

Faculté des Sciences Exactes et d'Informatique

Département de Mathématiques et informatique

Filière : Informatique

MEMOIRE DE FIN D'ETUDE

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Ingénierie des Systèmes d'Information**

Présenté par :

DOUBBI BOUNOUA Hadjer et DANI Kheira

THEME :

**La matrice de factorisation pour la recommandation
des POI**

Soutenu le : 19 / 06 / 2021

Devant le jury composé de :

Mr HENNI Fouad	MCB	Président
Mme DJAHAFI Fatiha	MAA	Examineur
Mme MAGHNI SANDID Zoulikha	MAA	Encadreur

Année Universitaire 2020-2021

Dédicaces

Je dédie ce mémoire à :

En premier lieu ceux que personne ne peut compenser les sacrifices qu'ils ont consentis pour notre éducation et notre bien-être à nos parents mon père DOUBBI BOUNOUA ABDELKADER et ma mère GUETARNI CHERIFA qui se sont sacrifiés pour nous prendre en charge tout au long de mes études et qui sont l'origine de notre réussite que dieu les gardes et les protèges inchaalah.

A mes frères ABDELWAFI et MOHAMED EL AMINE et mes sœurs WAFAA et ISMAHENE.

A mes nièces YASMINE et ABIR. A mes meilleures copines.

A toutes ma chère famille et mes proches qui nous ont accordé leur soutien dans les instants les plus difficiles.

Sans oublier ma chère collègue KHEIRA pour son soutien moral, sa patience, sa entente, sa sympathie et sa compréhension tout au long de ce projet.

HADJER

Dédicaces

Je dédie ce mémoire à :

A la mémoire de ma chère mère

A la mémoire de mon cher Oncle

Paix à leur âme

*A mon père DANI LAKHDAR qui a sacrifié sa vie pour
notre éducation, notre réussite et notre bien-être.*

Merci d'avoir compensé l'absence de ma mère.

*Merci de faire de moi ce que je suis aujourd'hui, Que dieu
te garde pour nous inchaallah.*

*A ma grand-mère SALAA HASNIA ceci est ma profonde
gratitude pour ton maternel amour*

*A mes sœurs et mon frère, qui m'ont toujours soutenu et
encouragé durant ces années.*

*A toute ma chère famille et mes proches qui nous ont
accordé leur soutien dans les instants les plus difficiles.*

*Sans oublier ma chère collègue HADJER, pour son soutien
moral, sa patience, sa entente, sa sympathie et sa
compréhension tout au long de ce projet.*

KHEIRA

Remerciements

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

Nous tenons à remercier notre encadreur Mme MAGHNI SANDID ZOULIKHA pour son encadrement son écoute, ses élucidations, ses conseils, ses directives et encouragements qu'elle nous a afflué.

Nos vifs remerciements vont également aux membres du jury qui ont accepté de juger notre travail.

Nous tenons également à remercier tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Résumé

Les systèmes de recommandation sont une sorte de filtrage des données qui permettent de guider l'utilisateur vers des ressources intéressantes et utiles au sein d'un espace de données important. Le système de recommandation de points d'intérêt (POI) utilise les informations d'enregistrement pour prédire les emplacements les plus potentiels. Ces systèmes ont rencontré plusieurs problèmes tels que le manque des données où les données d'enregistrements des POI sont très rares, et le passage à l'échelle qu'il désigne la capacité d'un algorithme à traiter avec la même efficacité les données nombreuses. Dans ce travail, nous avons intégré ces systèmes de recommandation dans un site web pour faire des recommandations de POI et résoudre ces problèmes en utilisant l'algorithme de factorisation matricielle du filtrage collaboratif qui se base sur la méthode réduction dimensionnelle et plus précisément la décomposition de la valeur singulière.

Mot clés : systèmes de recommandation, points d'intérêts, factorisation matricielle, filtrage collaboratif, SVD

Abstract

Recommendation systems are a kind of data filtering that can guide the user to interesting and useful resources within a large data space. The Points of Interest (POI) recommendation system uses the registration information to predict the most potential locations. These systems have encountered several problems such as the lack of data where POI record data are very rare and the scaling it denotes the ability of an algorithm to process the numerous data with the same efficiency. In this work, we integrated these recommendation systems into a website to make POI recommendations and solve these problems using the matrix factorization algorithm of collaborative filtering which is based on the dimensional reduction method and more precisely the decomposition of the singular value.

Keywords: recommendation systems, points of interest, matrix factorization, collaborative filtering, SVD

Liste des figures

Figure N°	Titre de la figure	Page
Figure 1	Etapes d'un système de recommandation	4
Figure 2	Filtrage basé sur le contenu	5
Figure 3	Filtrage collaboratif	6
Figure 4	Les méthodes de filtrage collaboratif	7
Figure 5	Filtrage hybride	9
Figure 6	Matrice de factorisation	13
Figure 7	La Décomposition de la valeur singulière SVD	14
Figure 8	Diagramme de cas d'utilisation	20
Figure 9	Diagramme de séquence « Inscription »	22
Figure 10	Diagramme de séquence « Login »	23
Figure 11	Diagramme d'activité Authentification	24
Figure 12	Diagramme de classe	25
Figure 13	Architecture de Firebase	28
Figure 14	Interface de l'accueil	34
Figure 15	Interface d'inscription	35
Figure 16	Interface d'identification	35
Figure 17	Interface d'accueil (après l'inscription)	36
Figure 18	Interface de détail de POI	36
Figure 19	Interface de détail de POI au-dessous	37
Figure 20	Interface de l'historique	37
Figure 21	Interface de favoris	38
Figure 22	Interface de recherche	38
Figure 23	Firebase	39

Figure 24	Logo du site	41
Figure 25	Illustrator	41
Figure 26	Création d'un compte Firebase	42

Liste des tableaux

Tableau N°	Titre du tableau	Page
Tableau 1	Tableau Utilisateur x Item	7
Tableau 2	Un exemple de points d'intérêt et ses informations associées	15
Tableau 3	Description textuelle du cas d'utilisation "inscription"	20
Tableau 4	Description textuelle du cas d'utilisation "login"	21
Tableau 5	Exemple de notation User / POI	29
Tableau 6	Utilisateur / Moyenne	29
Tableau 7	Matrice normalisé	30
Tableau 8	Matrice transposé	31
Tableau 9	Matrice User / POI	31
Tableau 10	Matrice U Orthogonal	32
Tableau 11	Matrice V Orthogonal	32
Tableau 12	Matrice S diagonal	32
Tableau 13	Matrice de recommandation	33

Liste des abréviations

Abréviation	Expression Complète	Page
SR	Système de Recommandation	1
LBSN	Les réseaux sociaux basés sur la localisation	1
POI	Point d'intérêts, Point Of Interest	1
FM	Factorisation Matricielle	12
SVD	Décomposition des valeurs singulière	1
PCA	Analyse des composants principaux	14
UML	Langage de modélisation graphique	2
FC	Filtrage collaboratif	6

Table des matières

Introduction générale.....	1
Chapitre 1 Les systèmes de recommandation	3
1.1 Introduction.....	3
1.2 Définitions des systèmes de recommandation.....	3
1.3 Types de données de système de recommandation	4
1.3.1 Collecte explicite	4
1.3.2 Collecte Implicite.....	5
1.4 Différentes approches de système de recommandation	5
1.4.1 Basé sur le contenu	5
1.4.2 Filtrage collaboratif.....	6
1.4.3 Filtrage Hybride.....	9
1.5 Les problèmes des Systèmes de Recommandation	9
1.6 Les Systèmes de Recommandation collaboratif.....	10
1.7 L'avenir des Systèmes de Recommandation.....	11
1.8 Conclusion	11
Chapitre 2 Factorisation matricielle pour les systèmes de recommandation.....	12
2.1 Introduction.....	12
2.2 La factorisation matricielle FM.....	12
2.2.1 Définition	12
2.2.2 Principe de fonctionnement du FM.....	12
2.2.3 Les modèles de la FM	13
2.3 Définition de point d'intérêts (POI).....	15
2.4 Les caractéristiques de point d'intérêts.....	15
2.4.1 Influence géographique.....	15
2.4.2 Données de fréquence et parcimonie	16
2.4.3 Influence sociale	16
2.5 Conclusion	17
Chapitre 3 Conception et modélisation du système	18
3.1 Introduction.....	18
3.2 Méthode d'Analyse et conception	18
3.2.1 Définition UML.....	18
3.2.2 Spécification des besoins du système	18
3.3 Les diagrammes UML	19

3.3.1	Diagramme de cas d'utilisation	19
3.3.2	Description textuelle	20
3.3.3	Diagramme de séquence.....	21
3.3.4	Diagramme d'activité.....	23
3.3.5	Diagramme de classe	24
3.4	Conclusion	25
Chapitre 4 Implémentation		26
4.1	Introduction.....	26
4.2	Environnement de développement	26
4.2.1	Draw.io.....	26
4.2.2	Flutter	26
4.2.3	Visual Studio Code	27
4.2.4	FireBase	27
4.2.5	Illustrator	28
4.3	Langages de programmation	28
4.3.1	Dart	28
4.3.2	JSON.....	28
4.4	Algorithme de recommandation utilisé :.....	29
4.5	Présentation de l'application	34
4.5.1	Interface de l'accueil.....	34
4.5.2	Interface d'inscription.....	34
4.5.3	Interface d'identification.....	35
4.5.4	Interface d'accueil (après inscription).....	36
4.5.5	Interface de détail de POI.....	36
4.5.6	Interface de favoris et historique	37
4.5.7	Interface de recherche	38
4.5.8	Base de données (Firebase)	39
4.6	Conclusion	39
Conclusion Générale		40
Annexe A		41
Annexe B		42
Références.....		43

Introduction générale

Notre quotidien nous expose fréquemment à des situations qui nous conduisent à prendre des décisions et à opérer des choix tels que la nourriture, les vêtements, les voitures, les lieux de vacances, qu'ils soient cruciaux ou secondaires, nos choix sont dictés par nos connaissances, préférences, et par les informations dont nous disposons.

Cette dernière décennie a été marquée par une accumulation des données dans tous les domaines. Cette accumulation des données a abouti à la création de nouveaux domaines de recherche dans le domaine de l'informatique. Le système de recommandations, un système complètement automatisé qui analyse les préférences des utilisateurs et prédit le comportement des utilisateurs, en est l'un parmi eux. L'intérêt de la recherche dans ce domaine est encore très élevé principalement en raison de la signification pratique du problème.

Les systèmes de recommandation ont été étudiés dans de nombreux domaines : la recherche d'informations, le web, le e-commerce, l'exploitation des usages du web et bien d'autres.

Les algorithmes de recommandation les plus connus sont ceux utilisés sur des sites web de commerce électronique, par exemple produits d'Amazon, les publicités de produits présentées par Google en fonction de l'historique des recherches, des recommandations de films de Netflix, Yahoo, Movies et MovieLens.

Parmi les approches des systèmes de recommandation, nous distinguons l'approche de filtrage collaboratif qui se base sur les notations des utilisateurs pour trouver ceux qui sont les plus similaires. Ces notations se trouvent dans une matrice de $n \times m$, n étant le nombre des utilisateurs et m le nombre d'items. Plus précisément, le filtrage collaboratif consiste à remplir la matrice de notation de sorte à prédire la note qu'un utilisateur va donner à un item en se basant sur les notations déjà fournies par d'autres utilisateurs.

Pour notre projet, nous nous intéressons aux systèmes de recommandations des points d'intérêt (POI recommender system) qui est une forme un peu plus avancée des systèmes de recommandations traditionnelles mais elle se base sur le filtrage collaboratif, les applications qui utilisent initialement ces approches sont les réseaux sociaux basés sur la localisation (LBSN).

D'autre part, deux problèmes influencés par le SR se présentent, d'un côté, le problème de passage à l'échelle ; et d'un autre côté, le manque des données, les données d'enregistrements des POI sont très rares.

L'objectif de notre travail est de remédier à ces deux problèmes. Pour cela, nous utilisons l'algorithme de factorisation matricielle en utilisant une de ses méthodes qui est la décomposition en valeur singulière « SVD », qui est l'un des méthodes utilisées par les services en ligne comme Netflix pour d'accélérer la recherche de recommandations de contenu pour les utilisateurs.

Pour réaliser notre travail, nous allons concevoir un site web qui intègre un système de recommandation de POI pour la recommandation des lieux touristiques de la ville d'Oran.

Outre l'introduction et la conclusion, ce rapport est articulé en 4 chapitres :

Le premier chapitre : définit les systèmes de recommandations. Ensuite, nous détaillons ses différentes approches. Enfin, nous citons les problèmes, et l'avenir de ces systèmes.

Le deuxième chapitre : a pour objectif de définir la factorisation matricielle en donnant ces modèles, après nous avons détaillé le concept de la recommandation de point d'intérêt et ses caractéristiques.

Le troisième chapitre : est une vue conceptuelle de notre système, il contient les diagramme UML

Et le dernier chapitre sera consacré à notre site « Oran Guide », nous commençons par présenter les environnements de développement et les langages de programmation utilisé, suivi par plusieurs interfaces qui représentent notre site web.

Chapitre 1

Les systèmes de recommandation

1.1 Introduction

Dans ce premier chapitre, nous allons, tout d'abord, introduire les principales notions de base de recommandation. Nous commençons par définir le système de recommandation et ses types puis, nous présentons les trois approches principales de filtrage et la comparaison entre eux, nous poursuivons avec les problèmes de ces systèmes. Enfin nous concluons par les systèmes de recommandation collaborative et l'avenir de ces systèmes.

1.2 Définitions des systèmes de recommandation

Les systèmes de recommandation est une forme spécifique de filtrage de l'information visant à présenter les éléments d'information (film, musique...) qui sont susceptibles d'intéresser l'utilisateur.

Les systèmes de recommandation sont des techniques des logiciels fournissent des suggestions d'« Item » à un « Utilisateur ». Généralement ils peuvent connaître les intérêts et les loisirs des utilisateur en fonction de leurs historiques puis prédire leurs notes ou leurs préférences pour un item donné [1].

Les systèmes de recommandation jouent un rôle important dans des sites Internet aussi bien classés que Amazon¹, YouTube², Netflix³, Yahoo⁴, TripAdvisor⁵, Last.fm⁶, IMDb⁷ et delicious⁸ pour des sites Web.

De plus, de nombreuses entreprises de médias développent et déploient actuellement des systèmes de recommandation dans le cadre des services qu'ils fournissent à leurs abonnés [2]. Par exemple Netflix, le service de location de films en ligne, a décerné un prix de 1 million de dollars à l'équipe qui a réussi à améliorer sensiblement les performances de son système de

¹[Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more](#)

²[YouTube](#)

³[Netflix Algérie - Regardez des séries TV en ligne, Regardez des films en ligne](#)

⁴[Yahoo](#)

⁵[Tripadvisor: Read Reviews, Compare Prices & Book](#)

⁶[Last.fm | Play music, find songs, and discover artists](#)

⁷[IMDb: Ratings, Reviews, and Where to Watch the Best Movies & TV Shows](#)

⁸[Agen Judi Bola Sbobet& Situs Daftar Slot Online \(delicious.com\)](#)

recommandation. L'entreprise a intégré les propositions les plus pertinentes dans sa version du système de recommandation mis en production.

