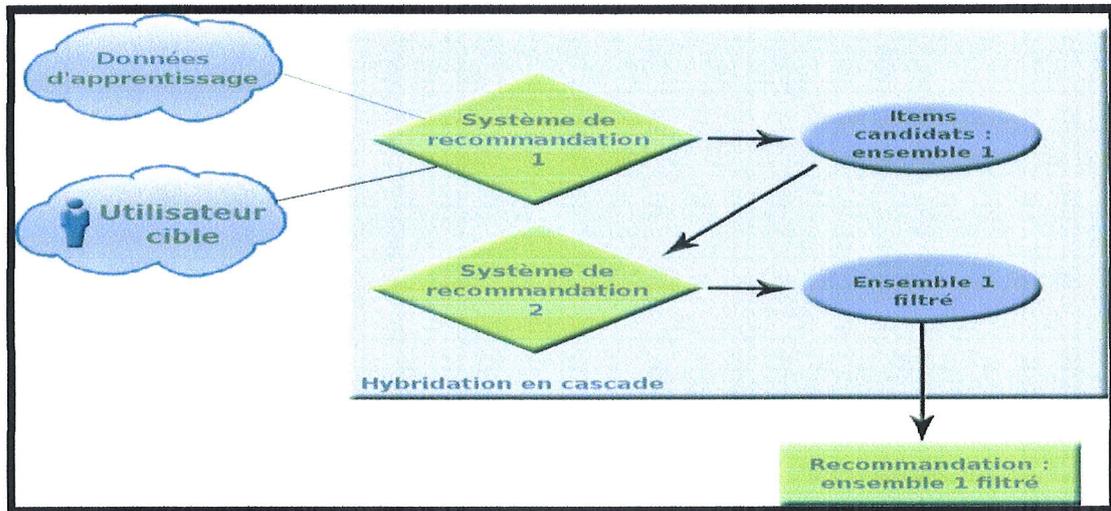


Figure 2.8 : Hybridation en cascade. [48]

- **Augmentation de caractéristiques**

L'augmentation de caractéristiques est semblable à la cascade, mais dans ce cas-là les résultats obtenus (le classement ou la classification) de la première technique sont utilisés par le deuxième comme une caractéristique ajoutée. Cette hybridation est décrite dans la figure 2.9.

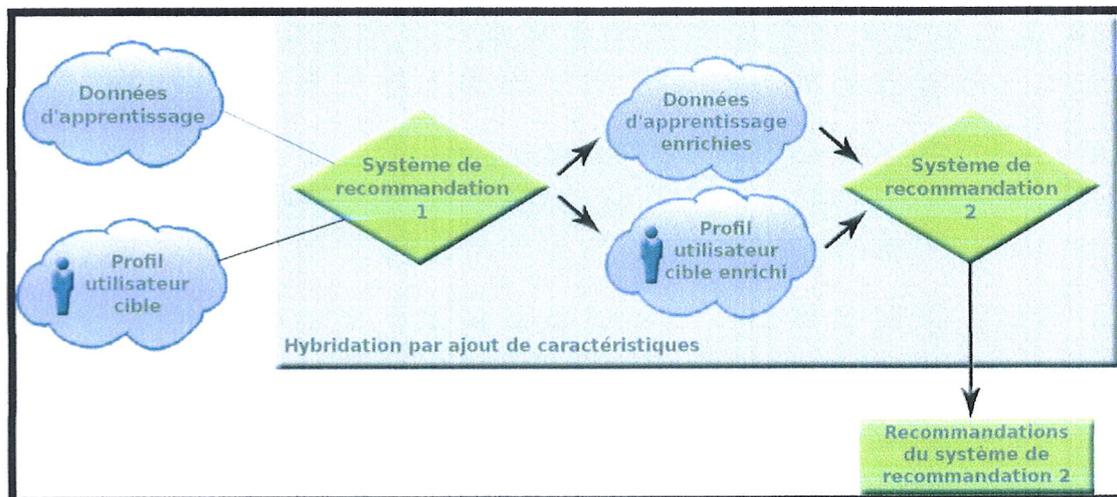


Figure 2.9 : Hybridation en augmentation de caractéristiques. [48]

- **Méta niveau**

Cette méthode est analogue à la méthode par augmentation de propriétés mais c'est le modèle appris qui est utilisé en entrée de la deuxième technique et non la liste résultat des recommandations. Cette hybridation est décrite dans la figure 2.10.

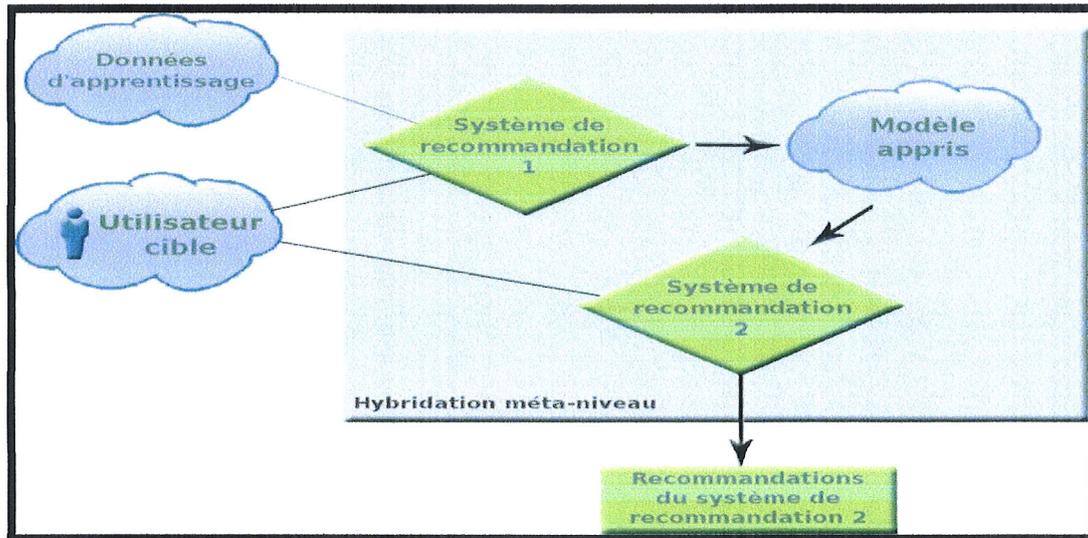


Figure 2.10 : Hybridation méta-niveau. [48]

2.5. Conclusion

Dans ce chapitre, nous avons passé en revue les différentes classes de systèmes de recommandations : les systèmes basés sur le filtrage collaboratif, les systèmes basés sur le filtrage à base de contenu et les systèmes basés sur le filtrage hybride, etc.

Les futurs systèmes de recommandation seront plutôt axés sur des méthodes multicritères, multidimensionnelles ou encore seront fondés sur des notions psychologiques comme les émotions et les opinions [27].

Pour notre système de recommandation de livres, nous allons utiliser deux approches de recommandation, à savoir : celle basée sur le contenu et celle basée sur le filtrage collaboratif. Le chapitre suivant décrit en détail la conception de notre SR.

Chapitre 3

Conception

- 1. Introduction**
- 2. Modélisation UML**
- 3. Etude préliminaire**
- 4. Capture des besoins**
- 5. Les cas d'utilisation**
- 6. Les diagrammes**
- 7. Les algorithmes et les méthodes de recommandation utilisée pour gérer l'application**
- 8. Classement des livres**
- 9. Conclusion**

3.1. Introduction

La conception est une étape préliminaire et primordiale qui doit précéder l'étape de développement de toute application informatique. Ce chapitre est organisé en deux grandes parties. D'abord, nous modélisons en utilisant le langage UML le coté système information dans notre application Web UBook (University Book). Ensuite, nous expliquons la démarche de recommandation des livres proposée.

3.2. Modélisations UML

UML est l'acronyme anglais pour «Unified Modeling Language». On le traduit par « Langage de modélisation unifié ». La notation UML est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du logiciel à développer [38].

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. Il offre un standard de modélisation, pour représenter l'architecture de logicielle. Les différents éléments représentables sont « Activité d'un objet/logiciel, Acteurs, Processus, Schéma de base de données, Composants logiciels et Réutilisation de composants ». Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie de code, par exemple en langage Java, à partir des divers documents réalisés [49].

Première partie : conception du système d'information

3.3. Etude préliminaire

3.3.1. Identification des acteurs

Un acteur est un utilisateur type du système qui représente une responsabilité par rapport au système ou un rôle plutôt qu'une personne physique. Un acteur peut être concrétisé par une personne, un système ou une machine. Les acteurs que nous avons réussi à identifier sont :

- **Le système de recommandation** : Le rôle principal du système est de fournir une recommandation des livres aux les utilisateurs de l'application.
- **L'administrateur du système** : le rôle principal de l'administrateur est l'administration de l'application web et de la base de données (ajouter, supprimer ou modifier) des livres, des utilisateurs et des spécialités.
- **L'utilisateur de l'application** : l'utilisateur peut être un étudiant ou un enseignant. En naviguant dans l'application, il peut consulter ou emprunter des livres de la bibliothèque.
- **Anonyme** : le rôle principal de l'anonyme est de consulter la liste et les détails des livres dans l'application.

3.3.2. Identification des messages

Un message est une information envoyée par un émetteur, dans le but de déclencher une activité chez le récepteur.

