

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El Ibrahimi de Borj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique



MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en informatique

Spécialité : Réseaux et multimedia

THEME

Conception d'un système intelligent de recommandation
d'activités pédagogiques basé sur l'apprentissage adaptatif

Présenté par :

Ilyes OUAREM

Islam CHETTOUAH

Soutenu publiquement le : 12/06/2025

Devant le jury composé de :

Président : Mme. Ramla BELALTA

Examineur : Dr. Nadjib BENAOUA

Encadreur : Mme. Safa Attia

2024/2025

Dédicace

Tout d'abord, **à Dieu**, louange à Allah, par Sa grâce s'accomplissent les bonnes actions. Ô Allah, ce n'est pas par mes efforts et mes diligences, mais par Ta guidance et Ta bénédiction.

Alhamdulillah au début et à la fin.

Je dédie ce travail à **mes parents** pour leur éducation exemplaire, leur amour et leur soutien qui ont été essentiels à ma réussite. Je suis profondément reconnaissant envers eux.

Ma grande famille Qui n'a jamais cessé de me soutenir tout au long de mon parcours.

J'exprime ma profonde gratitude pour leur soutien moral.

Mes amis Qui ont été de bons compagnons, ainsi que tous ceux que je respecte profondément.

À **Madame Safa Attia** et **Monsieur Nadjib Benaouda**, Je vous exprime toute ma reconnaissance pour votre encadrement, vos conseils précieux et votre disponibilité tout au long de ce projet. Votre accompagnement a été d'une grande valeur.

Enfin, Je tiens à remercier mon partenaire et ami **Islam**, car ce travail n'aurait pas pu être mené à bien sans son soutien inconditionnel.

À tous ceux qui ont contribué de près ou de loin à ce travail. je vous remercie sincèrement.

Ilyes OUAREM

Dédicace

À mon très cher père, Tu es mon roc, toujours attentif, compréhensif et digne de mon plus grand respect. Aucune dédicace ne saurait suffire à exprimer mes sentiments envers toi. Que Dieu te protège, te préserve et t'accorde une bonne santé ainsi qu'une longue vie. **À la lumière de mes jours, ma mère**, Je tiens à exprimer toute ma gratitude et mon amour envers cette femme qui m'a donné la vie, la tendresse et le courage. Aucun mot ni geste ne pourra véritablement traduire la profondeur de mes sentiments à son égard. Je lui dédie ce modeste travail en reconnaissance de ses sacrifices et de l'amour dont elle m'a toujours entouré. **À mon frère Sami**, À toi, force précieuse et compagnon d'enfance, je dédie ces lignes en témoignage de ma profonde estime. Merci pour ton aide, ton réconfort et tes encouragements dans les moments clés. **À ma sœur**, Que Dieu ait pitié de toi et t'accorde une place dans Son paradis. J'aurais tant souhaité que tu sois à mes côtés aujourd'hui pour partager cette réussite, car c'est en grande partie grâce à toi que je suis arrivé jusqu'ici. Je te dédie cet humble travail, ainsi que tout mon amour et ma gratitude. Je prie Dieu de nous réunir un jour au paradis. À Madame Attia Safa et Monsieur Nadjib Benaouda, Je vous exprime toute ma reconnaissance pour votre encadrement, vos conseils précieux et votre disponibilité tout au long de ce projet. Votre accompagnement a été d'une grande valeur. **À tous mes amis**, Merci pour votre présence, votre soutien moral et vos encouragements tout au long de ce parcours, Je tiens à remercier ma partenaire et ami **Ilyes**, car ce travail n'aurait pas pu être mené à bien sans son soutien inconditionnel.

Islam CHETTOUAH

Remerciement

Tout d'abord, nous remercions **Allah**, le Tout-Puissant, de nous avoir accordé la force, la patience et la volonté nécessaires pour mener à bien ce travail. C'est grâce à Sa bénédiction que nous avons fait face aux difficultés et persévéré jusqu'à la fin de ce projet. Nous exprimons ensuite notre profonde gratitude à **Mme Safa Attia**, notre encadrante, pour son accompagnement précieux tout au long de ce travail. Nous la remercions sincèrement pour sa disponibilité, ses conseils pertinents, sa bienveillance et son soutien constant, qui ont grandement contribué à la réussite de ce projet. Nous tenons aussi à remercier chaleureusement les membres du jury, **Mme Ramla BELALTA** et **notre professeur Nadjib BENAOUA**, qui ont accepté de juger et de discuter ce projet de fin d'études, en nous honorant par leur présence et leur précieuse contribution.

Merci à tous.

*Ilyes OUAREM
& Islam CHETTOUAH*

Résumé

Ce travail présente la création et le développement d'un système de recommandation intelligent (SRI) d'activités d'apprentissage pour l'étude de la langue anglaise chez les enfants âgés de 3 à 12 ans. Réalisé sous forme d'une application mobile Android, Développé sous forme d'une application mobile Android, le système s'appuie sur l'apprentissage adaptatif pour proposer des contenus personnalisés, en fonction d'un questionnaire de profil, de l'historique d'utilisation, des résultats aux tests et du comportement d'apprenants similaires. La modélisation du système a été réalisée avec UML pour structurer ses composants fonctionnels. Son résultat est une application fonctionnelle permettant de suggérer dynamiquement des activités pédagogiques adaptées qui accompagneront l'engagement de la progression des enfants.

Mots-clés : Apprentissage adaptatif – SRI – Application mobile – Langue anglaise – Enfants – Android – UML.

Abstract

This work presents the creation and development of an intelligent recommendation system (IRS) for learning activities to help children aged 3 to 12 learn the English language. Developed as a mobile Android application, this system is based on adaptive learning to offer personalized content, depending on a profile questionnaire, usage history, test results, and the behavior of similar learners. The system's design was created using UML to organize its functional elements. The result is a working application that can dynamically suggest suitable educational activities to help support children's learning progress.

Keywords : Adaptive learning – SRI – Mobile application – English language – Children – Android – UML.

ملخص

يعرض هذا العمل إنشاء وتطوير نظام ذكي للتوصية (SRI) بأنشطة تعليمية لتعلم اللغة الإنجليزية لدى الأطفال الذين تتراوح أعمارهم بين ٣ و١٢ سنة، وقد تم تطوير هذا النظام على شكل تطبيق أندرويد للهواتف المحمولة، ويعتمد على التعلم التكيفي لتقديم محتوى مخصص بناءً على استبيان الملف الشخصي، وسجل الاستخدام، ونتائج الاختبارات، وسلوك متعلمين مشابهيين. تم تصميم نموذج النظام باستخدام UML لتنظيم العناصر الوظيفية، والنتيجة هي تطبيق فعال قادر على اقتراح أنشطة تعليمية مناسبة بشكل ديناميكي، مما يدعم تقدم الأطفال في عملية التعلم.

الكلمات المفتاحية: التعلم التكيفي - SRI - تطبيق موبايل - اللغة الإنجليزية - الأطفال

- أندرويد - UML.

Table des matières

Liste des abréviations	xii
Liste des figures	xiv
Liste des tableaux	xv
1 Introduction Générale	1
1.1 Contexte	1
1.2 Probématiques	1
1.3 Solution	3
1.3.1 L'apprentissage adaptatif : une pédagogie personnalisée	3
1.3.2 Les systèmes de recommandation intelligents : personnalisation dynamique	4
1.3.3 Association des deux approches	4
1.3.4 Description du système proposé	4
1.4 Structure du rapport	6
2 Systèmes de recommandation et l'apprentissage adaptatif	7
2.1 Introduction	7
Première partie : Notions fondamentales	8
2.2 Systèmes de recommandation	8
2.2.1 Définition	8
2.2.2 Processus	8
2.3 Les systèmes de recommandation intelligents	10
2.4 Domaines d'expertise des systèmes de recommandation.	10

2.5	Collection des données	11
2.5.1	Types de feedback	11
2.5.1.1	Feedback explicite	11
2.5.1.2	Feedback implicite	12
2.6	Approches classiques	12
2.6.1	Approches de filtrage basées sur le contenu (FBC)	13
2.6.2	Approches basées sur le filtrage collaboratif	14
2.6.2.1	FC basé sur la mémoire	15
2.6.2.2	FC basé sur le modèle	16
2.6.2.3	Systèmes de recommandation hybrides	17
2.7	L'Évolution et les Défis des Systèmes de Recommandation Intelligents.	18
2.8	Les systèmes de recommandation adaptatifs en éducation	19
2.8.1	Définition	19
2.8.2	Fondements Théoriques	20
2.8.3	Modèles d'apprentissage adaptatif	21
2.8.4	Méthodes et algorithmes d'implémentation de l'apprentissage adaptatif	22
2.8.5	Critères d'adaptation contextuelle	23
2.8.5.1	Individualité dans l'adaptation	23
2.8.5.2	Relations	23
2.8.5.3	Temps	24
2.8.5.4	Activité	24
2.8.5.5	Localisation	24
2.8.6	Les objectifs des systèmes d'apprentissage Adaptatif	24
Deuxième partie : Mécanismes utilisés dans notre système		25
2.9	Mécanismes de recommandation utilisés	25
2.9.1	Recommandation basée sur le questionnaire de profil :	26
2.9.2	Recommandation basée sur l'historique de navigation (Nombre de clics, Temps cumulé, Feedback, etc.) :	26
2.9.3	Recommandation basée sur des tests de différents niveaux de difficulté :	27
2.9.4	Recommandation basée sur des apprenants similaires :	28
2.10	Résumé des types de filtrage utilisés	29
2.11	Modèles d'apprentissage adaptatif utilisés	30

2.11.1	Modèle basé sur la maîtrise des compétences	30
2.11.2	Modèle d'évaluation diagnostique	30
2.11.3	Modèle basé sur le profil de l'apprenant	30
2.11.4	Modèle prédictif et adaptatif	31
2.12	Conclusion	31
3	Modélisation et Architecture	32
3.1	introduction	32
3.2	Méthodologie de conception	33
3.2.1	Présentation d'UML	33
3.2.2	Vues UML	33
3.2.2.1	Vue fonctionnelle : diagramme de cas d'utilisation	34
3.2.2.2	Vue statique : diagramme de classes	48
3.2.3	Vue dynamique : diagrammes de séquence	50
3.3	Architecture du système proposé	55
3.3.1	Partie frontend	55
3.3.1.1	Interface Utilisateur	55
3.3.1.2	Appel au Serveur Backend	56
3.3.2	Partie backend	56
3.3.2.1	Stockage des Données Utilisateurs	56
3.3.2.2	Logique Métier de Recommandation des Ressources péda- gogiques	57
3.3.2.3	Exposition des fonctionnalités en tant que Services Web	57
3.4	Conclusion	59
4	Implémentation et Réalisation	60
4.1	Introduction	60
4.2	Les services web de type REST	61
4.3	Environnement de développement	62
4.3.1	Côté backend	62
4.3.1.1	Jakarta EE	62
4.3.1.2	TomEE	62
4.3.1.3	Eclipse Enterprise Java and Web Developers	62

4.3.1.4	L'API Jakarta RESTful Web Services	62
4.3.1.5	SGBDR MySQL	63
4.3.2	Côté front-end	63
4.3.2.1	Android	63
4.3.3	Android Studio	63
4.3.4	La bibliothèque HTTP	64
4.4	Schéma relationnel de notre base de données	64
4.5	Description de notre API proposée	67
4.6	Présentation de notre application mobile	69
4.6.1	Diagramme de flux utilisateur	69
4.6.2	Présentation de quelques interfaces graphiques	71
4.6.2.1	Processus d'inscription et création du profil personnalisé	71
4.6.2.2	Navigation après l'inscription et accès aux cours	73
4.6.2.3	Suivi de l'évaluation et adaptation des ressources	75
4.7	Conclusion	77
5	Conclusion générale	78
5.1	Contributions	78
5.2	Travaux futurs et perspectives	79
	Références	80
A	Exemples de codes pour notre API	86
1.1	Exemple : Enregistrement complet d'un enfant	86
1.2	Exemple : Recommandation personnalisée basée sur l'historique de navigation	90

Liste des abréviations

API Application Programming Interface.

APIs Application Programming Interfaces.

CU Cas d'utilisation.

FBC Filtrage basé sur le contenu.

FC Filtrage Collaboratif.

IDE Integrated Development Environment.

JAX-RS Java API for RESTful Web Services.

JSF JavaServer Faces.

JSON JavaScript Object Notation.

JSP JavaServer Pages.

MySQL Structured Query Language, langage de requête structuré.

PHP Hypertext Preprocessor.

REST Representational State Transfer.

SDK Software Development Kit.

SGBDR Système de Gestion de Base de Données Relationnelle.

SR Systèmes de Recommandation.

SRI Système de Recommandation Intelligent.

SW Services web.

UML Unified Modeling Language.

URI Uniform Resource Locator.

XML eXtensible Markup Language.