En gros, il existe deux entités de base dans les systèmes de recommandation sont :

« Item » est le terme général utilisé pour dénoter ce que le système recommande aux utilisateurs.

« Utilisateur » est la personne qui utilise un système de recommandation, donne son avis sur les différents items et reçoit les nouvelles recommandations du système.

Un système de recommandation requiert généralement 3 étapes :

Etape 1 : Recueillir de l'information sur l'utilisateur ;

Etape 2 : Construire une matrice contenant l'information recueillie ;

Etape 3 : Extraire à partir de cette matrice une liste de recommandations.

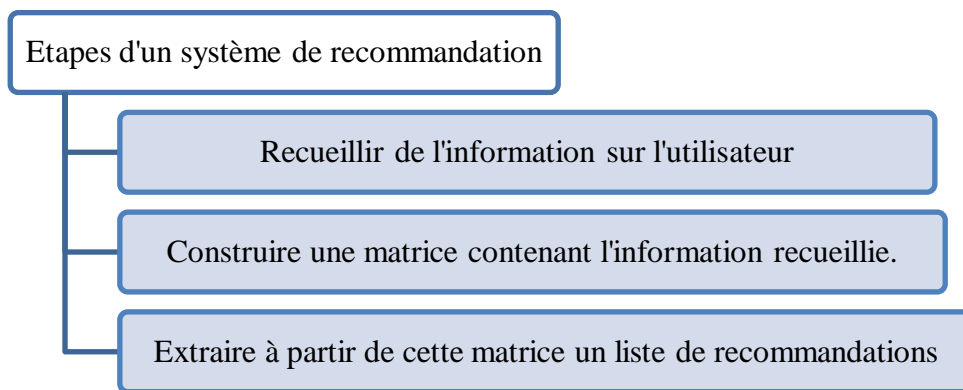


Figure 1 - Etapes d'un système de recommandation

1.3 Types de données de système de recommandation

Nous distinguons deux types de collecte des données : collecte explicite et collecte implicite :

1.3.1 Collecte explicite

Les utilisateurs sont sollicités pour émettre leurs avis sur des items via un système de notation (ex. une grille de 5 étoiles, un questionnaire de satisfaction), ou bien en publiant leurs avis sur un élément donné (ex. La fonction "J'aime" sur le réseau social Facebook⁹).

⁹Facebook

1.3.2 Collecte Implicite

La collecte implicite repose sur les interactions des utilisateurs sur le système. Par exemple le nombre de visites sur une page, le nombre de vues sur une vidéo, le temps passé sur une section donnée.

1.4 Différentes approches de système de recommandation

Les systèmes de recommandations permettent de fournir à l'utilisateur des informations personnalisées en fonctions de ses préférences. Il existe trois grandes approches de filtrage : basé sur le contenu, basé sur le filtrage collaboratif et le filtrage hybride.

1.4.1 Basé sur le contenu

Les systèmes de recommandation basés sur le contenu (content based) fonctionnent en analysant les caractéristiques des items à recommander (lieux, produits, etc.) puis en les regroupant [2].

Dans ce genre de SR, des mots-clés sont utilisés afin de décrire les ressources ; un profil est ensuite créé afin d'indiquer pour chaque utilisateur le type de ressources qu'il aime. Par exemple, nous construisons un profil d'un utilisateur A qui préfère les séries dont les genres préférés sont actions et romances (cas de Netflix). Et, nous essayons de recommander des produits qui de la même section que A préfère.

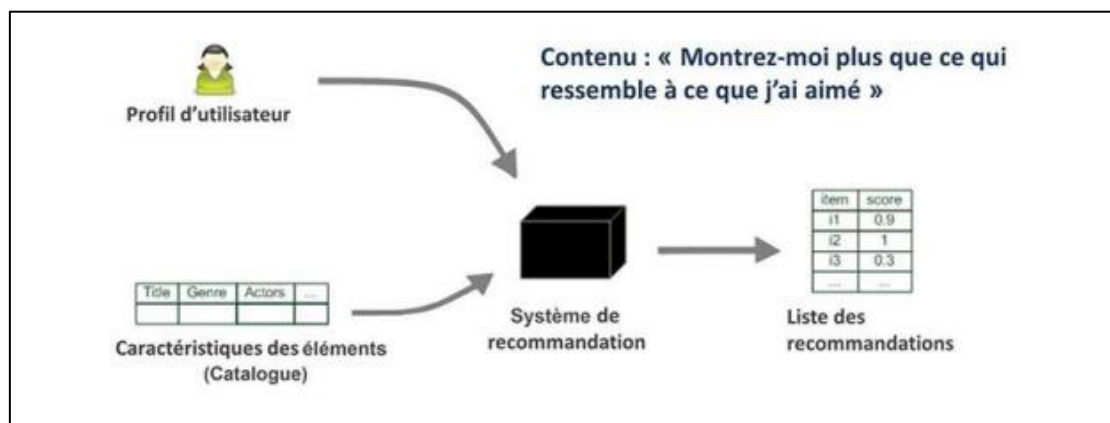


Figure 2 – Filtrage basé sur le contenu [5]

1.4.2 Filtrage collaboratif

Les méthodes basées sur le filtrage collaboratif sont basées sur la collection et l'analyse d'informations sur le comportement des utilisateurs, leurs activités et leurs préférences pour ensuite prédire ce que les utilisateurs sont susceptibles d'aimer d'après leurs similarités avec d'autres utilisateurs [1]. L'idée clef est que la note d'un utilisateur U pour un nouvel item i est susceptible d'être similaire à celle donnée par un autre utilisateur V , si U et V ont noté d'autres items d'une manière similaire. De même, U est susceptible de noter deux items i et j de la même façon, si d'autres utilisateurs ont donné des notes similaires à ces deux items.

Les approches collaboratives dépassent certaines limitations des approches basées sur le contenu. Par exemple, des items dont le contenu n'est pas défini, peuvent quand même être recommandés aux utilisateurs grâce aux feedbacks des autres utilisateurs. De plus, les recommandations collaboratives sont basées sur la qualité des items évaluée par les utilisateurs, au lieu de s'appuyer sur le contenu qui peut être un mauvais indicateur de qualité [1].

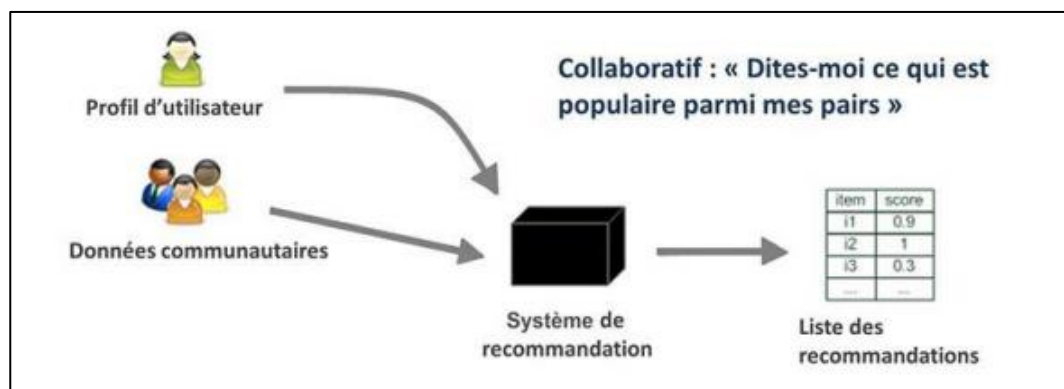


Figure 3 – Filtrage collaboratif [5]

Le filtrage collaboratif est composé de deux méthodes :

- FC Basé sur la mémoire
- FC basé sur le modèle

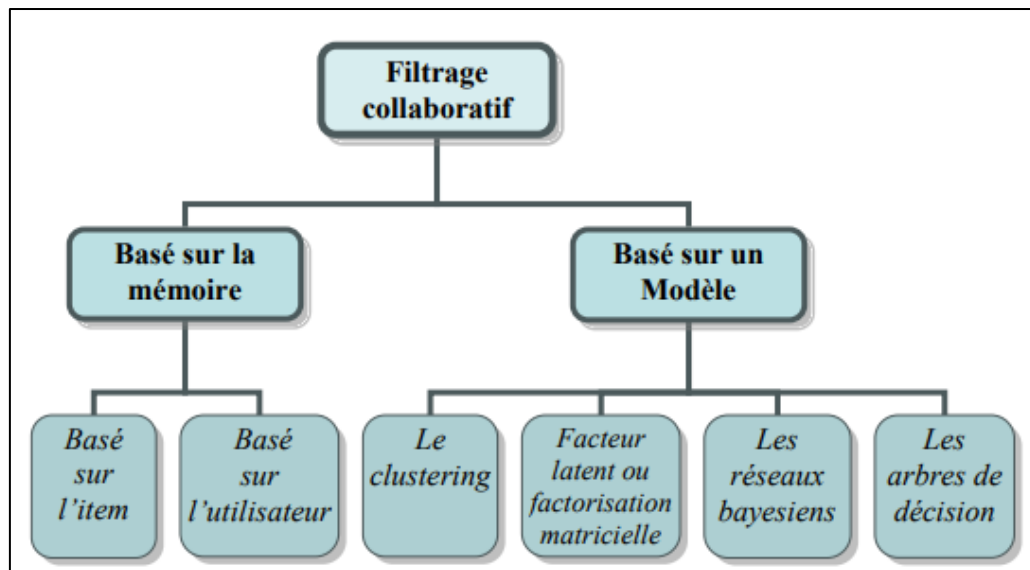


Figure 4 – Méthodes de filtrage collaboratif [2]

1.4.2.1 Filtrage collaboratif basé sur la mémoire

Cette méthode utilise un système de notation pour prédire les préférences d'un utilisateur en tenant compte des préférences d'un utilisateur similaire, ou du « voisin » [3]. Il existe deux façons de calculer les préférences ici, le FC basé sur l'utilisateur et le FC basé sur l'item. Le FC à base de mémoire souffre essentiellement de deux problèmes : le passage à l'échelle et le manque de données. Le passage à l'échelle fait que le temps de calcul est énormément long lors de son application sur des données nombreuses. Le manque de données vient du fait qu'il n'est pas toujours facile de trouver assez de notes communes entre les utilisateurs, particulièrement sur des grandes listes d'items [4].

- **FC basée sur l'utilisateur**

Cette approche est basée sur la similarité entre les utilisateurs. Nous construisons une matrice M [Utilisateur x Item] :

M	Item 1	Item 2	Item 3	Item 4	Item 5
Dina	3	8	9		7
Amelia	3		8		8
Rahim	2	5	7	9	

Tableau 1 –Utilisateur x Item

Dina et Amelia ont des avis similaires sur trois items (item 1, item3 et item 5) et Dina aime l'item 2 donc il sera une bonne recommandation pour Amelia.

- **FC basée sur l’item**

La similarité entre items ici est basée sur le jugement des utilisateurs. Pour que deux items soient considérés voisins (similaires), ils doivent être appréciés par les mêmes utilisateurs [4]. Nous construisons aussi la même matrice M. Dina et Amelia aiment l’item 3 et 5. Cela suggère que les personnes qui aiment l’item 3 aiment aussi l’item 5, donc l’item 5 pourra être recommandé à Rahim qui aime l’item 3.

1.4.2.2 Filtrage collaboratif basé sur le modèle

L’objectif principal du FC à base de modèle est de réduire l’impact du problème de passage à l’échelle et de celui du manque de données.

Dans cette méthode de filtrage collaboratif, différents algorithmes d’exploration de données et d’apprentissage automatique sont utilisés pour développer un modèle pour prédire l’évaluation d’un utilisateur d’un élément non estimé. Quelques exemples de ces modèles sont les réseaux bayésiens [6], les modèles de clustering [7], la factorisation matricielle [8] et les arbres de décision [9].

Dans notre travail, nous allons nous intéresser essentiellement sur la factorisation matricielle, qui consiste à décomposer une matrice en plusieurs autres matrices, et faire le produit de ces matrices entre elles, pour retrouver à la fin la matrice originale.

Comparaison entre le filtrage collaboratif et le filtrage basés sur le contenu

Les approches de recommandation présentées ont des avantages et des inconvénients :

- Le problème principal de ces deux approches réside dans le fait qu’il faut une base d’utilisateurs ayant déjà fait des choix pour faire des recommandations.
- Le SR basé sur le contenu requiert la connaissance de l’utilisateur seulement, contrairement aux systèmes basés sur le filtrage collaboratif qui nécessite une connaissance sur tous les utilisateurs.
- Le filtrage collaboratif se distingue par la diversité des items à recommander à l’utilisateur. Par exemple, un utilisateur qui a des voisins similaires du point de vue des politiques, peut se voir recommander des articles de sport si ces voisins aiment les articles de sport, même si cet utilisateur n’a jamais exprimé ce genre de favoris. Au contraire des SR basés sur le contenu si par exemple un utilisateur ne s’intéresse qu’aux articles parlant de la médecine, il ne se verra jamais proposer un article de cuisine.

1.4.3 Filtrage Hybride

Un système de recommandation hybride utilise des composants de différents types d'approches de recommandation ou s'appuie sur leur logique. En général, l'hybridation s'effectue en deux phases :

- Appliquer séparément le filtrage collaboratif et autres techniques de filtrage pour générer des recommandations candidates.
- Combiner ces ensembles de recommandations préliminaires selon certaines méthodes telles que la pondération, la mixtion, la cascade, la commutation, etc. afin de produire les recommandations finales pour les utilisateurs.

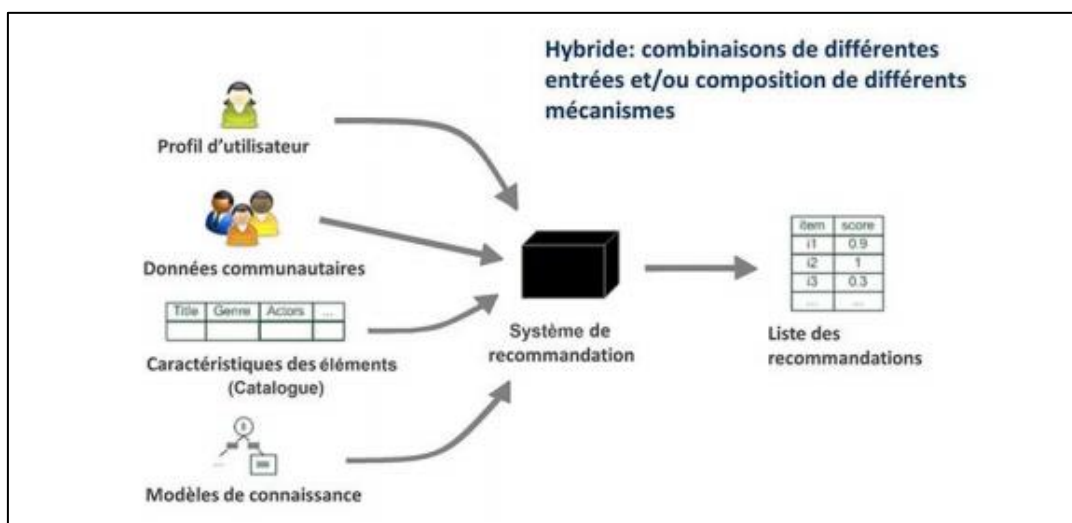


Figure 5 – Filtrage hybride [5]

1.5 Les problèmes des Systèmes de Recommandation

Démarrage à froid : Un problème des plus connus dans les systèmes de recommandation est lorsque l'on ne détient aucune ou peu d'informations sur un utilisateur, il y a deux grands types de ce problème :

- Cas du nouvel utilisateur : est qu'un nouvel utilisateur qui n'a pas encore accumulé suffisamment d'évaluations ne peut pas avoir de recommandations pertinentes.
- Cas du nouvel item : le système doit avoir suffisamment d'évaluations pour que l'item soit pris en considération dans le processus de recommandation.