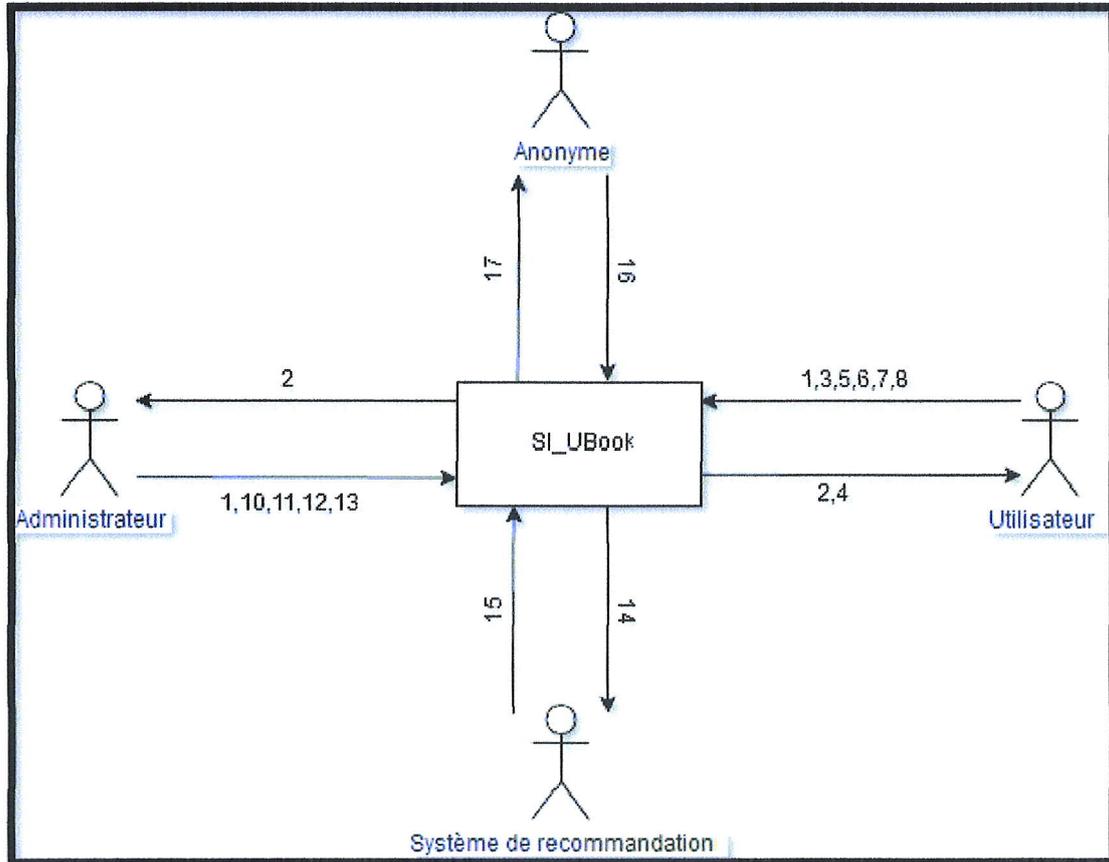


Figure 3.1 : Diagramme de contexte dynamique de système.

Numéro du message	Nature de message
1	Authentification
2	Valider l'authentification
3	Consulter un livre
4	Afficher la recommandation des livres
5	Noter un livre
6	Rechercher un livre
7	Consulter les recommandations
8	Réserver un livre
9	Afficher les réservations
10	Ajouter, supprimer, modifier, rechercher un utilisateur
11	Ajouter, supprimer, modifier, rechercher un livre
12	Gérer les emprunts
13	Ajouter, supprimer, modifier une spécialité
14	Demander les recommandations des livres
15	Donner les recommandations des livres
16	Accès a l'application UBook
17	Afficher la liste des livres de la bibliothèque

Tableau 3.1 : Spécification des messages.

3.4. Capture des besoins

L'analyse des besoins utilisateurs nous a permis de dégager les fonctionnalités qui seront offertes par notre application. Les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement seront décrites par la suite comme étant des besoins non fonctionnels (opérationnels).

3.4.1. Capture des besoins fonctionnels

L'étape de capture des besoins fonctionnels, est la première phase de la conception. Elle permet de produire un modèle des besoins en se basant sur la nature de l'acteur. Elle évite le risque de produire un système inadapté aux utilisateurs.

Le système doit répondre aux fonctionnalités suivantes :

- **Le système de recommandation fournit:**
 - Des recommandations des livres à des utilisateurs (étudiant/enseignant).
 - Une recommandation basée sur les informations des livres.

- Une recommandation basée sur les emprunts.
- Une recommandation basée sur la notation des livres.
- **Le système doit permettre au l'administrateur:**
 - De s'authentifier.
 - Gérer les livres : ajout, modification, suppression, recherche.
 - Gérer les spécialités : ajout, modification, suppression.
 - Gérer les utilisateurs : ajout, modification, suppression, recherche.
 - Gérer les emprunts : ajout, modification, suppression, recherche.
 - Gérer l'archive des emprunts.
- **Le système doit permettre à l'utilisateur :**
 - De s'authentifier.
 - Consulter les livres.
 - Rechercher un livre.
 - Noter un livre.
 - Consulter les recommandations des livres.
 - Réserver un livre.

3.4.2. Capture des besoins techniques

- L'authentification des utilisateurs par un login et un mot de passe pour accéder aux différentes fonctionnalités.
- La réservation de livre pendant 48 heures.

3.5. Les cas d'utilisation

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions réalisées par le système et produisant un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l'acteur concerné [38].

3.5.1. Identification des cas d'utilisation

Lors de la phase d'analyse des besoins, nous avons pu identifier les actions importantes que nous présentons ci-dessous et que nous les modélisons par la suite avec les diagrammes des cas d'utilisation :

Acteur	Cas d'utilisation	
Système de recommandation	Recommander des livres	Recommandation basée sur la notation des livres
		Recommandation basée sur les emprunts des livres
		Recommandation basée sur les informations des livres.

Tableau 3.2 : Identification des cas d'utilisation de système de recommandation.

Acteur	Cas d'utilisation	
Utilisateur	Authentification	Connexion
		Déconnexion
	Consulter la liste des livres	
	Noter un livre	
	Rechercher un livre	
	Consulter la recommandation des livres	
	Réserver un livre	

Tableau 3.3 : Identification des cas d'utilisation de l'utilisateur.

Acteur	Cas d'utilisation	
Administrateur	Authentification	Connexion
		Déconnexion
	Gérer les livres	Ajouter
		Rechercher
		Modifier
		Supprimer
	Gérer les utilisateurs	Ajouter
		Rechercher
		Modifier
		Supprimer
	Gérer les spécialités	Ajouter
		Modifier
	Gérer les emprunts	Rechercher
		Consulter la liste des livres réservés
		Annuler la réservation.
		Donner un livre

Tableau 3.4 : Identification des cas d'utilisation de l'administrateur.

Acteur	Cas d'utilisation	
Anonyme	Consulter la liste des livres	
	Rechercher un livre	

Tableau 3.5 : Identification des cas d'utilisation de l'anonyme.

3.5.2. Relations entre cas d'utilisation

➤ Inclusions :

Les relations d'inclusion sont identifiées par factorisation des traitements communs à plusieurs cas. Un exemple de cela est l'authentification requise pour chaque utilisateurs/administrateurs avant le début de toute utilisation du système.

➤ Extensions :

Les cas d'utilisation définis comme extensions a d'autre cas regroupent des traitements optionnels ou répondant à des conditions spécifiques. Un exemple de cela est l'extension de la récupération de livre par l'emprunt de livre.

➤ **généralisation/spécialisation :**

Ce type de relation est identifié lors de l'existence de traitements spécifiques ou modifiés d'un cas ou de plusieurs cas par rapport à un traitement normal.

3.5.3. Représentation de diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation (use cases).

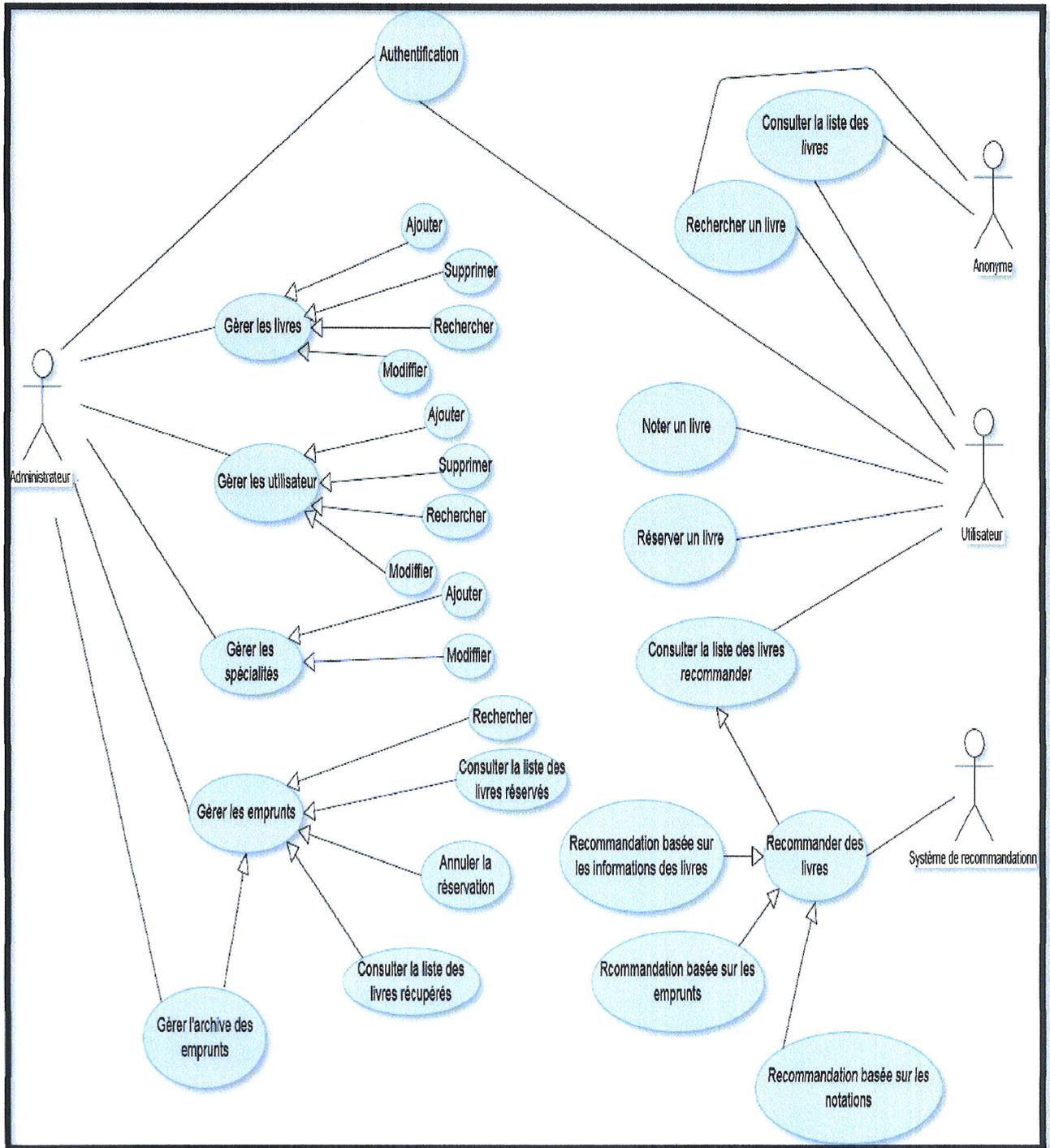


Figure 3.2 : Diagramme de cas d'utilisation de l'application.