Table des figures

2.1	Aperçu simplifié d'un processus de recommandation [1]	9
2.2	Stratégies populaires utilisées par les systèmes de recommandation [1]	13
2.3	Processus logique d'un SR basé sur le filtrage par contenu	14
2.4	Processus logique d'un SR sur le filtrage collaboratif	15
2.5	Processus logique d'un système de recommandation hybride.	17
3.1	Diagramme de cas d'utilisation du parent	35
3.2	Diagramme de cas d'utilisation de l'apprenant	41
3.3	Notre diagramme de classes	49
3.4	Diagramme de séquence « Création du compte et recommandations des cours basées sur le questionnaire du profil »	51
3.5	Diagramme de séquence « Authentification »	52
3.6	Diagramme de séquence « Recommandation d'activité pédagogique »	54
3.7	Architecteur de notre système	58
4.1	Schéma relationnel de notre base de données	65
4.2	Diagramme de flux utilisateur de notre application	70
4.3	Interfaces du processus d'inscription et de personnalisation	72
4.4	Interfaces de navigation après l'inscription	74
4.5	Interfaces liées aux tests et aux ressources recommandées	76

Liste des tableaux

3.1	Cas d'utilisation « Créer un compte »	36
3.2	Cas d'utilisation « Mettre à jour le profil de l'apprenant »	37
3.3	Cas d'utilisation « Changer le niveau de difficulté »	38
3.4	Cas d'utilisation « Consulter son historique de navigation »	39
3.5	Cas d'utilisation « Consulter son état d'avancement »	40
3.6	Cas d'utilisation « Consulter les cours recommandés basés sur le questionnaire du profil »	42
3.7	Cas d'utilisation « Consulter les ressources »	43
3.8	Cas d'utilisation « Voir les recommandations basées sur l'historique de navigation »	44
3.9	Cas d'utilisation « Effectuer le test recommandé en se basant sur la période de navigation »	45
3.10	Cas d'utilisation « Voir le résultat du test »	46
3.11	Cas d'utilisation « Voir cours suivant après réussite dans le test »	47
3.12	Cas d'utilisation « Voir les recommandations après échec dans le test »	48
4.1	Liste des opérations proposées	68

Chapitre 1

Introduction Générale

1.1 Contexte

Avec l'essor du numérique et dans un environnement éducatif de plus en plus numérique, les apprenants interagissent quotidiennement avec une multitude de ressources pédagogiques en ligne. Les plateformes d'apprentissage en ligne se sont multipliées, proposant une grande variété de ressources pédagogiques accessibles à distance. Les systèmes classiques d'apprentissage en ligne offrent généralement un ensemble fixe de contenus (cours, exercices, vidéos, quiz interactifs, forums) présentés de manière uniforme à tous les apprenants, sans distinction de niveau, de profil ou de rythme. Cette diversité, bien qu'enrichissante, peut également engendrer une surcharge d'informations, rendant difficile pour l'apprenant le choix des activités les plus adaptées à son profil, à ses préférences et à son niveau de progression. Bien qu'ils aient permis une large démocratisation de l'éducation, ces systèmes restent souvent rigides, peu interactifs, et insuffisamment adaptés à la diversité des styles d'apprentissage individuels, notamment chez les enfants [2].

1.2 Probématiques

Dans le domaine de l'éducation, particulièrement dans l'enseignement des langues étrangères aux jeunes enfants, un des principaux défis est la personnalisation de l'apprentissage pour répondre aux besoins uniques de chaque apprenant. L'enseignement de l'anglais aux enfants de 3 à 12 ans représente une situation particulièrement complexe, en raison des différences

marquées dans les stades de développement linguistique au sein de cette tranche d'âge. Ces différences influencent directement la manière dont les enfants interagissent avec les contenus pédagogiques, ce qui nécessite l'adoption de méthodes d'enseignement flexibles, adaptées et évolutives [3]. Parmi ces différences, on peut citer :

- À cet âge, les enfants sont à des stades différents de leur acquisition de la langue anglaise. Certains peuvent être débutants, d'autres plus avancés, et certains peuvent même avoir une expérience limitée avec la langue. - Les enfants plus jeunes (3-5 ans) sont souvent dans une phase d'apprentissage sensoriel, tandis que les enfants plus âgés (6-12 ans) sont capables d'approfondir leur compréhension grammaticale et lexicale. Cette diversité des niveaux linguistiques dans une même classe ou groupe d'enfants exige des méthodes pédagogiques flexibles et modulables.

- Chaque enfant possède un style d'apprentissage unique, certains apprennent mieux par le biais de l'audition, d'autres par la vue, d'autres encore par la pratique. Ces différences doivent être prises en compte lors de la création d'activités pédagogiques.

- Au-delà des différences dans les styles et les niveaux d'apprentissage, il y a aussi une grande variation dans les besoins d'éducation d'enfants. Certains peuvent avoir des problèmes spécifiques, comme un souci de compréhension orale ou des manques dans certaines règles de grammaire, tandis que d'autres pourraient être forts dans certaines compétences. Donc il est important d'identifier puis surveiller en vrai temps les points forts et faibles de chaque enfant pour donner un travail qui rend fort ses compétences tout en remplissant ses manques. [4].

- La motivation est un facteur clé de l'apprentissage, particulièrement chez les jeunes enfants. Si les activités proposées ne sont pas adaptées à leur niveau ou ne correspondent pas à leurs intérêts, ils risquent de se lasser rapidement. Les enfants entre 3 et 12 ans sont sensibles à des activités ludiques et interactives, mais aussi à des tâches qui leur permettent de voir des progrès tangibles.

- Avec l'essor des technologies éducatives (plateformes en ligne, applications mobiles, etc.), la problématique devient encore plus complexe. Les outils numériques, tout en offrant une flexibilité et une accessibilité accrues, exigent des approches pédagogiques encore plus adaptées. Les systèmes d'apprentissage en ligne doivent non seulement être interactifs et engageants, mais aussi intégrer des mécanismes d'adaptation intelligente qui suivent en temps réel les pro-

grès de l'enfant et ajustent les activités en fonction de ses résultats [5].

1.3 Solution

Face à ces limites, les recherches en intelligence artificielle (IA) et en sciences de l'éducation ont conduit au développement de systèmes d'apprentissage adaptatif intégrant des mécanismes de recommandation intelligents. Ces systèmes analysent les données issues des interactions de l'apprenant (temps passé, résultats, préférences, etc.) pour proposer des contenus personnalisés en temps réel [6].

L'approche adaptative, lorsqu'elle est intégrée dans un système de recommandation pédagogique, permet d'aller au-delà de la simple distribution de contenu : elle devient un outil intelligent d'accompagnement individualisé. Cela est particulièrement pertinent dans le contexte de l'apprentissage précoce des langues chez les enfants, et qui présente l'étude de cas de ce projet, où la motivation, l'interactivité et la progression personnalisée jouent un rôle clé dans la réussite du processus éducatif.

Donc, pour répondre aux limites des approches classiques et à la diversité des profils d'apprenants chez les enfants de 3 à 12 ans, une solution efficace consiste à combiner deux approches puissantes : l'apprentissage adaptatif et les systèmes de recommandation intelligents (SRI).

1.3.1 L'apprentissage adaptatif : une pédagogie personnalisée

L'apprentissage adaptatif repose sur l'idée que chaque apprenant doit suivre un parcours d'apprentissage unique, adapté à ses besoins, ses compétences et ses préférences [7]. Contrairement aux systèmes rigides, il ajuste dynamiquement les activités proposées en fonction :

- Du niveau de compétence détecté (débutant, intermédiaire, avancé);
- De la progression de l'apprenant (réussites, erreurs);
- Forces et faiblesses de l'enfant (compréhension orale, lecture, vocabulaire, grammaire);
- Du moment d'apprentissage (périodes de la journée, durée d'attention);
- Âge de l'enfant;

Cette approche favorise un meilleur engagement de l'enfant, renforce sa motivation et amé-

liore l'efficacité de l'apprentissage.

1.3.2 Les systèmes de recommandation intelligents : personnalisation dynamique

Les SRI sont des outils technologiques qui exploitent les données collectées sur l'utilisateur pour lui proposer automatiquement les contenus les plus pertinents [8]. Appliqués au domaine éducatif, ils permettent de recommander :

- Des activités adaptées au profil initial de l'enfant (via un questionnaire rempli par les parents) ;
- Des activités basés sur l'historique de navigation : nombre de clics, temps passé ;
- Des tests progressifs selon les moments de la journée (matin, après-midi, soir) pour s'adapter au niveau d'attention ;

1.3.3 Association des deux approches

L'intégration de l'apprentissage adaptatif dans un SRI donne naissance à une solution pédagogique hautement personnalisée, capable de :

- Détecter les besoins spécifiques de chaque enfant ;
- Adapter les activités proposés en fonction du comportement de l'enfant ;
- Augmenter la motivation et l'efficacité de l'apprentissage en donnant des activités intéressantes, utiles et ajustées à l'âge et au niveau de l'enfant ;

Cette solution offre un apprentissage sur mesure pour les enfants de 3 à 12 ans apprenant l'anglais, en mélangeant technologie futée, informations sur le comportement et idées d'éducation qui s'adaptent. Elle permet non seulement d'améliorer l'efficacité de l'enseignement, mais aussi de maintenir un haut niveau d'engagement chez les jeunes apprenants.

1.3.4 Description du système proposé

Le système proposé est une application mobile éducative intelligente visant à recommander automatiquement des activités pédagogiques adaptées pour faciliter l'apprentissage de la langue anglaise chez les enfants âgés de 3 à 12 ans. Il repose sur une approche d'apprentissage

adaptatif, qui consiste à personnaliser dynamiquement les contenus en fonction des besoins, des préférences et des performances de chaque apprenant.

Le cœur du système est un moteur de recommandation intelligent, qui s'appuie sur plusieurs sources de données combinées pour produire des recommandations pertinentes et évolutives. Le processus de personnalisation se base sur les éléments suivants :

- **Questionnaire de profil initial** : rempli par les parents, ce questionnaire permet de recueillir des informations sur l'enfant : âge, niveau actuel en anglais, forces et faiblesses, disponibilités horaires, etc. Ces données servent de base pour générer une première recommandation adaptée.

- **Historique de navigation et d'interaction** : le système suit les activités de l'enfant dans l'application : nombre de clics sur chaque activité, durée passée sur chaque ressource, réponses aux quiz, niveaux atteints, fréquence d'utilisation, etc. Ces informations permettent de détecter les contenus les plus engageants.

- **Résultats aux tests de niveaux variés** : des tests rapides, de difficultés différentes, sont proposés périodiquement (au début ou à la fin des sessions) afin d'évaluer les acquis de l'enfant. L'heure de navigation (matin, après-midi, soir) est également prise en compte pour identifier les moments les plus propices à certaines activités. Par exemple, si l'enfant est plus attentif le matin, des tests plus complexes peuvent lui être proposés à ce moment-là. Inversement, à d'autres moments des tests plus légers.

- **Comportement d'apprenants similaires** : le système compare les comportements et les résultats de l'enfant avec ceux d'autres utilisateurs aux profils similaires. Cela permet de recommander des activités qui ont été efficaces pour d'autres enfants ayant un profil proche.

En combinant différentes techniques de recommandation, voir chapitre 2, le système peut proposer, à chaque session, une sélection personnalisée d'activités pédagogiques : tests éducatifs, vidéos, audios, textes, adaptés non seulement au niveau de l'enfant, mais aussi à son style d'apprentissage et à ses habitudes d'utilisation. Ce système vise à optimiser la progression linguistique des enfants tout en maintenant leur motivation grâce à des recommandations intelligentes, pertinentes et évolutives.

1.4 Structure du rapport

Outre cette introduction générale, notre mémoire comprend :

- Un premier chapitre qui présente le contexte, les problématiques, la solution proposée ainsi qu'une vue globale sur notre SRI.
- Un deuxième chapitre qui traite des notions fondamentales des systèmes de recommandation et de l'apprentissage adaptatif, puis présente les mécanismes utilisés dans notre SR.
- Un troisième chapitre qui détaille la conception et la modélisation de notre système, en incluant la modélisation UML et l'architecture client-serveur proposée.
- Un quatrième chapitre qui décrit la réalisation pratique du système, depuis l'implémentation des services web jusqu'à l'application mobile.
- Et dernièrement, une conclusion générale qui récapitulera notre travail réalisé.

Chapitre 2

Systemes de recommandation et l'apprentissage adaptatif

2.1 Introduction

Dans ce chapitre, nous présenterons les notions fondamentales liées aux systèmes de recommandation et à l'apprentissage adaptatif, qui constituent le socle théorique de notre travail. Nous commencerons par définir ce qu'est un SR, puis nous expliquerons les différentes étapes du processus de recommandation, notamment la collecte des données utilisateurs, l'analyse des préférences et les méthodes de génération des suggestions.

Une catégorisation des principales approches de recommandation sera ensuite présentée, en mettant l'accent sur les techniques de filtrage collaboratif, de filtrage basé sur le contenu ainsi que sur les méthodes hybrides. Nous introduirons également la notion de système de recommandation intelligent (SRI), en insistant sur son rôle dans les environnements éducatifs.

Par la suite, nous aborderons l'apprentissage adaptatif, en exposant ses fondements pédagogiques, ses modèles théoriques, ainsi que les critères contextuels pris en compte pour adapter les contenus aux besoins des apprenants.

Enfin, dans une deuxième partie, nous décrirons les mécanismes de personnalisation que nous avons intégrés dans notre propre SR pédagogique. Nous mettrons en évidence comment les données issues du profil de l'enfant, de son historique de navigation, des tests passés, des

comportements similaires, ainsi que les modèles d'apprentissage adaptatif utilisés sont exploités pour offrir une expérience d'apprentissage personnalisée et évolutive.

Première partie : Notions fondamentales

2.2 Systèmes de recommandation

2.2.1 Définition

Un système de recommandation (SR) représente une forme spécifique de filtrage de l'information, réduisant le temps de recherche des utilisateurs en leur fournissant des suggestions personnalisées pour améliorer leur satisfaction globale [9].

Un SR est un environnement logiciel qui permet de fournir à un utilisateur un contenu pertinent en fonction de la connaissance de l'utilisateur, du contenu (des données sur l'utilisateur et l'item) et des interactions entre l'utilisateur et l'item (feedback) [10].

2.2.2 Processus

De manière générale, un système de recommandation fonctionne selon les trois phases consécutives suivantes : « la phase de la collection des données », la phase « d'apprentissage ou traitement des données » et la phase dite « recommandation ou prédiction ». La première phase de collection permet le recueil, en particulier, des différentes données relatives aux utilisateurs et de leurs préférences dans une base de données comme illustré dans la figure suivante. Dans la phase d'apprentissage, le SR applique des algorithmes d'apprentissage pour filtrer et exploiter les caractéristiques de l'utilisateur à partir des données recueillies lors de la phase de collection des données. Au cours de la phase de recommandation, le SR prédit le type d'items que l'utilisateur pourrait préférer. Cela peut être fait soit directement à partir de l'ensemble de données collectées, soit par les activités observées de l'utilisateur par le système [11, 1].