Rareté (Sparsity) : Quand des utilisateurs ont des goûts atypiques (qui varient de la norme), ils n'auront pas beaucoup d'utilisateurs en tant que voisins. Cela mènera à des recommandations pauvres.

Passage à l'échelle (Scalability) : C'est l'aptitude d'une solution à un problème de fonctionner même si la taille du problème augmente [10]. Dans notre contexte, il désigne la capacité d'un algorithme à traiter avec la même efficacité des volumes de données nombreuses.

Manque de données : l'identification des appréciations des utilisateurs est l'un des piliers de base du processus de recommandation. Les données relatives aux ces appréciations manquent et s'avèrent insuffisantes pour le bon fonctionnement du système de recommandation.

En effet, la quantité de données ou de notes disponible demeure toujours insuffisante pour pouvoir prédire correctement les notes manquantes. Donc, en raison de ce manque de données, la modélisation des utilisateurs devient complexe [11].

1.6 Les Systèmes de Recommandation collaboratif

Les systèmes de recommandation ont été étudiés dans plusieurs domaines : la recherche d'informations, le web, le e-commerce, et bien d'autres.

Pour le e-commerce plusieurs sites utilisent des algorithmes de recommandation pour personnaliser la boutique en ligne pour chaque client, tel que Amazon l'un des géants du Web. Elle utilise les recommandations comme un outil de marketing ciblé.

En effet, Amazon utilise des algorithmes de recommandation pour personnaliser son site web en fonction des intérêts de chaque client. Ces algorithmes sont basés sur le filtrage collaboratif Item-to-Item.

Netflix, le service de location de films en ligne a voulu améliorer leur propre moteur de recommandation. Pour ce faire Netflix a lancé une compétition de recommandation mettant en jeu un million de dollars. Autour de cet événement se sont cristallisées les principales avancées de la recherche de ces dernières années dans le monde de la recommandation.

YouTube aussi, le site web d'hébergement de vidéos et un média social sur lequel les utilisateurs peuvent envoyer, regarder, commenter, évaluer et partager des vidéos en streaming, son l'algorithme de recommandation ne se contente pas de guider le comportement des utilisateurs, il affecte également considérablement les créateurs de vidéos. Le sujet de la vidéo, la durée, ... etc. [12]

En effet, un réseau de neurones artificiel filtre les vidéos afin de déterminer si elles pourraient être retenues pour figurer dans la section « À suivre » de l'utilisateur (selon l'historique de ce dernier et le contenu visionné par des utilisateurs semblables).

L'idée n'est pas d'identifier de « bonnes » vidéos, mais de proposer des vidéos qui correspondent aux envies des utilisateurs, le but ultime étant qu'ils passent le plus de temps possible sur la plateforme (et donc visionnent le plus de publicités possible).

En 2018, le centre de recherche américain Pew Research a déduit que plus les utilisateurs passent de temps sur YouTube, plus l'algorithme leur proposera des vidéos longues et populaires.

1.7 L'avenir des Systèmes de Recommandation

La recherche dans le domaine des systèmes de recommandation n'en est qu'à ses débuts, l'engouement des chercheurs au cours de ces vingt dernières années a fait grandement avancer les choses. Il existe plusieurs livres qui présentent un bilan exhaustif des systèmes de recommandation, montrent déjà une certaine maturité du domaine de recherche, comme par exemple le livre *Recommender Systems Handbook*¹⁰.

Des workshops et des thèmes de recherche consacrés aux systèmes de recommandation ont fait leur apparition dans les principalement conférences internationales en apprentissage artificiel et en fouille de données ou dans des conférences spécialisées dans le domaine. L'évolution constante de sociétés sur Internet ayant de réels besoins en termes de recommandation assure un avenir florissant à ce domaine de recherche.

1.8 Conclusion

Dans ce chapitre, nous avons donné une définition des systèmes de recommandation en citant les différentes approches et les détaillées en donnant des exemples pour mieux comprendre la notion de ces systèmes. Après nous avons cité les problèmes de ces systèmes, tel que le manque de données et le passage à l'échelle. De là pour résoudre ces deux problèmes de SR nous allons utiliser la factorisation matricielle. Nous allons la détailler dans le chapitre suivant.

¹⁰[Recommender Systems Handbook | SpringerLink](#)

Chapitre 2

Factorisation matricielle pour les systèmes de recommandation

2.1 Introduction

Dans ce chapitre, nous commençons d'abord par définir ce qu'est une factorisation matricielle puis, nous présentons son principe de fonctionnement et ses modèles. Enfin nous définissons les points d'intérêts et ses caractéristiques.

2.2 La factorisation matricielle FM

Factorisation matricielle est parmi les algorithmes utilisés dans le filtrage collaboratif qui appartient aux méthodes basées sur le modèle.

La factorisation matricielle donne généralement une représentation plus compacte que l'apprentissage de la matrice complète [3].

2.2.1 Définition

La matrice de factorisation est une matrice d'évaluation qui permet d'identifier les utilisateurs et ses interattractions sous forme des notes par rapport aux items dans un matrice de taille $N \times M$ tel que M les notes des utilisateurs et N la listes des items.

Son principe est de décomposer une matrice en deux ou plusieurs autres matrices, et de calculer par la suite le produit de ces matrices pour obtenir la matrice des prédictions.

2.2.2 Principe de fonctionnement du FM

La plupart des modèles de factorisation matricielle sont basés sur le modèle du facteur latent. C'est l'approche la plus utilisé pour réduire le problème de sparsity et le manque de données dans la matrice d'un système de recommandation [1].

Tout d'abord, nous avons un ensemble d'utilisateurs \mathbf{T} et un ensemble des films \mathbf{F} .

Soit \mathbf{R} la matrice de taille $(m \times n)$ qui contient toutes les notes que les utilisateurs ont attribuées aux films (rating matrix).

La méthode consiste à trouver deux matrices \mathbf{U} ($\mathbf{m} \times \mathbf{d}$) et \mathbf{V} ($\mathbf{d} \times \mathbf{n}$), tel que \mathbf{d} est le nombre facteur latent (un entier positif) et leur produit qui se rapproche étroitement de la matrice originale \mathbf{R} est : $\mathbf{R} \approx \mathbf{V}^t \times \mathbf{U}$

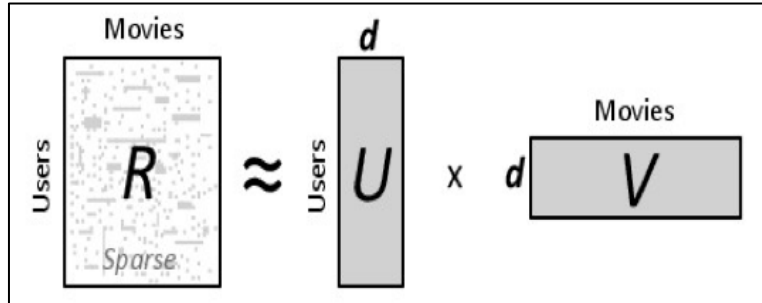


Figure 6 – Matrice de factorisation [8]

La note \mathbf{R} d'un utilisateur \mathbf{t} pour un film \mathbf{f} est :

$$R_{tf} \approx V_f^t \times U_t$$

Où l'élément \mathbf{t} est associé à un vecteur $\mathbf{U}_t \in \mathbf{R}_f$ et l'utilisateur \mathbf{t} est associé à un vecteur $\mathbf{V}_t \in \mathbf{R}_f$

L'objectif de cette méthode est d'apprendre les vecteurs \mathbf{U} et \mathbf{V} de la matrice de notation \mathbf{R} , où \mathbf{U} exprime la notation d'un élément en termes de facteurs \mathbf{d} et \mathbf{V} exprime l'intérêt des utilisateurs pour les facteurs.

L'identification de facteurs latents peut se faire à travers plusieurs modèles de factorisation matricielle, dont certains sont : la décomposition en valeur singulière (SVD), l'analyse des composantes principales (PCA), factorisation matricielle probabiliste (PMF), factorisation matricielle non négative (NMF) etc.

2.2.3 Les modèles de la FM

Dans les deux points suivants nous allons détailler les deux modèles du FM les plus utilisés :

2.2.3.1. La décomposition en valeur singulière (SVD)

La décomposition en valeur singulière (SVD) (en Anglais Singular Value Decomposition) est la technique la plus puissante de réduction de la dimensionnalité [11].

SVD d'une matrice \mathbf{A} de taille $\mathbf{m} \times \mathbf{n}$ est de la forme : $\mathbf{R} = \mathbf{U} \mathbf{S} \mathbf{V}^t$

Où \mathbf{U} et \mathbf{V} sont deux matrices orthogonales de taille $(\mathbf{m} \times \mathbf{k})$ et $(\mathbf{k} \times \mathbf{n})$ où \mathbf{k} est le rang de la matrice \mathbf{R} . \mathbf{S} est une matrice $\mathbf{k} \times \mathbf{k}$ et contient toutes les valeurs singulières de \mathbf{R} comme ses entrées diagonales.

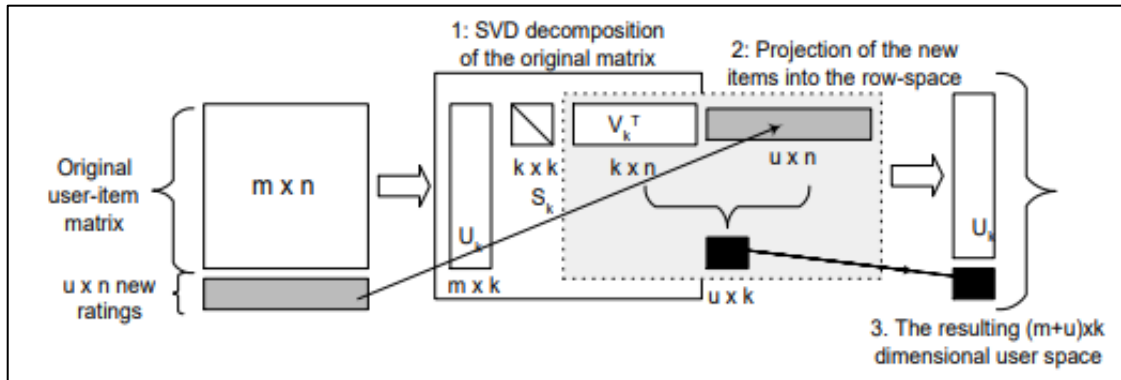


Figure 7 – La Décomposition en valeur singulière SVD [13]

La dimensionnalité des matrices peut être réduite en choisissant \mathbf{S} pour ne contenir que les \mathbf{k} plus grandes valeurs singulières de \mathbf{R} . Cela fait de \mathbf{S}_k une matrice $\mathbf{k} \times \mathbf{k}$, et puisque la réduction peut également être appliquée aux matrices \mathbf{U} et \mathbf{V} en supprimant les colonnes (\mathbf{r}_k) des lignes \mathbf{U} et (\mathbf{r}_k) de \mathbf{V} , la dimensionnalité de \mathbf{USV}^T peut être considérablement réduite en $\mathbf{U}_k \mathbf{S}_k \mathbf{V}_k^T$, ce qui permet d'économiser de l'espace et de la puissance de calcul en plus de supprimer les données de bruit de \mathbf{R} [1].

Le résultat de l'exécution de SVD sur \mathbf{R} est donc \mathbf{R}_k , une version de dimension réduite de \mathbf{R} [11].

2.2.3.2. Analyse des composants principaux PCA

L'analyse des composantes principales (ACP) (en Anglais Principal Component Analysis PCA) est une réalisation particulière de l'approche de la factorisation de la matrice (FM).

L'ACP est une procédure statistique qui utilise une transformation orthogonale pour convertir un ensemble d'observations de variables éventuellement corrélées en un ensemble de valeurs de variables linéairement non corrélées appelées composantes principales. Le nombre de variables d'origine est supérieur ou égal aux composants principaux.

Cette transformation est définie de telle sorte qu'une projection linéaire de données dimensionnelles élevées dans un sous-espace dimensionnel inférieur tel que la variance retenue est maximisée et l'erreur de reconstruction de moindre carré est minimisée. Les principaux composants sont orthogonaux car ils sont les vecteurs propres de la matrice de covariance [2].

PCA est sensible à la mise à l'échelle relative des variables d'origine. PCA permet d'obtenir une liste ordonnée de composants qui représentent la plus grande quantité de variance à partir des données en termes d'erreurs de moindres carrés [2].

La quantité de variance saisie par le premier composant est supérieure à la quantité de variance sur le deuxième composant et ainsi de suite. Nous pouvons réduire la dimensionnalité des données en négligeant ces composants [13].

2.3 Définition de point d'intérêts (POI)

Un point d'intérêt (en Anglais Point OF Interest) est un site spécifique identifié de manière unique généralement associé à une catégorie spécifique d'activités (par exemple, hôtel, restaurant, centre commercial, etc.).

De même pour les enregistrements dans les Réseaux sociaux basés sur la localisation (LBSN), un point d'intérêt est généralement également associé à un contenu qui correspond à l'ensemble de tous les commentaires, images, opinions que les utilisateurs ont téléchargés lors des enregistrements qu'ils ont effectués. Par exemple, dans le tableau 2, le point d'intérêt de la Tour Eiffel est associé à certaines des catégories auxquelles il aspire.

Cependant, dans de nombreux cas pratiques, les catégories ou autres descriptions de points d'intérêt ne sont pas divulguées pour diverses raisons (par exemple, vie privée, confidentialité, etc.). C'est pourquoi dans nos approches nous avons supposé que nous ne connaissions que les emplacements, c'est-à-dire les paires (latitude, longitude) pour tous les points d'intérêt [14].

point-of-interest name	Tour Eiffel
#Checkins	7.097.302
Location	Latitude: 48.858° Longitude: N, 2.294° E
Categories	Vues panoramiques, monument, divertissement, loisirs

Tableau 2 – Un exemple de points d'intérêt et ses informations associées [14]

2.4 Les caractéristiques de point d'intérêts

Les systèmes de recommandation POI présentent les caractéristiques uniques suivantes :

2.4.1 Influence géographique

Comme le rapportait la première loi de géographie de Tobler, « tout est lié à tout le reste, mais les choses proches sont plus liées que les choses éloignées » [15].

Pour les LBSN, la première loi de géographie de Tobler implique que les utilisateurs préfèrent visiter les emplacements à proximité plutôt que les emplacements distants et les utilisateurs peuvent être intéressés par les POI entourés d'un POI que les utilisateurs préfèrent.

L'influence géographique est la caractéristique la plus importante qui distingue les systèmes de recommandation de POI des systèmes de recommandation traditionnels et influe fortement sur les comportements de visualisation des utilisateurs [16].