3.5.4. Description textuelle des cas d'utilisation

Pour exprimer les cas d'utilisations de notre système, nous avons choisi le formalisme suivant :

Description sommaire :	
Titre	Authentification
But	Permettre à chaque utilisateur de l'application de s'authentifier vers ses rôles à travers un login et mot de passe.
Acteur	Administrateur/Utilisateur
Description de l'enchaînement :	
Pré-condition	Saisir le login et le mot de passe
Enchaînement	-saisir le login et le mot de passe. -Envoyer login et mot de passe. -Traitement des informations envoyées. -Accès à l'application.
Post-condition	
Alternative	En cas d'erreur, l'authentification est rejetée.

Tableau 3.6 : Description textuelle du cas d'utilisation d'authentification.

Description sommaire :	
Titre	Ajouter un utilisateur
But	Permettre à l'administration d'ajouter toutes les informations concernant l'utilisateur.
Acteur	Administrateur
Description de l'enchaînement	
Pré-condition	Authentification
Enchaînement	<ul style="list-style-type: none"> -Saisie des informations concernant l'utilisateur avec son rôle (étudiant ou enseignant). -Contrôle des données en temps réel, en cas de duplication. -Validation de la saisie. -Traitement des informations envoyées. -l'ajout est effectué avec succès.
Post-condition	
Alternative	En cas d'erreur, l'ajout est rejeté en précisant l'erreur effectuée.

Tableau 3.7 : Description textuelle du cas d'utilisation d'ajout un utilisateur.

Description sommaire :	
Titre	Ajouter un livre
But	Permettre de charger la bibliothèque.
Acteur	Administrateur
Description de l'enchaînement	
Pré-condition	Saisir le login et le mot de passe
Enchaînement	<ul style="list-style-type: none"> - Saisie de toutes les informations concernant le livre. -Contrôle des données en temps réel, en cas de duplication. -Validation de la saisie. -Traitement des informations envoyées. -l'ajout est effectué avec succès.
Post-condition	
Alternative	En cas d'erreur, l'ajout du livre est rejeté.

Tableau 3.8 : Description textuelle du cas d'utilisation d'ajout un livre.

Description sommaire :	
Titre	Consulter la liste des livres.
But	Permet à l'utilisateur de consulter tous les livres.
Acteur	Utilisateur/administrateur
Description de l'enchaînement	
Pré-condition	
Enchaînement	<ul style="list-style-type: none"> -Visualiser la liste des livres. -Accès au livre.
Post-condition	
Alternative	

Tableau 3.9 : Description textuelle du cas d'utilisation de consulter un livre.

Description sommaire :	
Titre	Rechercher un livre
But	Permet à l'utilisateur (enseignant étudiant) de rechercher un livre.
Acteur	Utilisateur/administrateur
Description de l'enchaînement	
Pré-condition	Authentification
Enchaînement	-Rechercher par auteur. -Rechercher par titre. -Rechercher par un mot clé. -La recherche a été effectuée.
Post-condition	
Alternative	En cas d'erreur, la recherche n'est pas effectuée.

Tableau 3.10 : Description textuelle du cas d'utilisation de rechercher un livre.

Description sommaire :	
Titre	Noter un livre
But	Permet à l'utilisateur (enseignant étudiant) de faire une notation d'un livre.
Acteur	Utilisateur
Description de l'enchaînement	
Pré-condition	Authentification
Enchaînement	-Consulter un livre. -Noter le livre.
Post-condition	
Alternative	

Tableau 3.11 : Description textuelle du cas d'utilisation de noter un livre.

Description sommaire :	
Titre	Recommander des livres
But	Permet au système de recommander des livres aux utilisateurs.
Acteur	Système
Description de l'enchaînement	
Pré-condition	Authentification
Enchaînement	- Consulter les informations des livres. - Consulter la notation des livres. - Consulter l'emprunt des livres.
Post-condition	
Alternative	

Tableau 3.12 : Description textuelle du cas d'utilisation de recommander des livres.

Description sommaire :	
Titre	Consulter les recommandations des livres
But	Permet à l'utilisateur (enseignant étudiant) de consulter des recommandations.
Acteur	Utilisateur
Description de l'enchaînement	
Pré-condition	Authentification
Enchaînement	
Post-condition	
Alternative	

Tableau 3.13 : Description textuelle du cas d'utilisation de consulter les recommandations.

Description sommaire :	
Titre	Réserver un livre
But	Permet à l'utilisateur (enseignant étudiant) d'emprunter un livre.
Acteur	Utilisateur
Description de l'enchaînement	
Pré-condition	Authentification
Enchaînement	<ul style="list-style-type: none"> -Faire une recherche ou une consultation d'un livre. -Faire choisir un livre. -Traiter le choix envoyés par l'administration. -L'emprunt est effectuer.
Post-condition	
Alternative	En cas d'erreur (Qte de livre=0), l'emprunt n'est pas effectué.

Tableau 3.14 : Description textuelle du cas d'utilisation réserver un livre.

Description sommaire :	
Titre	Récupérer un livre
But	Permet à l'utilisateur (enseignant étudiant) de récupérer les livre qu'il a déjà réservés.
Acteur	Utilisateur
Description de l'enchaînement	
Pré-condition	Authentification
Enchaînement	-Demander les livres déjà réservés. -L'administrateur rechercher ces emprunts. -L'administrateur donner les livres à l'utilisateur.
Alternative	Si l'emprunt n'est pas trouvé, le service n'est pas effectué.

Tableau 3.15 : Description textuelle du cas d'utilisation de récupération d'un livre.

3.6. Les diagrammes

3.6.1. Les diagrammes de séquence de l'application

Le diagramme de séquence permet de représenter les interactions entre objets, acteur et une instance d'objet en précisant la chronologie des échanges d'un message.

Il représente une instance d'un cas d'utilisation (les scénarios possible d'un cas d'utilisation donné).

Dans un diagramme de séquence, deux types de messages peuvent être échangé:

➤ **Les messages synchrones :**

Bloque l'expéditeur jusqu'à la prise en compte du message par le récepteur. Le contrôle est passé de l'émetteur ou récepteur qui devient à son tour émetteur (actif) pour la réponse ou la prise en compte. Un message de retour (en pointillé) peut être associé.

Représentation = Flèche pleine.

➤ **Les messages asynchrones :**

N'interrompt pas l'exécution de l'expéditeur qui peut émettre sans attendre la réponse du récepteur.

Représentation = Flèche creuse.

➤ **Les messages réflexifs :**

Message envoyé d'un objet vers lui-même.

Représentation = Flèche pleine.

Dans ce qui suit, nous présentons les diagrammes de séquence les plus importants.

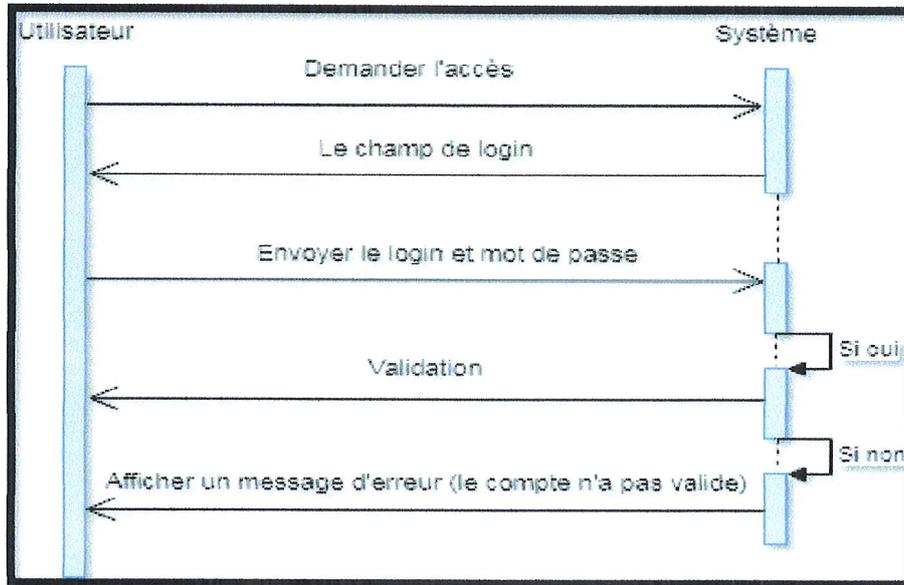


Figure 3.3 : Diagramme de séquence de l'authentification.

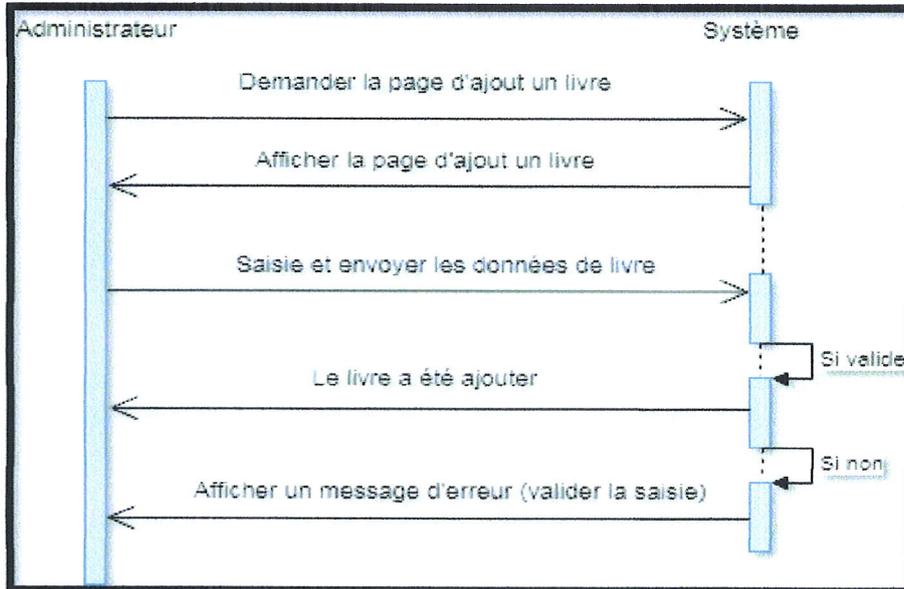


Figure 3.4 : Diagramme de séquence d'ajouter un livre.