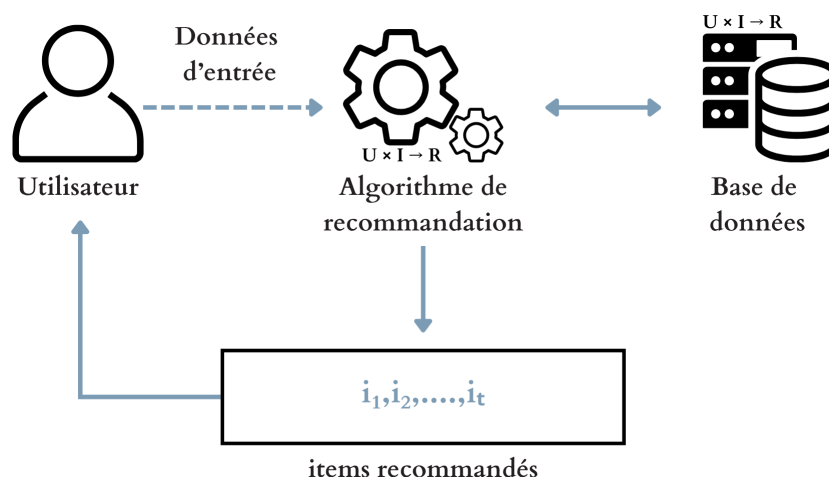


FIGURE 2.1 – Aperçu simplifié d'un processus de recommandation [1]

La tâche de recommandation peut être formulée comme suit : Soit $U = \{u_1, u_2, \dots, u_k\}$ l'ensemble des utilisateurs et $I = \{i_1, i_2, \dots, i_t\}$ l'ensemble de tous les items possibles qui peuvent être recommandés. Soit $f : U \times I \rightarrow \mathbb{R}$, une fonction d'utilité qui mesure l'utilité de l'item i pour l'utilisateur u , où \mathbb{R} est un ensemble totalement ordonné de valeurs d'utilité ou d'évaluations. Comme l'illustre la figure 2.1, les principaux éléments d'un SR sont comme suit [1] :

- **Les données d'entrée** : sont les informations saisies explicitement ou générées implicitement par l'utilisateur pour initialiser le processus de recommandation. Elles peuvent inclure des informations sur le type d'item demandé, ainsi que des informations sur le profil de l'utilisateur.
- **Une base de données** : est un système qui permet de stocker et d'organiser des informations relatives aux profils des utilisateurs ainsi qu'aux caractéristiques des items.
- **L'algorithme de recommandation** : cet algorithme utilise les données d'entrée fournies par l'utilisateur et les informations stockées dans la base de données pour générer une liste d'items recommandés à l'utilisateur actif¹.

De manière générale, deux profils sont distingués, celui de l'utilisateur et celui de l'item.

D'une part, les profils des utilisateurs comprennent des informations sur les caractéristiques personnelles telles que l'âge, le sexe, le pays, etc. ainsi que sur les préférences, qui peuvent être exprimées sous forme de scores d'évaluation pour les items consultés, achetés ou visités.

D'autre part, les profils des items contiennent des informations sur les caractéristiques des

1. L'utilisateur actif peut également être appelé utilisateur cible ou utilisateur courant.

items à recommander, tels que les produits, les lieux ou les activités. Par exemple, les profils des items peuvent inclure des détails sur les taxis, les musées, les restaurants, etc.

2.3 Les systèmes de recommandation intelligents

Les systèmes de recommandation intelligents (SRI) sont des logiciels conçus dans le but de suggérer des éléments pertinents aux utilisateurs en prenant en considération leurs préférences, leurs historiques d'interaction ou des paramètres contextuels. En recourant à des méthodes sophistiquées d'intelligence artificielle (IA), d'apprentissage automatique, de traitement du langage naturel (NLP) et d'analyse de mégadonnées, ces systèmes produisent des recommandations personnalisées [12]. Le but principal consiste à assister les utilisateurs dans la prise de décisions éclairées tout en améliorant leur expérience en abordant la question de la surcharge informationnelle.

les systèmes de recommandation intelligents peuvent utiliser des items, des vidéos, de la musique, des produits, des services et des activités pour adaptées aux besoins des utilisateurs et à rendre les clients plus satisfaits [13].

2.4 Domaines d'expertise des systèmes de recommandation.

Le développement de ces systèmes repose sur une approche multidisciplinaire. Il implique plusieurs domaines d'expertise, notamment :

1. **Intelligence artificielle (IA)** : Utilisée pour optimiser les algorithmes de recommandation afin d'améliorer la pertinence des suggestions.
2. **Interaction homme-machine (IHM)** : Pour concevoir des interfaces utilisateur intuitives et accessibles.
3. **Exploration de données (Data mining)** : Pour analyser d'énormes volumes de données et en extraire des informations utiles.
4. **Statistiques** : Pour modéliser les comportements des utilisateurs et affiner les recommandations.
5. **Analyse du comportement des utilisateurs** : Pour anticiper les besoins et les tendances des utilisateurs.

2.5 Collection des données

Pour qu'un SR soit créé, il est nécessaire de recueillir des données, étape centrale dans la conception de ces systèmes, qui dépendent principalement de ce que ces derniers obtiennent des utilisateurs. Ces données permettent de créer un profil utilisateur et les algorithmes de recommandation et sont généralement de trois types : celles d'éléments à recommander, d'utilisateurs et générées à partir des interactions. Plus précisément, il s'agit d'informations données en fonctionnement du comportement des utilisateurs du service, avec ce dernier étant le feedback dont les liens montrent une connexion entre les utilisateurs et les items recommandés. Il existe deux formes de feedback, explicites et implicites [14].

2.5.1 Types de feedback

Le feedback implicite et explicite sont deux types de données couramment utilisées dans les SR pour comprendre les préférences des utilisateurs et formuler des recommandations précises [10] :

2.5.1.1 Feedback explicite

Les feedbacks explicites montrent ce que les utilisateurs pensent du contenu. C'est l'information la plus utile car elle vient directement de l'utilisateur et montre son intérêt pour l'objet. Pour obtenir des avis clairs des utilisateurs, le système doit leur demander d'évaluer les éléments. Après avoir recueilli des avis, le système évalue à quel point un item correspond aux goûts des utilisateurs. Bien que cela aide la personne qui recommande à comprendre l'avis réel de l'utilisateur, il est souvent difficile de le collecter. Si nous prenons le cas d'une application de cinéma, il y a de fortes chances que certains utilisateurs ne votent jamais pour un seul film, ou ne donnent que des évaluations succinctes.

Il existe plusieurs façons de recueillir les avis des utilisateurs. Mettre en place une fonction pour aimer ou ne pas aimer permet aux gens d'évaluer facilement le contenu. Le système peut aussi demander aux utilisateurs de donner une note, qui montre à quel point ils ont aimé ou non le contenu.

2.5.1.2 Feedback implicite

Les feedbacks implicites ne nécessitent aucune participation de l'utilisateur et peuvent être générés en observant le comportement de l'utilisateur, par exemple quel élément il a vu, où il a cliqué ou combien de temps il a passé à afficher un type spécifique d'éléments. Les activités à suivre doivent être déterminées en fonction du domaine dans lequel le système de recommandation fonctionne.

Les systèmes de recommandation utilisent souvent une combinaison de rétroaction implicite et explicite pour obtenir une compréhension plus complète des préférences des utilisateurs. En analysant les deux types de rétroaction, les systèmes de recommandation peuvent générer des recommandations plus précises et personnalisées, améliorant la satisfaction des utilisateurs et la performance globale du système.

2.6 Approches classiques

Les systèmes de recommandation peuvent être personnalisés (SRP) ou basés sur le groupe (SRG) [15]. Dans le monde réel, nous rencontrons fréquemment des systèmes de recommandation personnalisés, en particulier sur les sites de commerce électronique, car ils anticipent les choses pour un individu tout en tenant compte des préférences de cet individu. Cependant, les systèmes de recommandation de groupe font des recommandations pour un groupe d'utilisateurs, chaque membre du groupe ayant ses préférences, ce qui rend difficile à gérer car nous devons satisfaire tous les membres du groupe.

Il existe trois stratégies populaires utilisées par les systèmes de recommandation pour générer des recommandations [10, 16] comme le montre la figure suivante :

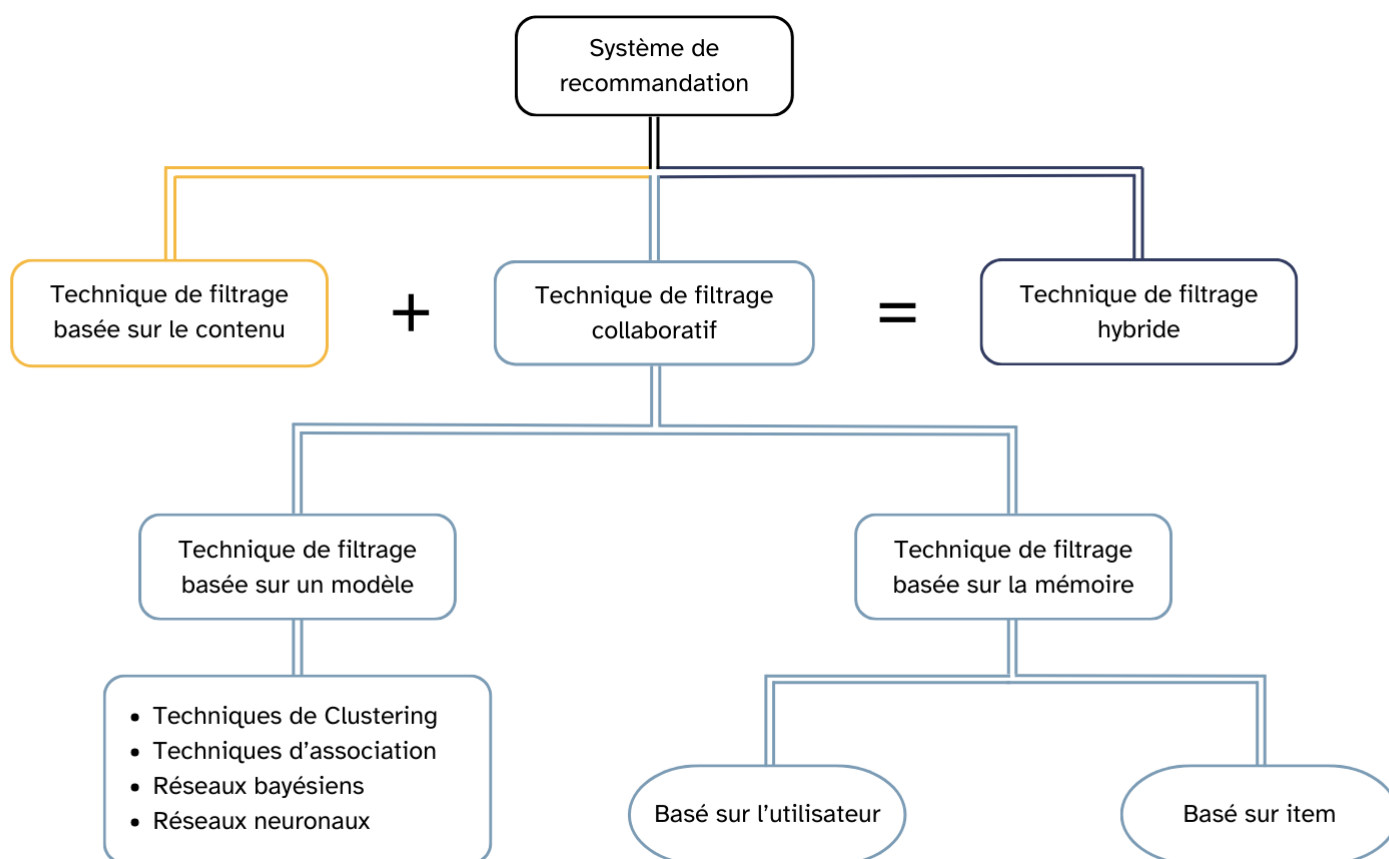


FIGURE 2.2 – Stratégies populaires utilisées par les systèmes de recommandation [1]

2.6.1 Approches de filtrage basées sur le contenu (FBC)

La prémisse fondamentale du filtrage basé sur le contenu est que **si l'utilisateur aime un élément, il voudra probablement un élément connexe** comme indiqué dans la figure 2.3.

Le SR utilisera les métadonnées des éléments précédents de l'utilisateur pour produire des recommandations. Les techniques de filtrage fondées sur le contenu tirent parti des renseignements sur les personnes et les items. Les caractéristiques, les descriptions textuelles et les étiquettes ne sont que quelques exemples de la façon dont cet élément pourrait être présenté. Le concept clé est de faire correspondre les profils des utilisateurs avec les profils des items en fonction des préférences de l'utilisateur quant aux qualités des items afin d'offrir des choses appropriées.

En conséquence, les utilisateurs reçoivent des recommandations pour des produits similaires à ceux avec lesquels ils ont interagi. Extraire les données pertinentes du contenu des éléments, créer des profils d'utilisateur et d'élément et filtrer les éléments en fonction des simi-

litudes de profil sont autant d'étapes nécessaires pour déployer une méthode de **FBC**. Cependant, pour utiliser cette stratégie, l'activité de l'utilisateur précédent est requise [16].

