Projet de fin d'études

Prénom Nom

21 juillet 2018

Table des matières

Table des figures

Remerciements

Je tiens tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce travail. Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce rapport par exprimer mes profondes gratitudes et respectueuse reconnaissance à tous ceux qui ont contribué de près ou de loin à ce travail :

À **Mme. Hajer FRADI** enseignante à l'ISSAT de Sousse, pour son encadrement, sa disponibilité, ses conseils précieux et pour la qualité de ses suivis durant toute la période du projet.

À **M. Dhaker ABDELJAWAD** mon encadrant, pour sa bonne volonté d'accepter de m'encadrer, pour le temps qu'il m'a octroyé et tous les conseils et les avis éclairés, qu'il m'a prodigué.

À **M. Kamel ALOUANI** le fondateur du société, À **MM. Salmen MESKA, Brahim KRAIM, Houssam MOHAMED** développeurs, pour leurs accompagnements durant tout le stage, leurs confiances, leurs soutiens, leurs admirables suivis, leurs conseils, leurs bienveillances et leurs disponibilités.

Introduction Générale

De nos jours, les formations professionnelles sont les meilleurs moyens pour améliorer les compétences des individus en premier lieu et d'augmenter les valeurs des entreprises en deuxième lieu. Pour cette raison la société **Sindibad group** a développé une plateforme d'écosystème professionnelle nommée « Takwinland » qui gère les relations entre les entreprises ou les particuliers et les formateurs indépendants ou les organismes de formations. Donc chaque acteur parmi ces derniers va utiliser la plateforme (recherche, navigation, transaction ...) pour répondre à ses besoins. Mais ce n'est pas tout à fait évident de trouver la meilleure réponse à ses besoins quand il s'agit de grandes bases de données. D'où l'idée de développer un système de recommandation (en anglais Recommender System RS) qui va recommander des *items* (formations, sessions ou projets de formations) pour tous les acteurs de l'application. Le présent rapport présentera les différentes étapes de la réalisation de ce projet et s'étalera sur cinq chapitres :

- Le premier chapitre « Présentation générale du projet » est un chapitre introductif dans lequel nous effectuons une brève description de l'entreprise. Ensuite, nous exposons le cadre général du projet ainsi que ses objectifs.
- Le second chapitre « Étude préalable » où nous décrivons les méthodes existantes pour mesurer la similarité des documents et nous introduisons les systèmes de recommandation existants. Ensuite, nous donnons quelques critiques par rapport aux solutions RS existantes pour présenter par la suite notre solution proposée. Enfin, nous décrivons les méthodes couramment utilisées pour évaluer les systèmes de recommandation.
- Le troisième chapitre « Analyse et spécification des besoins », où d'abord nous identifions les différents acteurs. Après, nous dégageons les besoins fonctionnels et non fonctionnels. Enfin,

nous illustrons les diagrammes de cas d'utilisations avec des descriptions textuelles.

- Le quatrième chapitre « La conception », dans lequel nous présentons l'architecture de l'application et le système de recommandation. Ensuite, nous expliquons le fonctionnement du RS avec les diagramme de classes, diagrammes de séquences, diagramme d'activités et les diagrammes de composants.
- Le cinquième chapitre nommé « Réalisation » est dédié à la présentation des technologies et des frameworks adoptés ainsi l'environnement matériel et logiciel. Nous visualisons par la suite les résultats du RS avec des scénarios différents.

Nous clôturons ce rapport par une conclusion générale dans laquelle nous évaluons les résultats atteints et nous exposons les éventuelles perspectives du présent projet.

Chapitre 1

Présentation Générale

Introduction

Ce chapitre est dédié à l'introduction du cadre général du travail. Au début, nous allons présenter l'organisme d'accueil et nous détaillons, par la suite, le contexte général du projet ainsi que ses objectifs et sa motivation.

1.1 Présentation de l'organisme d'accueil

1.1.1 C'est qui Sindibad Group

Sindibad Group est une société tunisienne fondée en 2016. Elle est leader de recherche, de développement et de conseil en matière d'expérience client. Sindibad Group est engagée socialement et désireuse de résoudre les problèmes de développement les plus urgents de la région.
Elle travaille en partenariat avec des entreprises, prestataires de services, centres de recherche, universités et tout agent de changement pour concevoir, construire et fournir des solutions à valeur ajoutée.

1.1.2 Missions

Sindibad Group responsabilise les individus et les organisations et leur permet de contribuer dans toute la mesure du possible au développement social et économique de leurs organisations et

communautés en améliorant sensiblement leurs capacités en termes de compétences techniques et professionnelles.

1.1.3 Approches

Sindibad group profite de nouvelles technologies et d'approches innovantes telles que la numérisation, l'automatisation des processus intelligents, l'analyse des processus métier et la refonte des processus maigres.

1.1.4 Solutions

Sindibad group construit une plateforme numérique globale nommée Takwinland pour être le premier écosystème de formation dans le monde, qui fournit à chaque partie prenante un bureau ou interface numérique intégrée qui cartographie le processus de gestion de formation de bout en bout pour trouver et collaborer avec le reste de l'écosystème membres. Pour un particulier, une entreprise, un organisme à but lucratif ou non lucratif, un fournisseur de formation, un formateur, une institution d'accueil et un fournisseur Giveaways, Takwinland offre un accès gratuit à un bureau numérique à domicile innovante. Vous serez équipé d'outils qui vous permettront de créer et gérer vos tâches quotidiennes, envoyer et recevoir des demandes, des retours, des documents et des offres, suivre en temps réel l'évolution de vos projets en cours et envoyer des rappels.

1.2 Présentation du projet

1.2.1 Cadre du projet

Ce projet de fin d'études de la formation d'ingénieurs en Informatique à l'Institut Supérieur des Sciences Appliquées et de Technologie de Sousse (ISSATSo), et qui a été effectué au sein de la société "Sindibad group" a pour objectifs : la conception et le développement d'un système de recommandation intelligent pour une plateforme de gestion de formations professionnelles basée sur l'apprentissage automatique et l'analyse de données.

1.2.1.1 Le problème de la sur-information

Les habitudes en ligne sont fortement évoluées depuis la création d'internet, pour se diriger vers un Web où les données sont au cœur des applications et où les utilisateurs sont incités à la fois implicitement et explicitement à ajouter de la valeur aux applications qu'ils utilisent : c'est l'émergence de ce qu'on appelle le Web 2.0. Cette transformation de l'utilisateur en un publicateur de contenu a amené la quantité d'informations disponible dans le monde [1]. On se retrouve dès lors face à un problème de sur-information, nuisible à l'utilisateur ainsi qu'illustré par la figure 1 [2].

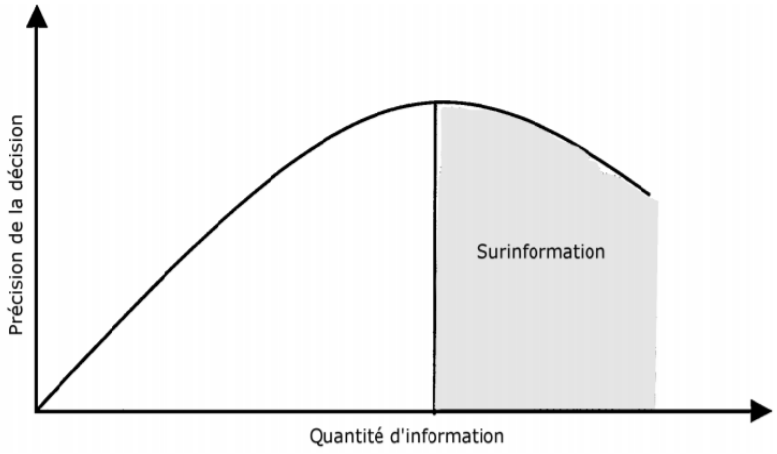

FIGURE 1.1 – Courbe en U inversé représentant le problème de sur-information

On peut trouver de nombreuses définitions pour la situation de sur-information ≪information overload≫ dans la littérature (Eppler & al. [3]), mais nous considérerons la définition suivante : Il y a sur-information lorsque la capacité d'un individu à intégrer de l'information (i.e., la quantité d'information qu'un individu peut intégrer dans la procédure de prise de décision au cours d'une période de temps spécifique) est inférieure à la capacité requise (i.e., la quantité d'information qu'un individu doit intégrer pour compléter une tâche) [3].

La nuisance apportée par les situations de sur-information touche aussi bien la recherche de données (la recherche devient moins systématique, l'identification d'information pertinente devient plus difficile...), que l'analyse des informations (omission, mauvaise interprétation...), la qualité des

décisions prises (efficacité diminuée, délai de décision augmenté...) ou l'utilisateur lui-même (stress, démotivation...) [3].

La quantité d'informations disponible n'est par ailleurs pas la seule cause pouvant mener à une situation de sur-information : les informations dupliquées, les informations complexes, la pression du temps ou la multiplicité des canaux de distribution de l'information sont également des facteurs de sur-information.

Les réseaux sociaux sont très certainement le type d'application web où cette situation est la plus flagrante. Un réseau social est un site web qui facilite les rencontres, la recherche d'esprits semblables, la communication et le partage de contenu, et la construction d'une communauté [4]. Ces sites encouragent la participation active et l'expression de leurs utilisateurs au travers des possibilités d'actions CRUD ce qui entrainent de très larges quantités de contenu créées par les utilisateurs, et qui contribuent fortement au phénomène de sur-information abordé précédemment.

1.2.1.2 Le cas des plateformes d'E-Learning

Les progrès des technologies informatiques, ainsi que le développement et la démocratisation d'Internet (plus de 3 milliards d'internautes en 2014 [5]) ont naturellement contribué au développement du Technoloy Enhanced Learning (TEL). Manouselis & al. [6] définissent le TEL comme ayant pour but la conception, le développement et l'expérimentation d'innovations socio-techniques qui pourront supporter et améliorer les pratiques d'apprentissage des individus et des organisations. Parmi ces innovations permises par le web, les plateformes d'e-learning se sont très largement développées au cours des dernières décennies, tout d'abord espaces de stockage et de médiatisation des données académiques, puis avec les possibilités du Web 2.0, Learning Managment System (LMS) et Personal Learning Environment (PLE), qui seront principalement considérés dans ce travail.

Le modèle général d'un LMS, ou Système de gestion de l'apprentissage, consiste à intégrer un ensemble d'outils (forums, quizz) et des données (étudiants, contenu) à l'intérieur d'un contexte de cours ou de module. Ce modèle suit le modèle d'organisation général de l'éducation en modules de cours, et l'isolation de l'apprentissage en unités discrètes. Les LMS sont des systèmes asymétriques (les formateurs ont plus de possibilités et de contrôle sur le système) centrés sur les formations. L'expérience que les apprenants ont de ce type de plateforme est très homogène.

Le modèle d'un PLE est quant à lui centré sur l'apprenant, et vise à lui offrir des possibilités de coordination avec un grand nombre de services. Les relations y sont symétriques puisque les utilisateurs sont tous invités à être à la fois producteurs et consommateurs de contenu, peuvent organiser leurs ressources comme ils l'entendent et choisir les outils qu'ils veulent. La figure 2 illustre le concept du PLE [7].

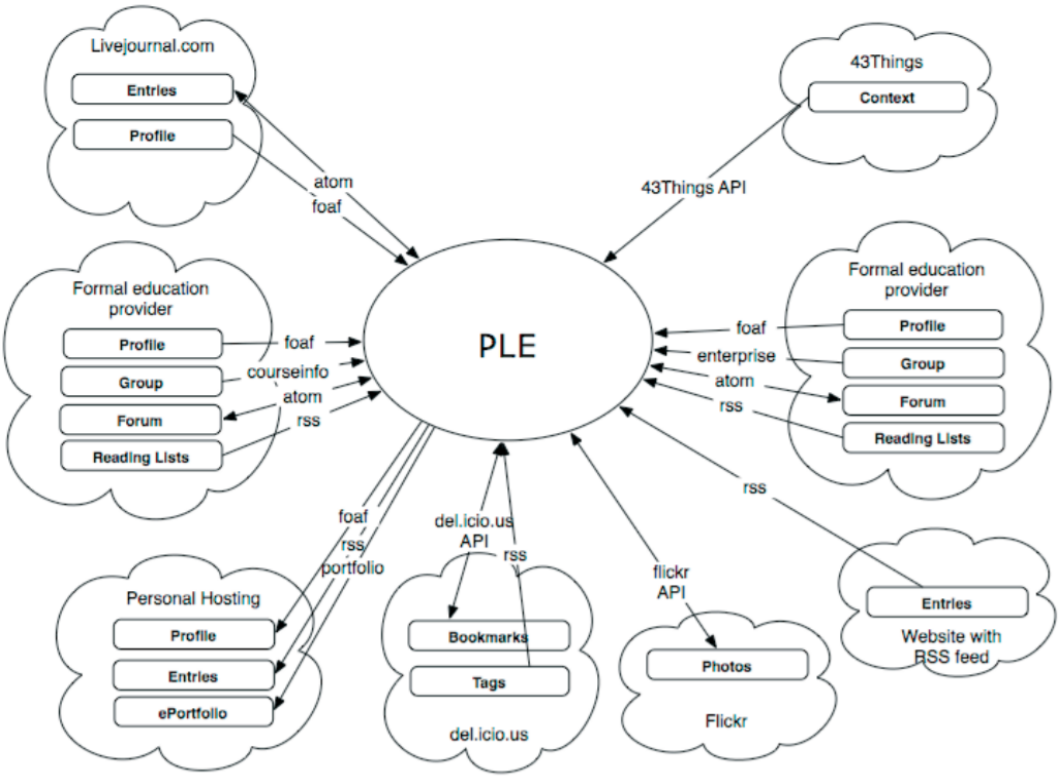

FIGURE 1.2 – Modèle conceptuel d'un PLE, d'après Wilson [7]

Cette évolution amène les utilisateurs de ces plateformes à être confrontés à un choix de données potentiellement trop grand, ce qu'en fait des candidats évidents aux problèmes de sur-information.

1.2.2 Objectifs

La portée la plus importante du projet sera de recommander aux apprenants et entreprises (des sessions et des formations), aux formateurs et organisme de formations (des formations, des sessions

et des projets de formations) dans différents situations (nouveau compte, faible historique, agit en temps réel avec ses historiques) tout en protégeant leurs informations personnelles.

1.2.3 Motivation

De nos jours, les données numériques sur Internet sont plus que jamais énormes comme nous avons dit précédemment, ce qui a créé un problème potentiel de surcharge d'informations, empêchant l'accès en temps réel à des éléments d'intérêt sur Internet. Ce surcharge sera utilisé dans la recommandation afin d'améliorer les résultats. Durant ce projet, nous envisagerons de développer un moteur de recommandation basé sur des informations de l'utilisateur c.à.d son profil, son historique de navigations, ses recherches et de différentes transactions sur la plateforme seulement.

Conclusion

Nous avons présenté dans ce chapitre l'organisme d'accueil. Ensuite, nous avons passé au cadre général du projet. Dans le chapitre suivant nous étudions quelques systèmes de recommandation (RS) existants afin de comprendre le processus du travail avec quelques critiques par rapport à ce qu'il existe. Enfin, nous présentons notre solution proposée.

Chapitre 2

Étude Préalable

Introduction

A travers ce chapitre nous faisons une étude préalable afin de mesurer les similarités entre des documents (des formations, des profils, des sessions, des projets de formations). En outre, nous nous occupons dans ce chapitre de présenter brièvement les solutions existantes en RS avec quelques critiques.

2.1 Similarité des documents

2.1.1 Les bases de données

L'ensemble de données doit être suffisamment grand pour avoir un degré de similarité plus précis entre les documents. Il doit également être d'une variabilité assez large pour que les documents des différentes catégories soient distincts les uns des autres afin de permettre une délimitation claire entre les catégories [8].

2.1.2 Pré-traitements

Le pré-traitement des données est une technique d'exploration de données qui implique la transformation des données brutes en un format compréhensible (selon les exigences du projet en question). Il comprend des différentes étapes :

- **Filtrage des données :** Les données sont filtrées par des traitements tels que le remplissage des valeurs manquantes, le lissage des données bruyantes ou la résolution des incohérences dans les données.
- **Intégration des données :** Les données avec une représentation différente sont rassemblées et les conflits au sein des données sont résolus.
- **Transformation de données :** Les données sont normalisées, agrégées et généralisées.
- **Réduction des données :** Cette étape vise à donner une représentation réduite des données dans un entrepôt de données.
- **Discrétisation des données :** Cela implique la réduction d'un certain nombre de valeurs d'un attribut continu en divisant la plage d'intervalle d'attributs.

2.1.3 Extraction des caractéristiques

2.1.3.1 Bag of Words

Le modèle de Bag of Words BOW est une représentation simplificatrice utilisée dans le traitement de la langue naturelle et la recherche d'information (IR). Dans ce modèle, un texte (tel que la phrase ou un document) est représenté comme un sac sans tenir compte de la grammaire et même de l'ordre des mots mais en gardant la multiplicité. Le modèle BOW est couramment utilisé dans les méthodes de similarité de documents où la fréquence ou l'occurrence de chaque mot est utilisée comme une information pertinente pour le classificateur. Il est comparable au modèle skip gram de l'unigramin le modèle de langage [15].

2.1.3.2 Extraction du vecteur de caractéristiques

Feature Vector ou en français vecteur de caractéristiques, est le processus de conversion du modèle de bag of words dans la forme vectorielle où chaque mot est représenté avec ses fréquences. Pour la création de vecteurs de caractéristiques, au départ, un vocabulaire est construit en utilisant tout le corpus disponible dans l'ensemble de données, ce qui aide à créer un modèle d'espace vectoriel pour les mots et le vecteur est dérivé de chaque corpus en conséquence [9].

2.2 Analyse de données

L'analyse de données [10] est une composante essentielle de l'exploration de données et de la veille économique et elle permet d'acquérir les connaissances nécessaires à la prise de décision. L'analyse des données est un moyen pour les organisations et les entreprises leur permettant d'obtenir les informations dont elles ont besoin afin de prendre des meilleures décisions, gérer leurs clients et augmenter la productivité et les revenus. En outre, avec la croissance de l'Internet, il y a tellement de données et d'informations numériques disponibles et l'analyse des données est devenue plus nécessaire que jamais. Certaines des techniques d'analyse de données sont :

- **Descriptif :** Ces sont les techniques d'analyse qui utilisent l'agrégation et l'exploration de données pour fournir un aperçu du passé et répondre ≪ Que s'est-il passé ? Il implique le calcul de simples mesures de composition et de distribution des variables. Ils sont souvent utilisés pour décrire la relation entre les données, telle que : stock total dans l'inventaire, l'argent moyen dépensé etc.
- **Prédictif :** Il s'agit du processus d'extraction d'informations à partir des ensembles de données existants afin de déterminer des modèles et de prédire les résultats et les futurs tendances. Il englobe une variété de techniques statistiques issues de la modélisation prédictive, de l'apprentissage automatique et du data mining. Par exemple, prédire quels articles les clients achèteront ensemble, comment les ventes pourraient se conclure à la fin de l'année, etc.
- **Prescriptif :** C'est une technique d'analyse de données qui permet à l'utilisateur de prescrire un certain nombre de différentes actions possibles et de les guider vers une solution. Ces analyses consistent donc à fournir des conseils.

  Système de recommandation : C'est l'une des techniques d'analyse de données prescriptives utilisées pour recommander un article à un utilisateur.

2.3 Système de recommandation

À l'origine, les systèmes de recommandation étaient définis comme ≪ les personnes fournissent des recommandations en tant que entrées, que le système agrège ensuite et les dirige vers les destinataires appropriés ≫, par exemple utiliser les connaissances des experts comme entrées pour enrichir sa

capacité à recommander aux utilisateurs du plateforme selon les connaissances acquises. Cependant, maintenant le terme a une connotation plus large, décrivant un système qui produit des recommandations individualisées en sortie ou qui a pour effet de guider l'utilisateur de manière personnalisée vers des objets intéressants ou utiles dans un large espace d'options possibles [11].

Les systèmes de recommandation sont des systèmes de filtrage d'informations qui traitent le problème de la surcharge d'informations en filtrant un fragment d'informations vitales à partir d'une grande quantité d'informations générée dynamiquement en fonction des préférences, des intérêts ou du comportement observé de l'utilisateur. Le système de recommandation a la capacité de prédire si un utilisateur particulier préfèrerait un élément ou non en fonction du son profil et son historique [12].