2.4.2 Données de fréquence et parcimonie

Dans les systèmes de recommandation traditionnels, les utilisateurs ont généralement exprimé leurs préférences en fournissant explicitement des évaluations pour des éléments (par exemple, un livre, un film, de la musique, etc.), qui sont convertis en matrice d'évaluation des éléments de l'utilisateur. Les notes sont souvent des valeurs numériques et tombent dans une plage numérique, telle que [1,5]. La note la plus élevée correspond à la meilleure satisfaction. Contrairement aux systèmes de recommandation traditionnels, les préférences d'un utilisateur sont reflétées par la fréquence d'enregistrement des emplacements, qui sont souvent transformées en matrice de fréquence d'enregistrement de l'emplacement de l'utilisateur. Les données de fréquence ont une large gamme par rapport aux évaluations. Par exemple, l'utilisateur peut s'enregistrer des milliers de fois à certains endroits, tandis que l'utilisateur peut s'enregistrer plusieurs fois pour d'autres emplacements.

En outre, la rareté de la matrice de fréquence d'enregistrement de l'emplacement de l'utilisateur est considérablement plus élevée que celle de la matrice d'évaluation des éléments de l'utilisateur, ce qui entraîne un défi majeur pour la recommandation de POI. Par exemple, la parcimonie de l'ensemble de données Netflix est d'environ 99%, tandis que la parcimonie de Gowalla¹¹ est d'environ $2,08 \times 10^{-4}$ [16].

2.4.3 Influence sociale

Partant du principe que les amis ont tendance à partager des intérêts plus communs et que les utilisateurs ont souvent tendance à demander à leurs amis des suggestions, les systèmes de recommandation traditionnels combinent les relations sociales avec les évaluations pour améliorer la qualité des recommandations.

Plusieurs études ont montré que les relations sociales se révèlent bénéfiques pour les SR. Cependant, dans les systèmes de recommandation des POI, des études antérieures ont montré

¹¹[SNAP: Network datasets: Gowalla \(stanford.edu\)](#)

qu'environ 96% des utilisateurs partagent moins de 10% d'intérêts communs visités, ont indiqué qu'un grand nombre d'amis ne partagent rien en termes de POI [16].

Par conséquent, l'influence sociale a des effets limités sur les comportements d'enregistrement des utilisateurs.

2.5 Conclusion

Dans ce chapitre, nous avons donné une définition de factorisation matricielle en citant sa méthode d'application et ses modèles. Après nous avons définissez le point d'intérêts en donnant un exemple. Enfin, nous avons donné ses caractéristiques.

Le prochain chapitre sera consacré à la partie conception et réalisation de notre système de recommandation.

Chapitre 3

Conception et modélisation du système

3.1 Introduction

Avant de passer au côté applicatif du projet, nous devons passer tout d'abord par la phase de conception. Dans ce chapitre nous allons présenter notre conception du projet en utilisant le langage de modélisation UML.

3.2 Méthode d'Analyse et conception

3.2.1 Définition UML

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer les divers aspects d'un système d'information. [17]

Ce langage est certes issu du développement logiciel, mais pourrait être appliqué à toute science fondée sur la description d'un système [18].

3.2.2 Spécification des besoins du système

Dans cette étape nous allons identifier les fonctionnalités de notre système, en déterminant les acteurs et les besoins fonctionnels du site.

3.2.2.1. Identification des acteurs

Dans notre système nous avons identifié un seul acteur qui est :

- ✚ **L'utilisateur** : C'est l'acteur qui utilise le système et qui peut accéder à tous les fonctionnalités du site.

3.2.2.2. Identification des besoins

➤ Les besoins fonctionnels

Le système doit permettre à l'utilisateur de :

- S'authentifier
- Consulter l'historique
- Rechercher des POI
- Afficher les informations sur un POI

- Ajouter des POI aux favoris
- Avoir des recommandations personnalisées

Le système doit permettre au visiteur de :

- S'inscrire
- Rechercher des POI
- Afficher les informations sur un POI

➤ **Les besoins non fonctionnels**

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

- a. **Fiabilité** : le site doit fonctionner de façon cohérente sans erreurs et doit être satisfaisant.
- b. **Les erreurs** : Les ambiguïtés doivent être signalées par des messages d'erreurs bien organisés pour bien guider l'utilisateur et le familiariser avec notre site web.
- c. **Ergonomie et bonne Interface** : le site doit être adaptée à l'utilisateur sans qu'il ne fournisse aucun effort (utilisation claire et facile).
- d. **Sécurité** : Notre solution doit respecter surtout la confidentialité des données personnelles des clients qui reste l'une des contraintes les plus importantes dans les sites web.

3.3 Les diagrammes UML

3.3.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. La Figure 8 montre le diagramme de cas d'utilisation de notre système.

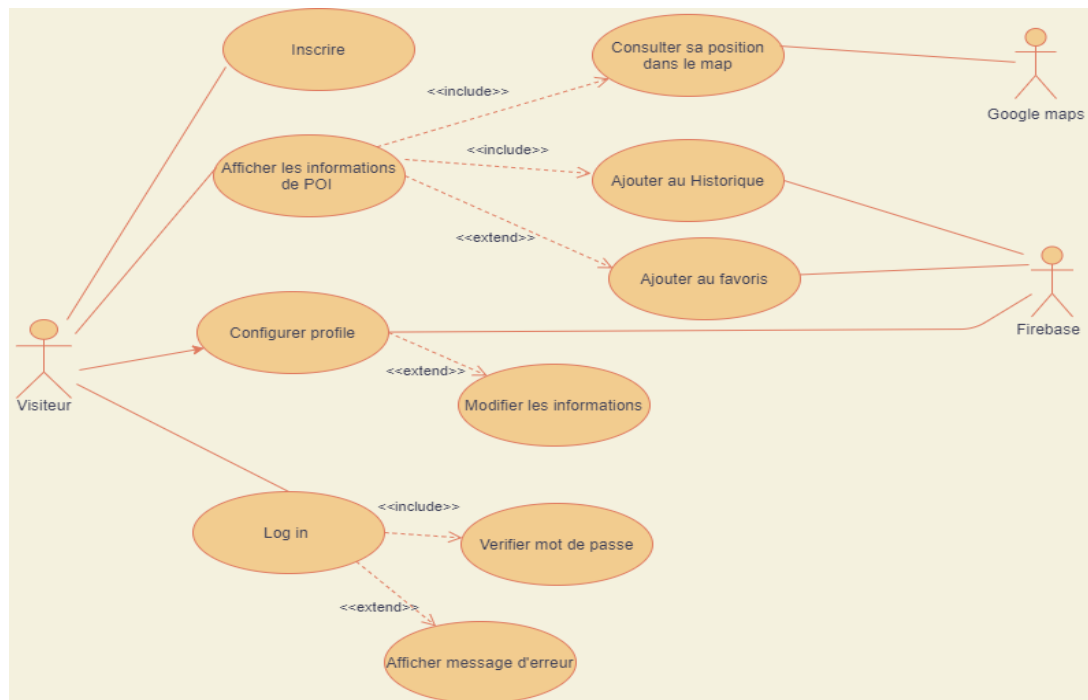


Figure 8 – Diagramme de cas d'utilisation

3.3.2 Description textuelle

3.3.2.1. Description textuelle du cas d'utilisation « Inscription »

Scénario 1	Inscription
Inscription Objectif	Créer un compte dans notre site
Acteurs principaux	Utilisateur
Pré conditions	Disposition d'une connexion Internet
Post conditions	Création d'un compte
Scénario nominal	<ol style="list-style-type: none"> 1. Le visiteur demande l'interface de création de compte 2. Le système affiche l'interface de création de compte 3. Le visiteur remplit les champs requis 4. Le système vérifie les champs introduit par le visiteur 5. Le visiteur accède à la page d'accueil
Scénario alternatif	<p>A1 : Champs obligatoires vide et/ou format incorrect</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur 2. Le système réaffiche l'interface de création de compte <p>A2 : Email déjà existant dans le système</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur « Utilisateur existe déjà » 2. Le système réaffiche l'interface de création de compte.

Tableau 3 – Description textuelle du cas d'utilisation "inscription"

3.3.2.2. Description textuelle du cas d'utilisation « Login »

Scénario 1	S'authentifier
Objectif	Authentification dans le site
Acteurs principaux	Utilisateur
Pré conditions	Avoir un compte
Post conditions	Accès à son compte
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède au site 2. Le système affiche l'interface d'authentification 3. L'utilisateur introduit son email et son mot de passe 4. Le système vérifie les champs introduit par l'utilisateur 5. Le système vérifie l'existence de l'utilisateur <p>Si les informations introduites sont correctes, le système affiche l'accueil du site</p>
Scénario alternatif	<p>A1 : Champs obligatoires vide et/ou non valide</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur 2. Le système reprend à l'étape 2 du scénario nominal <p>A2 : Email et/ ou mot de passe incorrect</p> <ol style="list-style-type: none"> 4. Le système affiche un message d'erreur « Vérifier vos informations et réessayer » 4. Le système réaffiche l'interface de login.

Tableau 4 – Description textuelle du cas d'utilisation "login"

3.3.3 Diagramme de séquence

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation.

3.3.3.1 Diagramme de séquence « Inscription »

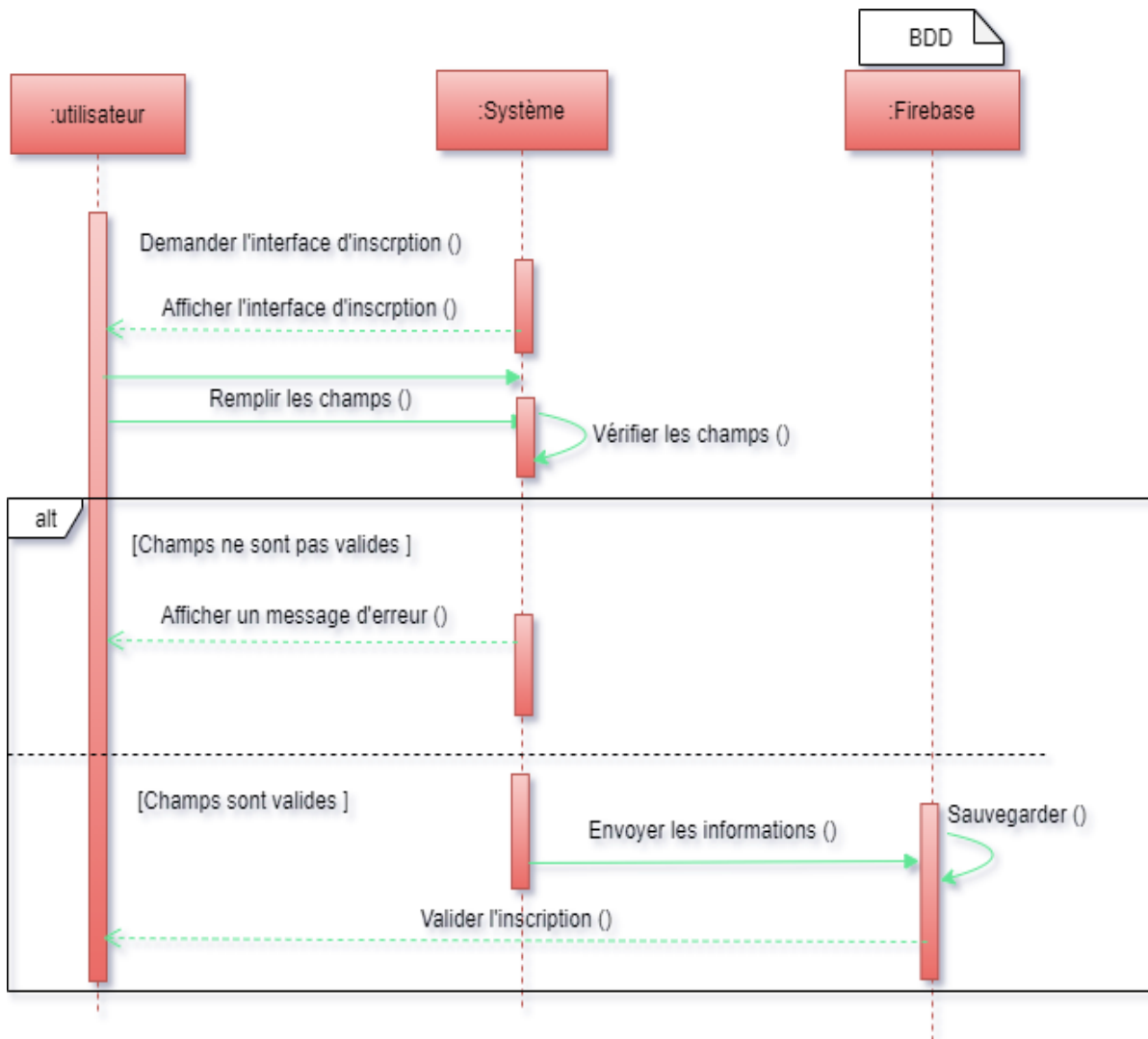


Figure 9 – Diagramme de séquence « Inscription »

3.3.3.2 Diagramme de séquence « login »

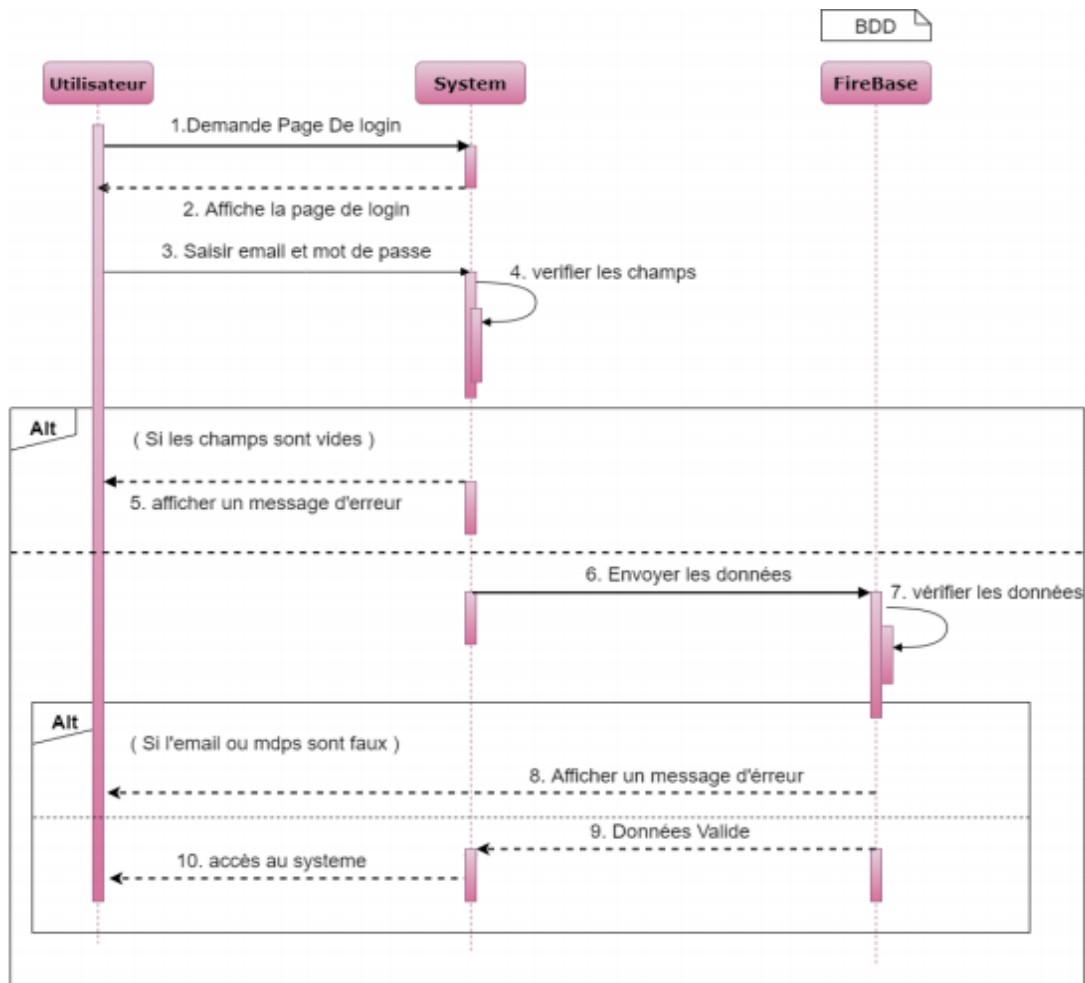


Figure 10 – Diagramme de séquence « Login »

3.3.4 Diagramme d'activité

C'est un Diagramme associé à un objet particulier ou à un ensemble d'objets, qui illustre les flux entre les activités et les actions. Il permet de représenter graphiquement le déroulement d'un cas d'utilisation. [19]. La Figure 11 montre le diagramme d'activité de notre site web.