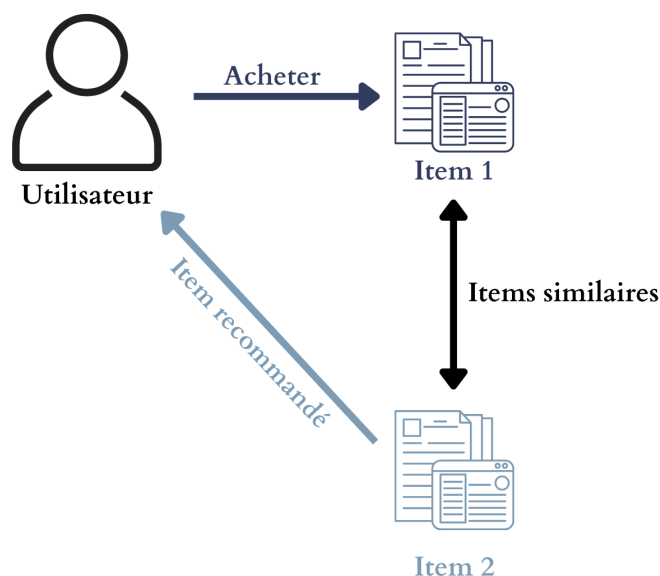


FIGURE 2.3 – Processus logique d'un SR basé sur le filtrage par contenu

Néanmoins, cette approche présente quelques limites, telles que le manque de diversité², la surspécialisation³ des recommandations, etc.

2.6.2 Approches basées sur le filtrage collaboratif

Le filtrage collaboratif (FC) est un processus de filtrage (ou d'évaluation) de l'information, basé sur le partage d'opinions entre les utilisateurs. Son but est de deviner les notes manquantes en utilisant les notes déjà données [17].

Les méthodes de FC peuvent être classées en deux types : celles qui utilisent la mémoire et celles qui utilisent un modèle, selon l'algorithme utilisé.

- Netflix propose des films selon les goûts des utilisateurs qui se ressemblent.

Il y a deux types principaux de filtrage collaboratif :

2. Il propose des contenus trop similaires, limitant la découverte.
3. Il recommande toujours le même type de contenu, empêchant l'exploration d'autres formes.

2.6.2.1 FC basé sur la mémoire

Contrairement aux méthodes de filtrage qui dépendent des modèles mathématiques et des algorithmes informatiques, les techniques de filtrage basées sur la mémoire utilisent les données d'interaction utilisateur-élément existantes pour générer des recommandations. La figure ci-après illustre comment ces méthodes conservent et examinent les interactions passées entre les utilisateurs et les éléments pour découvrir des modèles et des similitudes [16].

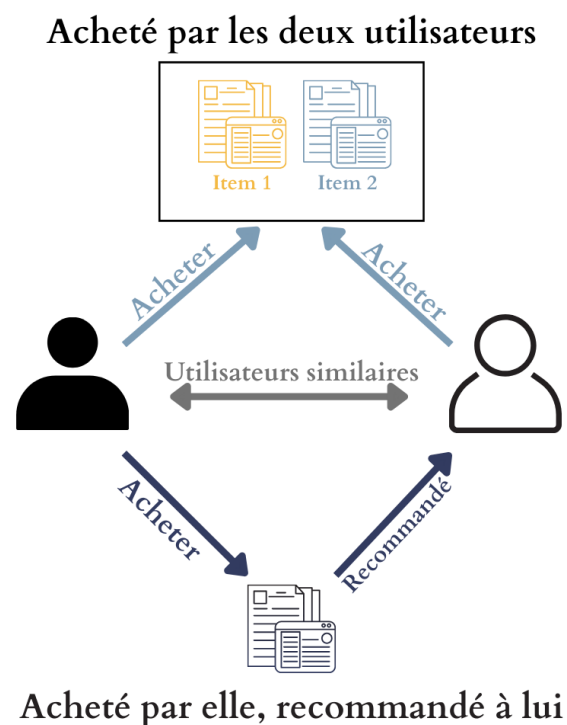


FIGURE 2.4 – Processus logique d'un SR sur le filtrage collaboratif

Il existe deux principaux types de techniques de filtrage basées sur la mémoire [10] :

- **FC basé sur les utilisateurs :**

Il s'agit d'une technique **qui permet de filtrer les éléments qui peuvent intéresser un utilisateur en fonction des réactions des autres utilisateurs**. Il fonctionne en recherchant un grand groupe d'individus ayant des intérêts similaires à un utilisateur spécifique, puis en considérant les éléments qu'ils apprécient pour générer une liste de recommandations classées.

- **FC basé sur les items :**

Qui suggère un item basé sur des items que **l'utilisateur a déjà acheté ou aimé**. Il recommande des items similaires à ceux que l'utilisateur a déjà aimés. En termes simples,

la collaboration basée sur les items se concentre sur les actions des autres utilisateurs concernant l'item que vous envisagez d'acheter. Les caractéristiques du produit n'ont pas d'importance pour faire une recommandation. Amazon a inventé le filtrage collaboratif basé sur l'item. Il est plus rapide et plus stable que le filtrage basé sur l'utilisateur dans les systèmes avec plus de personnes que d'objets. Il est efficace parce que la note moyenne attribuée à un item ne change généralement pas aussi rapidement que la note moyenne attribuée par un utilisateur à d'autres items [18].

Limitation de FC basé sur la mémoire :

Le FC basé sur la mémoire présente certains problèmes fréquents. Il ne fonctionne pas optimalement avec les nouveaux utilisateurs ou les nouvelles ressources, en raison d'un manque d'informations suffisantes pour fournir des conseils. Il est également restreint lorsque les données sont peu abondantes, car chaque utilisateur n'exploite qu'une fraction minuscule des contenus existants. En définitive, il privilégie principalement les contenus populaires, diminuant ainsi le degré de personnalisation [19].

2.6.2.2 FC basé sur le modèle

Méthode de filtration selon un modèle. Les techniques de filtrage basées sur un modèle changent la façon de faire des recommandations. Au lieu de se baser sur des informations détaillées sur les éléments ou les utilisateurs, ces méthodes utilisent des Techniques de Clustering⁴ et des réseaux de neurones. Elles se servent de formules et de modèles mathématiques pour trouver des tendances et des liens cachés dans les données des interactions entre les utilisateurs et les éléments. Ces modèles peuvent donner des conseils et des prévisions précises en trouvant des caractéristiques qui montrent ce que les utilisateurs aiment et ce qui décrit les objets. En tirant parti de la puissance des réseaux neuronaux et de la technologie de clustering, les méthodes de filtrage basées sur des modèles convertissent les données brutes en informations utiles, favorisant ainsi des recommandations personnalisées et augmentant ainsi la satisfaction des utilisateurs. Ces méthodes révèlent non seulement des préférences subtiles, mais identifient également de nouvelles associations entre les éléments, ouvrant la voie à des avancées dans les systèmes de recommandation dans de nombreux domaines [20].

Le FC a aussi des limites comme le démarrage difficile, le manque de données, et les diffi-

4. le clustering revient à diviser les personnes en groupes en fonction de leurs préférences

cultés à s'adapter à un grand nombre de cas. Pour résoudre ces problèmes, nous avons proposé des méthodes mixtes que nous expliquerons dans la prochaine section [20].

2.6.2.3 Systèmes de recommandation hybrides

Un SR hybride intègre le filtrage de contenu à des techniques collaboratives. En fusionnant ces approches, il est possible de remédier aux limitations rencontrées lorsque chacune est utilisée seule et, dans certains scénarios, cette combinaison peut s'avérer plus efficace[10]. Il existe de nombreuses façons de mettre en œuvre des systèmes de recommandation hybrides, notamment en générant des prédictions séparément via des méthodes basées sur le contenu et collaboratives avant de fusionner les résultats, ou en améliorant une méthode avec les capacités de l'autre.

Comme illustré dans la figure 2.5, un tel système de recommandation fonctionne. L'utilisateur interagit avec des items en exprimant son intérêt ou en évaluant leur qualité. Deux types de filtrage sont utilisés pour analyser ces interactions : basé sur le contenu, qui lui recommande des items similaires à ceux qu'il a aimés par le passé, et le filtrage collaboratif, qui suggère des items en fonction des préférences d'autres utilisateurs ayant des goûts similaires. De cette manière, la recommandation hybride permet à la fois un contenu plus compatible et mieux ajusté par rapport à l'utilisation.

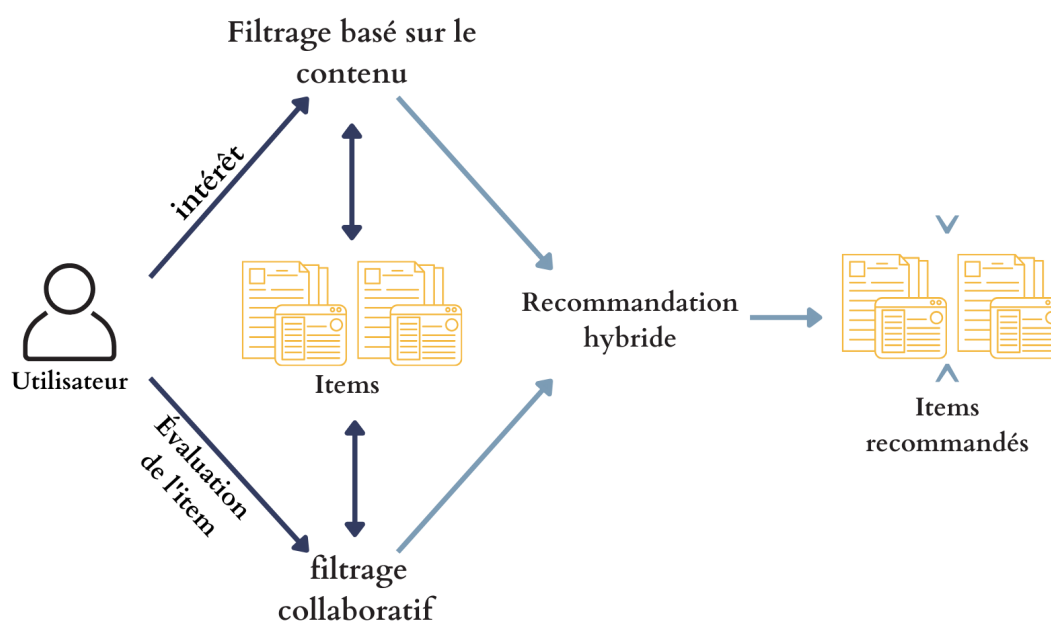


FIGURE 2.5 – Processus logique d'un système de recommandation hybride.

Plusieurs études comparent la performance des procédures conventionnelles aux méthodes hybrides et concluent que l'utilisation de méthodes hybrides permet d'obtenir des recommandations plus précises [21].

Exemple : Amazon se sert de deux méthodes pour suggérer des produits : le filtrage collaboratif et le filtrage basé sur le contenu [18].

2.7 L'Évolution et les Défis des Systèmes de Recommandation Intelligents.

Les premières entreprises à mettre en œuvre des systèmes intelligents étaient des entreprises du commerce électronique, des services en ligne et du contenu multimédia. En 1994, Amazon était l'une des premières à utiliser le filtrage collaboratif pour modifier le paysage des achats en ligne en suggérant des produits achetés par d'autres clients avec des tendances d'achat similaires [22]. Netflix en 1997 a accompagné l'annonce avec des recommandations de films et d'émissions de télévision, avec le célèbre Netflix Award de 2006 à 2009, un concours en ligne pour l'algorithme de filtrage collaboratif [23]. La plateforme a basé un système de recommandation avancé pour la musique en 2000 lorsque Pandora Media a été fondée et a commencé à utiliser les fonctions audio des titres pour créer des stations de radio personnalisées pour leurs abonnés [24]. De nombreuses autres éminentes sociétés de l'industrie ont participé à faire évoluer davantage les systèmes de recommandation en implémentant un filtrage collaboratif et des techniques basées sur le contenu, ce qui a permis d'améliorer la personnalisation et l'intelligence de ces systèmes.

Les SR améliorent l'expérience des utilisateurs grâce à l'intelligence artificielle. Selon Microsoft (Microsoft, 2024), ces systèmes analysent les préférences des utilisateurs pour leur proposer des suggestions pertinentes, renforçant ainsi leur satisfaction et leur intérêt [25]. Une étude de Polytechnique Montréal (Sarai, 2022) discute de l'utilisation du filtrage collaboratif pour recommander des produits, et mentionne les problèmes liés aux données manquantes [26]. L'Université du Québec (UQAM, 2021) montre que les recommandations deviennent plus efficaces quand on analyse ce que font les utilisateurs et les informations sur les produits [27]. En outre, IBM (2023) fournit une explication détaillée de la façon dont les moteurs de recommandation fonctionnent et pourquoi ils sont une solution nécessaire pour personnaliser le contenu.

Un article du Vector Institute indique comment les universités et les entreprises collaborent pour améliorer ces systèmes et mieux comprendre les préférences des utilisateurs [28]. Ainsi, on peut voir qu'en raison des progrès réalisés, l'IA a un impact considérable sur les recommandations, les rendant plus précis et adapté à chaque individu.

Du point de vue des chercheurs et des développeurs, les SRI sont des outils qui permettent de personnaliser l'interaction de l'utilisateur et d'augmenter l'engagement sur une plateforme web [9]. Ils fonctionnent sur la base du filtrage collaboratif et du filtrage basé sur le contenu pour recommander des produits, des films, de la musique et bien plus encore. Cependant, ils présentent aussi des défis, notamment la qualité et la diversité des recommandations, qui peuvent être limitées par des biais algorithmiques [20]. L'évolutivité est également un problème, car ils doivent être compétents pour traiter des quantités massives de données et maintenir une haute performance [29]. De plus, il est très important de protéger les données des utilisateurs. Ils doivent pouvoir contrôler leurs informations personnelles pour empêcher les abus [30]. Pour améliorer ces systèmes, beaucoup de chercheurs suggèrent de mélanger différentes méthodes pour mieux faire des recommandations aux nouveaux utilisateurs ou produits. L'avenir des recommandations dépend de l'intelligence artificielle et de l'apprentissage profond. Ces technologies aident à donner des conseils plus précis en comprenant mieux ce que veulent les utilisateurs. En résumé, même si ces systèmes ont beaucoup d'avantages, il faut les améliorer pour donner des conseils plus précis, variés et respectueux de la vie privée des utilisateurs [31].