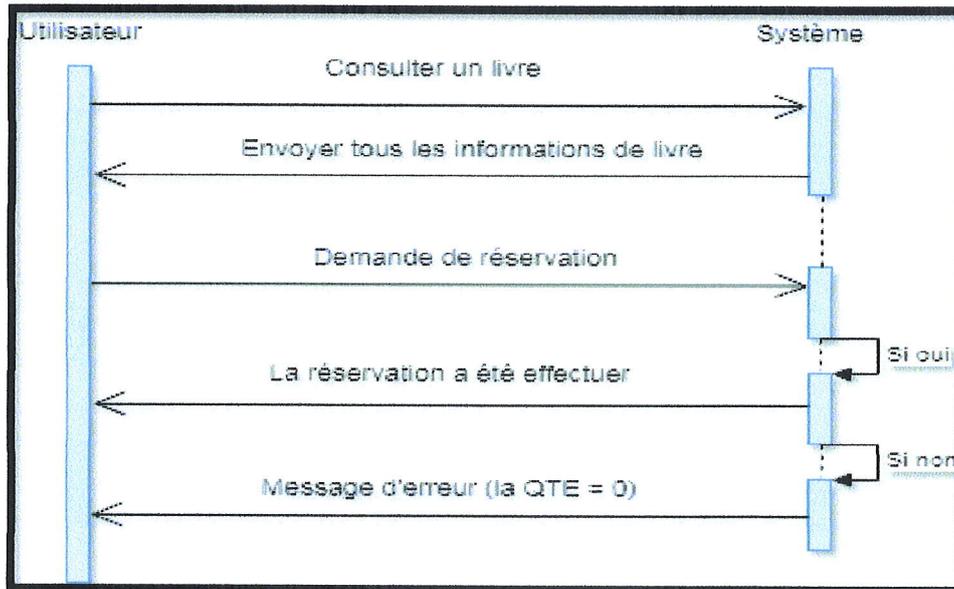


Figure 3.5 : Diagramme de séquence de réservation un livre.

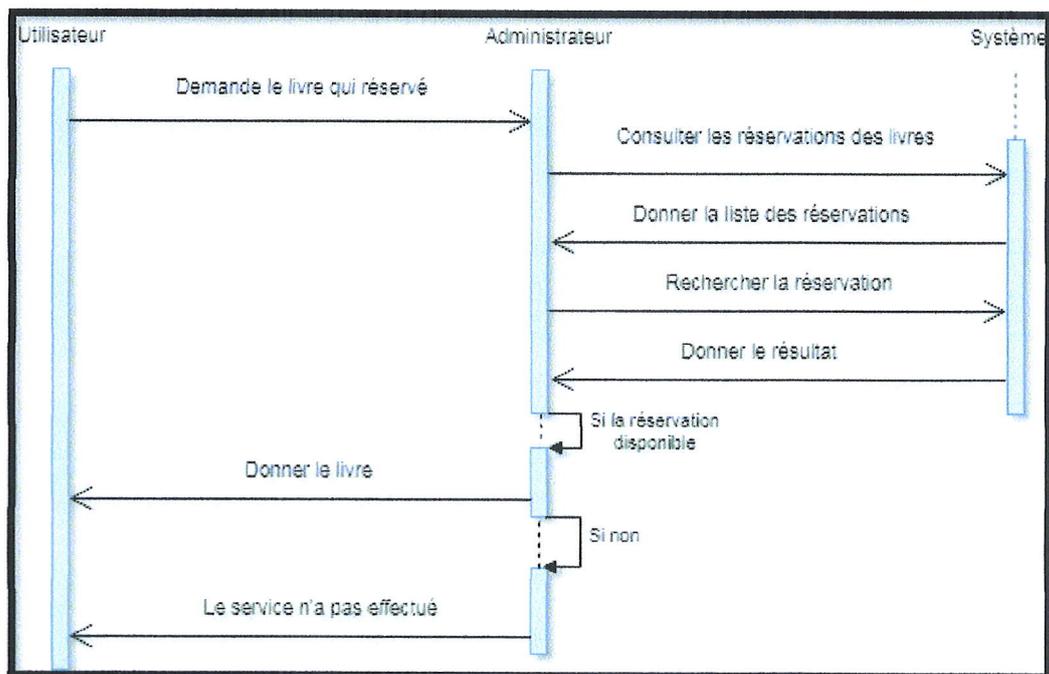


Figure 3.6 : Diagramme de séquence récupération des livres.

3.6.2. Le diagramme de classe de l'application

Le diagramme de classe est une description statique du système focalisé sur le concept de classe et d'association. Une classe représente un ensemble d'objets qui possèdent des propriétés similaires et des comportements communs décrivant en terme d'attributs et d'opérations.

Une association consiste à présenter les liens entre les instances de classe. La figure ci-dessous illustre le diagramme de classes de notre application UBook.

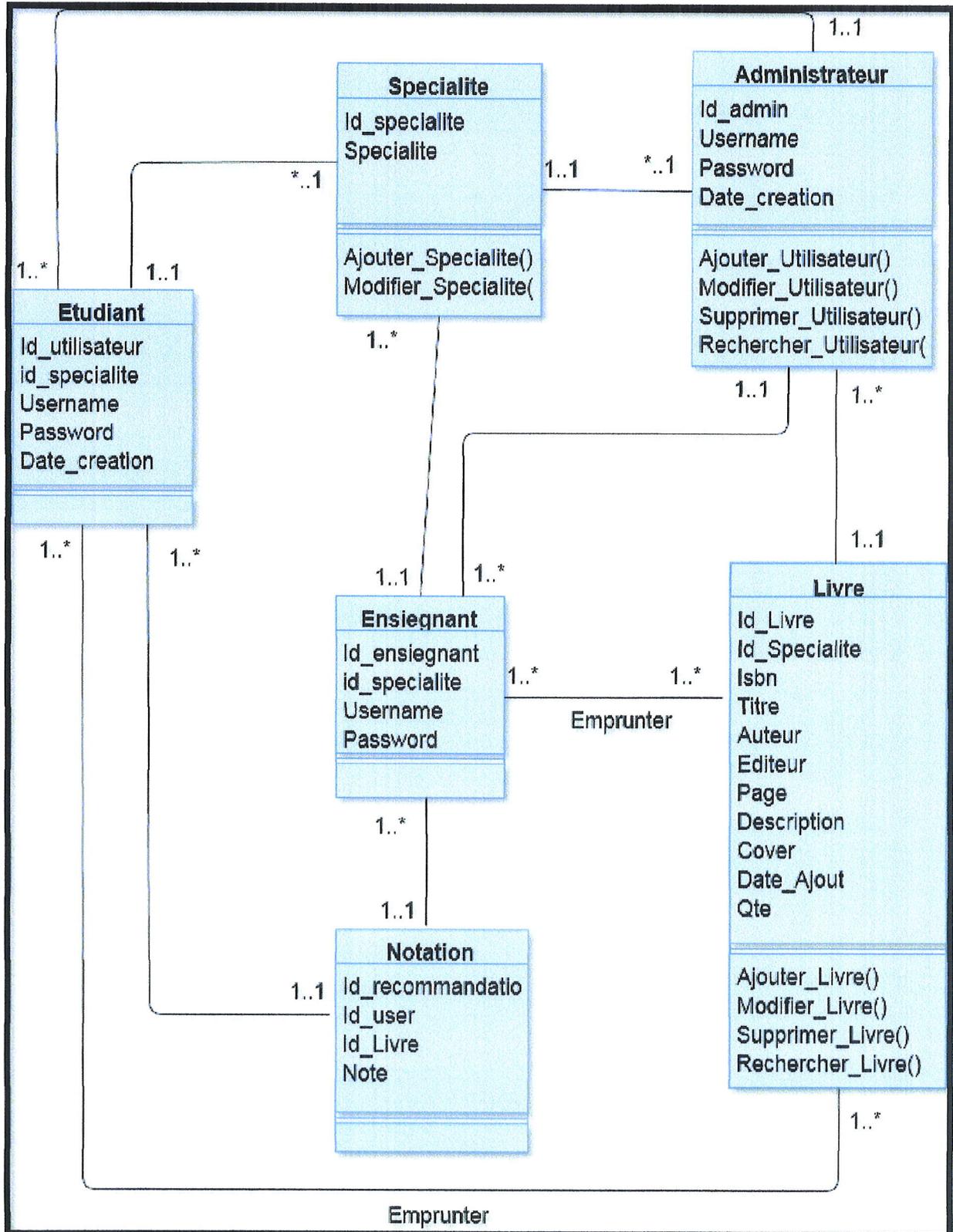


Figure 3.7 : Diagramme de classes de l'application UBook.

Deuxième partie : Les techniques de recommandations utilisées

3.7. Méthodes de recommandation utilisées

Au sein de notre application, nous avons utilisé différentes méthodes de recommandation. Dans les sous-sections suivantes, nous expliquons le principe de chaque méthode avec des exemples illustratifs.

3.7.1. La recommandation basée sur les informations des livres

Cette méthode s'inscrit dans le cadre de l'approche de recommandation basée sur le contenu. À chaque fois un utilisateur sélectionne un livre, nous lui affichons les livres les plus similaires à ce dernier selon l'éditeur, l'auteur, le titre et la description. Pour cela, nous utilisons la mesure de similarité de Dice.

- **Similarité par rapport à l'Auteur et l'éditeur**

Imagions que le livre avec l'id=2 (*Livre2*) est sélectionné. Nous devons chercher les livres ayant le même auteur et le même éditeur que *Livre2* parmi les livres appartenant à la même spécialité. Le travail de comparaison effectué est résumé dans le tableau 3.15 (à savoir : 1 indique livre avec même auteur ou livre avec même éditeur, sinon la valeur 0 est utilisée).