Diagramme d'activité d'authentification :

- L'utilisateur demande l'authentification en cliquant sur le bouton login.
- Le formulaire d'authentification s'affiche sur l'écran.
- L'utilisateur entre son email et son mot de passe.
- Le système vérifie les coordonnées d'utilisateur sur la base.
- La confirmation du succès ou échec est envoyée à l'utilisateur.

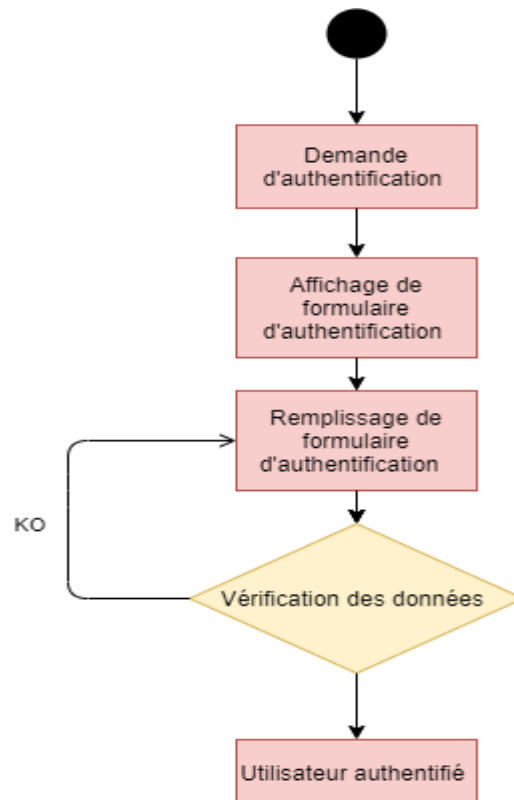


Figure 11 – Diagramme d'activité Authentification

3.3.5 Diagramme de classe

Le diagramme de classes est considéré comme le plus important dans la modélisation orientée objet, et c'est la seule étape obligatoire dans ce type de modélisation. Il représente les classes impliquées dans le système. Un diagramme de classes est une représentation statique des éléments qui composent le système et ses relations. Chaque application qui implémentera le système est une instance des différentes classes qui le composent. [20]. La Figure 12 montre le diagramme de classes de système.

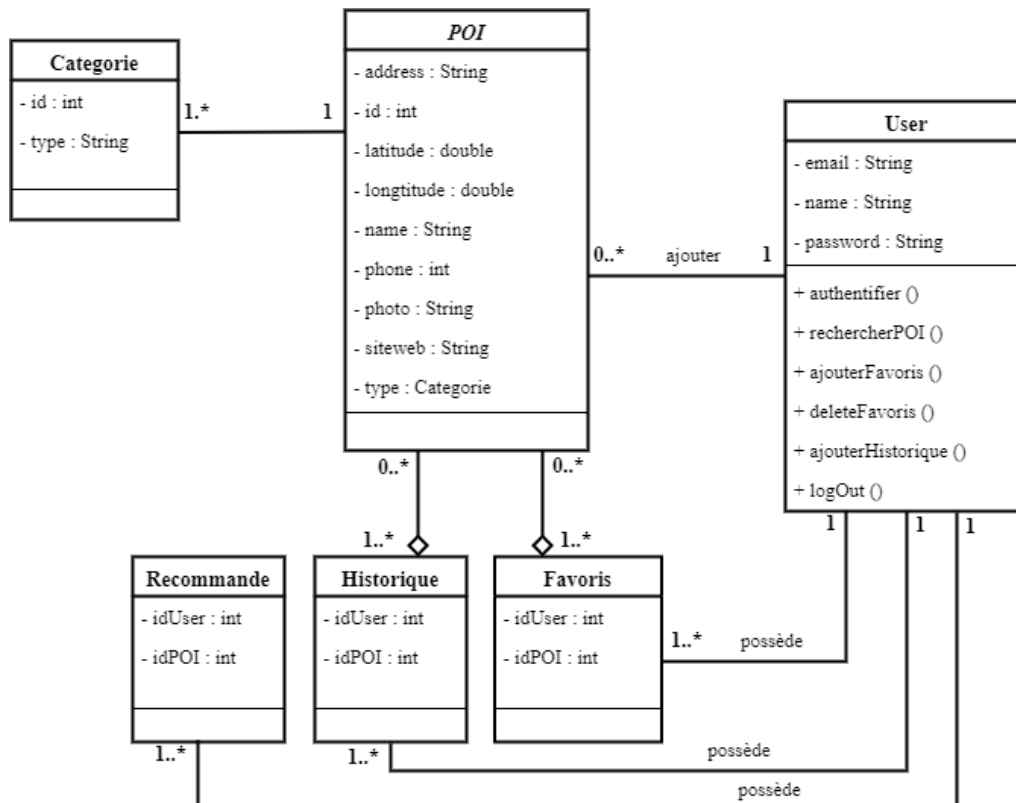


Figure 12 – Diagramme de classe

3.4 Conclusion

Dans ce chapitre, nous avons exposé et expliqué les différentes phases pour la conception et la réalisation de notre système de recommandation, qui consiste à recommander les POI à un utilisateur donné, en utilisant le filtrage collaboratif et le modèle de factorisation matricielle « la décomposition de la valeur singulière (SVD) ». Et modéliser les besoins du système en se basant sur le langage UML.

Le chapitre suivant représentera l’implémentation du système.

Chapitre 4

Implémentation

4.1 Introduction

Dans ce dernier chapitre nous présentons l'ensemble d'outils et d'environnements de programmation utilisées pour l'implémentation de notre site web, ensuite nous avons expliqué le choix de la méthode de recommandation et les calculs qui s'effectue pour l'algorithme du SVD en donnant un exemple. Et nous terminons par quelques interfaces qui présentent notre site.

4.2 Environnement de développement

4.2.1 Draw.io

Un éditeur de diagramme en ligne et sur desktop entièrement gratuit construit autour de Google Drive (TM) et basée sur HTML, qui vous permet de créer des organigrammes, UML, relation d'entité, diagrammes de réseau, maquettes et plus encore, très facile à utiliser, et un outil très léger aussi (environ 50 mb). [21]

4.2.2 Flutter

Flutter est un Framework de développement d'applications multiplateforme, conçu par Google. Flutter met à disposition une grande variété de bibliothèques d'éléments d'IU standard pour Android et iOS. Le Flutter SDK se base sur le langage de programmation Dart également développé par Google. Il se veut le successeur moderne du langage JavaScript classique et, tout comme ce dernier, il s'exécute directement sur les navigateurs, sous forme d'application web. Les programmes Dart peuvent aussi être exécutés directement sur un serveur. [22]

L'objectif est de permettre aux développeurs de fournir des applications hautes performances qui semblent naturelles sur différentes plates-formes. Flutter fonctionne avec le code existant, il est utilisé par les développeurs et les organisations du monde entier et est gratuit et open source.[23]

4.2.3 Visual Studio Code

Visual Studio Code ou VSC pour faire court est l'un des éditeurs de code source les plus populaires utilisés par les programmeurs. C'est rapide, léger et puissant aussi ! Hautement configurable, la fonctionnalité d'extension est sans aucun doute la caractéristique la plus utile. Microsoft a conçu VSC comme un éditeur de code multiplateforme pour l'écriture d'applications Web et cloud.[24]

4.2.4 FireBase

Firebase est une base de données No SQL hébergée dans le Cloud, c'est-à-dire qu'elle s'exécute sur un cloud et que l'accès à l'utilisateur est fourni en tant que service. Il stocke les données dans le format JSON (Javascript Object Notation), un format pour stocker ou transporter des données. Tous les utilisateurs qui y sont connectés peuvent accéder aux données en temps réel. Firebase fournis plusieurs services [25] :

- **Authentification** - connexion et identité de l'utilisateur
- **Base de données en temps réel** - en temps réel, hébergée dans le cloud, base de données No SQL
- **Cloud Firestore** - base de données No SQL en temps réel, hébergée dans le cloud
- **Cloud Storage** - stockage de fichiers massivement évolutif
- **Fonctions cloud** - sans serveur, back end événementiel
- **Hébergement FireBase** - hébergement Web mondial
- **Kit ML** : SDK pour les tâches ML courantes
- **Analytiques** - comprendre vos utilisateurs et comment ils utilisent votre application
- **Prédictions** - appliquez l'apprentissage automatique à l'analyse pour prédire le comportement des utilisateurs
- **Messagerie cloud** - envoyer des messages et des notifications aux utilisateurs
- **Messagerie In-App** - engagez vos utilisateurs actifs avec des messages ciblés

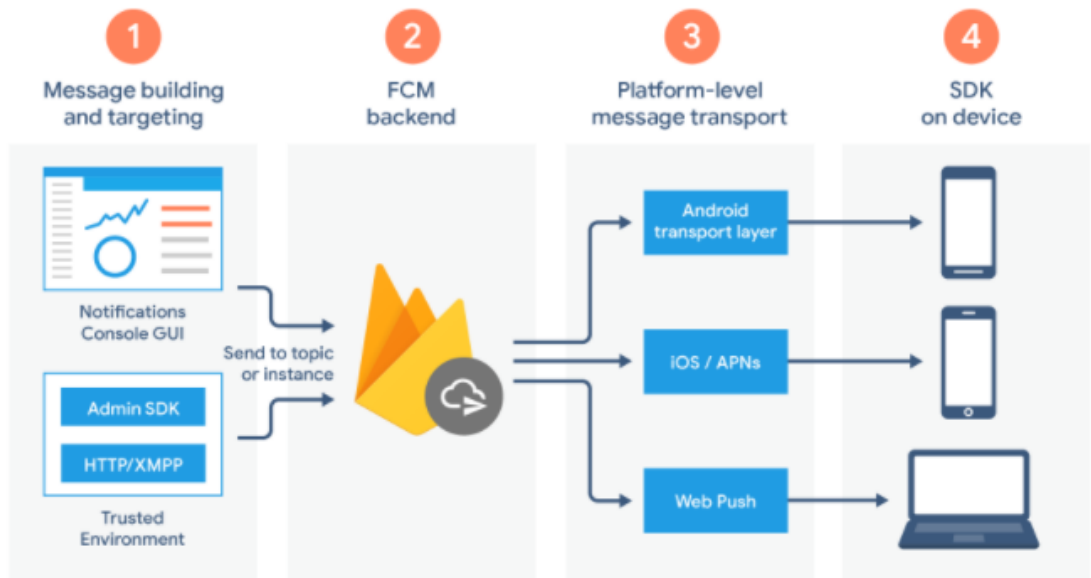


Figure 13 – Architecture de Firebase [26]

4.2.5 Illustrator

Entre les outils professionnels du dessin graphique nous allons trouver Adobe Illustrator comme un des plus utilisés. Ses différentes options l'ont érigé comme un standard entre les dessinateurs de graphismes vectoriels. Il permet de créer des logos, des icônes, des esquisses, des typos et des illustrations complexes pour toute sorte de support, imprimé ou numérique [29].

4.3 Langages de programmation

4.3.1 Dart

Un langage open source, purement orienté objet, typé en option et un langage basé sur les classes qui offre un excellent support pour la programmation fonctionnelle et réactive. Contrairement à C# ou Java. En fait, c'est une langue relativement simple, moderne et très efficace pour travailler avec, Dart est assez facile à apprendre et il rend l'expérience de développement très fluide. [27]

4.3.2 JSON

JSON (JavaScript Object Notation) est un format d'échange de données léger. Il est facile pour les humains de lire et d'écrire. Il est facile pour les machines d'analyser et de générer. Il est basé sur un sous-ensemble de la norme de langage de programmation JavaScript ECMA-262 3e

édition - décembre 1999. JSON est un format de texte qui est complètement indépendant du langage mais utilise des conventions qui sont familières aux programmeurs de la famille C des langages, y compris C, C ++, C #, Java, JavaScript, Perl, Python et bien d'autres. Ces propriétés font de JSON un langage d'échange de données idéal. [28]

4.4 Algorithme de recommandation utilisé :

Nous avons choisi le filtrage collaboratif basé sur le modèle qui réduit l'impact du problème de passage à l'échelle et de celui du manque de données. Et Nous avons utilisé la décomposition en valeur singulière qui est l'une des algorithmes de la factorisation matricielle.

Le SVD est une manière de factoriser une matrice non-carré qui généralise l'opération de décomposition en valeurs propres (pour une matrice symétrique) [30]. Cette décomposition est unique.

Notre constat que tout d'abord les calculs sont difficiles à mener avec des matrices très creuses (nombreuses valeurs nulles). Donc nous avons des traitements à faire avant d'appliquer l'algorithme SVD :

- Nous Construisons une matrice d'éléments utilisateur à partir des enregistrements d'interaction M_{ij} (User i, POI j).

Le tableau 5 représente un fragment de notre base de données dont nous allons utiliser pour effectuer les calculs :

	111 Restaurant	Planet Food	Pizza Up	Havana Coffee	Cafeteria 3d
U ₁	1	0	1	1	0
U ₂	1	0	0	1	1
U ₃	1	0	0	0	1
U ₄	1	0	1	1	1
U ₅	0	0	1	1	0

Tableau 5 – Exemple de notation USER / POI

Le 1 signifie qu'un POI j est favorisé par un User i, le 0 sinon.