2.8 Les systèmes de recommandation adaptatifs en éducation⁵

2.8.1 Définition

Les systèmes de recommandation ont également amélioré l'éducation en la rendant plus ouverte et adaptée à chaque élève. Les contenus et activités recommandés étaient faits sur commande et calibrés pour chaque élève en fonction de ses besoins et de ses préférences. Cela élimine les inconvénients de l'éducation classique où tous les élèves suivent le même

5. Les systèmes de recommandation adaptatifs sont aussi nommés systèmes de recommandation intelligents, car ils personnalisent les suggestions selon les besoins ou le comportement des utilisateurs.

programme, sans prendre en compte leurs différences, et les enseignants n'ont pas la force, le pouvoir ou le désir de donner assez d'attention à chaque élève. Les consoles spécialisées pour aider l'enfant à apprendre. Il est important pour les étudiants de trouver des ressources à l'avenir qui les guideront sur le chemin de l'éducation, les motiveront [32].

Par exemple :

- **Coursera et edX** : sont des plateformes de formation en ligne qui proposent des cours en fonction des intérêts, du parcours et des cours précédemment suivis par l'utilisateur. Grâce à des systèmes de recommandation, elles suggèrent des contenus pertinents afin de favoriser la progression personnalisée de chaque apprenant [33].
- **Duolingo** : adapte les exercices en fonction des réponses de l'utilisateur : en cas d'erreur, l'application propose des exercices similaires pour renforcer la compréhension ; en cas de réussite, elle passe à l'étape suivante. Ce système s'appuie sur un apprentissage adaptatif basé sur les performances en temps réel [34].
- **Knewton** : propose des parcours d'apprentissage entièrement personnalisés. Elle analyse les réponses de l'utilisateur, identifie les difficultés, puis ajuste dynamiquement les contenus afin d'optimiser la compréhension et la progression [35].
- **Smart Sparrow** : offre aux enseignants des outils pour concevoir des cours interactifs et adaptatifs. Les activités pédagogiques s'ajustent en fonction des réponses de l'élève, afin de mieux répondre à ses besoins individuels [36].

2.8.2 Fondements Théoriques

L'apprentissage adaptatif repose sur plusieurs théories pédagogiques clés qui expliquent comment l'individu apprend de manière efficace :

- **Le constructivisme (Piaget, Vygotsky)** : Cette théorie considère que l'apprenant construit activement ses connaissances à partir de ses expériences personnelles et de ses interactions sociales. L'enseignement doit donc permettre à l'élève de manipuler, explorer et construire du sens par lui-même. Les systèmes adaptatifs s'appuient sur cette approche en proposant des contenus interactifs et adaptés au niveau de compréhension de l'élève [37].
- **La théorie de la charge cognitive (Sweller)** : Elle insiste sur la capacité limitée de la mémoire de travail. Si une tâche est trop complexe ou trop chargée en informations,

l'apprenant peut être submergé. L'apprentissage adaptatif utilise cette théorie pour ajuster la difficulté des exercices afin de ne pas surcharger l'élève, en divisant les concepts en étapes progressives [38].

- **La zone de développement proximal (ZDP) (Vygotsky) :** C'est l'écart entre ce qu'un apprenant peut faire seul et ce qu'il peut accomplir avec de l'aide. Les systèmes adaptatifs tentent d'identifier cette zone pour proposer des tâches légèrement au-dessus du niveau actuel de l'élève, tout en lui fournissant un soutien (exemples, rappels, explications) pour progresser. [39]

2.8.3 Modèles d'apprentissage adaptatif

Les systèmes de recommandation adaptatifs utilisent des modèles d'apprentissage automatique capables d'ajuster leur comportement selon le niveau, les préférences ou les performances de l'utilisateur. Dans un contexte éducatif, cela permet de proposer des contenus, des exercices ou des parcours personnalisés.

1. **Modèle basé sur la maîtrise des compétences :** Ce modèle permet à l'apprenant de progresser à son propre rythme. Il ne passe à l'étape suivante que lorsque les connaissances précédentes sont maîtrisées. Les systèmes de tutorat intelligents (Intelligent Tutoring Systems – ITS) sont des exemples typiques de ce modèle [40].
2. **Modèle d'évaluation diagnostique :** L'objectif de ce modèle est d'identifier les points forts et les difficultés de l'apprenant afin de lui recommander des ressources ou des exercices adaptés. Ce principe est à la base des systèmes de recommandation pédagogique [41].
3. **Modèle basé sur le profil de l'apprenant :** Ce modèle utilise les caractéristiques individuelles de l'apprenant telles que son style cognitif, ses préférences ou encore son historique d'apprentissage. Les systèmes hypermédias adaptatifs (Adaptive Hypermedia Systems – AHS) reposent sur cette approche [42].
4. **Modèle prédictif et adaptatif :** Ce modèle intègre des techniques d'intelligence artificielle et d'apprentissage automatique (machine learning) pour prédire les besoins futurs de l'apprenant et ajuster les contenus proposés en conséquence. Il est couramment utilisé dans les plateformes modernes d'apprentissage numérique [43].

2.8.4 Méthodes et algorithmes d'implémentation de l'apprentissage adaptatif

Après avoir présenté les modèles d'apprentissage adaptatif, il est nécessaire d'explorer les méthodes et algorithmes qui rendent ces modèles opérationnels dans les systèmes éducatifs intelligents. Ces techniques permettent de personnaliser le parcours pédagogique, en s'appuyant sur des analyses des comportements d'apprentissage, des performances et des préférences de chaque apprenant.

Les principales approches utilisées dans la mise en œuvre de ces systèmes sont les suivantes [7] :

1. Modélisation cognitive de l'apprenant (User Modeling)

Cette méthode consiste à construire un profil dynamique de l'élève, en analysant ses connaissances, ses erreurs, ses stratégies d'apprentissage, et ses progrès au fil du temps. L'objectif est de comprendre comment l'apprenant pense, pour adapter les contenus pédagogiques en conséquence.

- Le système identifie les lacunes conceptuelles à partir des réponses de l'élève.
- Il ajuste la difficulté des contenus proposés selon le niveau de maîtrise.
- Il adapte ses explications à la logique de raisonnement de l'utilisateur.

Exemple : Un tuteur intelligent en mathématiques détecte qu'un élève confond les fractions équivalentes. Il adapte alors les exercices pour renforcer cette notion [44].

2. Moteurs de règles pédagogiques

Ce composant applique des règles prédéfinies pour adapter l'enseignement à chaque élève. Ces règles peuvent porter sur la progression, la difficulté, les rétroactions ou l'engagement de l'apprenant.

- Les exercices sont personnalisés en fonction du niveau de compétence détecté.
- **Retour formatif :** des explications détaillées sont fournies après chaque réponse.
- **Gamification :** intégration de mécanismes ludiques (badges, scores, défis) pour motiver.

Exemple : Dans Duolingo, si l'utilisateur échoue à plusieurs reprises sur un mot, l'applica-

tion renforce cet apprentissage avec des activités ciblées [45].

3. Intelligence artificielle et apprentissage machine

L'intelligence artificielle permet aux systèmes d'apprentissage adaptatif d'apprendre eux-mêmes à mieux aider les étudiants. Grâce aux algorithmes d'apprentissage automatique, ils peuvent analyser les données d'interaction, prédire les besoins futurs et recommander les contenus les plus appropriés.

- **Systèmes de recommandation** : suggestion automatique de cours ou d'activités selon le profil de l'élève.
- **Analyse prédictive** : détection anticipée des difficultés d'apprentissage.
- **Clustering et classification** : groupement des élèves par profil pour ajuster les parcours.

Exemple : Une plateforme adaptative propose à un élève un module de grammaire ciblé, car ses erreurs répétées suggèrent une faiblesse dans ce domaine [46].

2.8.5 Critères d'adaptation contextuelle

La mise en œuvre d'un système de recommandation adaptatif (SRA) nécessite la gestion de diverses informations contextuelles relatives à l'utilisateur, décrivant le contexte dans lequel une entité ou un service est utilisé. Ces informations contextualisées sont déclinées en cinq catégories [47] :

2.8.5.1 Individualité dans l'adaptation

L'individualité dans l'adaptation est relative aux caractéristiques de l'entité concernée, qui peuvent être : naturelles (éléments de l'environnement), humaines (préférences de l'utilisateur), artificielles (objets technologiques et leurs données), et de groupe (entités liées par des caractéristiques ou relations communes). Elles permettent de diversifier les recommandations et d'inclure des éléments plus spécifiques [47].

2.8.5.2 Relations

La dimension relationnelle regroupe les liens entre entités humaines ou objets, classés en relations sociales (ex : amis, collègues), fonctionnelles (ex : utilisation d'un objet) et de composition (ex : composants d'un appareil). Ces relations permettent d'enrichir les profils via des

rôles ou des usages, et varient en cardinalité selon leur nature (ex : un objet peut appartenir à plusieurs entités) [47].

2.8.5.3 Temps

La dimension temporelle est centrale pour l'adaptation des systèmes, que ce soit en guise de données temporelles brutes (date, heure ou catégorisées week-ends, saisons, plages horaires).L'intégration d'éléments récurrents et d'intervalles spécifiques enrichit les profils utilisateurs, tandis que l'historique des interactions permet d'identifier des habitudes et d'anticiper les comportements futurs [48].

2.8.5.4 Activité

Cette catégorie décrit ce qu'une entité fait ou envisage de faire, à travers des objectifs, tâches et actions. Une tâche, orientée vers un objectif, peut être décomposée en sous-tâches spécifiques à un domaine. L'objectif peut être déterminé manuellement ou automatiquement, en fonction d'une politique d'adaptation visant à sélectionner celui dont la probabilité de réussite est la plus élevée[49].

2.8.5.5 Localisation

Avec la généralisation des appareils mobiles, la prise en compte de la localisation devient essentielle pour l'adaptation. Un emplacement peut être exprimé de manière absolue (coordonnées GPS) ou relative (quartier, commerce...). Les modèles spatiaux utilisent à la fois des données quantitatives (coordonnées numériques) et qualitatives (nom de région, bâtiment). Le concept de superposition permet de convertir les données quantitatives en représentations qualitatives, structurant efficacement l'information spatiale pour une meilleure adaptation. Une entité dispose toujours d'un emplacement qualitatif associé à des données quantitatives [47].

2.8.6 Les objectifs des systèmes d'apprentissage Adaptatif

Ces systèmes sont conçus pour atteindre plusieurs objectifs fondamentaux : améliorer la qualité et l'efficacité de l'éducation, augmenter l'intérêt des utilisateurs, limiter les abandons dans les cours en ligne et aider les profs à suivre les élèves qui ont du mal. En résumé, ils aident à faire que l'éducation soit plus juste et inclusive tout en mettant les apprenants au centre du chemin éducatif [50].

Deuxième partie : Mécanismes utilisés dans notre système

2.9 Mécanismes de recommandation utilisés

L'enseignement de la langue anglaise aux enfants, notamment ceux âgés de 3 à 12 ans, nécessite une approche pédagogique adaptée à leurs besoins individuels, leur niveau de compétence et leurs préférences d'apprentissage. Les nouvelles stratégies éducatives de type intelligent, fondées sur les technologies éducatives ou sur l'utilisation de systèmes de recommandation à des fins pédagogiques deviennent apparentes pour proposer aux apprenants des activités en fonction de différents critères linguistiques, cognitifs ou psychologiques, capables d'agir comme un préalable à une démarche appropriée de façon flexible et interactive dans la continuité de l'apprentissage tout en privilégiant le plaisir d'apprendre pour maintenir l'attention des jeunes apprenants tout en optimisant leur progression.

Le système de recommandation que nous avons mis en place repose sur plusieurs éléments :

- Le questionnaire de profil personnelles initialement rempli par les parents,
- L'historique de navigation de votre enfant dans l'application,
- Des tests adaptatifs de difficulté en fonction des périodes de concentration de la journée,
- Des filtres liés au type d'activité (vidéo, texte, audio) et au temps passé sur chaque ressource.
- Les préférences d'apprentissage d'autres enfants similaires.

À partir de ces données, le système peut proposer des activités ciblées, adaptées aux besoins et aux préférences de chaque enfant, afin de leur permettre de se sentir davantage impliqués et de progresser dans l'apprentissage de la langue anglaise.

Dans cette section, nous expliquerons en détail les différents mécanismes utilisés dans notre système de recommandation, ainsi que les types de filtrage adoptés selon chaque approche pour proposer une expérience d'apprentissage personnalisée et efficace.

2.9.1 Recommandation basée sur le questionnaire de profil :

Description :

Lors de l'inscription, le parent remplit un questionnaire de profil concernant son enfant, dans lequel figurent des informations sur :

- L'âge de l'enfant.
- Le niveau d'anglais (débutant, intermédiaire, avancé).
- Les sujets préférés (par exemple : animaux, nourriture, nature, etc).
- Les points faibles de l'enfant (compréhension orale, lecture, vocabulaire, grammaire).
- Le moment de la journée où l'enfant est le plus concentré ou réceptif (le parent peut indiquer si l'enfant est plus disponible le matin, l'après-midi, le soir, etc.).

Type de recommandation :

Filtrage basé sur le contenu (Content-Based Filtering) :

Recommandations de situations pédagogiques correspondant aux caractéristiques et préférences renseignées dans le questionnaire.