Le système de recommandation est bénéfique aux fournisseurs de services et aux utilisateurs. Les systèmes de recommandation ont également prouvé qu'ils amélioraient le processus de décision et la qualité. Dans le cadre du commerce électronique, le système de recommandation aide à augmenter les revenus, pour le fait qu'ils sont des moyens efficaces de vendre plus de produits. Dans les bibliothèques scientifiques, le système de recommandation prend en charge les utilisateurs en leur permettant d'aller au-delà des recherches de catalogue. Par conséquent, la nécessité d'utiliser des techniques de recommandation efficaces et précises au sein d'un système qui fournira des recommandations pertinentes et fiables pour les utilisateurs ne saurait être surestimée.

Le système de recommandation produit généralement une liste de recommandations de l'une des deux manières suivantes : le filtrage collaboratif et le filtrage de contenu. Construire un modèle à partir du comportement passé d'un utilisateur (éléments précédemment achetés ou sélectionnés et notation numérique attribuée à ces éléments) ainsi que des décisions similaires prises par d'autres utilisateurs. Ce modèle est ensuite utilisé pour prédire des éléments (ou des évaluations pour des éléments) qui intéressent l'utilisateur. Les approches de filtrage basées sur le contenu utilisent une série de caractéristiques discrètes d'un élément, à savoir un profil basé sur l'historique d'achat de l'utilisateur, c'est-à-dire à l'aide du profil utilisateur pour recommander des articles. Le système de recommandation hybride est celui dans lequel le ou les systèmes recommandés sont combinés pour la recommandation. En outre, il existe plusieurs catégories du système de recommandation qui sont enrôlés ci-dessous :

Recommandation personnalisée : Elle implique la suggestion en ligne de données dans n'importe quel format qui est pertinent pour chaque utilisateur, basé sur le comportement implicite de l'utilisateur et les détails fournis.

(a) Recommandation basée sur la connaissance (recherche)

(b) Recommandation basée sur l'utilitaire

(c) Recommandation basée sur la démographie

(d) Recommandation basée sur le contenu

(e) Recommandation basée sur le filtrage collaborative

i. Basé sur la mémoire (basé sur l'utilisateur, basé sur l'article)

ii. Basé sur le modèle (techniques de regroupement, techniques d'association, réseau bayésien, réseaux de

(f) Recommandation hybride

Recommandation non personnalisée : Il s'agit d'un système de recommandation qui recommande des articles aux consommateurs en fonction de ce que les autres consommateurs ont dit sur le produit dans un contexte où les recommandations sont indépendantes du client, de sorte que tous les clients reçoivent la même recommandation.

2.3.1 Système de recommandation basé sur le contenu

Les RS basés sur le Contenu CBF ("Content based Filtering") analysent les descriptions des ressources pour identifier celles qui seront d'un intérêt particulier à l'utilisateur [13], c'est à dire que le système cherche à recommander des ressources semblables à celles que l'utilisateur apprécie. Par exemple, si l'utilisateur apprécie un livre, le système peut lui suggérer des livres du même auteur, ou du même genre. Ces RS nécessitent donc à la fois de dresser un profil de l'utilisateur et de représenter les ressources considérées d'une manière qui lui soit compréhensible.

Les PRS basés sur le Contenu fournissent donc leurs recommandations sur la base de la description des ressources, et d'un profil des intérêts de l'utilisateur, celui-ci ayant pu être entré directement par l'utilisateur, mais plus généralement au travers des retours que l'utilisateur fourni sur ces ressources [14]. Ce type de PRS est donc essentiellement adapté lorsque les ressources disposent d'informations textuelles riches, c'est à dire soit qu'il s'agisse de ressources de type intrinsèquement textuel (des

livres par exemple), soit disposant d'une description textuelle complète et si possible structurée. Ces profils peuvent prendre différentes formes, telles que [14] :

- Profils de mots-clefs, où le profil est un jeu de mots clefs le plus souvent pondérés.
- Profils de réseaux sémantiques, où le profil est un graphe pondéré où chaque nœud représente un concept représenté par un mot, ou un groupe de mots [15].
- Profils de concepts, où les sujets sont abstraits, et non rattachés à des mots ou groupes de mots spécifiques. Ils peuvent être représentés sous forme de graphes comme de vecteurs [16].

Les ressources sont représentées de la même manière, afin de pouvoir comparer leur similarité avec les profils utilisateurs établis. La méthode la plus courante pour pondérer ces profils est le tf*idf [123] (et ses variantes), où la fréquence d'apparition d'un terme dans une ressource (i.e. l'importance du terme dans la ressource) est mise en relation avec le nombre de ressources dans lequel ce terme apparaît (i.e. importance du terme dans l'ensemble des ressources, ce qui permet de limiter l'importance des termes très courants, porteurs de moins de sens).

Une fois obtenu le profil utilisateur et la représentation des ressources, il convient ensuite d'utiliser des algorithmes de classification pour prédire, pour chaque ressource, si l'utilisateur sera intéressé par cette ressource. Plusieurs types de classificateurs peuvent être utilisés, notamment des algorithmes classiques d'apprentissage automatique, tels que :

- Arbre de décision
- Arbre des forêts aléatoires
- Le K plus proches voisins
- Méthodes probabilistes (Naïve Bayes)
- Support à vecteurs machines
- Régression logistique

Ce type de RS est néanmoins peu efficace pour fournir des recommandations sur des ressources contenant peu d'informations textuelles ou des informations difficiles à discerner, comme par exemple l'humour présent dans des blagues. Dans certaines situations, l'utilisateur peut également accorder plus de valeur aux opinions des autres utilisateurs qu'aux informations structurées. Les RS basés sur le contenu souffrent aussi de la sur-spécialisation, qui les amène à rencontrer des difficultés à

recommander des ressources différentes de ce que l'utilisateur connaît déjà [17]. Ce type de RS risque donc à avoir une bonne sérendipité.

2.3.2 Système de recommandation basé sur le filtrage collaborative

Le filtrage collaboratif FC [18] est une technique de prédiction indépendante du domaine utilisée pour le contenu qui ne peut pas être décrit facilement et adéquatement par des méta-données telles que les formations. La technique de filtrage collaboratif fonctionne en construisant une matrice de préférence de base de données pour les articles par les utilisateurs. Il compare ensuite les utilisateurs ayant un intérêt et des préférences pertinentes en calculant les similitudes entre leurs profils pour faire des recommandations. Ces utilisateurs créent un groupe appelé voisinage. Un utilisateur obtient une recommandation pour les articles qu'il n'a pas notés auparavant mais qui ont déjà été positivement évalués par les utilisateurs de son quartier. Les recommandations produites par les FC peuvent être des prédictions ou des recommandations. La prédiction est une valeur numérique exprimant Rij, exprimant le score prévu de l'élément 'j' pour l'utilisateur 'i', recommandation qui est une liste des N éléments principaux que l'utilisateur aimera le plus. Le filtrage collaboratif peut être divisé en deux catégories :

- **Basé sur la mémoire :** Cette approche utilise les données d'évaluation des utilisateurs pour calculer la similarité entre les utilisateurs ou les éléments. Ceci est utilisé pour faire des recommandations. Cette première approche est utilisée dans de nombreux systèmes commerciaux. C'est efficace et facile à mettre en œuvre. Des exemples typiques de cette approche sont les recommandations de top N basées sur les FC et sur les items / utilisateurs. L'algorithme de recommandation N supérieur basé sur l'utilisateur utilise un modèle vectoriel utilisant la similarité pour identifier les k utilisateurs les plus similaires à un utilisateur actif. Après que les k utilisateurs les plus similaires sont trouvés, les matrices d'items utilisateur correspondantes sont agrégées pour identifier l'ensemble des éléments à recommander. Les avantages de cette approche comprennent : l'explicabilité des résultats, qui est un aspect important du système de recommandation, une création et une utilisation faciles, une facilitation facile des nouvelles données, l'indépendance du contenu des articles recommandés, une bonne mise à l'échelle des articles coréens. avec cette approche. Les performances diminuent lorsque peu des données

sont disponibles, ce qui se produit fréquemment avec les éléments liés au Web. Cela pourrait nuire à l'évolution de cette approche et créer des problèmes avec des grands ensembles de données.

- **Basé sur le modèle :** Dans cette approche, les modèles sont développés en utilisant différents algorithmes d'exploration de données et d'apprentissage automatique pour prédire l'évaluation par l'utilisateur des éléments non notés. Il existe de nombreux algorithmes CF basés sur des modèles tels que : Les réseaux bayésiens, les modèles de regroupement, les modèles sémantiques latents tels que la décomposition de la valeur singulière, l'analyse sémantique latente probabiliste, le facteur multiplicatif multiple, etc.

  Dans ce modèle, des méthodes comme la décomposition de la valeur singulière, l'analyse des composantes principales, connues sous le nom de modèles de facteurs latents, compressent la matrice de l'item utilisateur en une représentation de faible dimension en termes de facteurs latents. Un avantage de l'utilisation de cette approche est qu'au lieu d'avoir une matrice de haute dimension contenant un nombre abondant de valeurs manquantes, il s'agira de traiter une matrice beaucoup plus petite dans l'espace de dimension inférieure. Une présentation réduite pourrait être utilisée pour des algorithmes de voisinage basés sur l'utilisateur ou sur l'article. Elle gère mieux la parcimonie de la matrice originale que celle basée sur la mémoire.

2.3.3 Système de recommandation basé sur la connaissance

Ces systèmes cherchent à trouver des ressources correspondant aux spécifications ou aux besoins de l'utilisateur. La plupart de ces systèmes fonctionnent sur Case Based Reasoning (Raisonnement basé sur les cas) (CBR) ("case-based reasoning") [19], bien qu'il existe des solutions à partir de décision quantitative [20] ou d'arbres de décision. La grande force des RS de ce type est qu'ils ne nécessitent pas d'informations sur les utilisateurs, et ont donc l'avantage de ne pas souffrir de plusieurs des défauts des RS tels que le problème du démarrage à froid, la confidentialité ou le manque de données. Néanmoins, ils requièrent plus d'implication de l'utilisateur qui doit exprimer précisément sa recherche, et ils nécessitent des informations très complètes et structurées sur le domaine qu'ils traitent.

2.3.4 Système de recommandation basé sur la communauté

Ces systèmes fonctionnent à partir des préférences des amis des utilisateurs. Des études ont en effet montré que les utilisateurs préfèrent généralement les recommandations provenant de leurs amis à celles proposées par les RS [21]. La popularité croissante des réseaux sociaux de toutes sortes a été l'occasion idéale pour développer des RS basés sur la communauté, puisqu'ils apportent le plus souvent des moyens pour les utilisateurs de définir leurs relations avec les autres, ces relations permettant de déterminer la confiance qu'un utilisateur porte envers les autres. Cette relation de confiance entre utilisateurs étant corrélée avec la notion de similarité, elle est donc toute indiquée pour servir de base à des RS. Ce type d'approche est la plus récente, et la littérature ne s'accorde pas entièrement sur son efficacité par rapport aux autres types de RS. Néanmoins, il apparait qu'elle est moins sensible aux problèmes de démarrage à froid, et fournit des résultats plus homogènes pour différents types d'utilisateurs que les approches basées sur du CF.

2.3.5 Système de recommandation basé sur la démographie

Les systèmes sont basés sur le profil démographique de l'utilisateur, c'est à dire sa nationalité, son âge, ou sa langue. Un exemple très courant d'adaptation dynamique d'un site web selon le profil démographique de l'utilisateur est par exemple l'internationalisation, qui adapte automatiquement le contenu à la langue de l'utilisateur. Il existe néanmoins relativement peu de recherches spécifiques aux RS dans ce domaine [22][23].

2.3.6 Système de recommandation hybride

Chacune des techniques vues dans les sections précédentes présentent des avantages et des limites, ainsi qu'un fonctionnement optimal dans des situations différentes. L'idée des RS Hybrides consiste à combiner plusieurs de ces techniques afin de profiter de leurs synergies [24]. Par exemple, un système de CF rencontrant des difficultés à évaluer les nouvelles ressources (qui n'ont donc pas encore de notations, problème du démarrage à froid pour les nouvelles ressources) pourra être pallié par un système basé sur le contenu qui lui ne souffre pas de ce problème (puisque les nouvelles ressources sont évaluées sur la base de leur contenu).

Le terme de RS Hybride est utilisé pour n'importe quel RS combinant plusieurs techniques de recommandations dans le but d'améliorer ses performances. Ils peuvent combiner plusieurs techniques de même type [19], néanmoins, la plupart combinent des techniques possédant des sources d'information différentes (voir la figure 2 de la section 2.4.1 pour le détail des sources d'information des RS). La raison la plus courante pour la mise en place d'un RS hybride est la volonté de résoudre le problème du démarrage à froid.

Les RS Hybrides peuvent être classés dans sept catégories différentes, en fonction de la manière dont ils combinent les techniques qu'ils utilisent [25] :

- **Pondéré :** les résultats des différentes sources sont combinés numériquement.
- **Sélection :** le système choisit parmi les méthodes de recommandation et applique celle choisie.
- **Mixé :** les résultats provenant de différentes sources sont présentés ensemble.
- **Combinaison de caractéristiques :** des caractéristiques provenant de différentes sources d'information sont combinées ensemble et présentées à un seul algorithme.
- **Augmentation de la caractéristique :** une technique de recommandation est utilisée pour produire une série de caractéristiques, qui sont alors utilisées en entrée de la technique suivante.
- **Cascade :** les méthodes de recommandation sont hiérarchisées, celles de plus basse priorité servant à départager les égalités.
- **Méta :** une technique de recommandation est utilisée pour produire un modèle, qui est alors utilisé par la technique suivante.

Burke [26] a fait une étude comparative des différents types de PRS hybrides possibles, et de leurs performances. Il semblerait que, du moins pour des domaines semblables à celui étudié dans ses travaux, les méthodes d'hybridation les plus efficaces soient l'augmentation de la caractéristique et les RS en cascade. De plus, il montre également que, si les résultats des algorithmes basés sur le contenu ou sur la connaissance sont plutôt faibles lorsqu'ils sont utilisés seuls, ils viennent fortement améliorer les performances d'algorithmes plus performants (notamment basés sur le CF) lorsqu'ils sont combinés avec ceux-ci. Plusieurs études mettent en avant le gain de qualité général des RS Hybrides par rapport aux RS basés sur une seule technique, en particulier dans le cas du manque de données.

Néanmoins, il convient de faire attention aux contraintes matérielles lorsque on met en place une

méthode hybride. Les utilisateurs attendent généralement des réponses rapides de la part des RS et la scalabilité des algorithmes de recommandation est un problème d'autant plus présent pour les systèmes hybrides qu'ils impliquent plus de calculs de par leur nature.

2.4 Critiques sur les systèmes de recommandation existants

Les problèmes rencontrés par les systèmes de recommandation peuvent être décrits comme suit [27] :

- **Collecte de données :** Les données utilisées par les moteurs de recommandation peuvent être catégorisées en données explicites et implicites. Explicite est l'ensemble des données que l'utilisateur alimente lui-même dans le système. La collecte de données explicites ne doit pas être intrusive ou prendre beaucoup de temps. Source de données implicite est telle que les données de transaction. Les données implicites doivent être analysées avant de pouvoir être utilisées pour décrire les critères des utilisateurs ou les notations de ces derniers sur les différents articles.
- **Démarrage à froid :** Le problème de démarrage à froid se produit lorsque les données d'évaluation disponibles sont trop peu dans l'état initial. Le système de recommandation manque alors de données pour produire des recommandations appropriées. Ce problème se produit principalement dans les modèles d'apprentissage.
- **Stabilité Vs Plasticité :** L'inverse du problème du démarrage à froid est la stabilité par rapport à la plasticité. Lorsque les consommateurs ont évalué un si grand nombre d'éléments, leurs préférences dans les profils d'utilisateurs établis sont difficiles à changer.
- **Sparsity :** Dans la plupart des cas d'utilisation des systèmes de recommandation, le nombre d'évaluations déjà obtenu est très faible par rapport au nombre d'évaluations à prévoir. Mais les techniques de filtrage collaboratif mettent l'accent sur le chevauchement des notations et ont des difficultés lorsque l'espace de notation est clairsemé (peu d'utilisateurs ont évalué les éléments similaires). La rareté dans la matrice d'évaluation des items utilisateur dégrade la qualité des recommandations.
- **Confidentialité :** la confidentialité est un problème important dans les systèmes de recom-

mandation. Pour fournir des recommandations personnalisées, les systèmes de recommandation doivent savoir quelque chose concernant les utilisateurs. En fait, plus le système est riche d'informations, plus la recommandation peut être précise. Les utilisateurs sont préoccupés par les informations collectées, comment elles sont utilisées et si elles sont stockées. Cette confidentialité affecte à la fois la collecte de données explicites et implicites. En ce qui concerne les données explicites, les utilisateurs ne sont pas intéressés à divulguer des informations sur eux-mêmes et leurs intérêts. Si les questionnaires deviennent trop personnels, l'utilisateur peut donner de fausses informations afin de protéger sa vie privée.

2.5 Méthodes d'évaluation des systèmes de recommandation

Les mesures d'évaluation pour les systèmes de recommandation sont séparées en trois catégories :

- **Mesures l'exactitude de prédiction :** Ces mesures évaluent la proximité du système recommandé pour prédire les valeurs réelles d'évaluation ou d'utilité.
- **Mesures l'exactitude de classification :** Ces mesures évaluent la fréquence avec laquelle le système de recommandation prend des décisions correctes ou incorrectes concernant les articles.
- **Mesures l'exactitude de classement :** Ces mesures évaluent la justesse de classement des articles par le système de recommandation.

2.6 Solution proposée

Dans cette section, nous allons présenter la solution proposée afin d'atteindre nos objectifs en premier lieu et d'éviter les problèmes que nous avons détaillés dans la section précédente. Dans la suite, nous avons choisi l'approche hybride du système de recommandation.

2.6.1 La collection des données

Pour choisir un modèle d'apprentissage automatique permettant de prédire les articles préférés à l'utilisateur, nous avons besoin d'une base de données sous la forme d'une matrice dont les colonnes sont les articles, les lignes sont les utilisateurs et l'intersection est le rating.

Dans la suite de cette section, pour bien expliquer la méthode utilisée, on prend l'exemple de la recommandation des sessions aux apprenants et aux entreprises.

Notre mesure est donc le rating (ou notation en français) qui peut être calculé explicitement ou implicitement.

- **Explicite** : La plateforme donne une interface d'évaluation de la session sous forme d'étoiles de 1 à 5.
- **Implicite :** Les paramètres dans le cas de la recommandation des sessions aux apprenants et entreprises sont les visites des formations, les recherches, les réservations, les annulations de réservations, les profils, les projets des formations créés, et chacun de ces critères a son propre poids. Le diagramme suivant illustre mieux ces critères.

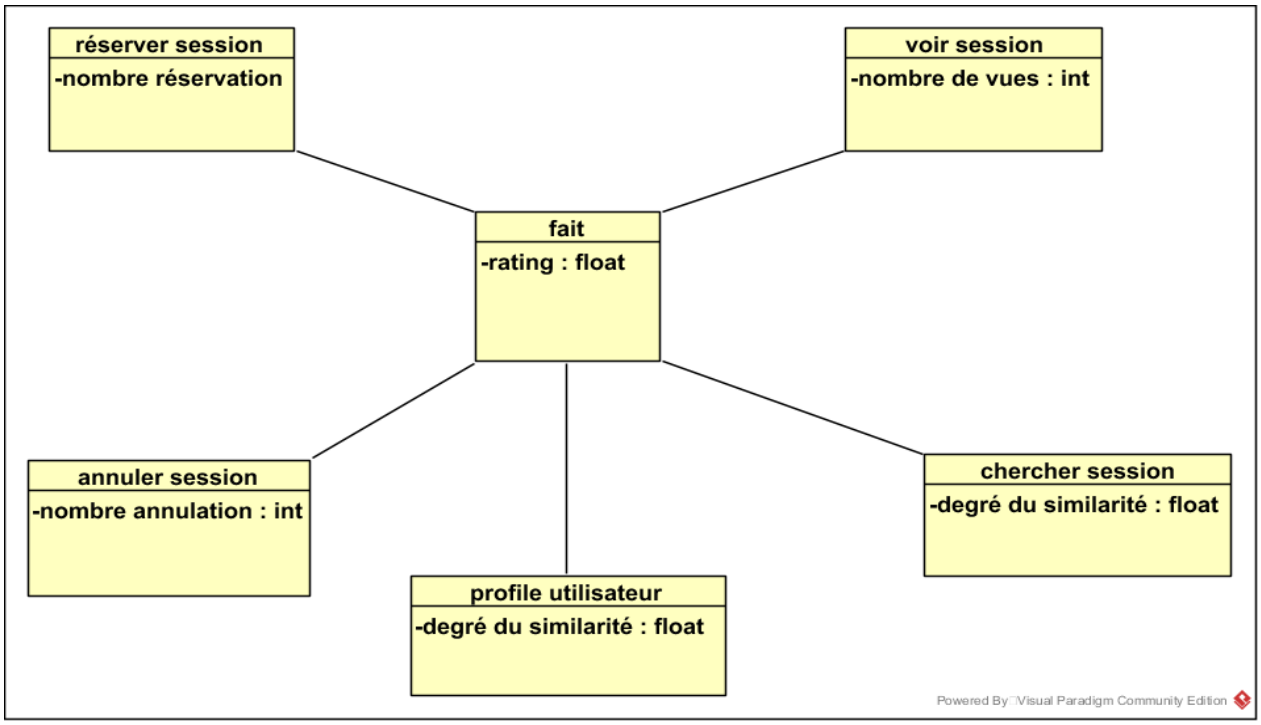

FIGURE 2.1 – Diagramme en étoile pour la mesure du rating

Donc comme l'indique ce diagramme, la fonction du rating s'exprime avec la formule suivante :

$$f(x, y, z, t, w) = \alpha x + \sigma y + \gamma z + \theta t + \Omega w$$

avec $\alpha < \sigma < \theta$ et $\sigma < |\Omega|$ et,

x : représente le nombre de vues d'un utilisateur u_1 pour une session donnée s_1.

y : représente le degré de similarité du mot dans la recherche par u_1 par rapport s_1.

z : représente le nombre de réservations de u_1 pour s_1.

t : représente le nombre d'annulation de réservations de u_1 pour s_1.

w : représente le degré de similarité du profil de u_1 par rapport s_1.