- Nous calculons la moyenne de chaque user par la formule suivante :

$$Moyenne_{user} = \frac{\text{Le nombre des poi favorisé de chaque user}}{\text{le nombre de toute les poi}}$$

Nous obtenons le tableau suivant :

Utilisateur	U ₁	U ₂	U ₃	U ₄	U ₅
La moyenne	0,6	0,6	0,4	0,8	0,4

Tableau 6 – Utilisateur / Moyenne

L'implémentation de cette étape en langage Dart :

```

for (int i = 0; i < doc.docs.length; i++) {
  users.add(doc.docs[i].id);
  usersname.add(doc.docs[i].data()['name']);
  await FirebaseFirestore.instance
    .collection('users') .doc(users[i]) .collection('favorit') .get() .then((docs) {
    if (docs.docs.isNotEmpty) {
      // ignore: deprecated_member_use
      List<String> fav = new List<String>();
      // ignore: deprecated_member_use
      List<String> favname = new List<String>();

      for (int d = 0; d < docs.docs.length; d++) {
        fav.add(docs.docs[d].id);
        favname.add(docs.docs[d].data()['name']);
      }
      sum = 0;
      for (int m = 0; m < listRow.length; m++) {
        listRow[m] = 0;
      }
      for (int k = 0; k < poiL.length; k++) {
        for (int l = 0; l < favname.length; l++) {
          if (poiL[k] == favname[l]) {
            listRow[k] = 1;
            sum++;
          }
        }
      }
    }
  });
  mean = sum / poiL.length;
}

```

- Ensuite la normalisation de la matrice :

La formule est : $M_{user,poi} - Moyenne_{user}$

	111 Restaurant	Planet Food	Pizza Up	Havana Coffee	Cafeteria 3d
U ₁	0,4	-0,6	0,4	0,4	-0,6
U ₂	0,4	-0,6	-0,6	0,4	0,4
U ₃	0,6	-0,4	-0,4	-0,4	0,6
U ₄	0,2	-0,8	0,2	0,2	0,2
U ₅	-0,4	-0,4	0,6	0,6	-0,4

Tableau 7 – Matrice normalisé

L'implémentation de cette étape en langage Dart :

```

for (int i = 0; i < matrixColumn.length; i++) {
  List<double> vnorm = [];
  for (int j = 0; j < matrixColumn[i].length; j++) {
    double n = matrixColumn[i][j] - minlista[i];
    vnorm.add(n);
  }
  print(vnorm);
  norm.add(vnorm);
}

```


- La matrice transposée de la matrice normalisée

	U ₁	U ₂	U ₃	U ₄	U ₅
111 Restaurant	0,4	0,4	0,6	0,2	-0,4
Planet Food	-0,6	-0,6	-0,4	-0,8	-0,4
Pizza Up	0,4	-0,6	-0,4	0,2	0,6
Havana Coffee	0,4	0,4	-0,4	0,2	0,6
Cafeteria 3d	-0,6	0,4	0,6	0,2	-0,4

Tableau 8 – Matrice transposé

```

for (int i = 0; i < norm[0].length; i++) {
    normVT = [];
    for (int j = 0; j < norm.length; j++) {
        normVT.add(norm[j][i]);
    }
    normT.add(normVT);
}

```

- Et la dernière étape avant d'appliqué SVD :

Nous divisons la matrice par :

$$\text{Dénominateur} = \sqrt{((\text{le nombre de tout les POI}) - 1)}$$

	U ₁	U ₂	U ₃	U ₄	U ₅
111 Restaurant	0,2	0,2	0,3	0,1	-0,2
Planet Food	0,2	-0,3	-0,2	-0,4	-0,2
Pizza Up	0,2	-0,3	-0,2	0,1	0,3
Havana Coffee	0,2	0,2	-0,2	0,1	0,3
Cafeteria 3d	-0,3	0,2	0,3	0,1	-0,2

Tableau 9 –Matrice User / POI

L'implémentation de cette étape en langage Dart :

```

denominateur = sqrt((matrixColumn.length) - 1);
print(denominateur);
for (int i = 0; i < normT.length; i++) {
    for (int j = 0; j < normT[i].length; j++) {
        normT[i][j] = normT[i][j] / denominateur;
    }
}
print(normT);

```

- Nous appliquons l'algorithme de SVD à la matrice obtenue

```

// SVD
var matrix = Array2d.empty();

for (int i = 0; i < normT.length; i++) {
    var line;
    line = normT[i];
    matrix.add(Array(line));
}
var svd = SVD(matrix);

```

Les résultats donnés par l'algorithme de SVD sont deux matrices orthogonales et une matrice diagonale.

```

var u = svd.U();
var s = svd.S();
var v = svd.V();
print('Mtrix U Orthogonal');
print(u);
print('Mtrix S Orthogonal');
print(s);
print('Mtrix V Diagonal');
print(v);

```

	X ₁	X ₂	X ₄
Y ₁	0,317	0,381	-0,394
Y ₂	0,0021	-0,8405	0,305
Y ₃	-0,6015	0,0392	-0,546
Y ₄	-0,317	0,381	0,4
Y ₅	0,601	0,0392	-0,546

Tableau 10 - Matrice U Orthogonal

	X ₁	X ₂	X ₃	X ₄	X ₅
Y ₁	-0,3717	0,3171	0,6015	0,000131	-0,6015
Y ₂	0,523	0,523	0,2745	0,549	0,274
Y ₄	0,475	0,475	-0,3021	-0,604	-0,3021

Tableau 11 - Matrice V orthogonale

	X ₁	X ₂	X ₄
Y ₁	0,809	0	0
Y ₂	0	0,765	0
Y ₄	0	0	0,252

Tableau 12 - Matrice S diagonale

- Nous faisons la multiplication de ces 3 matrice : $R = U S V^t$

	111 Restaurant	Planet Food	Pizza Up	Havana Coffee	Cafeteria 3d
U₁	-0,11	0,15	0,11	0,05	0,00
U₂	0,00	-0,34	0,00	0,04	0,37
U₃	-0,29	0,16	0,04	0,04	0,13
U₄	-0,40	-0,16	-0,41	-0,12	-0,13
U₅	-0,29	0,08	0,43	0,42	0,00

Tableau 13 – Matrice de recommandation

Et on obtient la matrice de recommandation

La matrice de recommandation obtenue contient des valeurs sur l'intervalle :] -1,1[

Puisque nous n'avons pas assez des données dans cet exemple nous avons pris tous les POI tel que leurs valeurs sont supérieur strictement à 0 (l'intervalle]0,1]) comme des POI recommandé à l'utilisateur

Par exemple Planet Food, Pizza up et Havana coffee sont des bonnes recommandations pour U1 et U5. Par contre ils ne sont pas recommandés pour l'utilisateur U4 car ses valeurs sont inférieures à 0.

L'implémentation de cette étape en langage Dart :

```
double sommeLigne = 0;
var resultat = [];
var resultatM = [];

for (int i = 0; i < u.length; i++) {
  resultat = [];
  for (int j = 0; j < s[0].length; j++) {
    sommeLigne = 0;
    for (int k = 0; k < u[0].length; k++) {
      sommeLigne = sommeLigne + (u[i][k] * s[k][j]);
    }
    resultat.add(sommeLigne);
  }
  resultatM.add(resultat);
}
```

4.5 Présentation de l'application

4.5.1 Interface de l'accueil

La figure 14 présente la page d'accueil, c'est la première page qui va être affichée au visiteur du site. Cette interface contient des sections divisées comme suit :

La première section contient le logo du site, le bouton de l'inscription et la barre de recherche où le visiteur peut faire une recherche sur n'importe quel endroit dans Oran.

La deuxième section contient le menu des catégories : nous avons des catégories commerciales et des catégories touristiques.

La troisième section contient des sous-sections, la première sous-section contient les POI les plus populaires selon la notion de la popularité de Google Maps et l'autre sous-section contient des POI filtrés selon nos catégories.

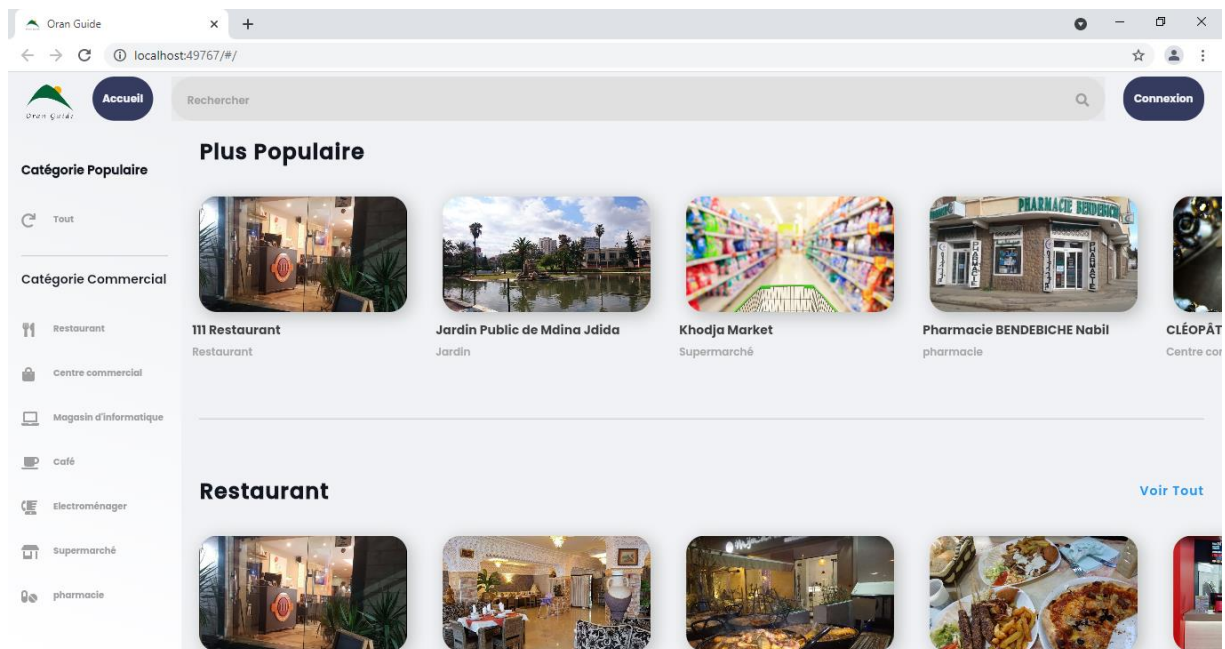


Figure 14 – Interface de l'accueil

4.5.2 Interface d'inscription

Sur cette interface l'utilisateur doit remplir le formulaire d'inscription pour créer un compte (nom d'utilisateur, Email et mot de passe).

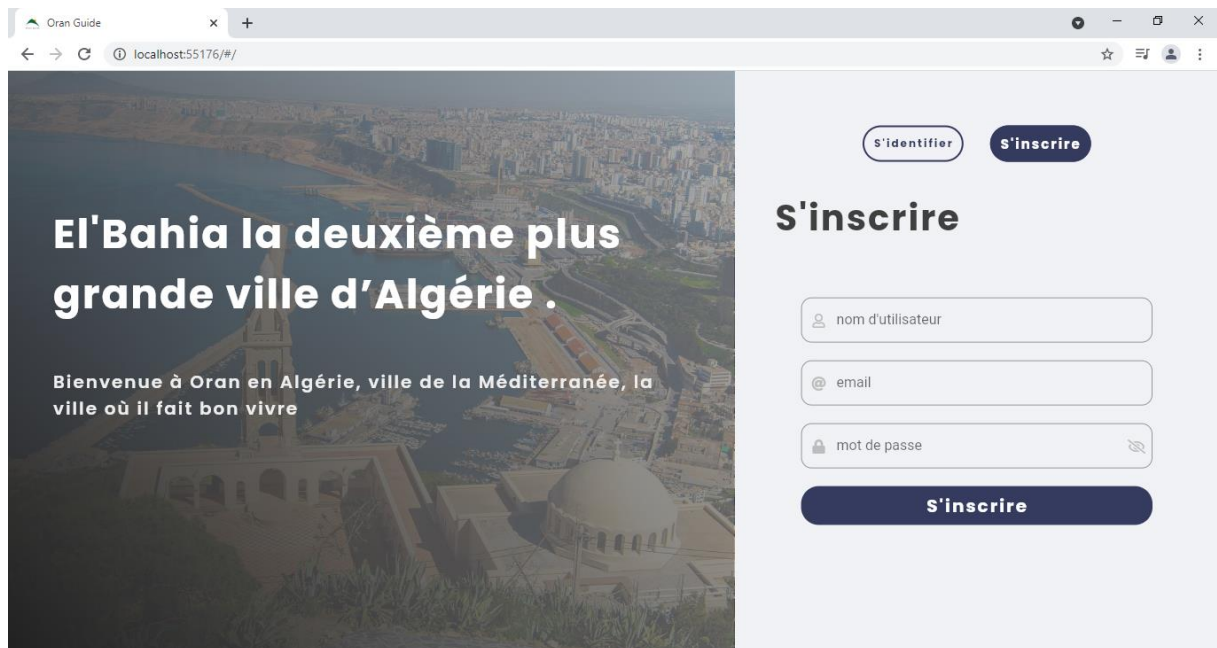


Figure 15 – Interface d'inscription

4.5.3 Interface d'identification

Cette interface permet au visiteur déjà inscrit dans le site d'accéder à leur espace personnel.

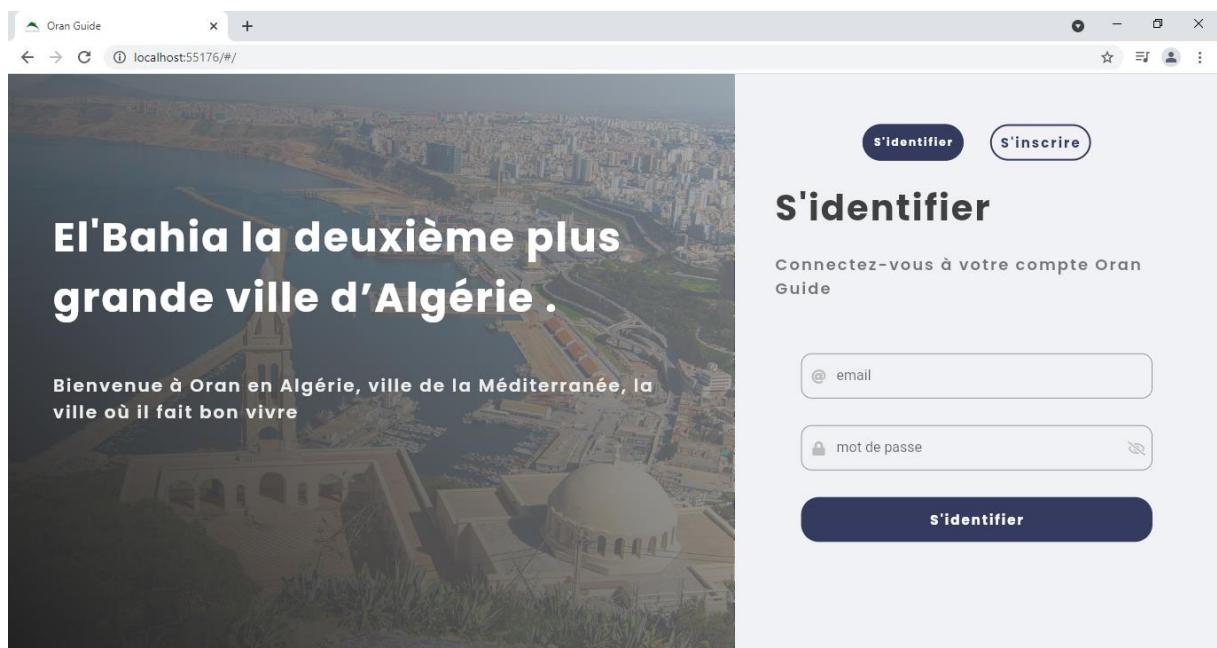


Figure 16 – Interface d'identification

4.5.4 Interface d'accueil (après inscription)

Après l'inscription, l'interface de l'accueil sera réaffichée mais avec plusieurs autres fonctionnalités. La section Just pour vous contient les POI recommandé à l'utilisateur.