Livre	Livre1	Livre3	Livre4	Livre5	Livre6	Livre7	Livre8	Livre11	Livre12
Auteur	0	1	1	1	1	1	1	0	0
Editeur	1	1	1	1	1	1	1	0	0

Tableau 3.16 : Comparaison d'un livre avec les autres livres de même spécialité.

Par la suite, la similarité de Dice est calculée comme suit :

$$sim(Livre1, Livre2) = \frac{2N_c}{N_1 + N_2}$$

N_c : Le nombre des termes communs entre les deux livres.

N_1 : Le nombre des termes de livre1.

N_2 : Le nombre des termes de livre2.

Le résultat d'application numérique sur notre exemple est récapitulé dans le tableau ci-dessous.

Livre	Livre1	Livre3	Livre4	Livre5	Livre6	Livre7	Livre8	Livre11	Livre12
Résultat	1/2	1	1	1	1	1	1	0	0

Tableau 3.17 : Résultat de similarité de Dice sur l'auteur éditeur.

- **Similarité par rapport au titre**

Continuons avec *re2*. Supposons que le titre de ce dernier est composé de trois mots. Nous mentionnons dans le tableau suivant, pour chaque livre : le nombre des mots dans le titre et aussi le nombre des mots en commun avec *Livre2*.

Livre	Livre1	Livre3	Livre4	Livre5	Livre6	Livre7	Livre8	Livre11	Livre12
Nombre des mots dans le titre	3	4	2	3	5	6	5	2	3
Nombre des mots en commun	2	1	0	0	3	2	1	0	2

Tableau 3.18 : L'intersection des mots entre livre2 et les autres livres.

Nous calculons la similarité entre le livre2 et les autres livres comme suit :

$$\begin{aligned} Sim(livre2, livre1) &= \frac{2*2}{3+3} = 2/3 & Sim(livre2, livre3) &= \frac{2*1}{3+4} = 2/7 \\ Sim(livre2, livre4) &= \frac{2*0}{3+2} = 0 & Sim(livre2, livre5) &= \frac{2*0}{3+3} = 0 \\ Sim(livre2, livre6) &= \frac{2*3}{3+5} = 3/4 & Sim(livre2, livre7) &= \frac{2*2}{3+6} = 4/9 \\ Sim(livre2, livre8) &= \frac{2*1}{3+5} = 1/4 & Sim(livre2, livre11) &= \frac{2*0}{3+2} = 0 \\ Sim(livre2, livre12) &= \frac{2*2}{3+3} = 2/3 \end{aligned}$$

- **Similarité par rapport à la description**

Le même principe utilisé pour le titre est utilisé pour calculer la similarité par rapport à la description.

Enfin, on résume dans un même tableau les résultats obtenus dans le calcul de similarité par auteur/éditeur, titre et description.

	Livre1	Livre3	Livre4	Livre5	Livre6	Livre7	Livre8	Livre11	Livre12
Auteur / Editeur	1/2	1	1	1	1	1	1	0	0
Titre	2/3	2/7	0	0	3/4	4/9	1/4	0	2/3
Description	3/42	5/56	12/75	13/60	2/44	10/45	8/60	7/65	4/90
Somme	52/42	77/65	87/75	73/60	79/44	75/45	83/60	7/65	64/90

Tableau 3.19 : La somme des similarités.

Après avoir sommé les mesures de similarité obtenues, nous ordonnons les livres par ordre décroissant : ce qui donne le classement suivant (livre6, livre7, livre8, livre1, livre5, livre3, livre4, livre12, livre11).

3.7.2. La recommandation basée sur les emprunts des livres

Ce type de recommandation appartient à l'approche du filtrage collaboratif. Le filtrage collaboratif est basé généralement sur deux types de recommandations : les modèles binaires et les modèles avec évaluations. Dans un modèle binaire, on ne se base que sur le fait qu'un utilisateur ait emprunté/sélectionné/coché ou non un item donné. Dans un modèle avec évaluation, les utilisateurs sont invités à noter (par exemple, sur une échelle de 1 à 10) les différents items. Dans notre application, nous nous utilisons un modèle binaire basé sur la consultation des livres. Particulièrement, nous appliquons l'approche par cosinus.

Prenons l'exemple de la matrice des emprunts ci-dessous (1 indique que l'utilisateur a déjà emprunté le livre en question, 0 sinon) :

	Livre1	Livre2	Livre3
Utilisateur1	1	1	0
Utilisateur2	0	1	0
Utilisateur3	0	1	1

Tableau 3.20 : L'emprunt des livres par utilisateur en binaire.

Alors, chaque livre peut être représenté par un vecteur constitué de zéros et de uns. Par la suite, si un utilisateur donné a emprunté par exemple le Livre1, on lui suggérera le livre dont le cosinus avec le Livre1 est le plus élevé. Notons que le cosinus de deux vecteurs binaires est une quantité nécessairement non négative.

Dans le cas de notre exemple, nous obtiendrons :

$$\cosinus(A, B) = \sum_{j=1}^{l1} \left(\frac{v_{A,j}}{\sqrt{\sum_{j=1}^{l1} v_{A,j}^2}} \times \frac{v_{B,j}}{\sqrt{\sum_{j=1}^{l1} v_{B,j}^2}} \right)$$

$$\cosinus(\text{Livre1}, \text{Livre2}) = \frac{(1 \times 1) + (0 \times 1) + (0 \times 1)}{\sqrt{1^2 + 0^2 + 0^2} \sqrt{1^2 + 1^2 + 1^2}} = \frac{1}{\sqrt{3}}$$

$$\cosinus(\text{livre1}, \text{livre3}) = \frac{(1 \times 0) + (0 \times 0) + (0 \times 1)}{\sqrt{1^2 + 0^2 + 0^2} \sqrt{0^2 + 0^2 + 1^2}} = 0$$

Ainsi, Livre2 sera recommandé à tout utilisateur qui a déjà emprunté le Livre1.

3.7.3. La recommandation basée sur la notation des livres

Après avoir consulté un livre, un utilisateur peut noter un livre sur une échelle 1 à 5. Dans la base de données, il existe une table des notes qui contient les champs : id_recommandation,

id_utilisateur, id_livre, et note. A chaque fois un utilisateur effectue une notation d'un livre, elle est enregistrée dans cette table (voir le tableau ci-dessous pour un exemple).

	Livre1	Livre2	Livre3	Livre4	Livre5
Utilisateur1	1	5	4	?	2
Utilisateur2	3	2	?	5	5
Utilisateur3	4	3	?	5	4
Utilisateur4	?	?	3	?	1
Utilisateur5	4	3	?	5	?

Tableau 3.21 : Notation des livres par les utilisateurs.

A base de ces notations, on calcule la moyenne \bar{v}_i pour chaque utilisateur i :

$$\bar{v}_{\text{utilisateur } 1} = \frac{1+5+4+2}{4} = 3$$

$$\bar{v}_{\text{utilisateur } 2} = \frac{3+2+5+5}{4} = 3.25$$

$$\bar{v}_{\text{utilisateur } 3} = \frac{4+3+5+4}{4} = 4$$

$$\bar{v}_{\text{utilisateur } 4} = \frac{3+1}{2} = 2$$

$$\bar{v}_{\text{utilisateur } 5} = \frac{4+3+5}{3} = 4$$

Ensuite, nous utilisons la corrélation de Pearson (la formule est déjà vue dans le chapitre 2) pour calculer le poids entre les utilisateurs $w(U_A, U_B)$.

Donnons un exemple pour l'utilisateur5 :

$$w(\text{Utilisateur5}, \text{Utilisateur1}) = \frac{-2}{\sqrt{2 \times 10}} = -0.447$$

$$w(\text{Utilisateur5}, \text{Utilisateur2}) = \frac{3}{\sqrt{2 \times 6.75}} = 0.816$$

$$w(\text{Utilisateur5}, \text{Utilisateur3}) = \frac{2}{\sqrt{2 \times 2}} = 1$$

D'après ces résultats, utilisateur5 a plus de corrélation avec utilisateur3 et utilisateur2 (1 et 0.816)

Nous remarquons qu'utilisateur5, utilisateur3 et utilisateur2 partagent les mêmes préférences.

Pour estimer la note que l'Utilisateur5 aurait pu donner pour le Livre5, il faut appliquer la

Formule 2.4.2.2.

$$P_{\text{utilisateur5}, \text{livre5}} = 4 + \frac{(-0.447 \times -1) + (0.816 \times 1.25) + (1 \times 0)}{|-0.447| + |0.816| + |1|} = 4.648$$

$$P_{\text{utilisateur5}, \text{livre3}} = 4 + \frac{(-0.447 \times 1) + (0 \times 1)}{|-0.447| + |0.816| + |1|} = 3.802$$

De la même façon, nous estimons la note du livre 3. Enfin, le livre qui aura la note estimée la plus élevée et donc celui qui correspond le mieux aux préférences d'*Utilisateur*5 doit lui être recommandé.