Par exemple :

- Si un enfant est en niveau débutant en anglais et jour de difficultés en compréhension orale, le système recommandera des activités interactives à partir de vidéos et audios simples, à un rythme et à un vocabulaire simples, en adéquation avec son profil.

Mécanisme :

Matching : le système utilise les données du profil pour construire un **profil d'intérêt** pour l'enfant et recommande des activités à partir de ce profil.

2.9.2 Recommandation basée sur l'historique de navigation (Nombre de clics, Temps cumulé, Feedback, etc.) :

Description :

Le système suit l'interaction de l'enfant avec les activités proposées, en collectant des données telles que :

- **Nombre de clics** sur les activités.
- **Temps passé** sur chaque activité.
- **Feedback fourni** par l'enfant (activité réussie, score obtenu, etc.).
- **Fréquence des interactions** avec certaines catégories d'activités (par exemple, vidéos, audios, textes, etc.).

Type de recommandation :

- **Filtrage d'activité** : Le système identifie le type de ressource le plus interactives (vidéos, audios, textes). **par exemple**, Si un enfant utilise surtout des vidéos, le système montrera surtout ce genre de contenu dans les recommandations d'avenir.
- **Filtrage par temps d'engagement** : Le système mesure le temps passé sur chaque type de ressource. Par exemple, passer beaucoup de temps sur des vidéos montre mieux l'intérêt et change ce qui est recommandé après.

Mécanisme :

Le système analyse les informations concernant les clics et le temps passé sur chaque activité afin de déterminer ce que l'enfant aime vraiment. Il se construit ainsi une image claire des préférences de l'enfant et identifie le type d'activités les plus intéressantes pour lui.

2.9.3 Recommandation basée sur des tests de différents niveaux de difficulté :

Description :

Le système adapte les tests selon une **difficulté** ressentie susceptible de varier selon certains facteurs :

- **Le niveau de l'enfant** : l'application peut moduler la difficulté des tests en fonction du niveau d'anglais de l'enfant.
- **La période de navigation** : les performances peuvent varier selon l'heure de la journée (par exemple, les enfants peuvent être plus concentrés le matin, l'après-midi ou le soir), ce qui peut alors collaborer au choix du test.

Type de recommandation :

- **Filtrage l'heure** : Analyse les moments d'apprentissage, notamment les périodes de la journée et le temps d'attention, à partir des informations renseignées au questionnaire de profil (comme le moment de la journée où l'enfant est le plus concentré ou réceptif), sous entend, que les capacités de concentration, d'énergie ou d'engagement de l'enfant se font très variables selon ces déterminations.

Mécanisme :

Filtrage selon l'heure de la journée (filtrage de l'heure) : le système adapte dynamiquement les propositions en fonction des plages horaires.

- **Matin (6h – 12h)** : si l'enfant est plus attentif dans la matinée, le système propose des tests de difficulté élevée exploiter sa meilleure attention.
- **Après-midi (13h – 16h)** : des tests de difficulté modérée sont proposés, équilibrant stimulation et fatigue.
- **Soir (17h – 21h)** : on privilégiera des tests plus simples et interactifs pour une attention de l'enfant probablement moins présente à ce moment de la journée.

Ce mécanisme d'adaptation temporelle contribue à une meilleure gestion de la charge cognitive de l'enfant, tout en maintenant une progression pédagogique cohérente et personnalisée.

2.9.4 Recommandation basée sur des apprenants similaires :

Description :

Les enfants ayant des comportements ou des profils similaires peuvent bénéficier des mêmes types d'activités. Le système peut exploiter les données issues des **comportements d'apprentissage** pour identifier des apprenants similaires à un enfant donné.

Type de recommandation :

- **Filtrage collaboratif basé sur l'utilisateur (User-Based Collaborative Filtering)** : Ce type de filtrage repose sur l'hypothèse que si deux enfants ont des comportements ou des profils d'apprentissage similaires, leurs préférences en matière d'activités seront

également proches. **Par exemple**, si un enfant échoue à un test deux fois, le système détecte ce comportement comme un signe de difficulté. Il cherche ensuite d'autres enfants qui ont été réussis dans la même situation avec certaines activités. Le système recommande alors ces activités spécifiques (par exemple, des vidéos éducatives, des audios et des textes simples) à l'enfant en difficulté pour l'aider à progresser.

Mécanisme :

- **Identification des similarités entre utilisateurs** : le système calcule des mesures de similarité entre les profils d'apprentissage des enfants, en se basant sur leurs préférences de contenu, leurs scores et leurs interactions. Les enfants ayant des profils proches bénéficieront de recommandations issues des activités ayant bien fonctionné pour d'autres.
- **Recommandation ciblée après échecs répétés** : Lorsqu'un enfant échoue deux fois à un test, le système active un mécanisme de recommandation basé sur l'expérience d'enfants similaires ayant réussi ces mêmes types de tests. Il suggère alors les ressources qui ont contribué à leur réussite.

2.10 Résumé des types de filtrage utilisés

- **Filtrage basé sur le contenu** pour personnaliser les recommandations en fonction des informations du profil de l'enfant (âge, niveau d'anglais, points faibles, moment de la journée. etc.).
- **Filtrage collaboratif (User-Based)** pour recommander les activités qui ont réussi avec des enfants présentant un profil ou des comportements similaires.
- **Filtrage d'activité** pour privilégier les ressources (vidéos, audios, textes) les plus interactives par l'enfant, en analysant ses choix préférés.
- **Filtrage par temps d'engagement** pour adapter les recommandations aux types de contenus sur lesquels l'enfant passe le plus de temps, indiquant un plus grand intérêt ou engagement.
- **Filtrage temporel** pour ajuster la difficulté des tests en fonction des tranches horaires (matin, après-midi, soir) afin de respecter le rythme cognitif de l'enfant.

Chaque mécanisme de filtrage utilise le type de donnée qu'elle exploite de façon à maximiser la personnalisation, et à répondre aux besoins éducatifs de chacun.

Chaque approche de filtrage exploite les informations d'une manière particulière pour maximiser la personnalisation et l'engagement des enfants, en considérant leurs exigences d'apprentissage individuelles.

2.11 Modèles d'apprentissage adaptatif utilisés

Notre système de recommandation utilise plusieurs modèles d'apprentissage pour s'adapter au profil et aux besoins de chaque enfant. Ces modèles permettent de proposer la bonne activité au bon moment selon les résultats, les préférences et le rythme de l'enfant. Il s'agit des modèles suivants.

2.11.1 Modèle basé sur la maîtrise des compétences

Ce modèle permet à l'enfant d'avancer étape par étape. Il ne passe à l'activité suivante que lorsqu'il a bien compris celle d'avant.

Applications :

- Recommander des contenus spécifiques en cas d'échecs répétés.
- Adapter les exercices selon les moments de la journée où l'enfant est le plus concentré (matin, après-midi, soir).

2.11.2 Modèle d'évaluation diagnostique

Ce modèle identifie les points forts et les points faibles de l'enfant pour mieux adapter les activités.

Applications :

- Personnaliser les activités à partir des réponses du parent dans le questionnaire.
- Modifier les contenus selon les résultats obtenus aux tests.

2.11.3 Modèle basé sur le profil de l'apprenant

Ce modèle prend en compte des informations personnelles comme l'âge, les goûts ou les difficultés de l'enfant.

Applications :

- Choisir des contenus adaptés au profil initial de l'enfant.
- Analyser l'historique d'utilisation pour affiner les recommandations.

2.11.4 Modèle prédictif et adaptatif

Ce modèle utilise l'intelligence artificielle pour deviner ce dont l'enfant pourrait avoir besoin.

Applications :

- Proposer des activités qui ont bien fonctionné avec d'autres enfants similaires.
- Ajuster automatiquement les contenus selon les comportements passés.

2.12 Conclusion

Notre étude a porté, au long de ce chapitre, sur les notions fondamentales liées aux systèmes de recommandation et à l'apprentissage adaptatif. Nous avons présenté l'essentiel des concepts relatifs aux SRs, notamment leur définition, les étapes de leur fonctionnement, les approches de recommandation, ainsi que l'émergence des SRI.

Nous avons ensuite introduit l'apprentissage adaptatif en tant que paradigme complémentaire aux SRI dans le domaine de l'éducation, en exposant ses fondements théoriques, ses modèles, ainsi que les critères contextuels permettant de personnaliser l'expérience d'apprentissage.

Enfin, nous avons présenté les principaux mécanismes de personnalisation intégrés dans notre propre système, basés sur le profil, l'historique, les tests, les comportements similaires des apprenants, ainsi que les modèles d'apprentissage adaptatif utilisés.

Chapitre 3

Modélisation et Architecture

3.1 introduction

Avant de se lancer dans le développement de l'application, il est indispensable d'établir une représentation abstraite du système à mettre en place. On appelle cela la modélisation, qui consiste à représenter un système réel par un modèle descriptif dans le but de le comprendre, l'analyser et anticiper ses besoins. Nous avons employé le langage UML pour réaliser cela. C'est un instrument graphique conçu pour représenter de manière structurée les éléments du système, ses composants, les intervenants impliqués ainsi que leurs interactions. Ce modèle permet de concevoir minutieusement la structure et le fonctionnement de l'application avant sa mise en œuvre. Pour cela, nous avons utilisé trois types de diagrammes UML : le diagramme de classes, le diagramme des cas d'utilisation et le diagramme de séquences.

Ce chapitre présente dans une dernière partie l'architecture globale de notre système. Ce dernier est basé sur un modèle client-serveur, où nous avons développé une application mobile comme client, interagissant avec le serveur qui héberge notre partie backend. Cette dernière partie implémente les fonctionnalités nécessaires pour accéder aux données, les stocker et générer des recommandations.

3.2 Méthodologie de conception

Dans ce qui suit nous allons présenter le langage UML :

3.2.1 Présentation d'UML

UML (Unified Modeling Language) est un langage de modélisation visuelle normalisé, employé pour définir, illustrer, élaborer, construire et consigner les composantes d'un système logiciel. Il propose un ensemble de symboles graphiques uniformes qui facilitent une représentation précise et organisée de la structure statique, du comportement dynamique, des interactions et de l'architecture générale d'un système informatique [51].

3.2.2 Vues UML

En UML, une vue (ou view en anglais) est une perspective particulière sur un système logiciel, mettant l'accent sur un certain aspect (fonctionnel, statique et dynamique). Chaque vue regroupe un ou plusieurs diagrammes UML qui illustrent les différents éléments d'un système sous cette perspective. L'idée est de séparer les différents aspects du système pour faciliter la compréhension, l'analyse, la conception et la communication entre les intervenants du projet [51]. Il est crucial de maîtriser les trois vues (perspectives) de modélisation : fonctionnelle, statique et dynamique

- **Vue fonctionnelle** : La perspective fonctionnelle utilise, entre autres, des diagrammes de cas d'utilisation pour décrire les fonctionnalités du système.
- **Vue statique** : la perspective statique utilise, entre autres, des diagrammes de classes pour représenter sa structure .
- **Vue dynamique** : la perspective dynamique se concentre sur le comportement du système et utilise, entre autres, des diagrammes de séquence pour décrire les interactions entre les différents composants.

En utilisant ces différentes vues, il est possible de représenter tous les aspects importants du système et de communiquer efficacement ces informations aux parties prenantes du projet [52].

3.2.2.1 Vue fonctionnelle : diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente les interactions entre les acteurs et le système, en illustrant les cas d'utilisation. Les acteurs, qu'ils soient des utilisateurs humains ou des systèmes externes, interagissent directement avec le système et jouent un rôle dans au moins un cas d'utilisation spécifique. Les cas d'utilisation décrivent les actions que le système doit exécuter pour répondre aux besoins des acteurs [53].

Les Figures 3.1 et 3.2 représentent le diagramme de cas d'utilisation de notre système, qui met en évidence les différentes fonctionnalités et interactions.

Diagramme de cas d'utilisation du parent :

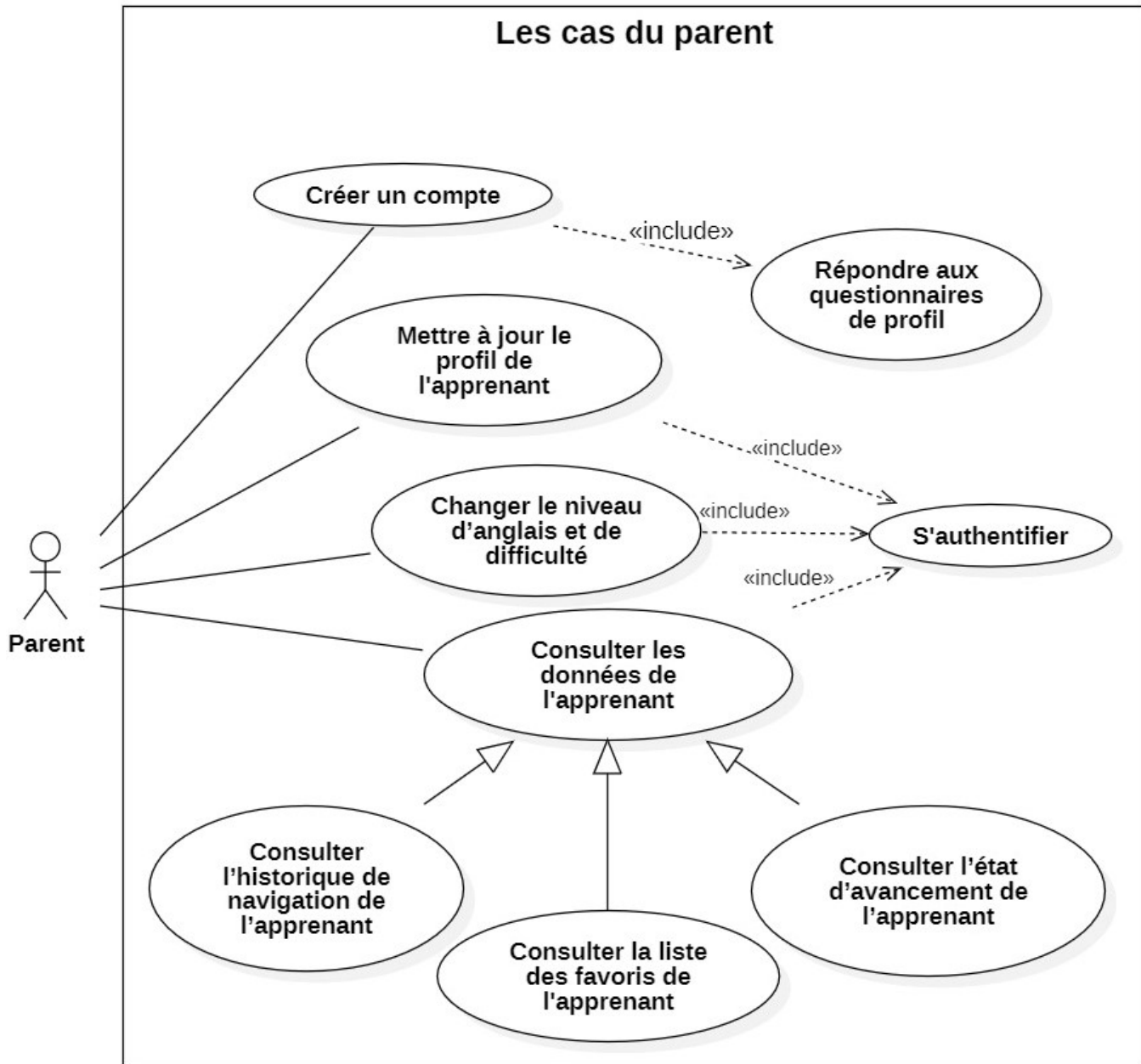


FIGURE 3.1 – Diagramme de cas d'utilisation du parent

Description textuelle des cas d'utilisation du parent :

Un cas d'utilisation (CU) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié [52].