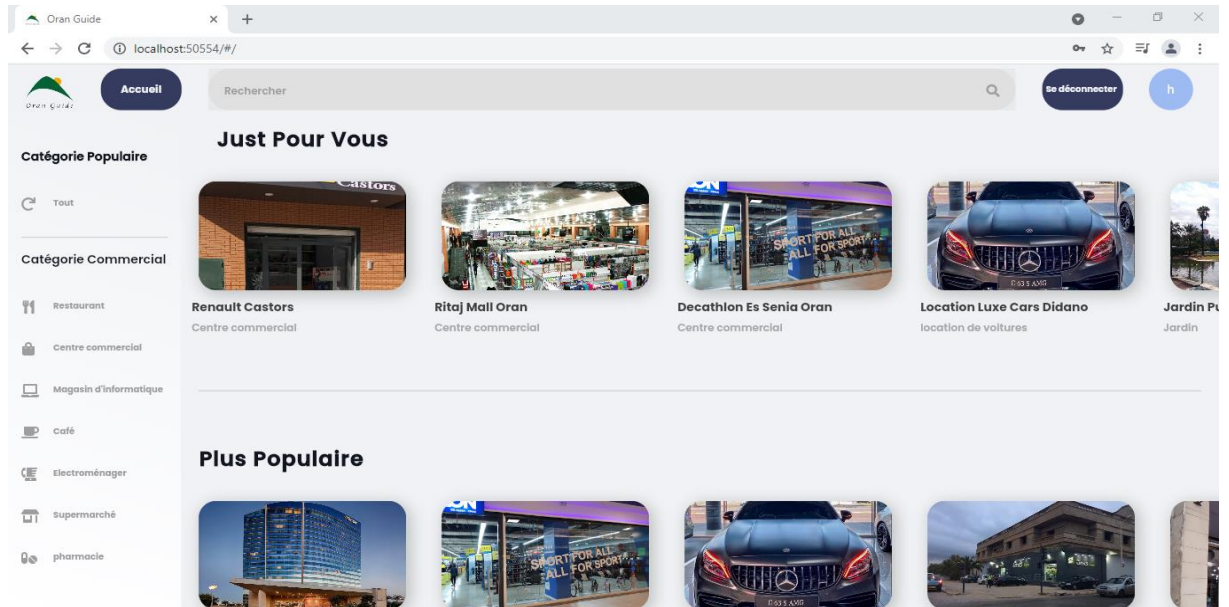


Figure 17 – Interface d'accueil (après l'inscription)

4.5.5 Interface de détail de POI

Cette interface (figure 18) représente la page de détail de POI dans laquelle nous avons fournis les informations nécessaires pour l'utilisateur.

L'utilisateur peut ajouter le POI à ses favoris en cliquant sur le cœur en haut de l'image.

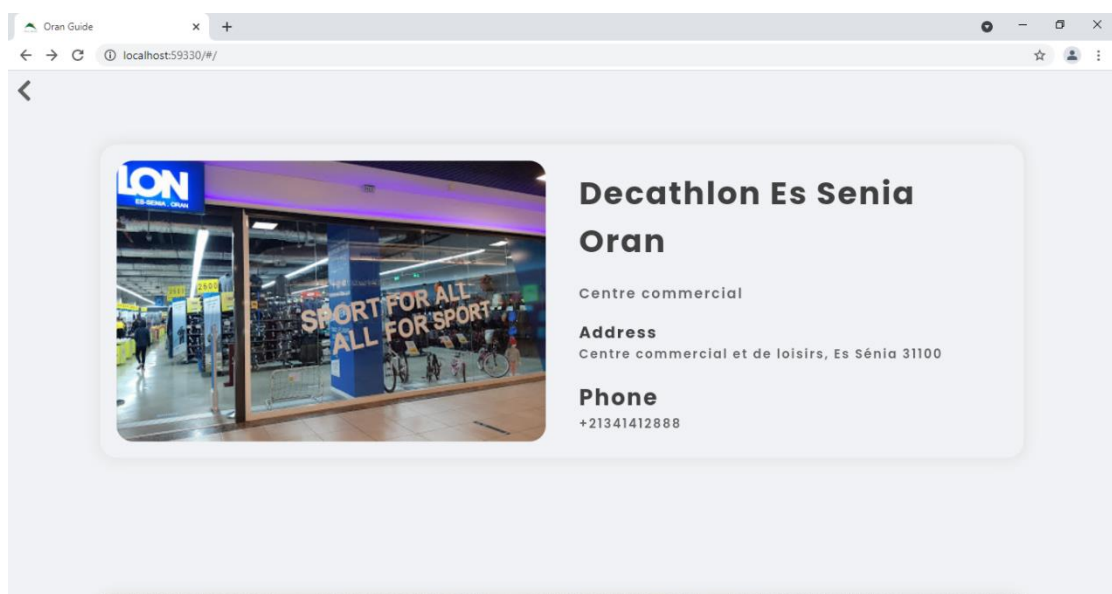


Figure 18 – Interface de détail de POI

La figure 19 représente la position du ce POI dans Google maps et au-dessous une section qui contient des POI filtré selon la catégorie de POI sélectionné.

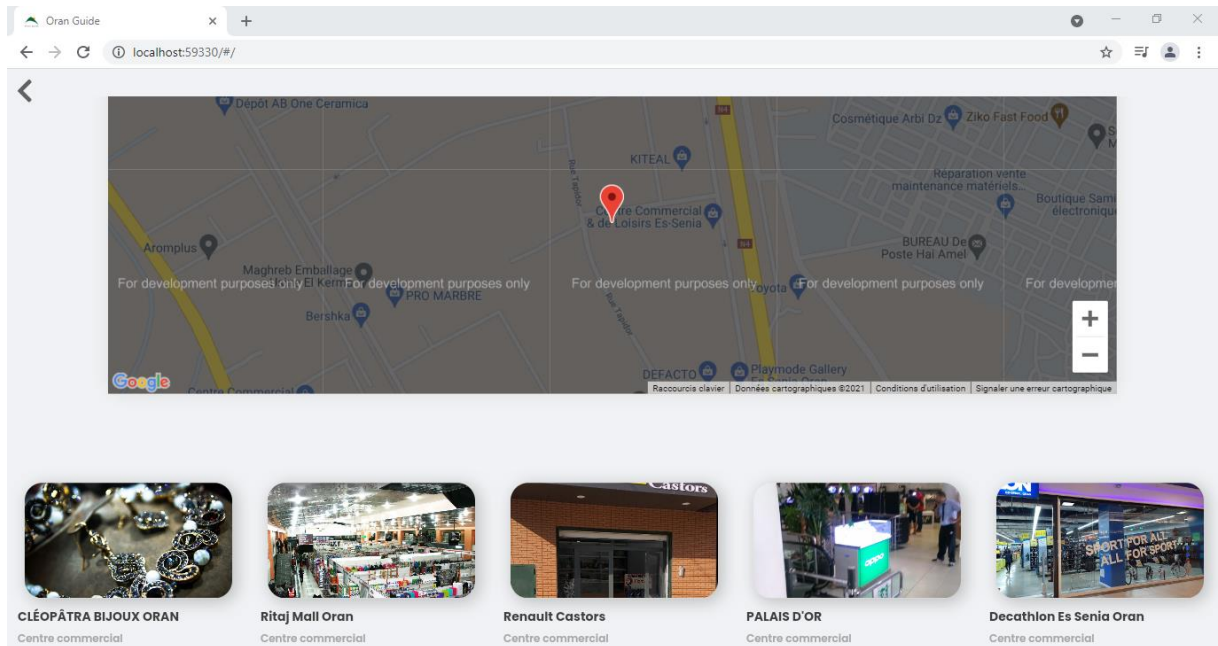


Figure 19 – Interface détaille de POI au-dessous

4.5.6 Interface de favoris et historique

La figure 20 représente l’interface de l’historique. Quand l’utilisateur appuie sur un POI, il sera ajouté à son historique.

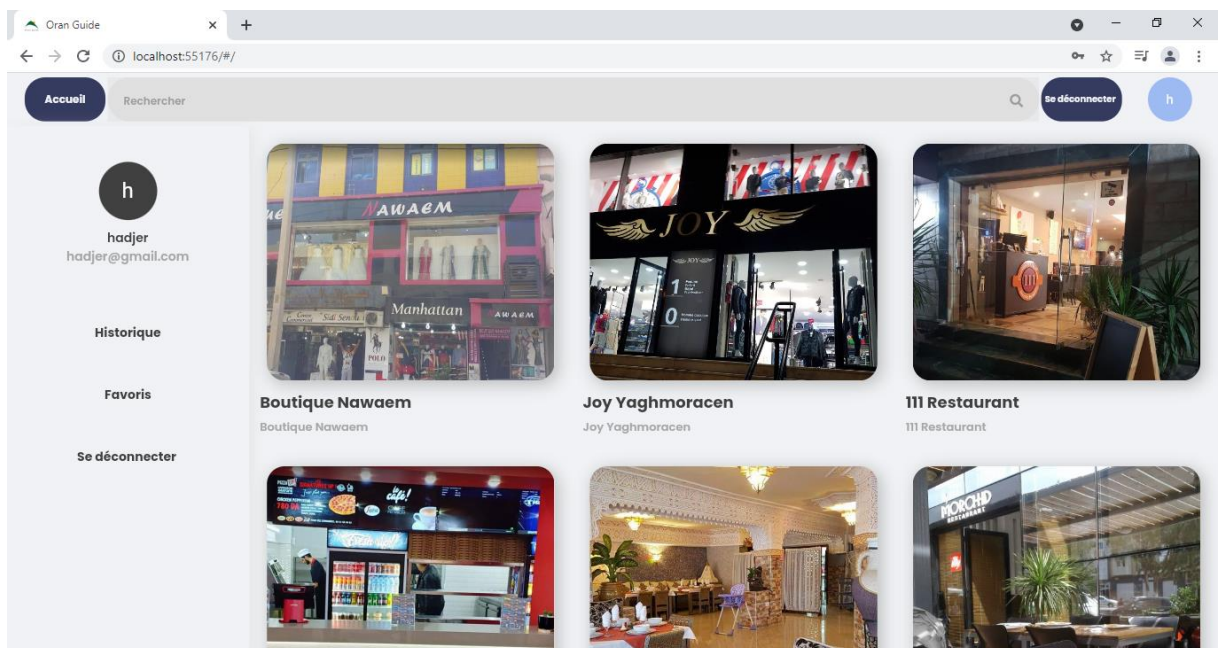


Figure 20 – Interface de l’historique

La figure 21 représente la liste des POI favorisé par l'utilisateur.

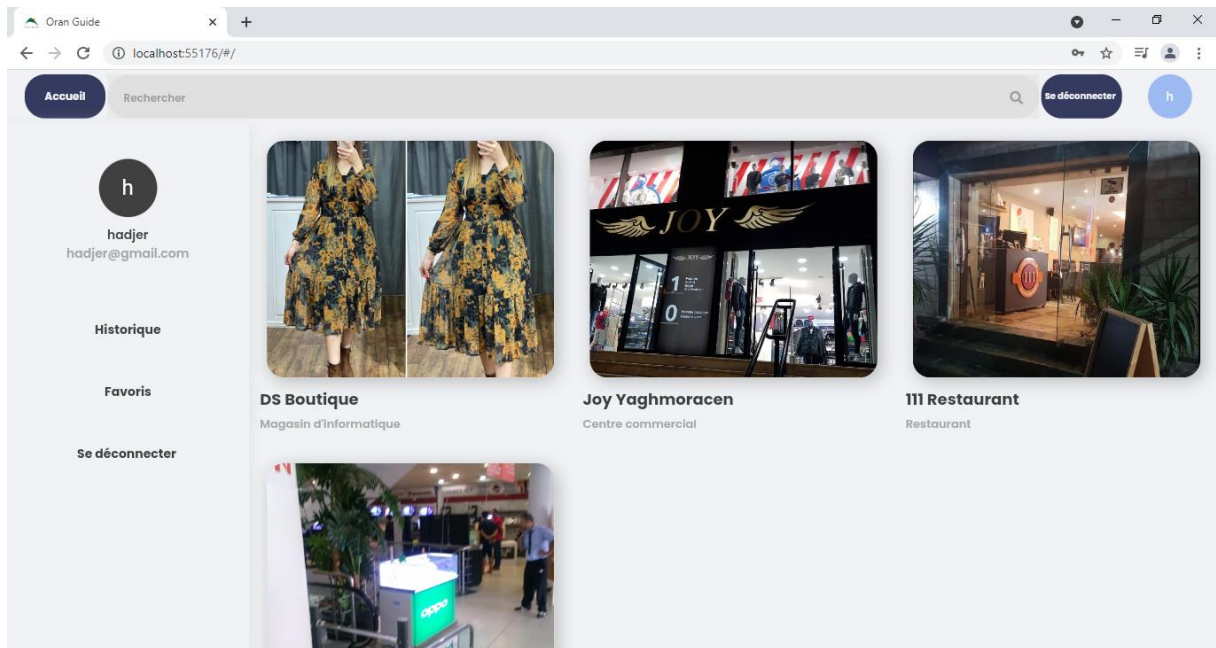


Figure 21 – Interface de POI favorisé

4.5.7 Interface de recherche

La recherche s'effectue par catégorie et par le nom de POI.