Enfin, nous aurons comme résultat une matrice d'usage sous la forme suivante :

$$M = \begin{pmatrix} & s1 & s2 & s3 & s4 & s5 & s6 \\ u1 & 2 & 3.3 & 1 & 0 & 0 & 0 \\ u2 & 0 & 33 & 0 & 0 & 0 & 0 \\ u3 & 2.5 & 0 & 4.4 & 5 & 0 & 0 \\ u4 & 0 & 5 & 0 & 0 & 4.6 & 0 \\ u5 & 0 & 0 & 0 & 1 & 6.6 & 7 \end{pmatrix}$$

- $u_1 ... u_5$: les utilisateurs qui sont les apprenants et les entreprises.
- $s_1 ... s_6$: les sessions dans la plateforme.
- $M(i, j)$: l'évaluation ou le rating du l'utilisateur $\underset{i}{u}$ sur la session $\underset{j}{s}$.

2.6.2 Résolution des problèmes

- **Démarrage à froid et la stabilité :** Nous avons choisi l'approche hybride du RS, qui combine entre l'approche du BCF pour éviter le démarrage froid et l'approche du CF pour résoudre la stabilité et recommander en temps réel avec les échanges des utilisateurs.
- **Sparcity :** Sparcity est le taux des éléments nuls dans la matrice. Par exemple soit la matrice suivante :

$$M = \begin{pmatrix} 11 & 22 & 0 & 0 & 0 & 0 & 0 \\ 0 & 22 & 33 & 0 & 0 & 0 & 0 \\ 0 & 0 & 33 & 44 & 0 & 0 & 0 \\ 0 & 0 & 0 & 44 & 55 & 0 & 0 \\ 0 & 0 & 0 & 0 & 55 & 66 & 0 \\ 0 & 0 & 0 & 0 & 0 & 66 & 77 \end{pmatrix} \quad Sparcity(M) = 26/42 = 0.74$$

Donc pour éviter ce problème, nous avons calculé le degré de similarité entre les documents avec des algorithmes du **Text Mining**. Par exemple : la colonne $M(0,2) = 0$, supposons que la similarité $S(M(0,2), M(0,0)) = 0.4$ et $S(M(0,2), M(0,1)) = 0.3$, alors $M(0,2) = 0.4M(0,0) + 0.3M(0,1) = 11$. Donc pour généraliser, notre formule s'écrira sous la forme :

$$M(i,j) = \sum_{k,l} M(k,l) * S(M(i,j), M(k,l)) \quad \forall k, l \in N$$

// $M(k,l) \neq 0$

- **Le problème de confidentialité :** Chaque utilisateur a ses informations personnelles qu'il souhaite protéger. Donc nous n'avons pas utilisé les cookies pour le problème de confidentialité des données, seul les informations développées dans la plateforme seront exploitées. De plus, si l'utilisateur désactive les cookies du navigateur on ne peut plus avoir l'historique. C'est pourquoi, nous développons nos propres fonctions et évènements pour enregistrer les navigations, les recherches, les transactions...

Conclusion

Dans ce chapitre, nous avons spécifié les démarches pour mesurer les similarités des documents, les approches de RS, et enfin notre solution proposée. Dans le chapitre suivant, nous allons identifier les acteurs. Et par la suite, nous allons dégager les besoins fonctionnels et non fonctionnels. Enfin, nous donnerons les diagrammes de cas d'utilisations pour mieux expliquer le processus de recommandation.

Chapitre 3

Analyse et Spécification des Besoins

Introduction

Dans ce chapitre nous allons présenter en premier lieu la plateforme « Takwinland », et en second lieu, les acteurs et leurs privilèges associés. Ensuite, les besoins fonctionnels et non fonctionnels seront dégagés. Enfin, nous expliquerons le fonctionnement de notre RS par des diagrammes de cas d'utilisations.

3.1 Présentation de la plateforme

Takwinland est une plateforme qui met en relation tous les acteurs de l'écosystème de la formation professionnelle et de l'éducation. À travers un réseau (Market Network) rassemblant tous les intervenants directs ou indirects du secteur de la formation et de l'éducation, Takwinland cherche avant tout à :

- Démocratiser la formation professionnelle.
- Améliorer la qualité globale des formations offertes sur le marché.
- Faciliter l'accès aux formations pour les étudiants et les entreprises.
- Réguler le marché des formations professionnelles.

Sur cette plateforme il y a trois types d'articles (*items*) qui sont les formations, les sessions, les projets de formation :

- **La formation :** Contient la description, la catégorie et les programmes, l'audience, les prérequis du cour de la formation. Une formation peut avoir plusieurs sessions.
- **La session :** Est une formation avec un hôte (local de la formation) et date de début.
- **Le projet de formation :** Lorsque l'apprenant ou l'entreprise cherche à faire une formation personnalisée, il crée un projet de formation contenant la description, la catégorie, l'hôte de la formation et les critères du formateur.

3.2 Identification des acteurs

Notre solution est constituée de deux types d'acteurs : la machine (ou système) et l'homme que nous allons identifier.

- **Système de recommandation :** C'est le logiciel qui analyse les navigations de chaque utilisateur sur le plateforme et interprète ses évaluations et ses notations sur les différentes articles. Ensuite il va recommander une liste d'articles à chaque utilisateur.
- **Apprenant :** Individus cherchant ou voulant prendre des formations professionnelles.
- **Entreprise :** Entreprises cherchant des formations professionnelles pour ses employés qui peuvent être des apprenants.
- **Formateur :** Individus, experts donnant des formations à titre personnel, sans matricule fiscale ni agrégation par l'état.
- **Organisme de formations :** Entreprises, cabinets de formations agréés par l'état qui fournissent des sessions des formations professionnelles pour des individus particuliers ou des entreprises.

3.3 Spécification des besoins

Une spécification des besoins est une description du système de logiciel à développer. Elle consiste à présenter les exigences fonctionnelles et non fonctionnelles. Autrement, elle décrit ce que le produit logiciel est censé faire et quoi ne pas faire, tout en faisant l'appel aux exigences suffisantes et nécessaires pour le développement du projet. Elle aide principalement à décrire la portée du travail et à fournir aux concepteurs de logiciels une forme de référence.

3.3.1 Besoins fonctionnels

La spécification des exigences fonctionnelles du projet est principalement catégorisée en fonction des exigences de l'utilisateur, des exigences de sécurité et de l'exigence de l'appareil, chacune étant expliquée ci-dessous :

- Permet :
  - En tant qu'apprenant ou entreprise de visualiser la recommandation des sessions et les formations qui peuvent être respectivement réserver ou abonner.
  - En tant que formateur ou organisme de formations de visualiser les propositions des formations, des sessions et des projets de formations qui peuvent être respectivement créer ou postuler.
- Le système doit s'adapter avec les mises à jour dans le profil et l'historique de l'utilisateur au moins chaque période.
- le système doit être lancé sur le navigateur Web.

3.3.2 Besoins non fonctionnels

Les exigences non fonctionnelles du système peuvent être résumées comme suit :

- **Performance :** Le système doit avoir des résultats pertinents, précis et fiables.
- **Confidentialité :** Le système doit protéger les données personnelles des utilisateurs.
- **Temps réel :** Le système doit recommander des articles aux utilisateurs en temps réel.
- **Disponibilité :** Le système doit être disponible pour l'utilisateur à tout moment lorsqu'il est connecté.
- **Récupération :** En cas d'indisponibilité du serveur, le système doit pouvoir récupérer et éviter toute perte de données ou redondance.
- **Aptitude à la maintenance :** Le code doit être lisible, organisé pour offrir la possibilité des prochaines maintenances. Notre RS doit être évolutive afin de s'adapter aux prochains besoins des différents acteurs.
- **Ergonomie :** Le RS doit offrir une interface conviviale, simple et facile à utiliser.

3.4 Diagrammes de cas d'utilisation

3.4.1 Diagramme de cas d'utilisation : Recommandation des sessions aux apprenants et entreprises

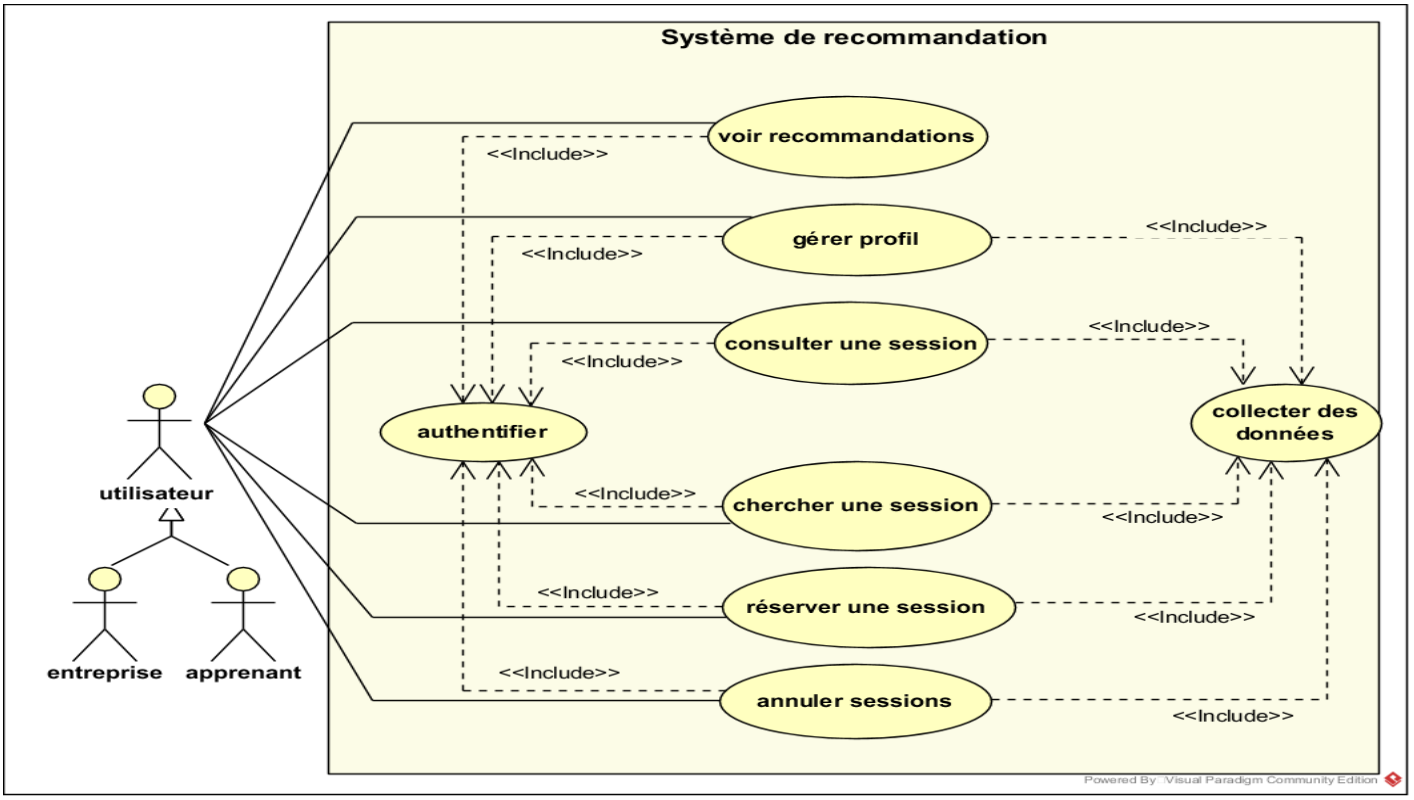

FIGURE 3.1 – Diagramme de cas d'utilisation : recommandation des sessions aux apprenants et entreprises

Cas d'utilisation	Recommandation des sessions aux apprenants et entreprises.
Acteurs	Apprenant et entreprise.
Pré-conditions	L'utilisateur doit avoir un compte sur la plateforme au début et il doit être connecté.
Post-conditions	Des sessions recommandées seront visualisées sur le plateforme pour chaque utilisateur.
Scénario nominal	• L'utilisateur peut faire un ensemble des actions suivantes (voir des sessions, chercher sur des sessions, réserver des places dans des sessions ou bien annuler ses réservations). • En parallèle, il gère son profil. • Le système de recommandation collecte l'historique des navigations de l'utilisateur et le continu du son profil s'il n'a pas d' historique. • Le système crée un modèle d'apprentissage automatique. • Le système calcule la performance du modèle afin de choisir le meilleur modèle. • Le système entraine le modèle chaque deux semaines. • Enfin, le système prédit les sessions recommandées à l'utilisateur à base des données collectées.

3.4.2 Diagramme de cas d'utilisation : Recommandation des formations aux apprenants et entreprises

Ce cas d'utilisation est fait par analogie au cas d'utilisation précédent juste nous changeons ≪ réserver une session ≫ par ≪ abonner en une formation ≫ et ≪ demander le devis d'une formation ≫

3.4.3 Diagramme de cas d'utilisation : Recommandation des formations aux formateurs et organismes de formations

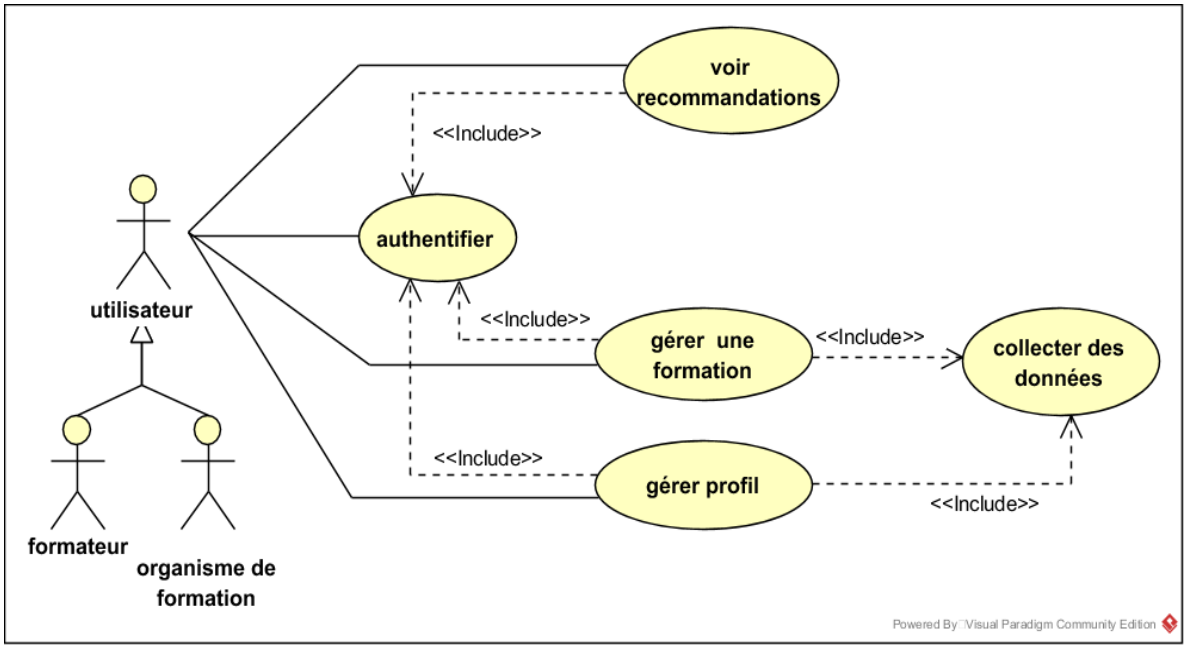

FIGURE 3.2 – Diagramme de cas d'utilisation : recommandation des formations aux formateurs et organismes de formations

Cas d'utilisation	Recommandation des formations aux formateurs et organismes de formation.
Acteurs	Formateur et Organisme de formations.
Pré-conditions	Chaque utilisateur doit avoir un compte sur la plateforme et doit être connecté.
Post-conditions	Des formations recommandées seront visualisées sur le plateforme à l'utilisateur.
Scénario nominal	• L'utilisateur peut faire un ensemble des actions suivants (créer, modifier ou bien supprimer des formations). • En parallèle, il gère son profil. • Le système de recommandation collecte les historiques des apprenants et des entreprises pour extraire ses besoins et le continu du son profil s'ils n'ont pas d'historique. • Le système crée un modèle d'apprentissage automatique. • Le système calcule la performance du modèle afin de choisir le meilleur modèle. • Le système entraine le modèle chaque deux semaines. • Enfin, le système prédit les formations recommandées à l'utilisateur à base des données collectées..

3.4.4 Diagramme de cas d'utilisation : Recommandation des sessions aux formateurs et organismes de formations

Ce cas d'utilisation est fait par analogie au cas d'utilisation précédent juste nous changeons « gérer une formation » par « gérer une session ».

3.4.5 Diagramme de cas d'utilisation : Recommandation des projets de formation aux formateurs et organismes de formations

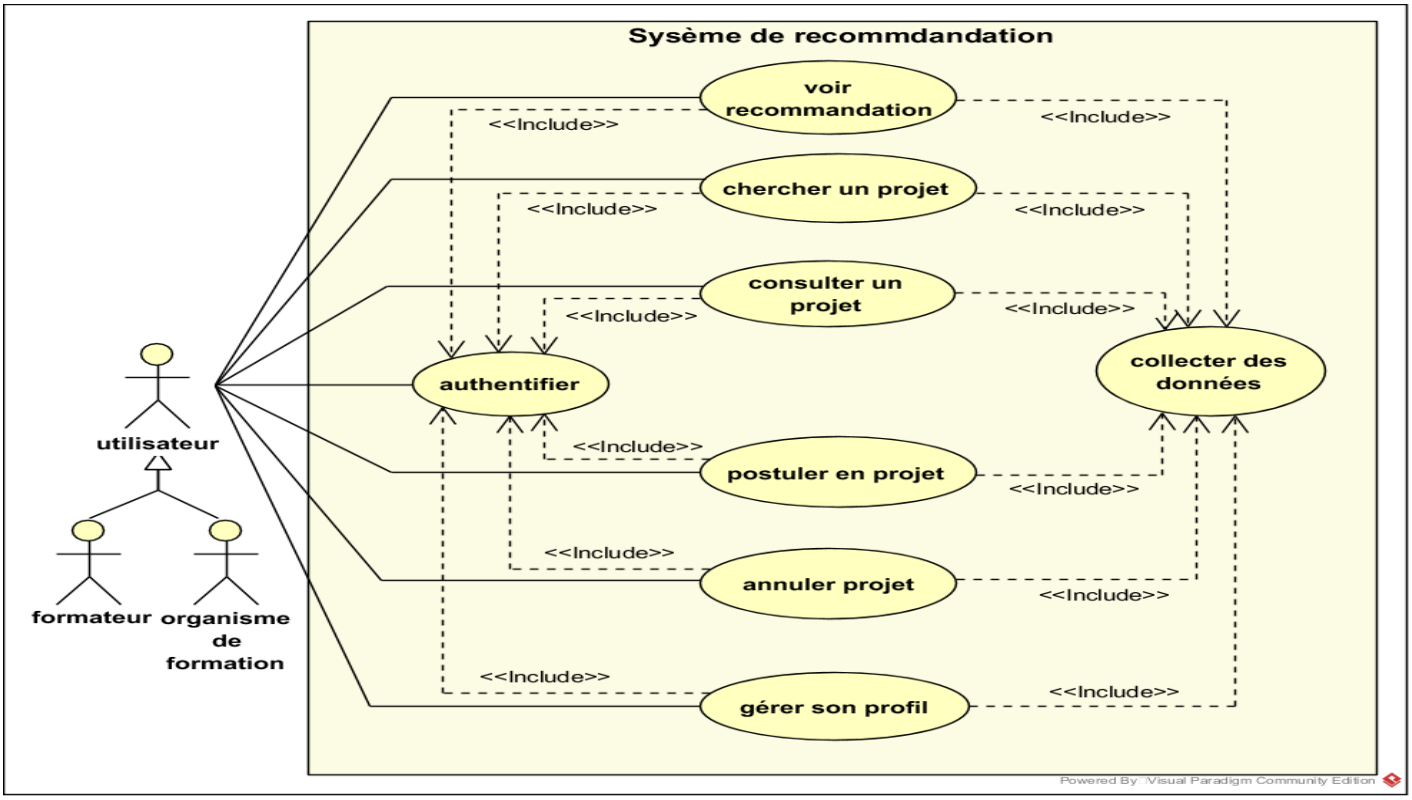

FIGURE 3.3 – Diagramme de cas d'utilisation : recommandation des projets de formation aux formateurs et organismes de formations

Cas d'utilisation	Recommandation des projets de formations pour le formateur et l'organisme de formation.
Acteurs	Formateur et Organisme de formations.
Pré-conditions	Chaque utilisateur doit avoir un compte sur la plateforme et doit être connecté.
Post-conditions	Des projets de formations recommandées seront visualisées sur la plateforme à l'utilisateur.
Scénario nominal	• L'utilisateur peut faire un ensemble des actions suivantes (voir des projets, chercher sur des projets, postuler dans des projets ou bien annuler ses postulations). • En parallèle, il gère son profil. • Le système de recommandation collecte l'historique des actions de l'utilisateur et le continu du son profil s'il n'a pas d'historique. • Le système crée un modèle d'apprentissage automatique. • Le système calcule la performance du modèle afin de choisir le meilleur modèle. • Le système entraine le modèle chaque deux semaines. • Enfin, le système prédit les projets recommandés à l'utilisateur à base des données collectées.