3.8. Classement des livres

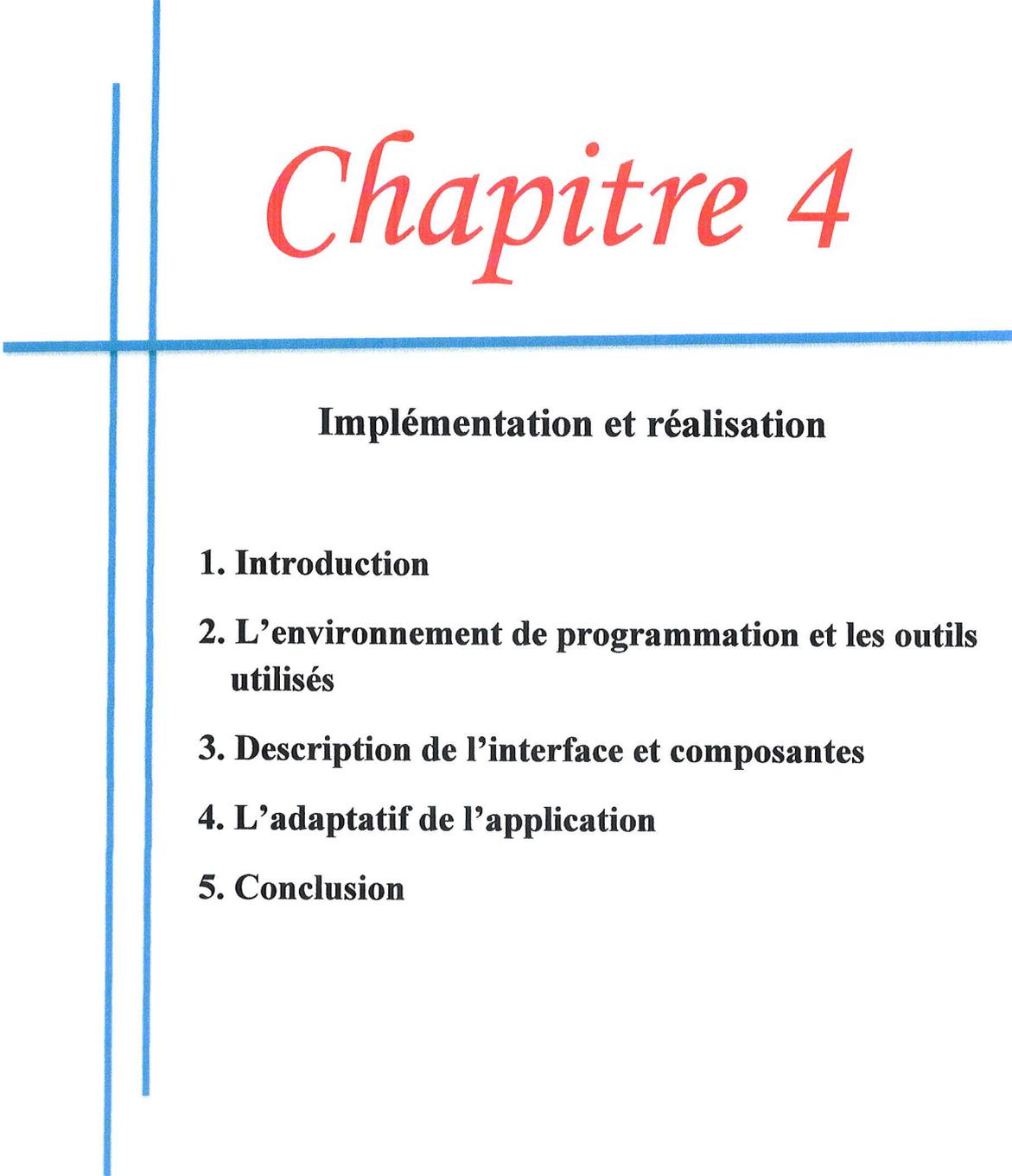
Nous proposons deux méthodes pour classer les livres :

- Les livres sont classés des plus empruntés aux moins empruntés.
- Les livres sont classés dans un ordre décroissant selon la moyenne des notations qui leur

3.9. Conclusion

Dans ce chapitre, nous avons présenté la conception de notre système de recommandation de livres au sein d'une bibliothèque universitaire. Les différents diagrammes UML illustrant le fonctionnement de notre application ont été présentés en premier lieu. Par la suite, nous avons expliqué notre approche de recommandation.

Le chapitre suivant, quant à lui, sera consacré à la description de la phase d'implémentation de notre application.



Chapitre 4

Implémentation et réalisation

- 1. Introduction**
- 2. L'environnement de programmation et les outils utilisés**
- 3. Description de l'interface et composantes**
- 4. L'adaptatif de l'application**
- 5. Conclusion**

4.1. Introduction

Ce chapitre est réservé à la description de la partie implémentation de notre travail. Nous déterminons d'abord les langages de programmation et les outils utilisés. Ensuite, nous illustrons le fonctionnement de l'application Web UBook à l'aide de quelques captures d'écran.

4.2. Langages de programmation et outils utilisés

Parmi les langages de programmation déjà présentés au premier chapitre, nous avons utilisé pour le développement de notre application : PHP, SGBD MySQL, HTML, JavaScript, CSS et AJAX. En outre, nous avons fait recours aux outils: Notepad++ et BOOTSTRAP.

4.3. Description de l'interface et composantes

- Page d'accueil



Figure 4.1 : Page d'accueil de l'application UBook.

- **Liste des livres les plus empruntés** : Cette liste sur la page d'accueil contient les livres fréquemment empruntés, classés de gauche à droit



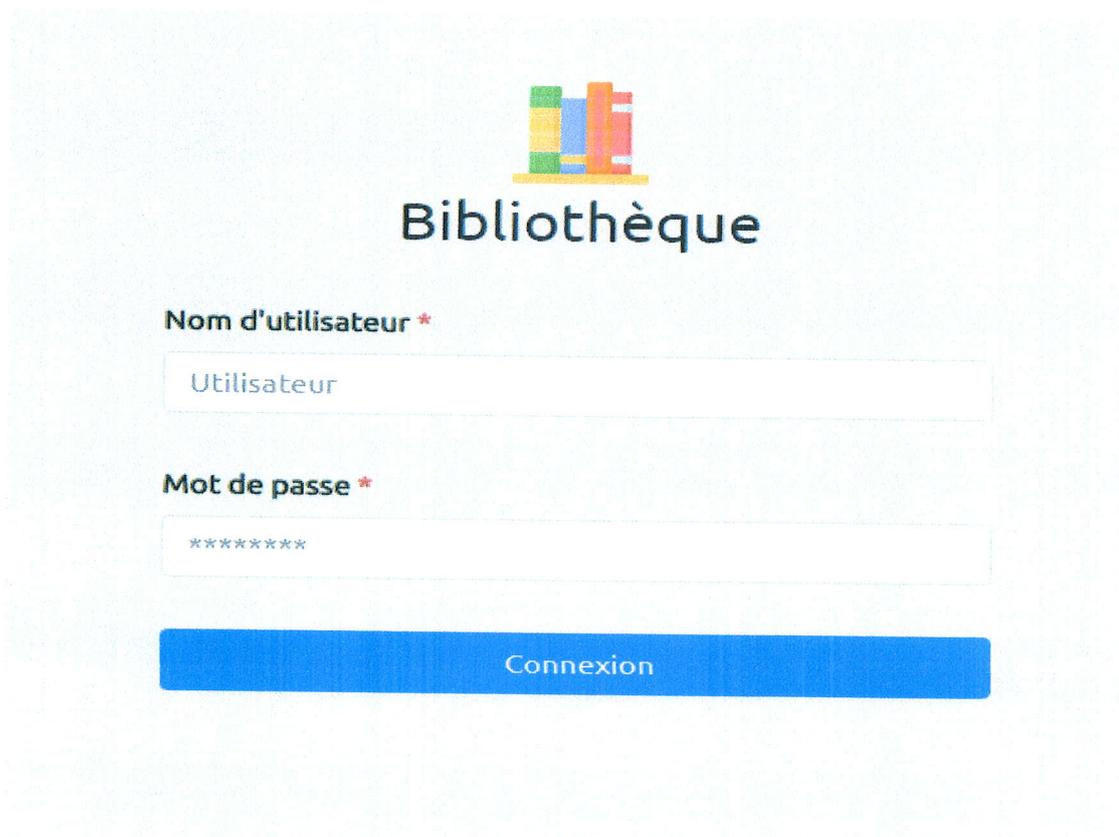
Figure 4.2 : Livres les plus empruntés.

- **Liste des livres les mieux notés** : Cette liste sur la page d'accueil contient les livres avec les notations les plus élevées, classés de gauche à droite.



Figure 4.3 : Les livres de meilleure notation.

- **Boite de connexion :** Sur la page d'authentification, vous êtes inscrit en tant qu'administrateur ou en tant qu'étudiant.



The image shows a web page for a library's authentication system. At the top center, there is a logo consisting of four colorful books (green, blue, orange, and red) stacked on a yellow base. Below the logo, the word "Bibliothèque" is written in a large, dark blue font. Underneath, there are two input fields: the first is labeled "Nom d'utilisateur *" and contains the text "Utilisateur"; the second is labeled "Mot de passe *" and contains seven asterisks "*****". Below these fields is a prominent blue button with the text "Connexion" in white.

Figure 4.4 : La page d'authentification.

- Information sur les livres

The screenshot shows a library interface with a search bar and navigation links. The main content area displays the book cover for 'PHP 7 Cours et exercices' by Jean Engels, published by Eyrolles. To the right of the cover is a table of book details:

Titre de livre : php 7 cours et exercices	
ISBN	16
Auteur	jean engels
Editeur	eyrolles
Pages	230
Date d'ajout	2019-09-09 15:00:29
Quantité	4

Below the table is a green 'Emprunter' button and a rating section with five stars (the first three are filled) and 'Enregistrer' and 'Annuler' buttons. A description bar at the bottom reads 'Description de livre' and 'php 7 cours et exercices corrigées des exercices et leur code source'.

Figure 4.5 : La page des informations des livres.

Cette page contient toutes les informations sur un livre. A partir de cette page, l'étudiant peut emprunter un livre et peut également l'évaluer. Cette page contient également trois listes de livres recommandés qui sont les suivantes :

- La liste des livres recommandés à base de notation

The screenshot shows a section titled 'Livres similaires par notes' with a grid of six book covers. Each cover includes the book title, author, and a 'disponible' status with a number of books available:

LARAVEL	PHP	PHP 7 Cours et exercices	PHP et MYSQL	Algorithmique et Programmation	L'art du développement Android
4 Emprunts	3 Emprunts	1 Emprunts	3 Emprunts	1 Emprunts	2 Emprunts
disponible 13	disponible 8	disponible 4	disponible 4	disponible 9	disponible 12

An 'Afficher plus' link is located at the bottom of the list.