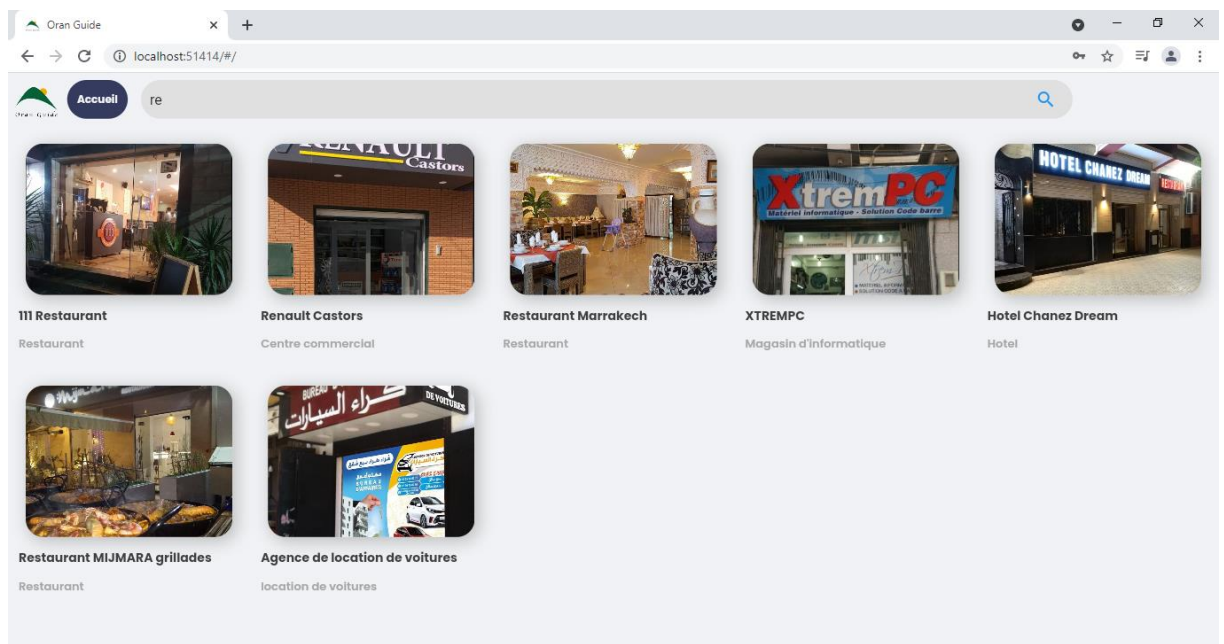


Figure 22 – Interface de recherche

4.5.8 Base de données (Firebase)

Comme nous avons expliqué précédemment que nous avons utilisé Firebase comme base de données pour notre application, cette base de données en ligne va nous assurer que si un utilisateur change de dispositif, son compte sera encore sauvegardé dans la base de données, voici une capture d'une page de Firebase.

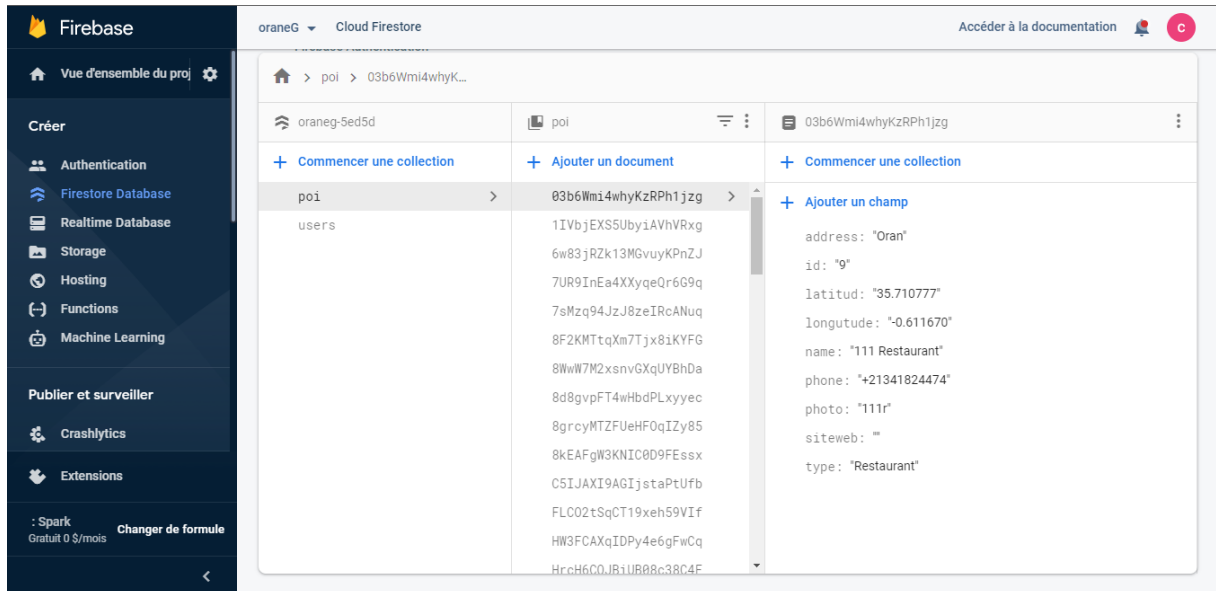


Figure 23 – Firebase

4.6 Conclusion

Dans ce chapitre, nous avons présenté les technologies et les différents langages et logiciels utilisés dans l'implémentation de notre application, nous avons présenté un exemple démonstratif sur SVD et les calculs qui s'effectue. Enfin nous avons présenté les différentes interfaces de notre site.

Conclusion Générale

Avec l'occurrence des systèmes de recommandations basées sur les POI, nous pouvons faire des recommandations personnalisées aux utilisateurs, et ils n'aident pas seulement les Utilisateurs à explorer de nouvelles endroits mais encore ils aident les propriétaires des POI d'augmenter leurs revenus en attirant des utilisateurs.

Dans ce travail nous avons présenté quatre chapitres, nous avons vu dans le premier chapitre la définition du système de recommandation et ses types le filtrage basé sur le contenu, le filtrage collaboratif, et la combinaison des deux types d'approches nous avons détaillée sur les types de FC et nous avons remarqué que les FC basé sur le modèle sont plus efficaces que celles qui sont basées sur la mémoire. Les FC basé sur modèle utilisent plusieurs algorithmes, et pour notre cas les algorithmes de factorisation matricielle qui nous intéressent, Pour cela, nous allons les détaillés dans le deuxième chapitre. En fin nous avons parlé sur les points d'intérêt et ses caractéristiques.

Le troisième chapitres est consacré pour la conception et la modélisation du notre système et finalement nous allons présentés les outils de l'implémentation et les interfaces du notre site.

Dans ce cadre, notre travail repose sur le développement d'un site web de l'attractivité commercial et touristique de la ville d'Oran qui repose sur le système de recommandation de POI.

Comme perspectives nous envisageons d'apporter quelques améliorations à savoir :

- Intégrer le contexte « localisation » de l'utilisateur (à la maison, au travail) pour but de lui fournir des suggestions pertinentes selon l'endroit où il se trouve.
- Applique la recommandation par l'algorithme de SVD++.
- Rendre notre site en ligne.

Annexe A

Création de logo du site



Figure 24 – Logo du site

Nous avons utilisé le logiciel Illustrator pour créer notre logo.

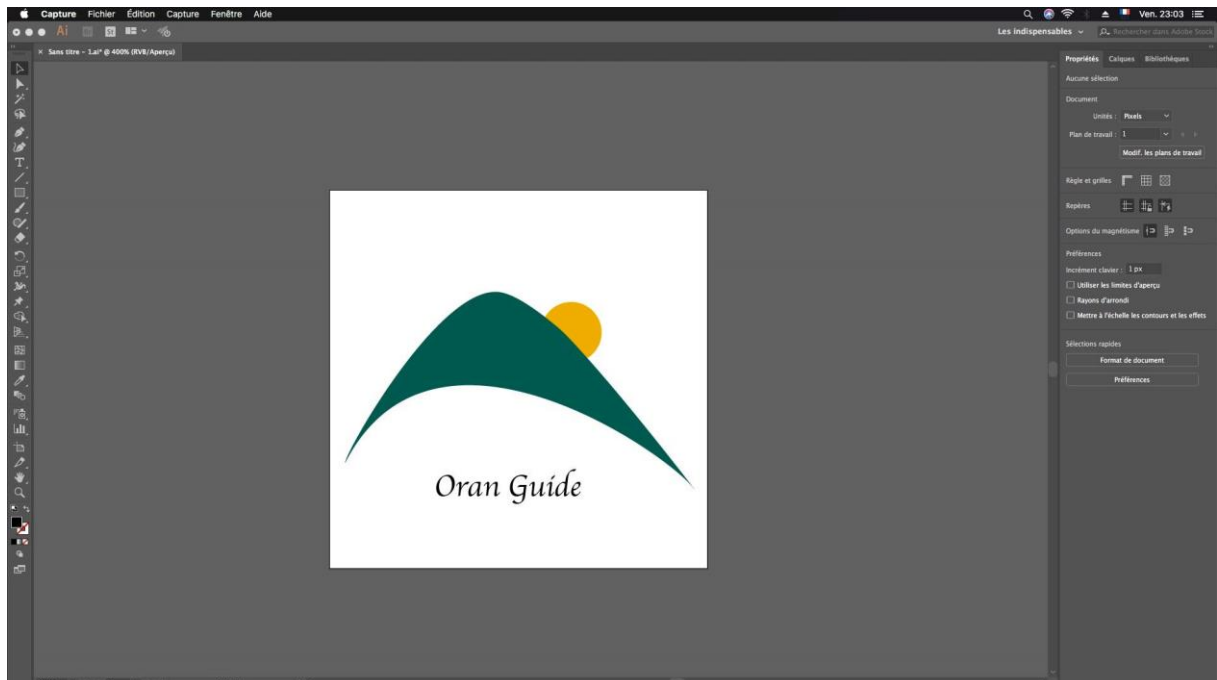


Figure 25 – Illustrator

Annexe B

Obtenir votre base de données Firebase et Accéder aux données de votre site.

La création d'un compte Firebase est très facile, il vous suffit d'accéder au site de [Firebase console \(google.com\)](https://firebase.google.com) et de vous connecter à l'aide des informations d'identification de votre compte Gmail. Puis cliquer sur créer projet et donner le « nom de projet » et « Pays /région »

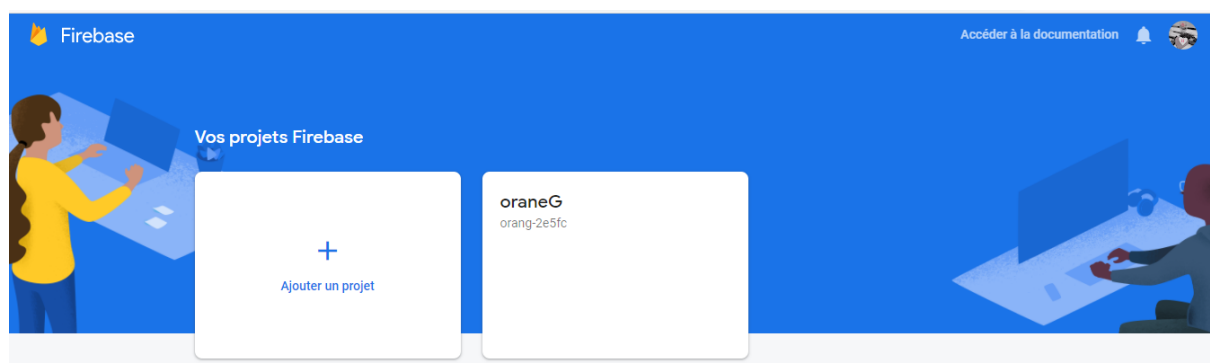


Figure 26 – Création d'un compte Firebase

Références

- [1] Ain Picot-Clemente. Une architecture générique de Systèmes de recommandation de combinaison d'items. Application au domaine du tourisme. Recherche d'information [cs.IR]. Université de Bourgogne, 2011. Français. tel-00688994v1.
- [2] S.Taouli, W.Benachenhou. Utilisation de factorisation matricielle sans les systèmes de recommandation sensible au contexte. Mémoiremaster, université Abou Bakr Belkaid – Tlemcen, 2017.
- [3] Filtrage collaboratif. All You Need to Know About Collaborative Filtering (digitalvidya.com) (consulté le 02/02/2021)
- [4] Charif ALCHIEKH HAYDAR. Les systèmes de recommandation à base de confiance. Université de Lorraine .2014. Thèse de doctorat
- [5] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. RecommenderSystems: An Introduction. Cambridge UniversityPress, New York, NY, USA, 1st edition, 2010.
- [6] Les réseaux bayésiens. PowerPoint Presentation (cnrs.fr) (consulté le 03/02/2021)
- [7] Les modèles de clustering. Microsoft PowerPoint - ClusteringAssas.ppt (lip6.fr) (consulté le 03/02/2021)
- [8] Factorisation matricielle. Aide de PTC Mathcad (consulté le 03/02/2021)
- [9] Les arbres de décision. Cours - Arbres de décision — Cours Cnam RCP209(consulté le 03/02/2021)
- [10] Laurent Andrey, Olivier Festor, Abdelkader Lahmadi. Evaluation du passage à l'échelle des systèmes de gestion : métriques et modèles. Colloque Francophone sur l'Ingénierie des Protocoles - CFIP 2006, Eric Fleury and Farouk Kamoun, Oct 2006, Tozeur/Tunisia. ffinria-00105550f
- [11] Ilham Esslimani. Vers une approche comportementale de recommandation : apport de l'analyse des usages dans un processus de personnalisation. Interface homme-machine [cs.HC]. Université Nancy II, 2010. Français.

- [12] Algorithm YouTube : Le comprendre pour obtenir plus de vues (hootsuite.com) (consulté le 20/05/2021)
- [13] Bokde D., Girase S., Mukhopadhyay D., Role of Matrix Factorization Model in Collaborative Filtering Algorithm: A Survey. IJAFRC, 2014.
- [14] Jean-Benoît Griesner. Scalable Models for Points-Of-Interest Recommender Systems. Artificial Intelligence [cs.AI]. Telecom ParisTech, 2018. English. fftel-02085091
- [15] Tobler's First Law and Spatial Analysis. <https://www.jstor.org/stable/3693985> (consulté le 11/03/2021)
- [16] Yonghong Yu, Xingguo Chen, A Survey of Point-of-Interest Recommendation in Location-Based Social Networks, Nanjing University of Posts and Telecommunications. 2015.
- [17] Roques, P. UML 2: Modéliser une application web. Editions Eyrolles. (2008).
- [18] Roques, P., & Vallée, F. UML en action: de l'analyse des besoins à la conception en Java. Eyrolles. (2000).
- [19] A. RAISSI Conception et développement d'un site web de e-commerce pour le compte de LSAT_Nokia. Mémoire master, université virtuelle de Tunis, 2013.
- [20] diagramme de classe doc_num.php (univ-saida.dz) (consulté le 02/04/2021)
- [21] Définition de draw.io disponible à l'adresse : diagrams.net - Chrome Web Store (google.com)(consulté le 20/05/2021)
- [22] Définition de draw.io disponible à l'adresse : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/flutter-cest-quoi/>(consulté le 26/05/2021)
- [23] Définition de Flutter disponible à l'adresse : <https://flutter.dev/docs> (consulté le 26/05/2021)
- [24] Définition de Visual Studio Code disponible à l'adresse : <https://code.visualstudio.com/docs> (consulté le 26/05/2021)
- [25] Définition de FireBase, disponible à l'adresse : <https://www.geeksforgeeks.org/firebaserealtimedatabase-with-operations-in-android-with> (consulté le 26/05/2021)

[26] L'architecture Firebase. Présentation de l'architecture FCM | Firebase (google.com) (consulté le 26/05/2021)

[27] 10 good reasons to learn Dart. If you already know C++, C#, or Java... | by NafisFuad | HackerNoon.com | Medium (consulté le 26/05/2021)

[28] Définition du langage JSON (consulté le 26/05/2021)

[29] Définition de Illustrator disponible à l'adresse : Adobe Illustrator CC 2021 - Télécharger pour PC Gratuitement (malavida.com) (consulté le 01/06/2021)

[30] Décomposition en valeurs singulières (SVD) - Fouille de données avancée (2016-2017) (beedotkiran.github.io) (consulté le 11/06/2021)