3.4.6 Diagrammes de cas d'utilisations détaillés

3.4.6.1 Diagramme de cas d'utilisation : chercher une session

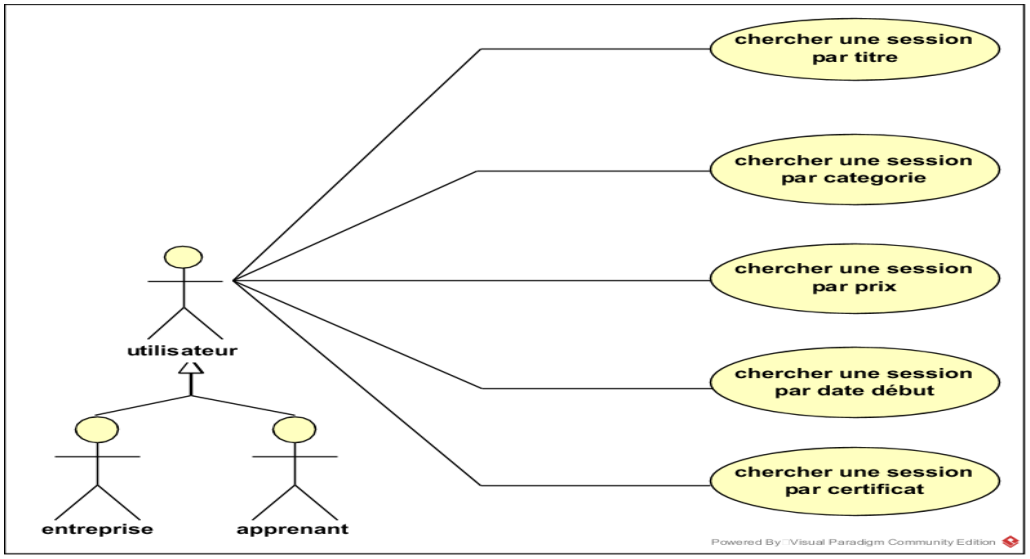

FIGURE 3.4 – Diagramme de cas d'utilisation : chercher une session

Cas d'utilisation	Chercher une session.
Acteurs	Apprenant et entreprise.
Pré-conditions	Chaque utilisateur doit avoir un compte sur la plateforme et doit être connecté.
Post-conditions	Des sessions seront sélectionnées et affichées à l'utilisateur.
Scénario nominal	• L'utilisateur cherche sur les sessions par le titre, la date de début, par catégorie, par le prix à partir d'un minimum x jusqu'à un maximum y, par certificat c.à.d des sessions certifiées ou non.

3.4.6.2 Diagramme de cas d'utilisation : gérer son profil

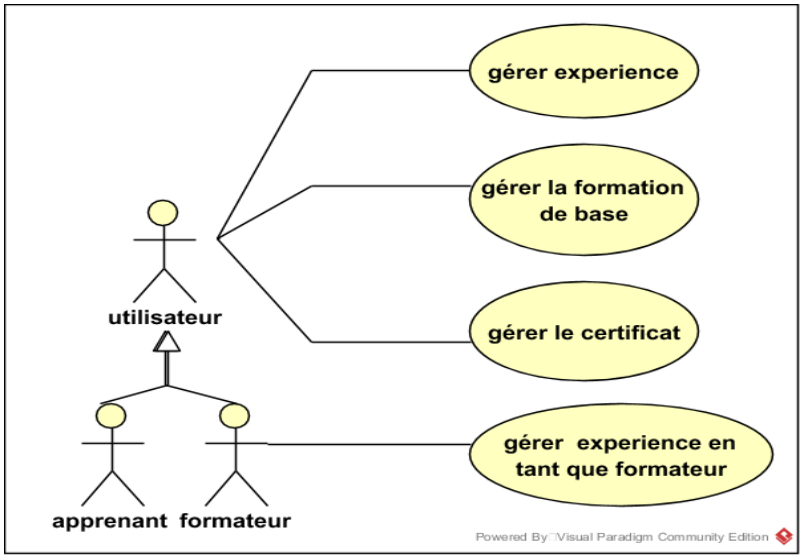

FIGURE 3.5 – Diagramme de cas d'utilisation : gérer son profil aux formateurs et apprenants

Cas d'utilisation	Gérer son profil.
Acteurs	Apprenant et formateur.
Pré-conditions	Chaque utilisateur doit avoir un compte sur la plateforme et doit être connecté.
Post-conditions	Le profil modifié sera affiché de nouveau.
Scénario nominal	L'utilisateur peut ajouter, modifier ou bien supprimer une expérience du travail, expérience en tant que formateur, la formation de base c.à.d ses études, certificats.

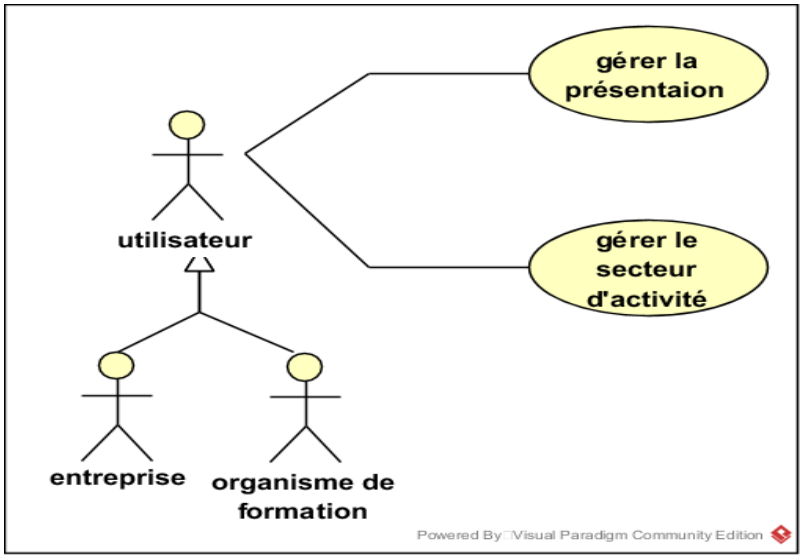

FIGURE 3.6 – Diagramme de cas d'utilisation : gérer son profil aux entreprises et organismes de formations

Cas d'utilisation	Gérer son profil.
Acteurs	Organisme de formations et entreprise.
Pré-conditions	Chaque utilisateur doit avoir un compte sur la plateforme et doit être connecté.
Post-conditions	Le profil modifié sera affiché de nouveau.
Scénario nominal	L'utilisateur peut ajouter, modifier ou bien supprimer une présentation de sa société et ses secteurs d'activité.

3.4.6.3 Diagramme de cas d'utilisation : gérer une formation

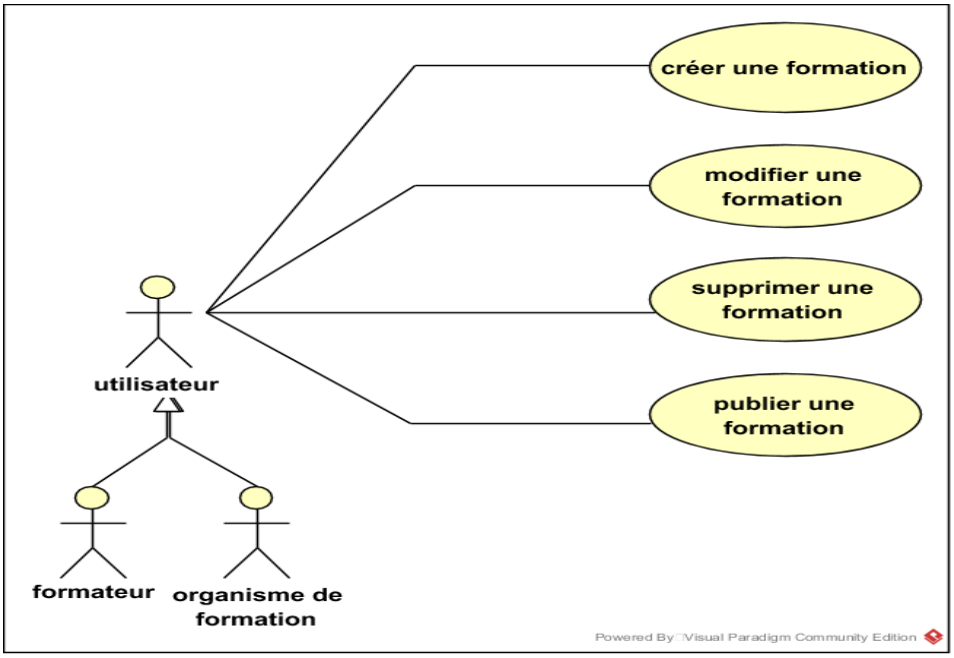

FIGURE 3.7 – Diagramme de cas d'utilisation : gérer une formation

Cas d'utilisation	Gérer une formation.
Acteurs	Formateur et arganisme de formations.
Pré-conditions	Chaque utilisateur doit avoir un compte sur la plateforme et doit être connecté.
Post-conditions	La formation modifiée sera affichée de nouveau.
Scénario nominal	L'utilisateur peut créer, modifier, supprimer ou bien dupliquer des formations.

Conclusion

Dans ce chapitre, nous avons identifié les acteurs de la plateforme, les besoins attendus, et nous avons présenté les diagrammes de cas d'utilisations. Dans le chapitre suivant nous allons présenter l'architecture de l'application et le système du recommandation, ensuite, la conception en détaillée avec le diagramme de classes, les diagrammes de séquences, d'activités et de composants.

Chapitre 4

Conception

Introduction

La réalisation d'un système de recommandation doit obligatoirement satisfaire à différentes exigences. Cela nécessiterait en premier lieu, une analyse clairement développée des éléments du système telle qu'elle fut présentée dans le chapitre précédent. La seconde exigence consiste en une conception clairement détaillée du fonctionnement du système du point de vue de la définition des différentes interactions entre les composantes du système, ceci fera l'objet du présent chapitre.

4.1 Langage de modélisation

Pour modéliser la conception de notre projet, nous avons eu recours au langage UML qui est devenu la norme utilisée par l'OMG et UML est avant tout un support de communication qui facilite la représentation et la compréhension et qui :

- Limite les ambiguïté par son aspect formel.
- Permet d'exprimer visuellement une solution objet par sa notation graphique.
- Est indépendant des langages de programmation, domaine d'application et au processus de développement.

UML n'est pas un simple outil de représentation, c'est un langage commun, universel, normalisé, simple, adapté à toutes les phases de développement et compatible avec toutes les techniques de réalisation. Il est indispensable pour la programmation objet unifiant les différentes approches et donnant une définition plus formelle.

4.2 Architecture globale d'application

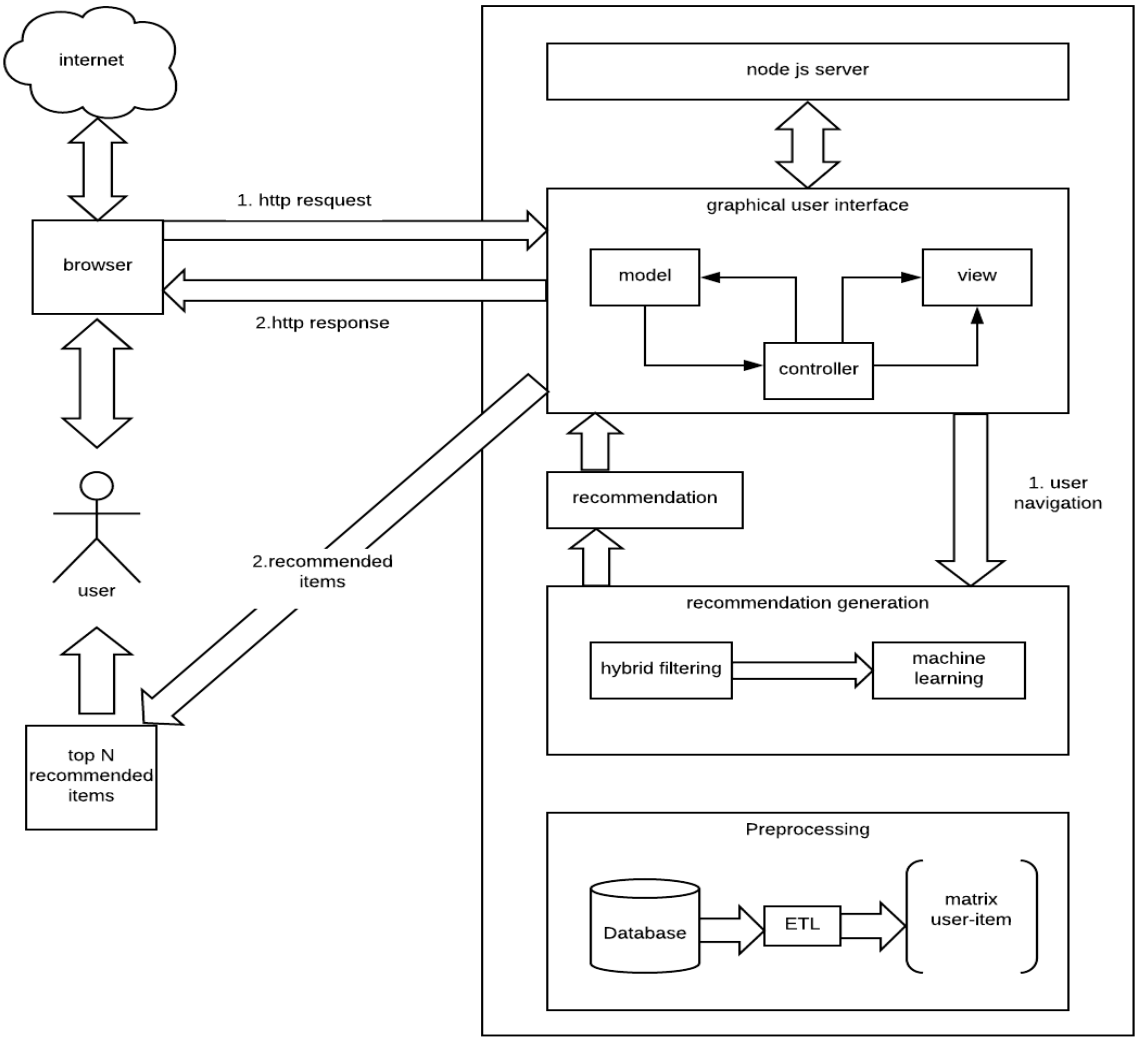

FIGURE 4.1 – Diagramme présentant l'architecture de l'application

Ce diagramme illustre l'architecture de notre application. Les outils du développement utilisés sont le serveur node js, l'architecture model-view-controller, base de données MongoDB, l'approche hybride du recommandation et le machine learning pour la prédiction. Donc l'utilisateur navigue, interagit avec les différents composantes sur le plateforme. En temps réel le système analyse son historique et son profil afin de lui proposer des articles qui peuvent être potentiellement intéressants pour lui.

4.3 Architecture globale du système de recommandation

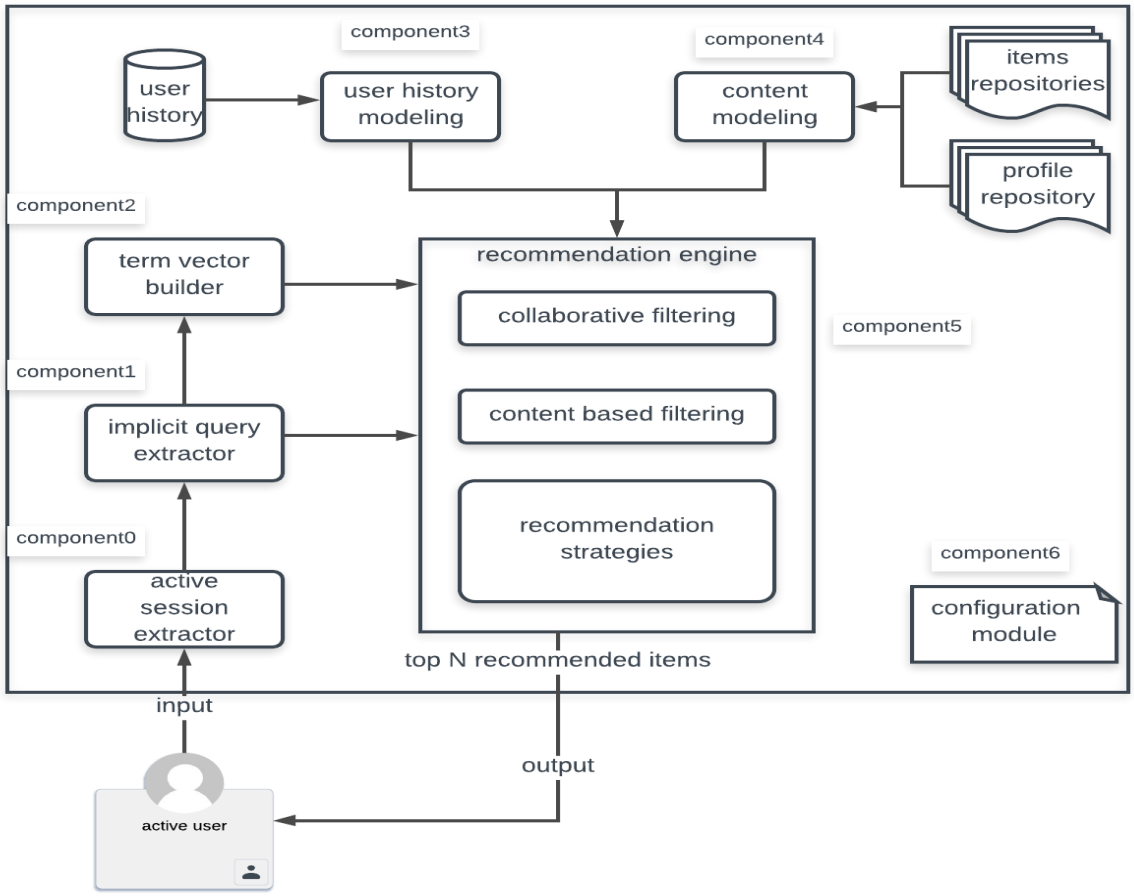

FIGURE 4.2 – Diagramme présentant l'architecture du système de recommandation hybride

Ce diagramme représente les différents composants dans notre système et les interactions entre eux. Parmi les composants on a :

- **active session extractor :** C'est le composant responsable de l'extraction des données développées dans la session courante.
- **implicite query extractor :** Il transforme les transactions courantes en notations et évaluations de l'utilisateur courant sur les articles avec des requêtes implicites.
- **term vector builder :** Il vectorise les résultats du composant 1 en deux vecteurs un pour les articles et l'autre pour l'évaluation de ces derniers.
- **user history modeling et content modeling :** Permet d'analyser les historiques et les profils pour extraire les critères et les utiliser en modèle du machine learning.
- **recommendation engine :** Il prend les données analysées et donne comme résultat des articles recommandés.
- **configuration module :** Gère la configuration du modèle d'apprentissage et fixe ses paramètres.