- **Pré condition :** définissent les conditions qui doivent être satisfaites pour que le CU puisse démarrer.
- **Post condition :** définissent ce qui doit être vrai lorsque le CU se termine avec succès, qu'il s'agisse d'un scénario nominal ou alternatif.

a. Cas d'utilisation « Créer un compte » :

Le tableau illustre le cas d'utilisation de création d'un nouveau compte et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Créer un compte</p> <p>But : Permettre la création d'un nouveau compte utilisateur</p> <p>Acteur : Parent</p>
<p>Séquencement</p> <p>Précondition : Aucune.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur lance l'application. – L'utilisateur accède à l'espace de création de compte. – L'application affiche un formulaire de saisie d'informations. – L'utilisateur remplit le formulaire avec les informations demandées. – L'application vérifie la validité des données saisies. – Si les données sont valides, l'application redirige l'utilisateur vers une série de questions de profil (ex : child age group, english level, child stuggles, etc.). – L'utilisateur répond au questionnaire de profil. – L'application enregistre les réponses et crée le nouveau compte. – L'utilisateur est redirigé vers la page d'accueil. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Des données invalides sont saisies. – Un ou plusieurs champs obligatoires sont vides. – Le compte existe déjà dans la base de données. <p>Post-conditions :</p> <ul style="list-style-type: none"> – Le compte utilisateur est enregistré dans la base de données avec les informations de profil associées.

TABLE 3.1 – Cas d'utilisation « Créer un compte »

b. Cas d'utilisation « Mettre à jour le profil de l'apprenant » :

Le tableau illustre le cas d'utilisation de mettre à jour le profil de l'apprenant et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Mettre à jour le profil de l'apprenant</p> <p>But : Permettre au parent de mettre à jour les informations personnelles de l'apprenant</p> <p>Acteur : Parent</p>
<p>Séquencement</p> <p>Précondition : Authentification réussie. Le parent doit être sur l'interface profil.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur accède à l'espace de profil « Child Profile ». – L'utilisateur clique sur le bouton « Profile information » du profil de l'apprenant. – L'application demande la saisie du mot de passe. – L'utilisateur saisit le mot de passe. – Si le mot de passe est correct, l'utilisateur modifie les informations personnelles de l'apprenant. – L'application enregistre les modifications et affiche la page d'accueil. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Le mot de passe saisi est incorrect : l'accès à la modification du profil est refusé. <p>Post-conditions :</p> <ul style="list-style-type: none"> – Les informations personnelles de l'apprenant sont mises à jour dans la base de données.

TABLE 3.2 – Cas d'utilisation « Mettre à jour le profil de l'apprenant »

c. Cas d'utilisation « Changer le niveau d'anglais et de difficulté » :

Le tableau illustre le cas d'utilisation de changer le niveau d'anglais et de difficulté et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Changer le niveau d'anglais et de difficulté</p> <p>But : Adapter le niveau d'anglais et de difficulté pour offrir un apprentissage personnalisé et progressif selon les capacités de l'enfant</p> <p>Acteur : Parent</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à l'espace de profil « Child Profile ». – L'utilisateur clique sur le bouton « Level of English and Difficulty » pour changer le niveau d'anglais et de difficulté. – L'application demande la saisie du mot de passe. – L'utilisateur saisit le mot de passe. – Si le mot de passe est correct, l'utilisateur choisit le niveau d'anglais et la difficulté. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Le mot de passe saisi est incorrect : l'accès à la modification est refusé. <p>Post-conditions :</p> <ul style="list-style-type: none"> – Les informations de niveau et de difficulté de l'apprenant sont mises à jour dans la base de données.

TABLE 3.3 – Cas d'utilisation « Changer le niveau de difficulté »

d. Cas d'utilisation « Consulter l'historique de navigation de l'apprenant » :

Le tableau illustre le cas d'utilisation de consulter son historique de navigation et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Consulter l'historique de navigation de l'apprenant</p> <p>But : Suivre les ressources consultées afin d'évaluer les habitudes et le parcours d'apprentissage de l'apprenant</p> <p>Acteur : Parent</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à l'interface d'accueil. – L'utilisateur clique sur le bouton « History » dans la page d'accueil. – L'application affiche l'historique de navigation de l'apprenant. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Aucune. <p>Post-conditions :</p> <ul style="list-style-type: none"> – L'historique de navigation de l'apprenant est affiché au parent.

TABLE 3.4 – Cas d'utilisation « Consulter son historique de navigation »

e. Cas d'utilisation « Consulter l'état d'avancement de l'apprenant » :

Le tableau illustre le cas d'utilisation de consulter son état d'avancement et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Consulter l'état d'avancement de l'apprenant</p> <p>But : Suivre la progression de l'apprenant afin d'évaluer son évolution et son engagement dans l'apprentissage</p> <p>Acteur : Parent</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à l'interface de profil « Child Profile ». – L'utilisateur clique sur le bouton « Progress Report » dans la page de profil – L'application affiche l'état d'avancement de l'apprenant. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Aucune. <p>Post-conditions :</p> <ul style="list-style-type: none"> – L'état d'avancement de l'apprenant est affiché au parent.

TABLE 3.5 – Cas d'utilisation « Consulter son état d'avancement »

Diagramme de cas d'utilisation de l'apprenant :

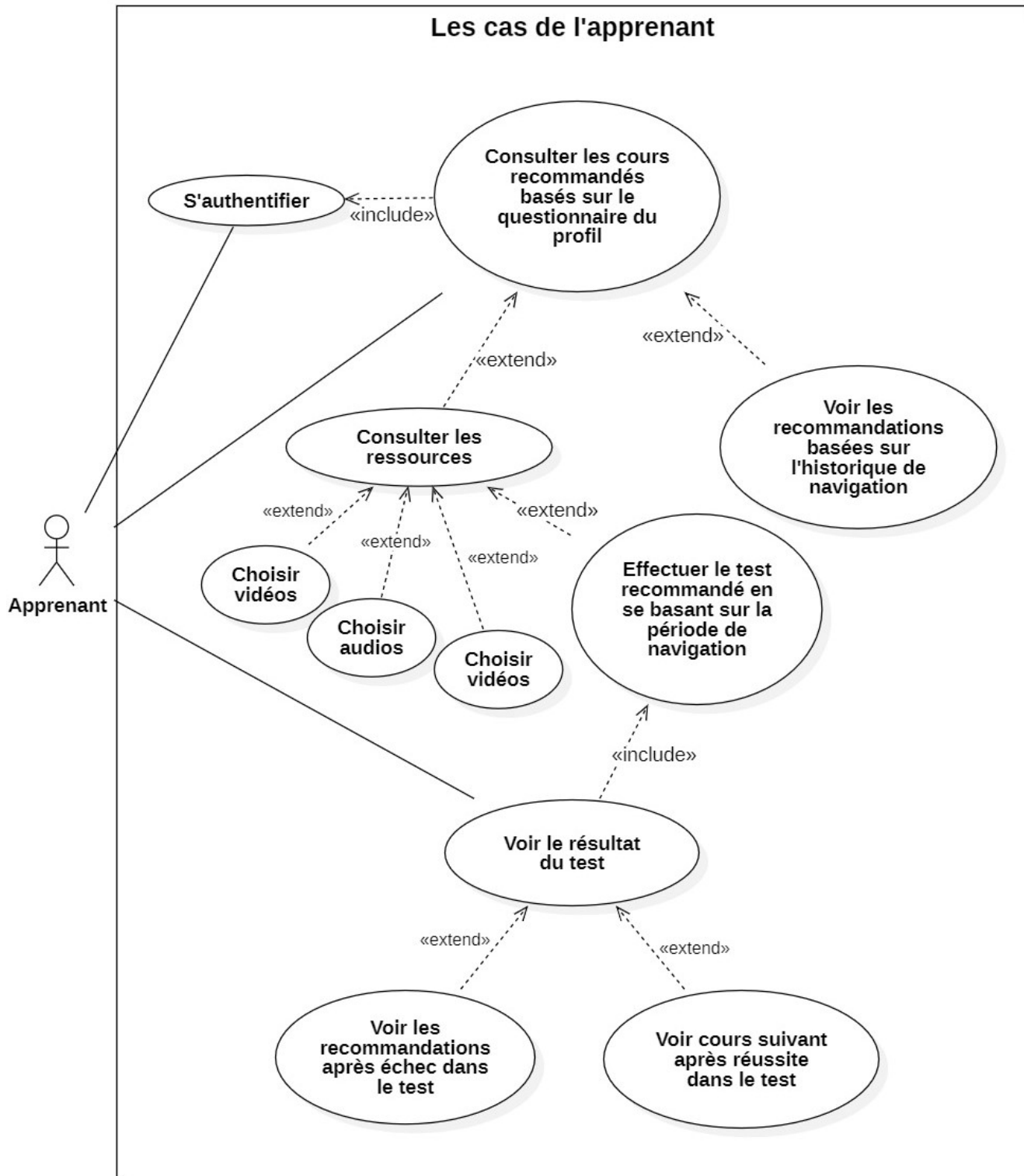


FIGURE 3.2 – Diagramme de cas d'utilisation de l'apprenant

Description textuelle des cas d'utilisation de l'apprenant :**a. Cas d'utilisation « Consulter les cours recommandés basés sur le questionnaire du profil » :**

Le tableau illustre le cas d'utilisation de consulter les cours recommandés basés sur le questionnaire du profil et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Consulter les cours recommandés basés sur le questionnaire du profil</p> <p>But : Recevoir des cours adaptés à son profil afin de mieux apprendre selon ses besoins</p> <p>Acteur : Apprenant</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à la page d'accueil. – L'application affiche la liste des cours recommandés selon le profil. – L'utilisateur clique sur un cours pour le consulter. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Aucune. <p>Post-conditions :</p> <ul style="list-style-type: none"> – Le cours sélectionné est affiché à l'utilisateur.

TABLE 3.6 – Cas d'utilisation « Consulter les cours recommandés basés sur le questionnaire du profil »

b. Cas d'utilisation « Consulter les ressources » :

Le tableau illustre le cas d'utilisation de consulter les ressources et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Consulter les ressources</p> <p>But : Permettre à l'apprenant d'accéder à des contenus pédagogiques variés afin de favoriser son apprentissage selon ses préférences</p> <p>Acteur : Apprenant</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à la page d'accueil. – L'application affiche la liste des cours disponibles. – L'utilisateur sélectionne un cours. – L'utilisateur clique sur le bouton « Explore Resources » pour consulter les ressources associées (vidéo, audio, texte). <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Aucune. <p>Post-conditions :</p> <ul style="list-style-type: none"> – L'historique de navigation est enregistré dans la base de données.

TABLE 3.7 – Cas d'utilisation « Consulter les ressources »

c. Cas d'utilisation « Voir les recommandations basées sur l'historique de navigation » :

Le tableau illustre le cas d'utilisation de voir les recommandations basées sur l'historique de navigation et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Voir les recommandations basées sur l'historique de navigation</p> <p>But : Offrir à l'apprenant des recommandations personnalisées en fonction de son historique de navigation afin d'optimiser son apprentissage</p> <p>Acteur : Apprenant</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à la page d'accueil. – L'application affiche la liste des cours disponibles. – L'utilisateur sélectionne un cours. – L'utilisateur clique sur le bouton « Recommended for You ». – L'utilisateur voit les recommandations proposées. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Aucune. <p>Post-conditions :</p> <ul style="list-style-type: none"> – Aucune.

TABLE 3.8 – Cas d'utilisation « Voir les recommandations basées sur l'historique de navigation »

d. Cas d'utilisation « Effectuer le test recommandé en se basant sur la période de navigation » :

Le tableau illustre le cas d'utilisation d'effectuer le test recommandé en se basant sur la période de navigation et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Effectuer le test recommandé en se basant sur la période de navigation</p> <p>But : Permettre à l'apprenant de réaliser un test adapté à sa période de concentration (matin, après-midi, soir) afin de maximiser sa concentration, d'évaluer ses acquis et d'améliorer l'efficacité de son apprentissage</p> <p>Acteur : Apprenant</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à la page d'accueil. – L'application affiche la liste des cours. – L'utilisateur sélectionne un cours. – L'utilisateur clique sur le bouton « Test Time ! » pour effectuer le test recommandé après la consultation des ressources. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Aucune. <p>Post-conditions :</p> <ul style="list-style-type: none"> – L'application vérifie les réponses soumises par l'apprenant.