Figure 4.6 : La liste des livres recommandés à base de notation

```

1  <?php
2  //=====moyenn=====
3  function moyenn($person){
4      $sum=0;
5      $cmp=0;
6      foreach($person as $key){
7          if($key!=0){
8              $sum=$sum+$key;
9              $cmp=$cmp+1;
10         }
11     }
12     if($cmp==0 or $sum==0){
13         return 0;
14     }else{
15         return $sum/$cmp;
16     }
17 }
18
19
20 //=====distance =====
21 function distance($person){
22     $sum=0;
23     foreach($person as $key){
24         if($key!=0){
25             $sum=$sum+ (($key- moyenn($person)) * ($key- moyenn($person)));
26         }
27     }
28     return sqrt($sum);
29 }
30
31

```

Figure 4.7 : Le code PHP des fonctions distance et moyenne.

```

32
33 //=====corrilation w(person1, person2)=====
34 function corrilation($person1, $person2){
35     $cor=0;
36     $i=0;
37     if($person1==$person2){return 0 ;}
38     else{
39         if((distance($person1)*distance($person2)) !=0){
40             for($i=0;$i<count($person1) ;$i++){
41                 $key2 = $person2[$i];
42                 $key1 = $person1[$i];
43                 if($key1 !=0 and $key2 !=0){
44                     $cor=$cor+ (($key1-moyenn($person1)) * ($key2-moyenn($person2)));
45                 }
46             }
47             return $cor/(distance($person1)*distance($person2));
48         }
49     }
50 }
51
52 //===== facteur de normalisation k =====
53
54 function facteur($matrix , $person1){
55     $k=0;
56     foreach($matrix as $person){
57         if($person1 != $person){
58             $k=$k+abs(corrilation($person1, $person));
59         }
60     }
61     if($k==0){return 0;}
62     else{return(1/$k); }
63 }
64

```

Figure 4.8 : Le code PHP des fonctions corrélation et facteur de normalisation.

```

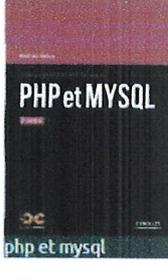
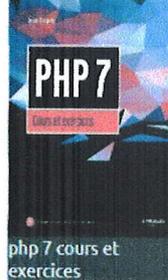
64
65 //=====p(person,object)=====
66 function similaire($matrix,$person1){
67     $som=0;
68     $pred=0;
69     $arr=array();
70     $j=0;
71     $i=0;
72     while($i<count($person1)){
73         if($person1[$i]==0){
74             foreach($matrix as $person2){
75                 if($person1!=$person2){
76                     $som=$som+correlation($person1,$person2)*($person2[$i]-moyenn($person2));
77                 }
78             }
79             $som=$som*facteur($matrix,$person1);
80             $pred=moyenn($person1)+$som;
81             $arr[0][$j]=$pred;
82             $arr[1][$j]=$i;
83             $j++;
84             $som=0;
85             $pred=0;
86         }
87         $i++;
88     }
89     return ($arr);
90 }
91 }

```

Figure 4.9 : Le code PHP de la similarité (Pearson).

- La liste des livres recommandés par les emprunts

Livres similaires par emprunts

 <p>programmez en orienté objet en php</p> <p>3 Emprunts</p> <p>disponible 2</p>	 <p>php et mysql</p> <p>3 Emprunts</p> <p>disponible 1</p>	 <p>php 7 cours et exercices</p> <p>1 Emprunts</p> <p>disponible 4</p>	 <p>Algorithmique et Programmation</p> <p>1 Emprunts</p> <p>disponible 2</p>	 <p>Architecture de l'ordinateur 4 edition</p> <p>1 Emprunts</p> <p>disponible 2</p>	 <p>android</p> <p>2 Emprunts</p> <p>disponible 12</p>
---	---	---	---	---	---

[Afficher plus](#)

Figure 4.10 : La liste des livres recommandés par les emprunts.

```

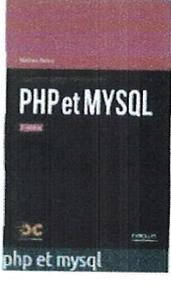
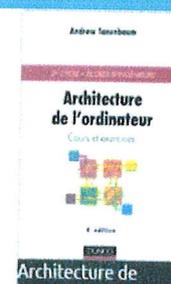
1  <?php
2  function distance_e($livre){
3      $som=0;
4      foreach($livre as $key=>$value){
5          $som=$som+($value*$value);
6      }
7      if($som==0){
8          return $som;
9      }else{
10         return sqrt($som);
11     }
12 }
13 function cosinuce($matrix,$livre){
14     $somme=0;
15     $sim=0;
16     $table_sim=array();
17     foreach($matrix as $key=>$livre1){
18         $arraykey=array_keys($livre1);
19         for($i=0;$i<count($livre);$i++){
20             $x=$livre1[$arraykey[$i]];
21             $somme=$somme+($livre[$i]*$x);
22         }
23         if(($somme==0)or(distance_e($livre)==0)or(distance_e($livre1)==0)){
24             $sim=0;
25         }else{
26             $sim=$somme/((distance_e($livre))*(distance_e($livre1)));
27             $table_sim[$key]=$sim;
28             $sim=0;
29             $somme=0;
30         }
31     }
32     arsort($table_sim);
33     return $table_sim;
34 }
35 ?>

```

Figure 4.11 : Le code PHP de la fonction Cosinus.

- La liste des livres recommandés par les informations des livres

Livres similaires à ce livre par contenu

 <p>php et mysql</p> <p>3 Emprunts</p> <p>disponible</p>	 <p>programmez en orienté objet en php</p> <p>3 Emprunts</p> <p>disponible</p>	 <p>php 7 cours et exercices</p> <p>1 Emprunts</p> <p>disponible</p>	 <p>L'art du développement Android</p> <p>android</p> <p>2 Emprunts</p> <p>disponible</p>	 <p>Architecture de l'ordinateur 4 edition</p> <p>1 Emprunts</p> <p>disponible</p>	 <p>Au cœur de Java 2 Notions fondamentales JDK 5.0</p> <p>java 2</p> <p>1 Emprunts</p> <p>disponible</p>
---	---	---	--	---	--

[Afficher plus](#)

Figure 4.12 : La liste des livres recommandés par les informations des livres.

```

1  <?php
2
3  function simulation_dice($matrix) {
4      $sarr=array();
5      $intersect=0;
6      foreach($matrix as $key => $valeur) {
7          foreach($valeur as $key1=>$valeur1) {
8              foreach($valeur1 as $k=>$v) {
9                  if($v==1){
10                     $intersect++;
11                 }
12             }
13             $dice=(2*$intersect)/(2*count($valeur1));
14             $sarr[$key1]=$dice;
15             $dice=0;
16             $intersect=0;
17         }
18     }
19     return $sarr;
20 }
21
22 function dice_content_text($title1,$title2) {
23     $a1 = preg_split("/[\s, . ; : ! ? ] + /", strtolower($title1));
24     $a2 = preg_split("/[\s, . ; : ! ? ] + /", strtolower($title2));
25     $sarr1=array_unique($a1);
26     $sarr2=array_unique($a2);
27
28     $res=array_intersect($sarr1,$sarr2);
29     $cmp= count($res);
30     $dice=(2*$cmp)/(count($sarr1)+count($sarr2));
31     return $dice;
32 }
33
34 ?>

```

Figure 4.13 : Le code PHP des fonctions de similarité de Dice.

4.4. L'adaptabilité de l'application

- Pour les écrans larges ($\geq 1200\text{px}$)



Figure 4.14 : L'affichage de l'application en large écran.

- Pour les PC ($\geq 992\text{px}$)



Figure 4.15 : L'affichage de l'application en PC.

- Pour les tablettes ($\geq 768\text{px}$)



Figure 4.16 : L'affichage de l'application en tablette.

- Pour les mobiles (< 768px)

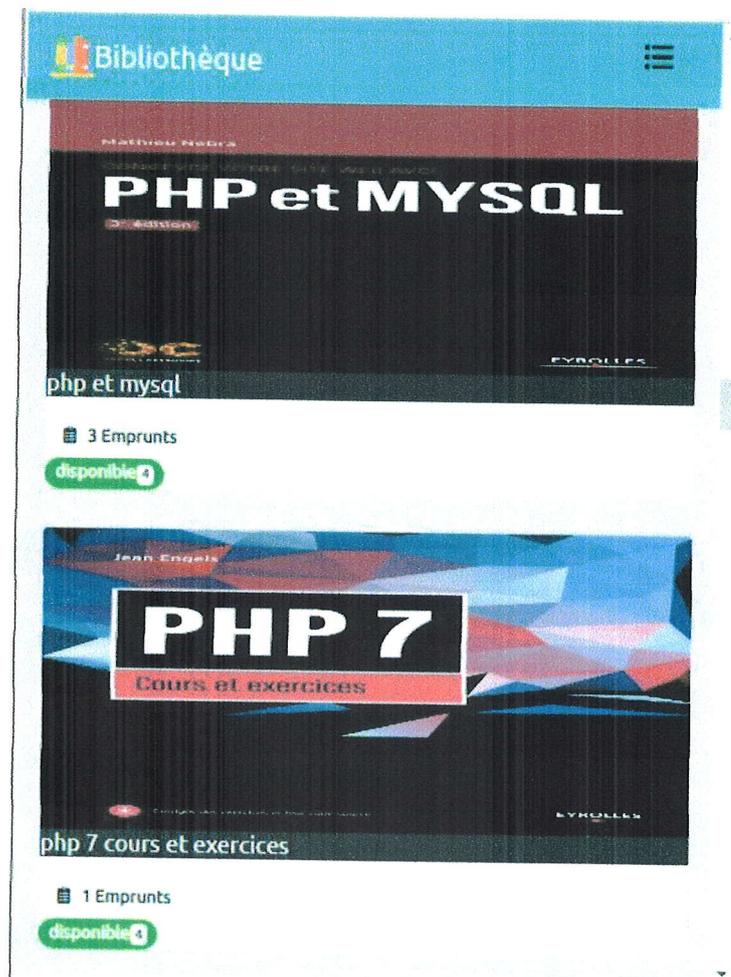
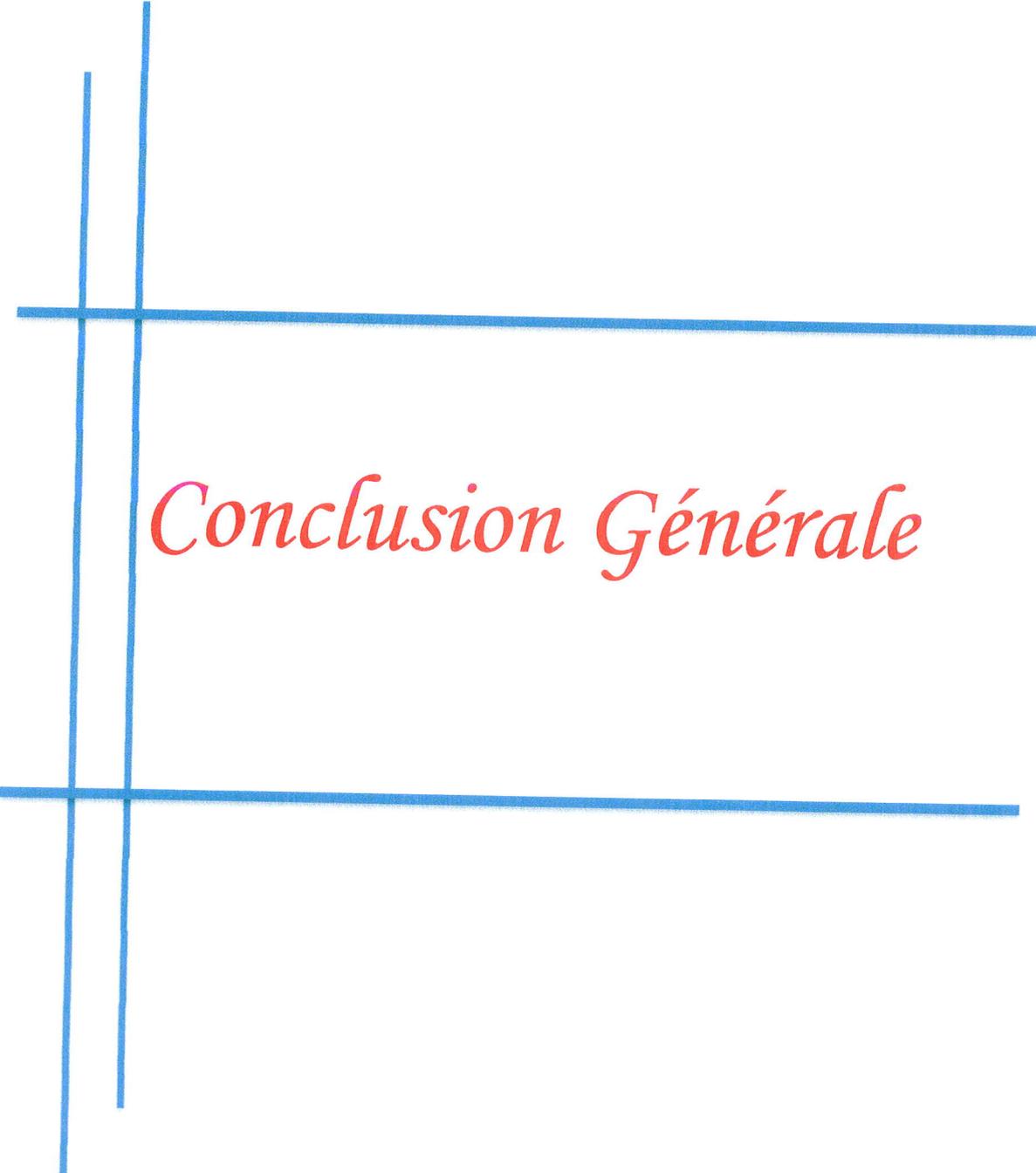