4.3.1 Diagramme de block

Un diagramme de définition de block permet d'exprimer la structure d'un système, d'un sous-système ou d'un composant. Les blocs peuvent représenter par des entités physiques ou logiques (à définir dans le méta-modèle). Ils sont décomposables et ils peuvent posséder des propriétés et un comportement. Ils permettent aussi de représenter les flux.

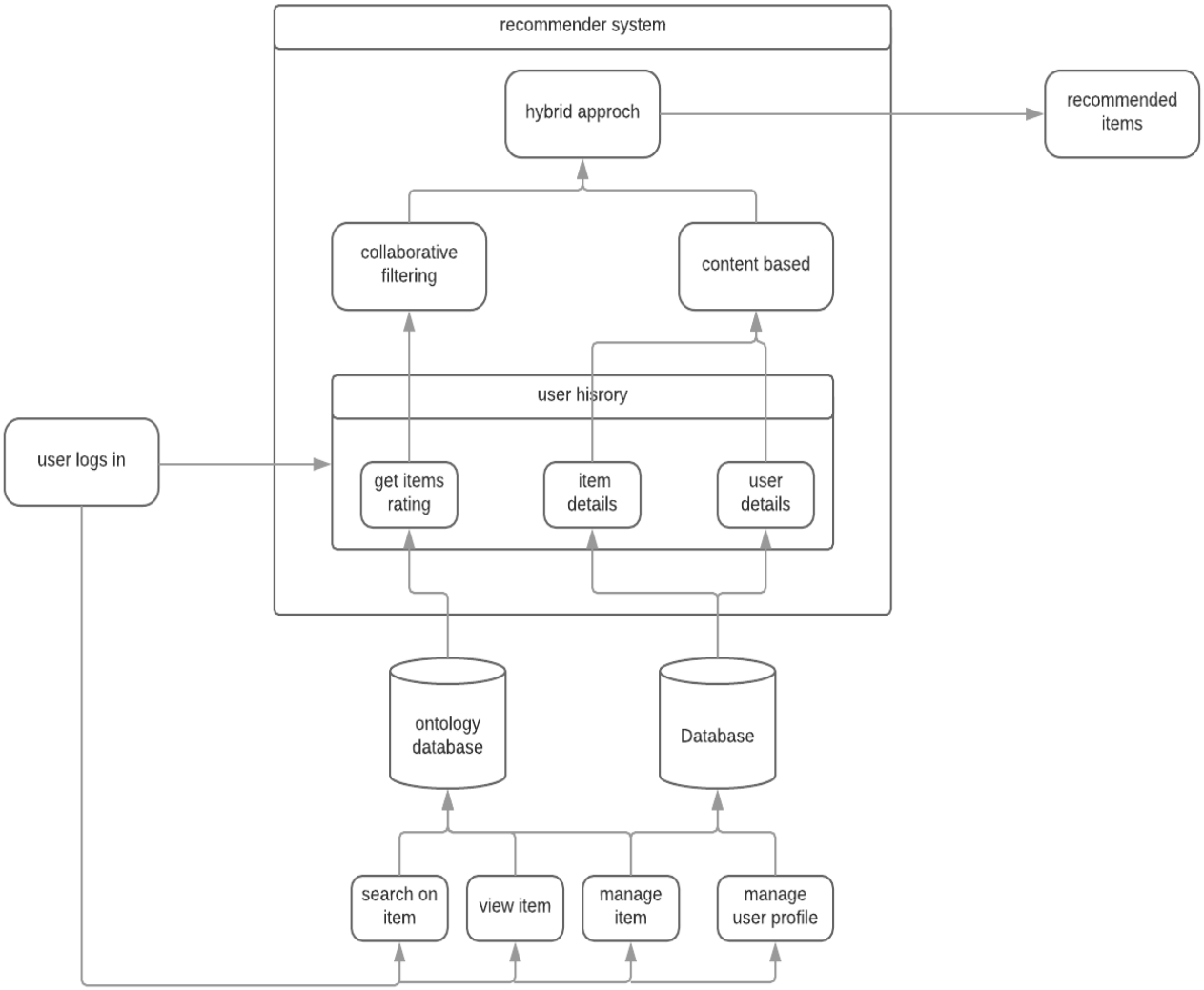

FIGURE 4.3 – Diagramme du block pour le système de recommandation

Ce diagramme sert à présenter le fonctionnement du système de recommandation avec l'approche proposée et les données exploitées.

4.3.2 Diagramme de flux de données

Le Data Flow Diagram DFD (en français, diagramme de flux de données) est un type de représentation graphique du flux de données à travers un système d'information. Cet outil est souvent utilisé comme étape préliminaire dans la conception d'un système afin de créer un aperçu de ce système d'information. De plus, il est également utilisé pour visualiser le traitement de données (structured design). Il montre quel type d'informations le système utilise (input ou output), d'où elles proviennent et où elles sont stockées. Cependant, il n'indique ni la temporalité des transmissions de données, ni l'ordre dans lequel les données circulent.

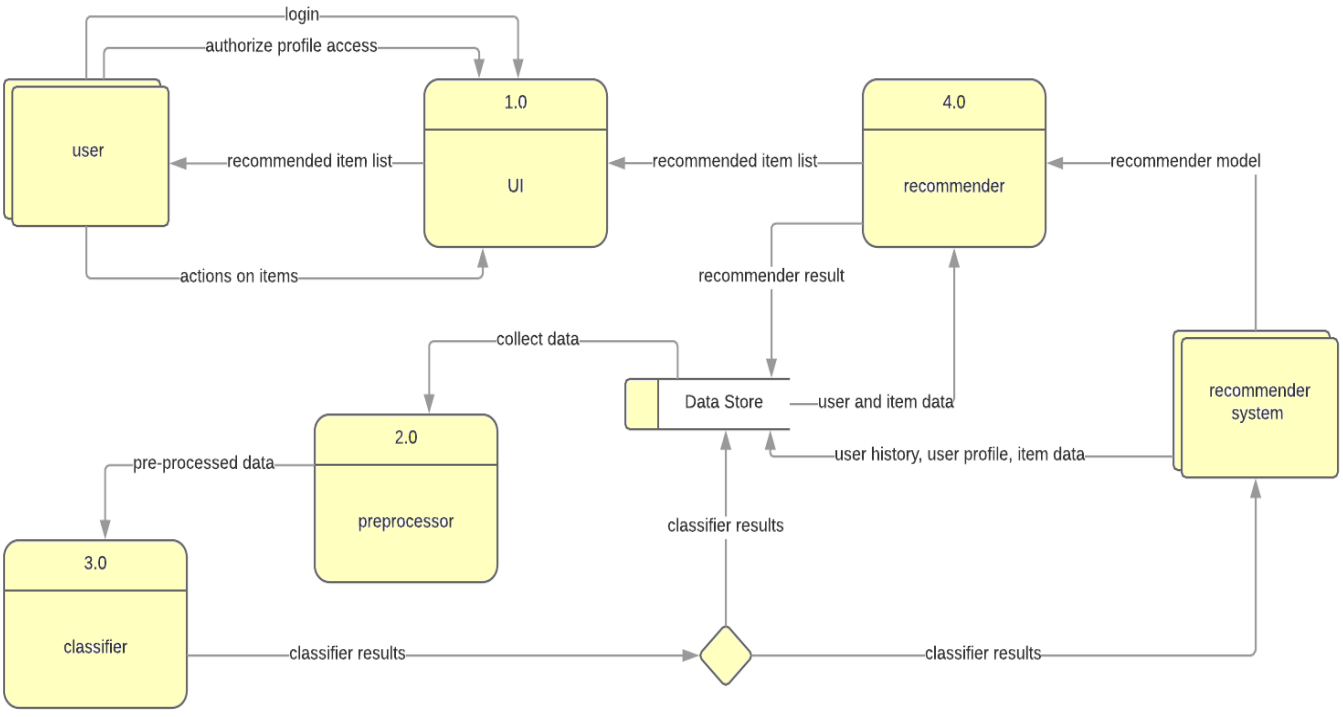

FIGURE 4.4 – Diagramme de Data Flow

4.4 Conception détaillée

4.4.1 Diagramme de classes

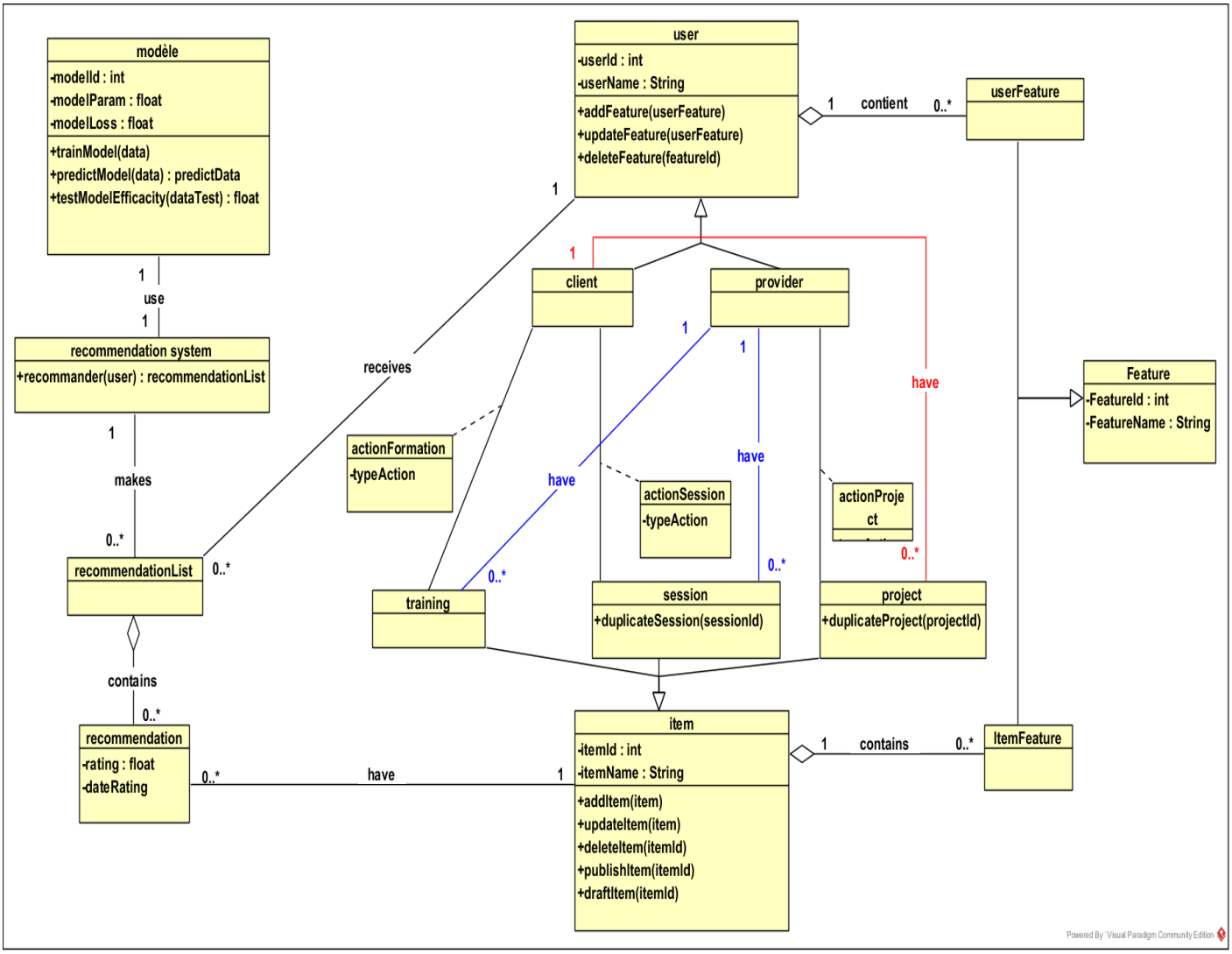

FIGURE 4.5 – Diagramme de classes

4.4.2 Diagramme de composants

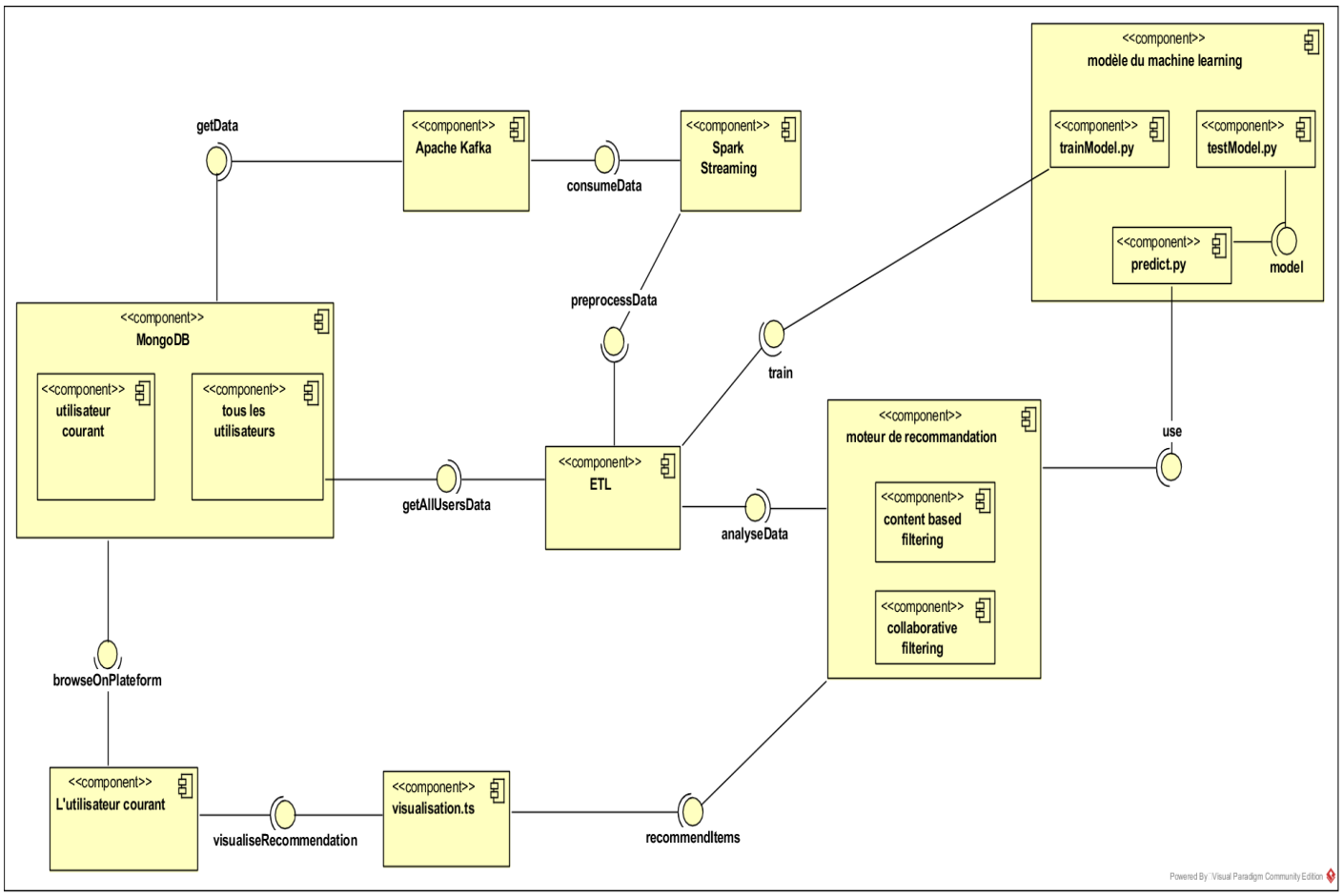

FIGURE 4.6 – Diagramme de composants

4.4.3 Diagrammes de séquences

4.4.3.1 Diagramme de séquence : Recommandation des sessions aux apprenants et entreprises

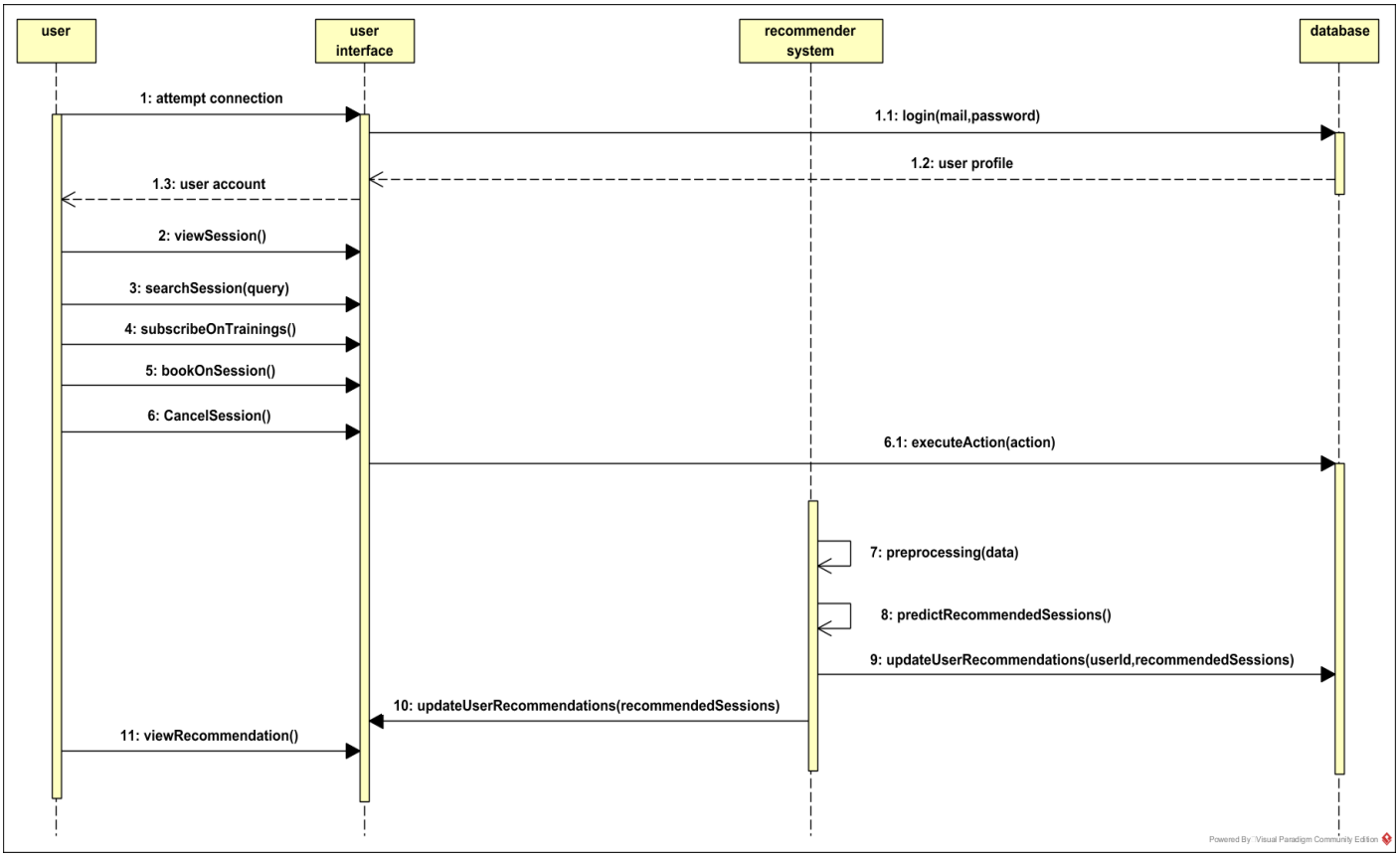

FIGURE 4.7 – Diagramme de séquence : recommandation des sessions aux apprenants et entreprises

4.4.3.2 Diagramme de séquence : Recommandation des formations aux apprenants et entreprises

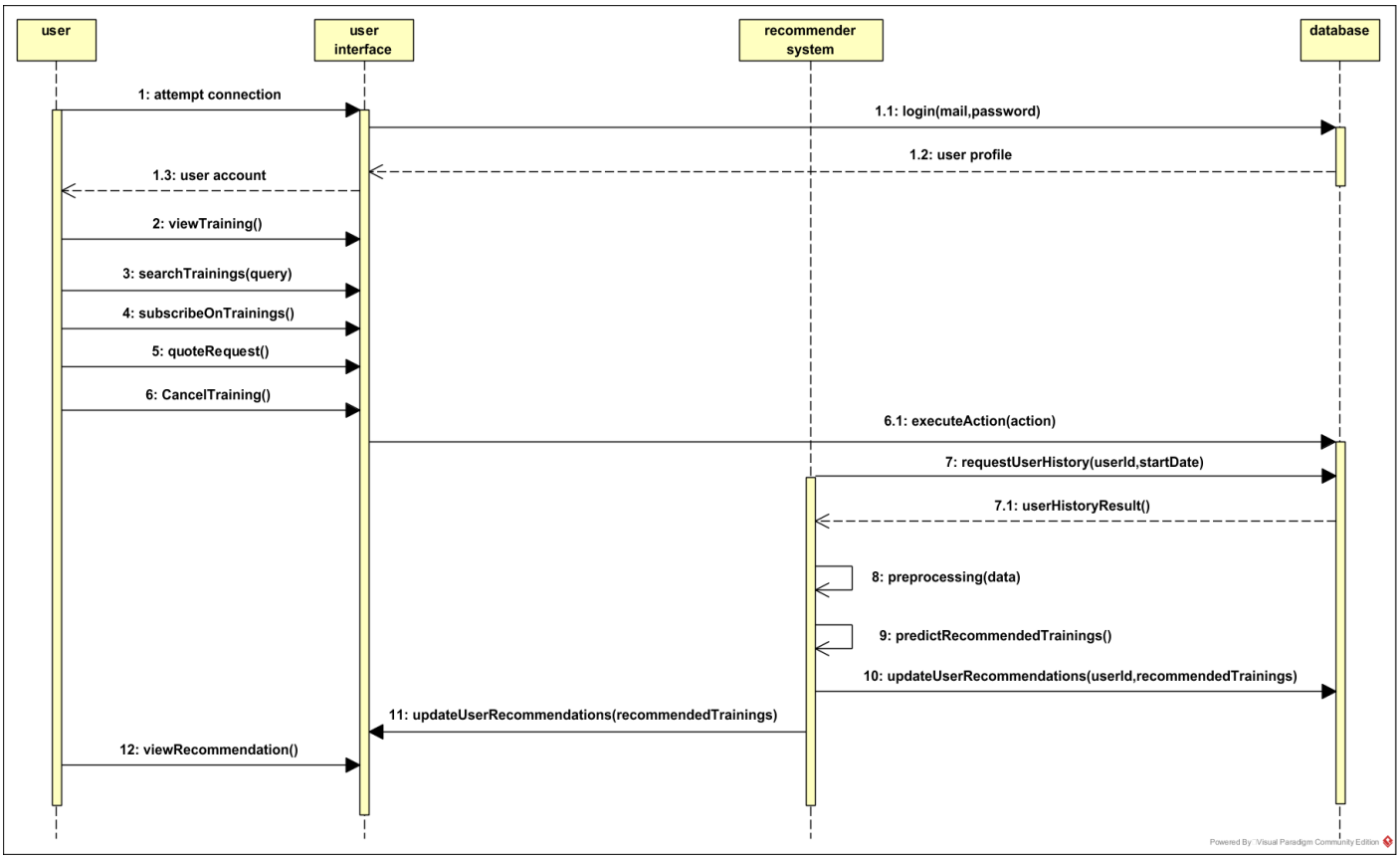

FIGURE 4.8 – Diagramme de séquence : recommandation des formations aux apprenants et entreprises

4.4.3.3 Diagramme de séquence : Recommandation des sessions aux formateurs et organismes de formations

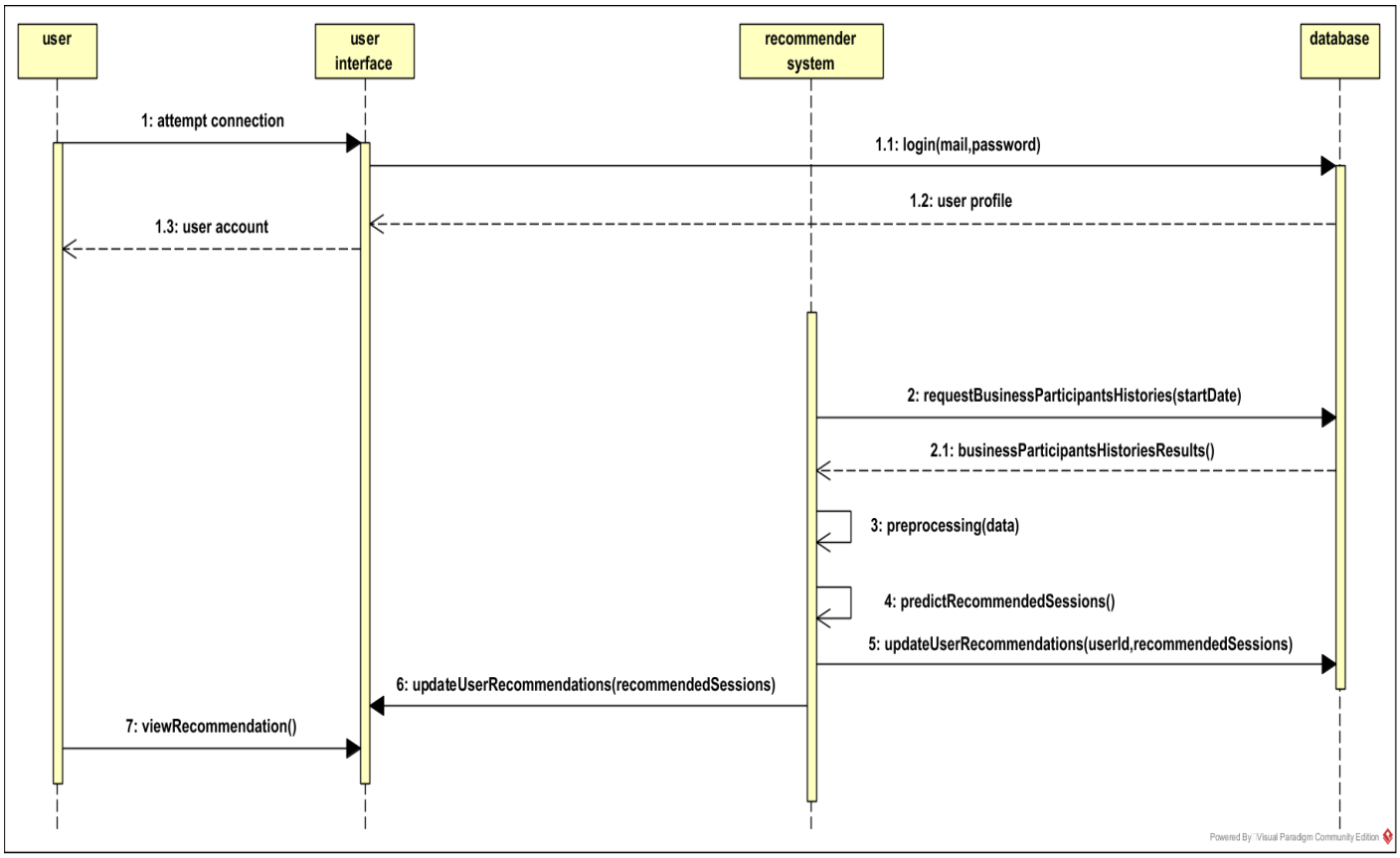

FIGURE 4.9 – Diagramme de séquence : recommandation des sessions aux formateurs et organismes de formations

4.4.3.4 Diagramme de séquence : Recommandation des formations aux formateurs et organismes de formations

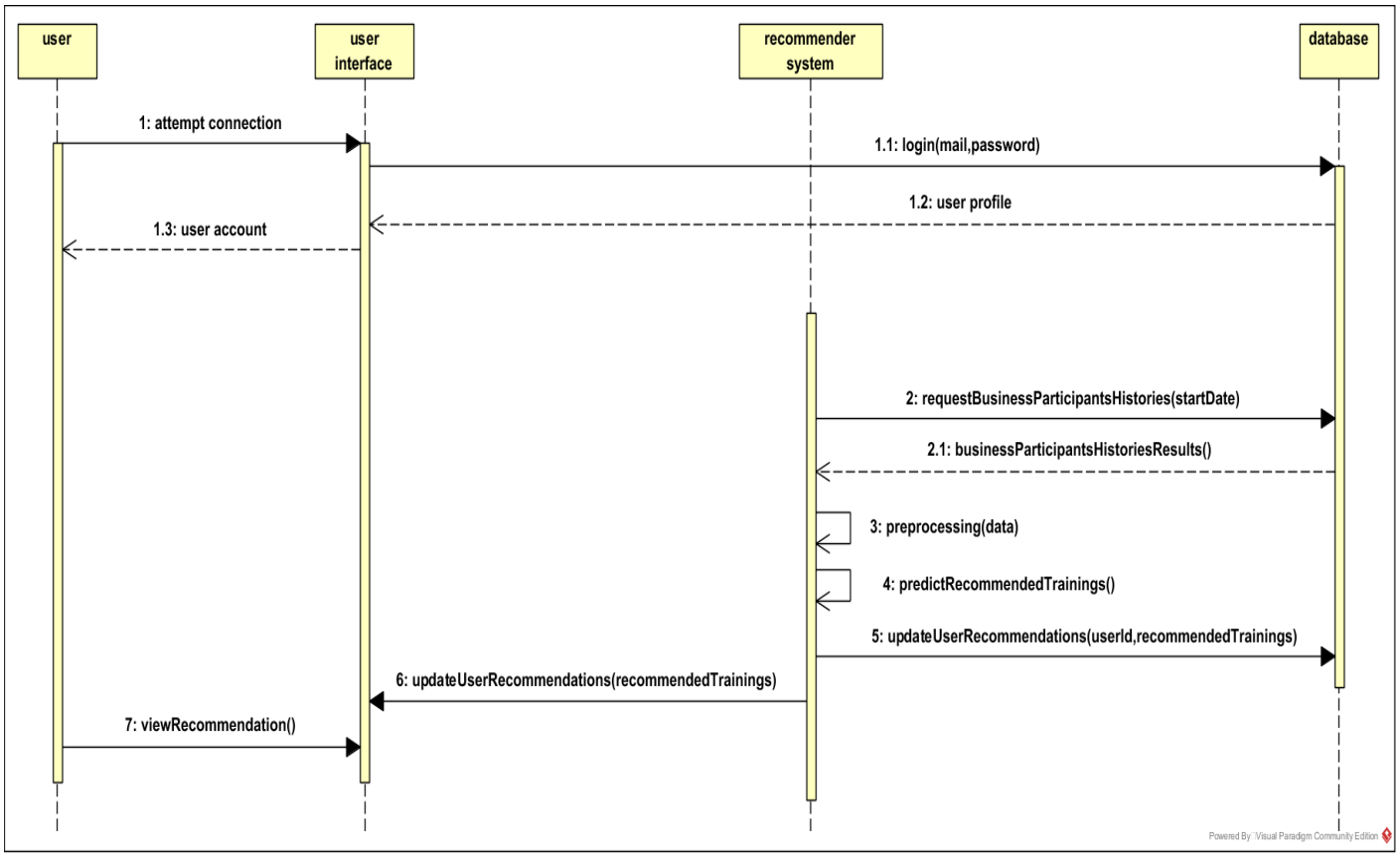

FIGURE 4.10 – Diagramme de séquence : recommandation des formations aux formateurs et organismes de formations

4.4.3.5 Diagramme de séquence : Recommandation des projets de formations aux formateurs et organismes de formations

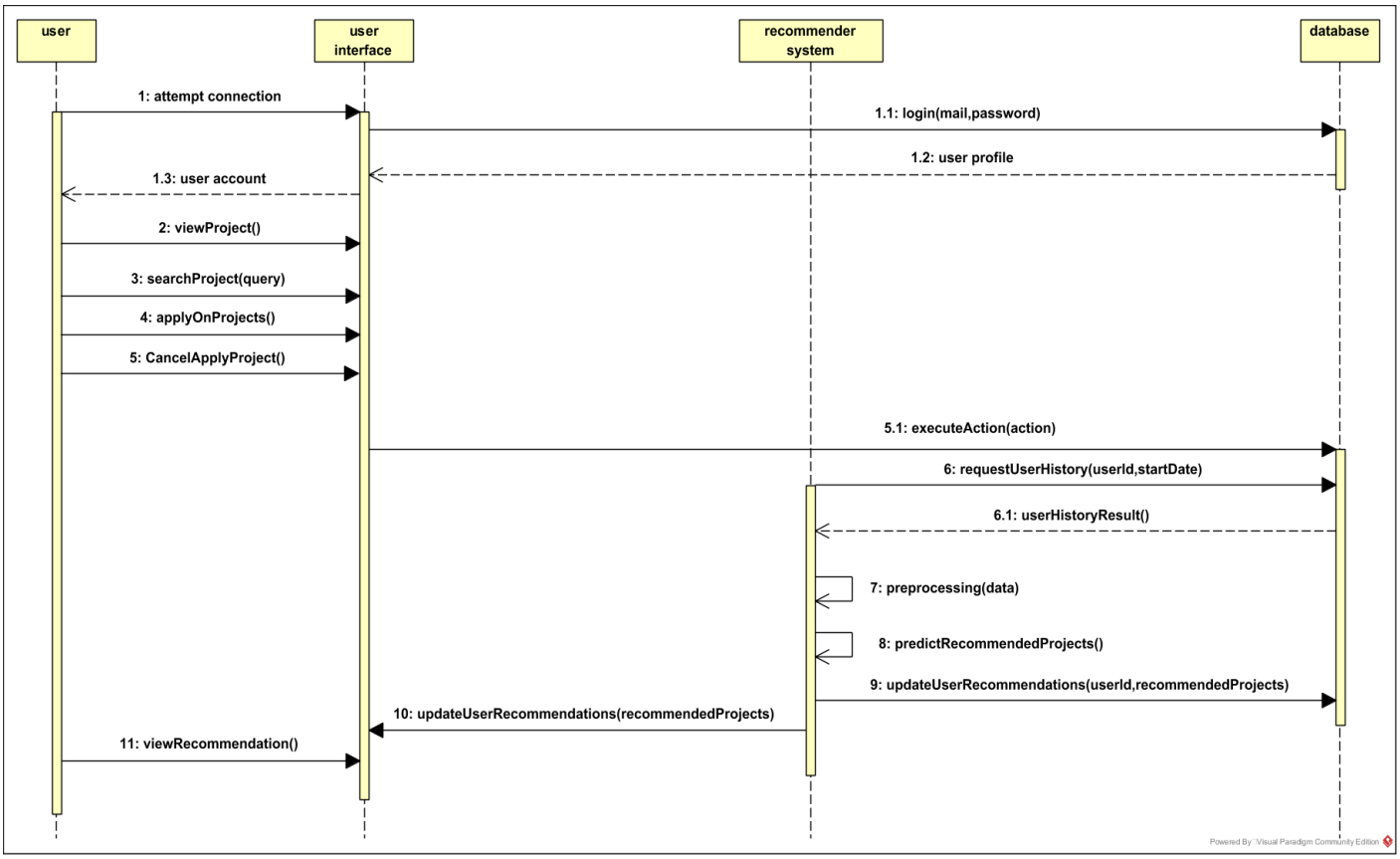

FIGURE 4.11 – Diagramme de séquence : recommandation des projets de formations aux formateurs et organismes de formations

4.4.4 Diagrammes d'activités

4.4.4.1 Diagramme d'activités : Entrainer le modèle

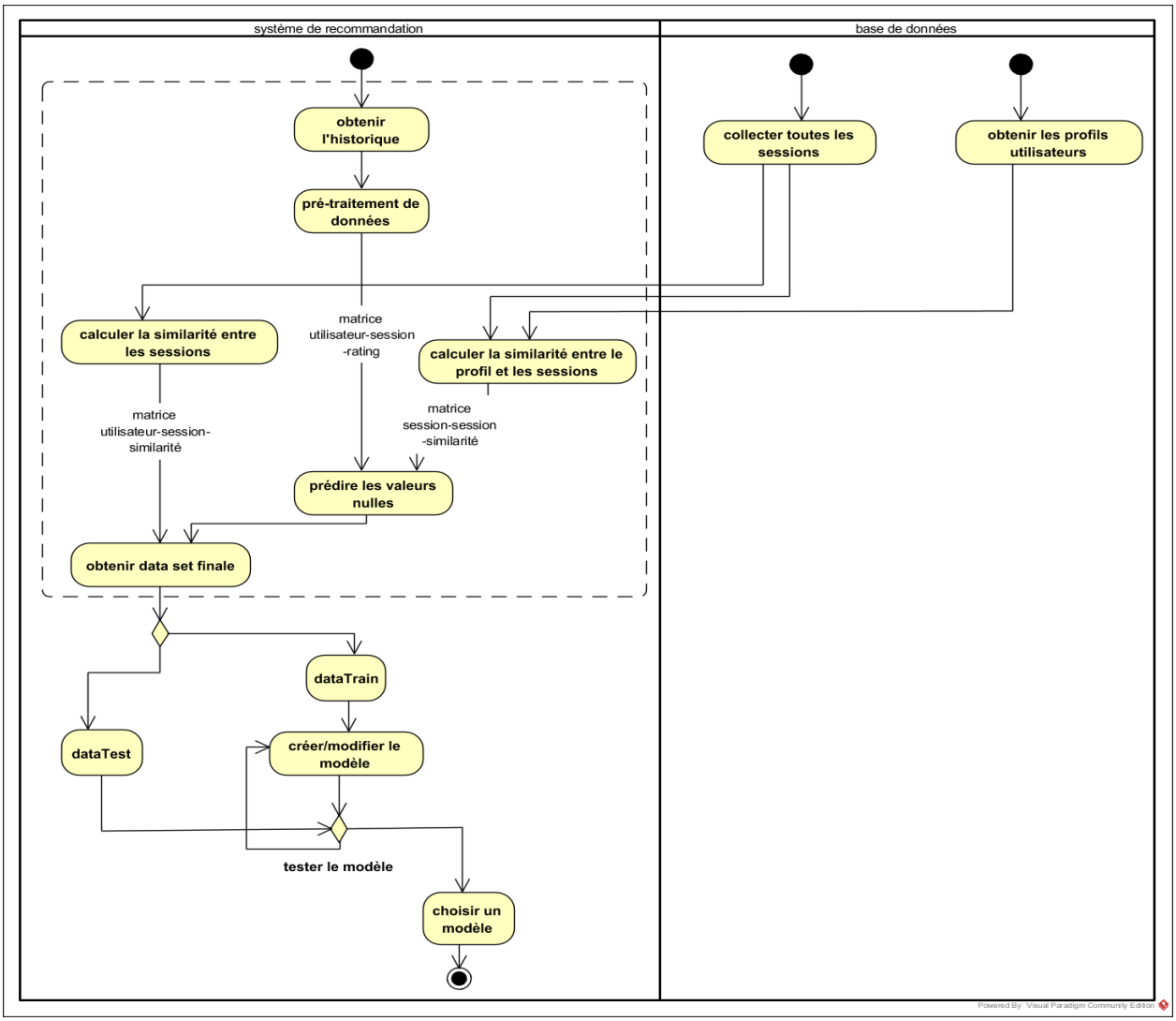

FIGURE 4.12 – Diagramme d'activités : Entrainer le modèle

4.4.4.2 Diagramme d'activités : Recommandation des sessions aux apprenants et entreprises

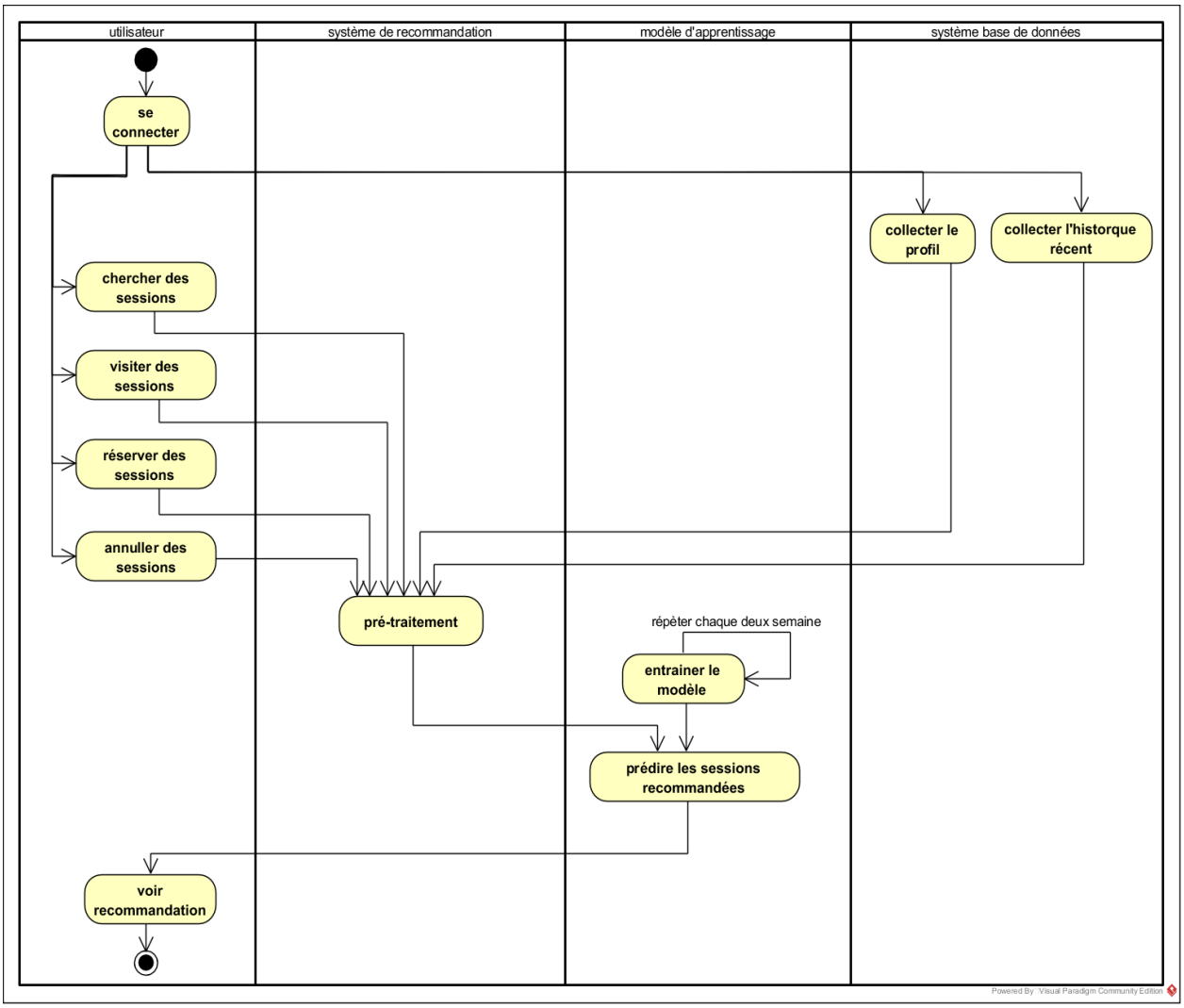

FIGURE 4.13 – Diagramme d'activités : Recommandation des sessions aux apprenants et entreprises

Conclusion

Dans ce chapitre nous avons présenté l'étude conceptuelle de notre application, il nous reste alors la partie réalisation qui sera présentée dans le chapitre suivant.