Figure 4.17 : L'affichage de l'application en mobile.

4.5. Conclusion

Dans ce chapitre, nous avons présenté l'application web UBOOK qui permet à ses utilisateurs de profiter de plusieurs types de recommandation de livres. L'application permet la réservation des livres souhaités en ligne afin de les récupérer par la suite au niveau de la bibliothèque. Un atout important de l'application UBOOK est qu'elle se prête à s'afficher correctement sur tout type d'écran (PC, mobile, tablette, etc.)



Conclusion Générale

Conclusion Générale

Un système de recommandation a pour but de fournir à un utilisateur les ressources pertinentes en fonction de son profil. Le sujet de ce mémoire porte sur la problématique de recommandation de livres dans une bibliothèque universitaire. En effet, les étudiants trouvent des difficultés à choisir les livres les plus appropriés à cause du problème de surcharge d'information. Les opinions orales des collègues et/ou des enseignants restent insuffisantes et peu pratiques.

Pour résoudre ce problème, nous avons développé « *l'application Web adaptatif UBook* » qui facilite aux étudiants le choix des bons livres à emprunter. Pratiquement, trois types de recommandation sont suggérés par UBook :

- Recommandation basée sur les informations de livre (le titre, l'auteur, l'éditeur et la description)
- Recommandation basé sur la fréquence des emprunts des livres.
- Recommandation basé sur la notation des livres.

L'application permet alors aux étudiants de rechercher des livres, de bénéficier de plusieurs recommandations mais aussi d'effectuer une réservation des livres désirés en ligne.

En perspective, nous envisageons d'appliquer d'autres approches de recommandation plus ou moins intelligentes, notamment celles basées connaissances.

Bibliographie

- [1]: [Camille Roux, L'histoire et l'avenir du Web, page 2].
- [2]: [<https://recherchemid.wordpress.com/2016/11/29/quest-ce-que-le-web-2-0-histoire-caracteristiques-et-perspectives/>].
- [3]: [<https://www.ideematic.com/dictionnaire-digital/application-web>].
- [4]: [<https://www.symetris.ca/sites/default/files/application-web.png>].
- [5]: [BOUKLI HACENE abdelhafid Conception et Développement d'une application Java sous Android (Donner du sang) le 26 mai 2015 page 13].
- [6]: [<https://www.symetris.ca/sites/default/files/application-web.png>].
- [7]: [http://www.foad-mooc.auf.org/IMG/pdf/Creation_sites_web.pdf].
- [8]: [<http://www.wisibilis.com/themeGraphique/surMesure/image/composition-offre-site-web-boutique.png>].
- [9]: [<https://archive.awt.be/contenu/tel/res/res.fr.fic,040,000.pdf>].
- [10]: [L. SHKLAR ET R. ROSEN. Architecture: Principles, Protocols and Practices. Edition: John WILEY ET SONS, 2003].
- [11]: [Samir ADOUANE. Intégration des moyens de modification dynamique des contenus sur le web. Université el Hadj Lakhdar Batna, Batna, 2007].
- [12]: [<http://salihayacoub.com/420Ke9/SQLite/SqliteIntroduction.pdf>].
- [13]: [Philippe Rigaux, Pratique de MySQL et PHP, 4e Edition, Paris, 2009 (ISBN 978-2-10-053752-5)].
- [14]: [Olivier Guibert, Bases de Données Oracle Administration, l'Université Bordeaux 1, Édité le 8/4/14].
- [15]: [<https://docs.postgresql.fr/8.3/preface.html>].
- [16]: [Jérôme Lafosse, Java EE guide de développement d'application web en java].

- [17]: [Balagurusamy, Programming In C#, Tata McGraw-Hill Education - 2008, (ISBN 9780070667570)].
- [18]: [Anders Hejlsberg Mads Torgersen Scott Wiltamuth Peter Golde, The C# Programming Language (Covering C#4.0), Portable Documents, Addison-Wesley Professional-2010, (ISBN 9780132481724)].
- [19]: [Daniel Charnay, Philippe Chaléat. HTML et Javascript. Eyrolles, pp.450, 1998, Best of Eyrolles, 2-212-11157-6. fhal-00001356].
- [20]: [Thomas Bodin, Bootstrap, 14 octobre 2017].
- [21]: [Pearson France, Ben Frain, Responsive Web Design, 2013].
- [22]: [<https://www.journaldunet.com/developpeur/client-web/responsive-design/>].
- [23]: [Mohamed Belmokhtar, Germain Souquet, Responsive Web Design, Université de La Rochelle, Avril 2014].
- [24]: [Ivan MADJAROV, Arnaud FÉVRIE, Aix Marseille Université, Département Réseaux et Télécommunications, Marseille, France, 17-20 novembre 2014].
- [25]: [<https://fr.yeeply.com/blog/developpement-d-application-web-pour-votre-site/>].
- [26]: [NAAK AMINE, Papyrus : Un système de gestion et de recommandation d'articles de recherche, Université de Montréal, Juillet 2009, page 10].
- [27]: [NICOLAS Béchet, Etat de l'art sur les Systèmes de Recommandation. Projet AxIS de l'INRIA, dans le cadre du projet Addictrip, page 03].
- [28]: [JONATHAN LOUËDE, Stratégies de Bandit pour les Systèmes de Recommandation, le 04 Novembre 2016 (04/11/2016)].
- [29]: [BENOSMAN IMENE, CHARIF NESRINE, Utilisation de l'approche pré filtrage contextuel des systèmes de recommandation sensible au contexte, le 26 JUIN 2016].
- [30]: [<https://interstices.info/les-systemes-de-recommandation-categorisation/>].
- [31]: [Idir Benouaret, Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels, le 25 janvier 2017].

- [32]: [Salton, 1989 Salton, G. (1989). Automatic text processing : The transformation, analysis, and retrieval of. Reading : Addison-Wesley].
- [33]: [Mooney, et al. 2000 R. J. Mooney et L. Roy (2000). "Content based book recommending using learning for text categorization", 2000].
- [34]: [Naak Amine, 2009, Un système de gestion et de recommandation d'articles de recherche, page 17-18].
- [35]: [Mme KARAOUZENE Meryem, Système de recommandation des services web sémantiques, le 14 Juin 2015].
- [36]: [Idir Bounaoine, (2017),Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels, le 25 Janvier, page 38].
- [37]: [Goaffray Bonnin(2010), des systèmes de recommandation robustes pour la navigation Web : inspiration de la modélisation statistique du langage, le 23 Novembre 2010, page 27].
- [38]: [Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering, *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence* (pp. 43-52)].
- [39]:[Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J., GroupLens: an open architecture for collaborative filtering of netnews, 1994].
- [40]: [Naak Amine, Un système de gestion et de recommandation d'articles de recherche, 2009, page28-29].
- [41]: [An Te NGUYEN, THÈSE pour obtenir le grade de DOCTEUR DE L'UNIVERSITÉ JOSEPH FOURIER – GRENOBLE I, le 23/11/2006].
- [42]: [HAFOUDA AMIRA, BASSIMANE RIHAB, BENSACI HAYAT, Réalisation d'une application de recommandation, le 2017/2018].
- [43]:[Bouchindhomme and Rochlitz, 1992] Bouchindhomme and Rochlitz (1992). Dans le flou artistique, *Eléments d'une théorie de la "rationalité esthétique"*, page 203–238.
- [44]: [Majda MAATALLAH] Une Technique Hybride pour les Systèmes de Recommandation, theseUnivercité Badji Mokhtar Annaba.