Chapitre 5

Réalisation

Introduction

Après avoir achevé l'étape de conception de l'application, nous entamons dans ce chapitre la phase de réalisation. Nous allons décrire, en premier lieu le l'environnement du travail, en second lieu les technologies choisies. Ensuite, nous allons donner un aperçu sur le travail accompli à travers des captures d'écran.

5.1 Environnement de développement

La réalisation de l'application, comme on a mentionné dans les chapitres précédents, a nécessité un ensemble d'outils logiciels afin de satisfaire les exigences demandés dans le cahier des charges.

5.1.1 Environnement matériel

Durant la réalisation de notre projet nous avons utilisé :

- Ordinateur portable ayant les caractéristiques suivantes :
  - Marque : Lenovo.
  - Système d'exploitation : Windows 10 Entreprise 64 bits.
  - Processeur : Intel i5, 2.40 GHz.
  - RAM : 6 Go.

- Mémoire : 1000 Go.

5.1.2 Environnement logiciel

- **Visual Studio Code :**

  Visual Studio Code est un éditeur de code source léger mais puissant qui est disponible sous Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Nodejs et possède un riche écosystème d'extensions pour d'autres langages (tels que C++, Java, Python, PHP) et des runtimes (.NET et Unity).

- **Pycharm :**

  PyCharm est un environnement de développement intégré utilisé pour programmer en Python. Il offre l'analyse de code, un débogueur graphique, la gestion des tests unitaires, l'intégration de logiciel de gestion de versions, et supporte le développement web avec Django. Il est développé par l'entreprise tchèque JetBrains. Il est multi-plateformes et fonctionne sous Windows, Mac OS X et Linux. Il est décliné en édition professionnelle, réalisé sous licence propriétaire, et en édition communautaire réalisé sous licence Apache.

5.2 Choix de technologies

5.2.1 Technologies de programmation

5.2.1.1 Python

Python est un langage de programmation de haut niveau largement utilisé en développement. Python propose un système de type dynamique et une gestion automatique de la mémoire et prend en charge plusieurs paradigmes de programmation, notamment les styles orienté objet, impératif, programmation fonctionnelle et procédural. Il a une grande bibliothèque standard complète.
Au fil des années Python est devenu un outil quotidien pour les ingénieurs et les chercheurs de toutes les disciplines scientifiques. Grâce a des nombreuses librairies d'une grande qualité il permet aujourd'hui d'égaler, voire de surpasser des solutions propriétaires les plus performantes du marché. Il est devenu un des outils incontournables de Data Science.

5.2.1.2 Pourquoi utiliser Python pour le calcul scientifique ?

Python est devenu une alternative viable aux solutions propriétaires leader du marché, comme MatLab, Maple, Mathematica, Statistica, SAS...

Il offre plusieurs avantages par rapport ces outils :

- Il calcule aussi vite. La plupart des librairies scientifiques de Python peuvent être compilées pour tirer partie des architectures vectorielles et multi-cœurs/multi-thread des processeurs modernes.Python dispose de nombreuses librairies de calcul distribué, permettant de répartir la charge des applications sur de nombreuses machines.
- Il couvre probablement tous les domaines scientifiques. Python ne se limite pas aux mathématiques et statistiques, il dispose de nombreuses librairies permettant d'aborder de multiples domaines comme :
  - Le traitement du signal.
  - La génétique.
  - L'apprentissage automatique.
  - Le langage naturel.
  - Le traitement d'images.
  - ...
- Il ne se limite pas à la simulation et peut être déployé en production.
- Il est facile, même pour les non informaticiens.
- Il est gratuit et open source.

5.2.1.3 Les bibliothèques python utilisés dans notre projet

Numpy Numpy est une librairie d'algèbre linéaire permettant de manipuler des tableaux à N dimensions. Principalement utilisée pour manipuler des nombres réels, elle peut traiter tout type de données, même les types natifs de Python. Elle se présente sous la forme d'un package majoritairement écrit en langage C, alliant ainsi toute la rapidité de ce langage avec la souplesse de la syntaxe Python. Elle apporte une syntaxe simple et puissante qui est indéniablement à l'origine de son adoption massive.

Scipy Scipy propose un ensemble de librairies mathématiques regroupées par disciplines (analyse numérique, algèbre linéaire, statistiques, traitement du signal, traitement d'images, ...). Elle s'appuie sur Numpy et offre ainsi des outils spécialisés dans plusieurs disciplines des mathématiques.

MatplotLib MatplotLib est une librairie de tracé de courbes en 2D et 3D. Elle a une syntaxe largement inspirée de celle des librairies Matlab. C'est une des plus anciennes librairies de visualisation en Python. Les graphiques peuvent être interactifs incluant des widgets et animés. Un autre de ses atouts est sa compatibilité avec de nombreuses librairies graphiques comme QT, GTK, TK, Wx, etc. Elle peut être ainsi s'intégrée dans de multiples applications clientes.

Pandas Pandas est une librairie d'analyse de données tout simplement prodigieuse. Tout comme Numpy elle permet de manipuler des tableaux de données avec une aisance sans précédent. Là ou elle se diffère de Numpy c'est qu'elle sait gérer des tableaux dont les colonnes ou les lignes sont de types différents (dates, nombres, texte, ...). Elle propose de nombreuses fonctionnalités de recherche, filtre, agrégation, fusion qui en font un outil hors du commun. On peut l'imaginer comme une librairie fournissant toutes les fonctions d'un tableur Excel /LibreOffice mais bien plus rapide, souvent plus facile à utiliser et pouvant manipuler des millions d'enregistrements.

5.2.2 Technologies en Big data

5.2.2.1 Apache Hadoop

La bibliothèque de logiciels Apache Hadoop est un cadre qui permet le traitement distribué de grands ensembles de données à travers des grappes d'ordinateurs en utilisant des modèles de programmation simples. Il est conçu pour évoluer à partir de serveurs uniques vers des milliers de machines, chacune offrant un calcul et un stockage local. Plutôt que de compter sur le matériel pour fournir une haute disponibilité, la bibliothèque elle-même est conçue pour détecter et gérer les défaillances au niveau de la couche d'application, fournissant ainsi un service hautement disponible sur un cluster d'ordinateurs, chacun pouvant être sujet à des défaillances.
Hadoop inclut les modules suivants :

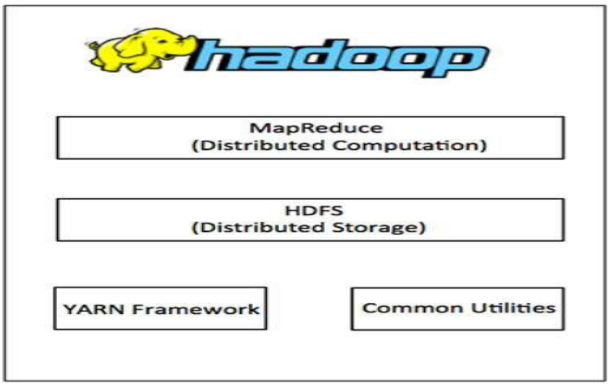

FIGURE 5.1 – L'architecture du Hadoop

5.2.2.2 Apache Spark

Apache Spark est une technologie de calcul en cluster rapide, conçue pour un calcul rapide. Il est basé sur Hadoop MapReduce et étend le modèle MapReduce pour l'utiliser efficacement pour plusieurs types de calculs, ce qui inclut les requêtes interactives et le traitement de flux. La principale caractéristique de Spark est son informatique en grappe en mémoire qui augmente la vitesse de traitement d'une application. Spark est conçu pour couvrir un large éventail de charges de travail telles que les applications batch, les algorithmes itératifs, les requêtes interactives et le streaming. En plus de supporter toutes ces charges de travail dans un système respectif, cela réduit le fardeau de la gestion du maintien d'outils séparés. Spark inclut les modules suivants :

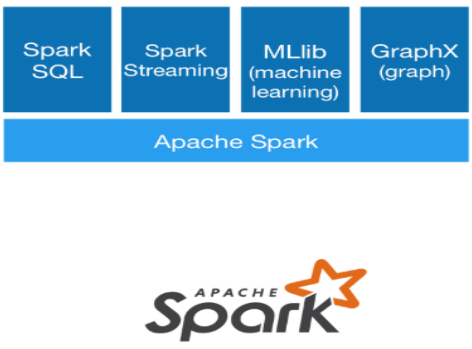

FIGURE 5.2 – L'architecture du Spark

5.2.2.3 Étude comparative entre Hadoop et Spark

	Hadoop	**Spark**
Plateforme	Java, python	Java, python, scala, R
License	Open source	Open source
Traitement de l'information	Hadoop traite les informations en disque	Spark calcul les informations en mémoire
Temps d'exécution	Un peu lent	Rapide
Analyse en temps réel	Hadoop échoue quant il y a un traitement en temps réel	Spark peut traiter des données en temps réel.
Facilité d'utilisation	Hadoop augmente le latence et ralentit le traitement des graphes. Hadoop n'a pas un mécanisme d'échange de messages	Spark est livré d'une bibliothèque de calcul graphique appelée GraphX. Netty et Akka permettent à Spark de distribuer des messages à travers les exécuteurs.
Tolérance aux pannes	Oui	Oui
Sécurité	Kerberos, LDAP pour l'authentification. ACL et modèle d'autorisations de fichiers traditionnel.	L'authentification via le secret partagé
Compatibilité	Hadoop est compatible à Spark	Spark est compatible à Hadoop
Cloud Services	Oui	Oui
Bibliothèque d'apprentissage automatique	Non	Oui, c'est Spark MLib

Puisque dans notre projet nous somme besoin des traitements en temps réel nous avons choisi Spark.

5.2.3 SkLearn pour le machine learning

5.2.3.1 Présentation du SkLearn

Sklearn ou Scikit-learn est une bibliothèque libre Python dédiée à l'apprentissage automatique. Elle est développée par de nombreux contributeurs notamment dans le monde académique par des instituts français d'enseignement supérieur et de recherche comme Inria et Télécom ParisTech. Elle contient un bon nombre d'algorithmes fréquemment utilisés associés à des tâches en machine Learning et en data Mining à titre d'exemples : le regroupement, la régression et la classification, la réduction des dimensions, la sélection du modèle et le pré-traitement pour analyser des datasets.

5.2.3.2 Pourquoi utiliser SkLearn

Cette librairie est intéressante vu que :

- Elle dispose d'une excellente documentation fournissant de nombreux exemples.
- Elle dispose d'une API uniforme entre tous les algorithmes, ce qui fait qu'il est facile de basculer de l'un à l'autre.
- Elle est très bien intégrée avec les Librairies Pandas et Seaborn[1].
- Elle dispose d'une grande communauté et de plus de 800 contributeurs référencés sur GitHub !
- C'est un projet open source.
- Son code est rapide.

5.2.3.3 L'architecture du SkLearn

SkLearn est une librairie d'apprentissage automatique couvrant l'ensemble de la discipline :

- **Les types d'apprentissage :** supervisé, non supervisé, par renforcement, par transfert
- **Les algorithmes :**
  - Linear Regression (régression linéaire).
  - Logistic Regression (régression logistique).

- Decision Tree (arbre de décision).
- SVM (machines à vecteur de support).
- Naive Bayes (classification naïve bayésienne).
- KNN (Plus proches voisins).
- Dimensionality Reduction Algorithms.
- Gradient Boost et Adaboost.
- Réseaux de neurones.

Voir la figure ci-dessous pour plus d'informations :

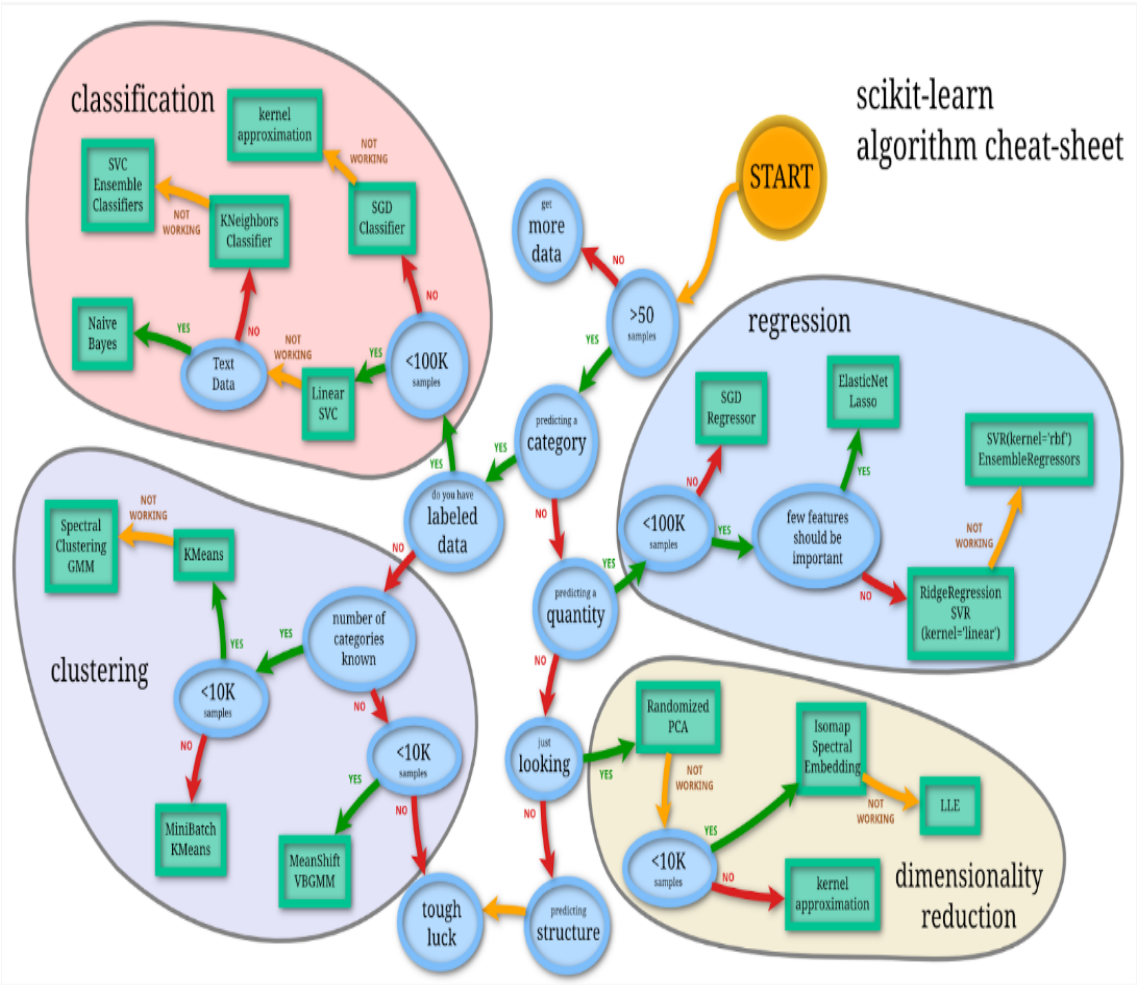

FIGURE 5.3 – Vue d'ensemble de Scikit-learn

5.3 Les algorithmes du Text Mining

Le traitement du langage naturel (NLP) est un domaine de l'informatique et de l'intelligence artificielle qui s'intéresse aux interactions entre les ordinateurs et les langues (naturelles) humaines, en particulier comment programmer des ordinateurs pour traiter et analyser de grandes quantités de données en langage naturel. Dans notre travail nous avons besoin de connaitre la similarité entre des textes par exemple entre les titres des formations, entre le continu des profils utilisateurs et le continu des formations. Pour cela nous avons proposer quelques algorithmes de similarités afin de choisir le plus performant.

5.3.1 NLTK

Natural Language Toolkit (NLTK) est une bibliothèque logicielle en Python permettant un traitement automatique des langues, développée par Steven Bird et Edward Loper du département d'informatique de l'Université de Pennsylvanie. En plus de la bibliothèque, NLTK fournit des démonstrations graphiques, des données-échantillon, des tutoriels, ainsi que la documentation de l'interface de programmation (API).

NLTK est une plate-forme leader pour la construction de programmes Python pour travailler avec des données de langage humain. Elle fournit des interfaces faciles à utiliser pour plus de 50 ressources corporelles et lexicales telles que WordNet, ainsi qu'une suite de bibliothèques de traitement de texte pour la classification, tokenization, stemming, étiquetage, analyse et raisonnement sémantique, wrappers pour les bibliothèques NLP de puissance industrielle, et un forum de discussion actif. Grâce à un guide pratique présentant les fondamentaux de la programmation ainsi que des sujets en linguistique informatique, ainsi qu'une documentation API complète, NLTK convient aux linguistes, ingénieurs, étudiants, enseignants, chercheurs et utilisateurs de l'industrie. NLTK est disponible sous Windows, Mac OS X et Linux. NLTK est un projet libre, open source, axé sur la communauté.
Les figures ci-dessous montrent des exemple de testes pour NLTK.

```
In [7]: runfile('C:/Users/oussama/Desktop/premierappJS/simText.py', wdir='C:/Users/
oussama/Desktop/premierappJS')
('Sentence 1: ', 'data science')
('Sentence 2: ', 'machine learning.')
('Similarity index value : ', 0.64)
Somewhat Similar

In [8]: runfile('C:/Users/oussama/Desktop/premierappJS/simText.py', wdir='C:/Users/
oussama/Desktop/premierappJS')
('Sentence 1: ', 'data analytic')
('Sentence 2: ', 'machine learning.')
('Similarity index value : ', 0.62)
Somewhat Similar

In [9]: runfile('C:/Users/oussama/Desktop/premierappJS/simText.py', wdir='C:/Users/
oussama/Desktop/premierappJS')
('Sentence 1: ', 'deep learning')
('Sentence 2: ', 'machine learning.')
('Similarity index value : ', 0.73)
Somewhat Similar

In [72]: runfile('C:/Users/oussama/Desktop/premierappJS/simText.py',
wdir='C:/Users/oussama/Desktop/premierappJS')
('Sentence 1: ', 'deep learning with NLTK  ')
('Sentence 2: ', 'NLTK')
('Similarity index value : ', nan)
Not Similar
```

FIGURE 5.4 – Test de similarité avec NLTK

5.3.2 Cosinus

La similarité cosinus (ou mesure cosinus) permet de calculer la similarité entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux. Cette métrique est fréquemment utilisée en fouille de textes.

Soit deux vecteurs A et B, l'angle θ s'obtient par le produit scalaire et la norme des vecteurs :

$\cos(\theta) = \frac{A.B}{\|\vec{A}\|.\|\vec{B}\|}$

Comme la valeur $\cos(\theta)$ est comprise dans l'intervalle [-1,1], la valeur -1 indiquera des vecteurs résolument opposés, 0 des vecteurs indépendants (orthogonaux) et 1 des vecteurs similaires (colinéaires de coefficient positif). Les valeurs intermédiaires permettent d'évaluer le degré de similarité.

Le cas d'une comparaison des documents textuels :

La similarité cosinus est fréquemment utilisée en tant que mesure de ressemblance entre deux documents. Il pourra s'agir de comparer les textes issus d'un corpus dans une optique de classification (regrouper tous les documents relatifs à une thématique particulière), ou de recherche d'information (dans ce cas, un document vectorisé est constitué par les mots de la requête et est comparé par mesure de cosinus de l'angle avec des vecteurs correspondant à tous les documents présents dans le corpus. On évalue ainsi lesquels qui sont les plus proches). La mesure d'angle entre deux vecteurs ne

pouvant être réalisée qu'avec des valeurs numériques, il faut donc imaginer un moyen de convertir les mots d'un document en nombres. On partira d'un index correspondant aux mots présents dans les documents puis on attribuera à ces mots des valeurs. La forme la plus simple pourrait être de compter le nombre d'occurrences des mots dans les documents.

```
>>> fuzz.partial_ratio("machine learning", "deep learning")
83
>>> fuzz.partial_ratio("machine learning", "data science")
29
>>> fuzz.partial_ratio("CNN", "deep learning with CNN")
100
```

FIGURE 5.5 – Test de similarité avec cosinus

5.3.3 Solution Hybride

Après quelques tests, nous avons de décider de combiner ces deux algorithme (NLTK, Cosinus) en un seul sous la forme suivante :

Algorithme

similarité1=nltk(phrase1,phrase2)

si similarité < 0.5 alors

similarité2=cosine(phrase1,phrase2)

return sup(similarité1,similarité2)

finsi

return similarité1

fin algorithme

5.4 Les algorithmes de machine learning

Dans notre travail nous testons plusieurs algorithmes d'apprentissage automatique pour retenir la plus performante dans notre solution. Précisément on a travaillé avec les algorithmes suivants :

5.4.1 Arbre de décision

Les arbres de décision sont développés pour essayer de diviser ainsi les données sur les paramètres des vecteurs :

- Chaque nœud interne décrit un test sur un paramètre d'apprentissage.
- Chaque branche représente un résultat du test.
- Chaque feuille contient la valeur de la variable cible :
  - Une étiquette de classe pour les arbres de classification.
  - Une valeur numérique pour les arbres de régression.

La pertinence de l'algorithme construisant l'arbre se mesure à sa capacité de trouver les paramètres qui permettent de maximiser le partage à chaque nœud et peuvent être un bon complément aux régressions logistiques.
Les contraintes des arbres de décision :

- L'influence de l'ordre des paramètres prédicteurs dans le graphe.
- La difficulté de représenter des règles comme le Ou Exclusif.

5.4.2 Les forêts aléatoires

Les forêts d'arbre décisionnels/aléatoires sont basées sur le concept de bagging /inférence statistique et d'arbres décisionnels. L'idée étant d'apprendre sur de multiples arbres de décision travaillant sur des sous-ensembles de données les plus indépendants possible. Cela permet de régler plusieurs problèmes inhérents aux arbres de décision uniques comme l'altération du résultat selon l'ordre des paramètres prédicteurs dans les nœuds, ou encore de réduire leur complexité.

5.4.3 Les machines à vecteur de support

Les machines à vecteur de support sont très utilisées dans les problèmes de régression et offrent une extension aux régressions linéaires lorsque les données présentes des niveaux de séparation plus tordus et on peut l'utiliser pour la classification. Elles permettent de calculer des données quand leurs labels ne sont pas séparables par une équation linéaire. Elles proposent de séparer les données

avec des équations plus riches comme des polynômes, gaussiennes, etc...
Les SVM connaissent un très grand succès, pour de multiples raisons :

- Elles peuvent travailler avec des données disposant d'un très grand nombre de paramètres.
- Elles utilisent peu d'hyper-paramètres.
- Elles garantissent de bons résultats théoriques.
- Elles peuvent égaler ou dépasser en performance les réseaux de neurones ou modèles gaussiens.

5.4.4 Naïve Bayes

La classification naïve bayésienne est basée sur le théorème de Bayes permettant de déterminer la distribution d'une loi binomiale. Il s'agit d'une équation décrivant la relation entre des probabilités conditionnelles de quantités statistiques. Elle s'inscrit dans le groupe des classifieurs linéaires.
Nous souhaitons ici trouver la probabilité d'un label à partir de paramètres observés, noté P($L|PARAMS$) :
$p(L|PARAMS) = \frac{p(L|PARAMS)*p(L)}{p(PARAMS)}$
La génération d'un modèle sur cette loi se fait pour chaque label et peut être ardue. Le modèle est dit naïf car il simplifie grandement cette tâche en procédant à plusieurs approximations naïves. Il est de ce fait très rapide et c'est un bon modèle pour commencer une classification.

5.4.5 K plus proche voisins

L'algorithme des plus proches voisins est relativement simple. Une de ses forces est de ne calculer aucune information dans le processus d'apprentissage. Il recherche les N plus proches voisins (par un calcul de distance) entre la donnée à prédire et les données connues. Il retourne alors la classe de la majorité des voisins. Assez simple à mettre en œuvre il peut générer beaucoup de calculs et ne pas être adapté à de fortes volumétries notamment si le nombre de paramètres est très grand.

5.4.6 Gradient Boost et Adaboost

Le gradient boosting est un algorithme s'appliquant aux problèmes de classification et régression. L'idée est d'améliorer la prédiction et la vitesse en combinant un ensemble d'algorithmes d'apprentissages plus simples au travers d'un arbre de décision. Le travail revient alors à identifier la fonction permettant de maximiser le choix des différents algorithmes.

AdaBoost est une variante du gradient boost. Il combine via une somme pondérée le résultat de différents algorithmes d'apprentissage plus simples. Il est adaptatif dans le sens où il peut jouer sur le poids des différents algorithmes simples en fonction de la qualité de leurs résultats.
Après avoir tester ces algorithme nous avons choisi les forêts aléatoires.

5.5 Les interfaces homme machine

Dans cette section nous allons créer différents scénarios pour tester l'efficacité du notre système de recommandation.

5.5.1 Système de recommandation à base de continu

5.5.1.1 Le cas pour l'apprenant

Pour tester la performance de notre système on prend à titre d'exemple un apprenant qui a comme expériences Data Scientist chez Copenhagen Robots, stage Data Analyste chez l'université du new york et d'autres. De plus il est diplômé en tant que ingénieur génie logiciel de l'Université de Madrid. Il a un certificat dans le domaine de data science. Il est un nouveau utilisateur n'a pas d'historique. Les figures suivantes montrent les recommandations des sessions et formations.

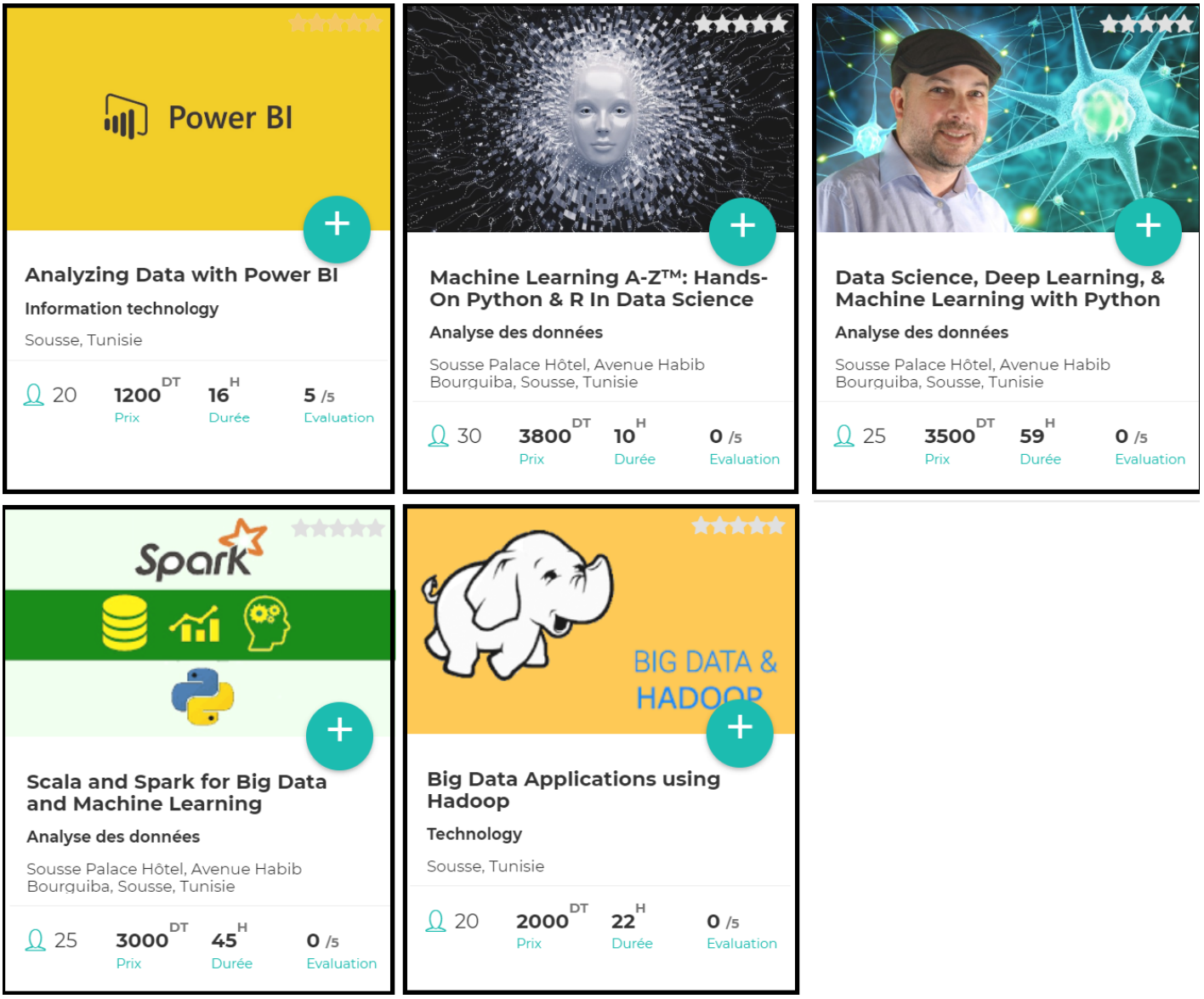

FIGURE 5.6 – Des sessions recommandées à cet apprenant

5.5.1.2 Le cas pour le formateur

5.5.2 Système de recommandation à base collaborative

Dans la section précédente nous avons testé le cas d'un utilisateur avec profil et sans historique, dans cette partie nous prenons le cas contraire, donc les figures suivantes montrent un cas d'un profil vide, les historiques de recherche, de visite de différent sessions et formations, les sessions réservées, les formations abonnées ...

5.5.2.1 Le cas pour l'entreprise

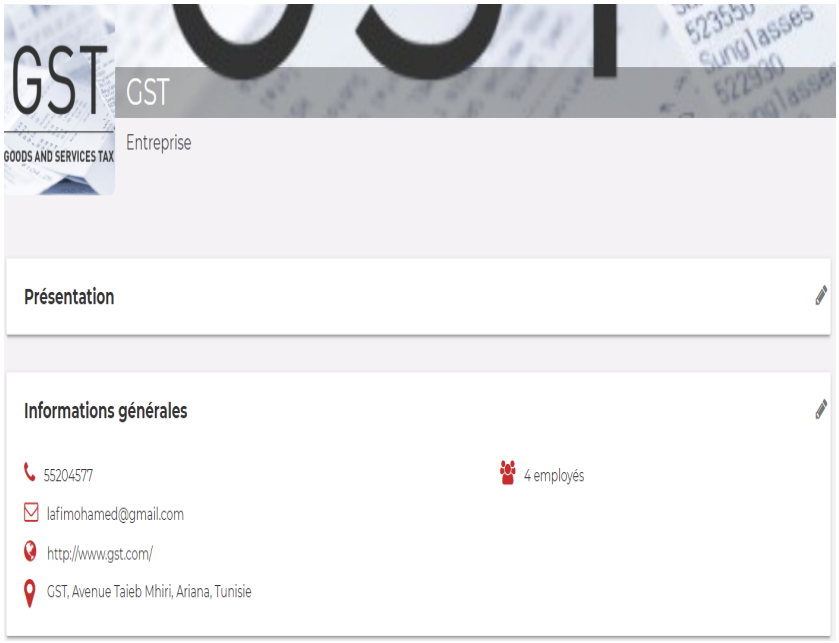

FIGURE 5.7 – Vue d'ensemble de Scikit-learn

5.5.2.2 Recommandation des sessions

5.5.2.3 Recommandation des formations

Conclusion

Pendant ce chapitre, nous avons présenté l'environnement du travail,le choix de technologies et ainsi que quelques captures écrans pour des scénarios différents afin de prouver l'efficacité de notre système de recommandation.

Conclusion Générale

En plein dans le phénomène du Big Data, on assiste à l'utilisation conjointe de quantités massives d'informations et d'algorithmes d'apprentissages. Ce duo rend possible la solution de comment comprendre la personnalité de l'utilisateur afin de répondre à ses besoins d'une façon automatique.

C'est dans ce cadre que s'est inscrit ce projet de fin d'études au sein de la société Sindibad group. Ce projet consiste à concevoir et développer un système de recommandation intelligent pour une plateforme de gestion de formation professionnelle basée sur l'apprentissage automatique et l'analyse de données.

Nous avons décomposé notre projet en quatre tâches : le récolte de données (nous avons créé un système de tracking pour enregistrer les navigations de l'utilisateur sur la plateforme), la préparation d'une dataset correcte pour entrainer et tester le modèle, Choisir un modèle du machine learning et enfin visualiser la recommandation sur la plateforme.

Pour atteindre cet objectif, nous avons commencé par l'étude préalable qui a permis de décrire et de comprendre les principaux concepts autour desquels tourne notre projet. Puis nous avons passé à l'analyse et la spécification des besoins et exigences. Ensuite, nous avons élaboré la forge logicielle de l'outil à développer en commençant par l'architecture adoptée, pour aboutir par la suite à la conception, qui met l'accent sur l'aspect dynamique et statique du système. Enfin, nous avons abordé l'étape de la réalisation durant laquelle nous avons traduit la modélisation conceptuelle en une implémentation physique moyennant les différentes technologies choisies.

Ce travail nous a été très instructif les multiples connaissances acquises. Il nous a procuré une opportunité pour, d'une part aborder un domaine métier et d'autre part confirmer une fois de plus nos compétences dans le raisonnement des problèmes, dans le développement en Python et toucher près plusieurs aspects du machine learning et analyse de données.

Néanmoins, nous tenons à présent à souligner quelques extensions ou perspective intéressantes de notre projet. Grâce à son caractère extensible et sa modularité, notre outil pourra être amélioré en ajoutant d'autre algorithmes d'apprentissage basés sur le deep learning ou bien combiner entre des algorithmes afin de créer un nouveau algorithme plus performant , le feedback de l'utilisateur sur nos recommandations.

Chapitre 6

Annexe

6.1 Définition de la sur-information

	Hadoop	**Spark**
Facilité d'utilisation	Hadoop augmente le latence et ralenti le traitement des graphes. Hadoop n'a pas un mécanisme pour l'échange de messages	Spark est livré d'une bibliothèque de calcul graphique appelé GraphX. Netty et Akka permettent à Spark de distribuer des messages à travers les exécuteurs
Tolérance aux pannes	oui	oui
Sécurité	Kerberos, LDAP pour l'authentification. ACL et modèle d'autorisations de fichiers traditionnel	l'authentification via le secret partagé
Compatibilité	Hadoop est compatible à Spark	Spark est compatible à Hadoop
Cloud Services	oui	oui
Bibliothèque d'apprentissage automatique	non	Oui, c'est Spark MLib

Bibliographie

[**1**] Badrul Sarwar et al.Item based collaborative filtering recommendation algorithms.Proceedings of the 10th international conference on World Wide Web. 2001

[**2**] Martin Eppler. Informative action : An analysis of management and the information overload. Thèse de doct. HEC Management Studies,University of Geneva, 1998.

[**3**] Martin J Eppler et Jeanne Mengis.The concept of information overload : A review of literature from organization science, accounting, marketing, MIS, and related disciplines. Dans : The information society (2004).

[**4**] Xujuan Zhou et al. The state-of-the-art in personalized recommender systems for social networking.Artificial Intelligence Review (2012).

[**5**] Internet World Stats. 2014. url : www.internetworldstats.com/stats.htm

[**6**] Nikos Manouselis et al.Recommender Systems in Technology Enhanced Learning. Dans : Recommender Systems Handbook. 2011.

[**7**] Scott Wilson et al.Personal Learning Environments : Challenging the dominant design of educational systems. Dans : Journal of e-Learning and Knowledge Society (2007).

[**8**] G. Parsa. (2017, July). Document Classi ?cation. Retrieved from http ://www.kdnuggets.com/2015/01/tex analysis-101-document-classification.html

[**9**] Textual data and vector space model. (2017, July). Retrieved from http ://www.calpoly.edu/dsun09/lesson

[**10**] Data Analysis. (2017, February). Retrieved from https ://www.ngdata.com/what-is-data-analysis

[**11**] Recommender System.(2017, July). Retrieved from http ://recommendersystem.blogspot.com/20121001

[**12**] F. O. Isinkaye, Y. O. Folajimi and B. A. Ojokoh. (2015). Recommendation Systems :Prin-

ciples, methods and evaluation Egyptian Informatics Journal

[**13**] Michael J Pazzani et Daniel Billsus.Content-based recommendation systems.The adaptive web. 2007

[**14**] Mouzhi Ge, Carla Delgado-Battenfeld et Dietmar Jannach.Beyond accuracy : evaluating recommender systems by coverage and serendipity. Proceedings of the fourth ACM conference on Recommender systems. ACM. 2010, p. 257260

[**15**] Anna Stefani et C Strappavara. Personalizing access to web sites : The SiteIF project. Dans : Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT. 1998

[**16**] Joana Trajkova et Susan Gauch.Improving Ontology Based User Profiles.RIAO. 2004

[**17**] RcMcCrae and O.John.(1992) :Journal of Personality.

[**18**] F. O. Isinkaye, Y. O. Folajimi and B. A. Ojokoh. (2015). Recommendation Systems :Principles, methods and evaluation Egyptian Informatics Journal

[**19**] Robin Burke.Knowledge based recommender systems.Encyclopedia of library and information systems (2000)

[**20**] Hemant K Bhargava, Suresh Sridhar et Craig Herrick.Beyond spreadsheets : tools for building decision support systems. Dans : Computer (1999)

[**21**] Rashmi R Sinha et Kirsten Swearingen. Comparing RecommendationsMade by Online Systems and Friends. Dans : DELOS workshop : personalisation and recommender systems in digital libraries. 2001.

[**22**] Tariq Mahmood et Francesco Ricci.Improving recommender systems with adaptive conversational strategies. Dans : Proceedings of the 20th ACM conference on Hypertext and hypermedia. 2009.

[**23**] Bruce Krulwich. Lifestyle : Intelligent user profiling using largescale demographic data. Dans : AI magazine (1997).

[**24**] Robin Burke. Hybrid web recommender systems.The adaptive web. 2007

[**25**] Robin Burke.Hybrid recommender systems : Survey and experiments : User modeling and user adapted interaction (2002).

[**26**] Robin Burke.Hybrid web recommender systems. Dans : The adaptive web. 2007

[27] A. Tejal. (2015). A Survey on Recommendation System. International Journal of Innovative Research in Advanced Engineering(IJIRA).