TABLE 3.9 – Cas d'utilisation « Effectuer le test recommandé en se basant sur la période de navigation »

e. Cas d'utilisation « Voir le résultat du test » :

Le tableau illustre le cas d'utilisation, voir le résultat du test et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Voir le résultat du test</p> <p>But : Aider l'apprenant à comprendre ses résultats pour reconnaître ses réussites, détecter ses difficultés et suivre son évolution</p> <p>Acteur : Apprenant</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à l'interface d'accueil. – L'application affiche la liste des cours. – L'utilisateur sélectionne un cours – L'utilisateur clique sur le bouton « Test Time ! » pour effectuer le test après la consultation des ressources. – L'utilisateur passe le test recommandé. – Une fois le test terminé, l'utilisateur voit son résultat. – L'utilisateur passe le test. – L'utilisateur voit son résultat. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Aucune. <p>Post-conditions :</p> <ul style="list-style-type: none"> – Aucune.

TABLE 3.10 – Cas d'utilisation « Voir le résultat du test »

f. Cas d'utilisation « Voir cours suivant après réussite dans le test » :

Le tableau illustre le cas d'utilisation, voir cours suivant après réussite dans le test et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Voir cours suivant après réussite dans le test</p> <p>But : Permettre à l'apprenant d'accéder au cours suivant après avoir réussi le test, afin de continuer son apprentissage étape par étape</p> <p>Acteur : Apprenant</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à la page d'accueil. – L'application affiche la liste des cours. – L'utilisateur électionne un cours. – L'utilisateur clique sur le bouton « Test Time ! » après consultation des ressources. – L'utilisateur passe le test recommandé, voit son résultat. – Après la réussite, l'utilisateur retourne à la liste des cours. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Aucune. <p>Post-conditions :</p> <ul style="list-style-type: none"> – Envoi du cours suivant.

TABLE 3.11 – Cas d'utilisation « Voir cours suivant après réussite dans le test »

g. Cas d'utilisation « Voir les recommandations après échec dans le test » :

Le tableau illustre le cas d'utilisation, voir les recommandations après échec dans le test et présente les différents acteurs qui ont accès à ce service.

<p>Identification</p> <p>Nom du cas d'utilisation : Voir les recommandations après échec dans le test</p> <p>But : Aider l'apprenant à progresser en lui proposant des ressources adaptées après un échec au test</p> <p>Acteur : Apprenant</p>
<p>Séquencement</p> <p>Précondition : Authentification.</p> <p>Enchaînements nominaux :</p> <ul style="list-style-type: none"> – L'utilisateur s'authentifie. – L'utilisateur accède à la page d'accueil. – L'application affiche la liste des cours. – L'utilisateur choisit un cours. – L'utilisateur clique sur le bouton « Test Time ! » après consultation des ressources. – L'utilisateur passe le test recommandé, voit son résultat. – En cas d'échec, l'utilisateur reçoit des recommandations basées sur les ressources avec lesquelles il interagit fréquemment. – En cas de second échec, des ressources ayant été efficaces pour d'autres utilisateurs lui sont proposées. <p>Enchaînements alternatifs :</p> <ul style="list-style-type: none"> – Aucune. <p>Post-conditions :</p> <ul style="list-style-type: none"> – Aucune.

TABLE 3.12 – Cas d'utilisation « Voir les recommandations après échec dans le test »

3.2.2.2 Vue statique : diagramme de classes

Le diagramme de classes est essentiel en modélisation orientée objet. Il représente de façon statique les éléments du système et leurs relations, afin de mettre en œuvre les scénarios d'utilisation. Le diagramme de classes présenté ci-dessous Figures 3.3 modélise les données nécessaires au bon déroulement des cas d'utilisation évoqués précédemment. Il représente, dans ce contexte, un modèle conceptuel des données, décrivant les éléments essentiels à la mise en œuvre du système.

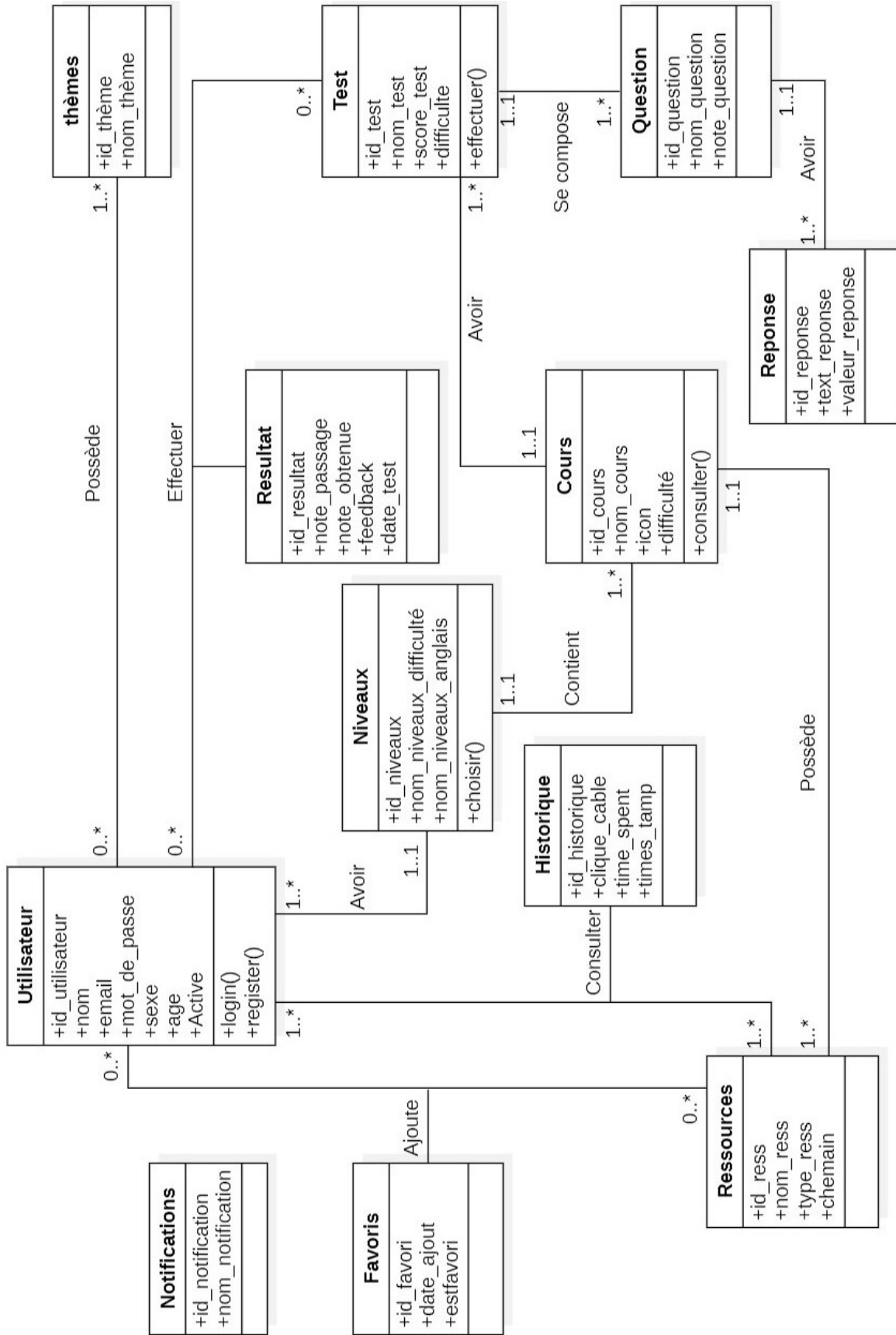


FIGURE 3.3 – Notre diagramme de classes

3.2.3 Vue dynamique : diagrammes de séquence

Le diagramme de séquence montre l'enchaînement des messages entre les objets du système dans le temps, afin de modéliser son comportement dynamique dans un scénario donné. Les figures 3.4, 3.5 et 3.6 illustrent respectivement les diagrammes de séquence systèmes des cas d'utilisation essentiels : Création du compte et recommandations des cours basées sur le questionnaire du profil, S'authentifier et recommandation d'activité pédagogique.

Diagramme de séquence de Création du compte et recommandations des cours basées sur le questionnaire du profil :

Lorsque l'utilisateur lance l'application, il demande à créer un compte. L'application lui affiche un formulaire d'inscription qu'il doit remplir. Les données saisies sont vérifiées. Si elles sont incorrectes (par exemple une adresse email invalide), l'utilisateur doit les ressaisir jusqu'à ce qu'elles soient valides. Ensuite, l'application envoie les données à la base de données pour vérifier si le nom d'utilisateur existe déjà.

Si le nom est déjà pris, un message d'erreur s'affiche. Sinon, un questionnaire de profil est envoyé à l'utilisateur. Une fois les réponses soumises, elles sont envoyées à la base de données. L'application utilise alors ces réponses pour recommander des cours adaptés au profil de l'utilisateur.

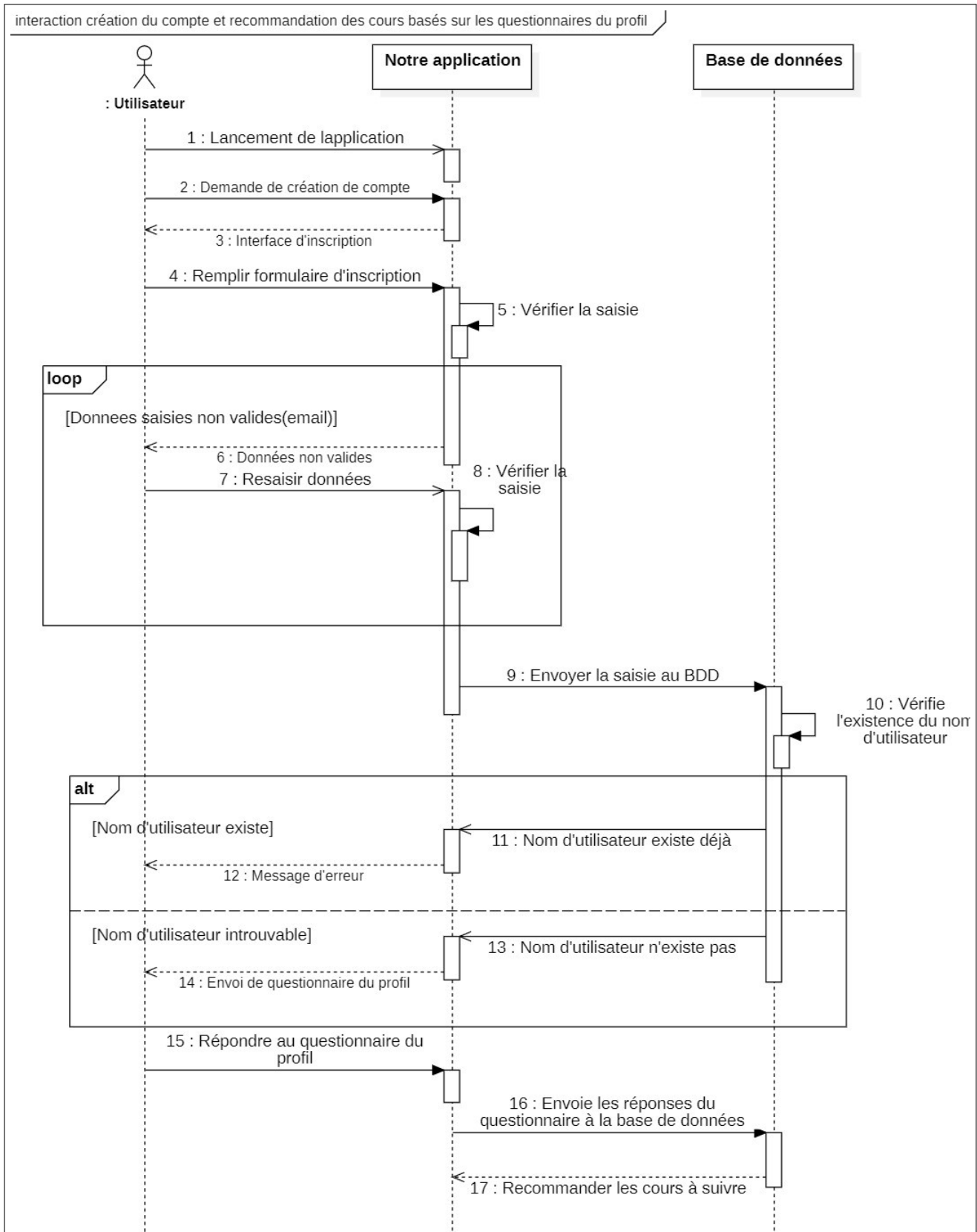


FIGURE 3.4 – Diagramme de séquence « Création du compte et recommandations des cours basées sur le questionnaire du profil »

Diagramme de séquence de « Authentification » :

L'authentification consiste à assurer la confidentialité des données, elle se base sur la vérification du nom d'utilisateur et du mot de passe. Ces informations sont préétablies dans une base de données. Lors de l'authentification de l'utilisateur, deux cas peuvent se présenter : informations correctes ou incorrectes, ce qui explique l'utilisation de l'opérateur « alt ». Si les informations fournies sont correctes, alors le système accorde l'accès à l'interface d'accueil. En revanche, si l'utilisateur saisit des informations incorrectes, le système génère un message d'erreur et réaffiche la page d'authentification d'où l'utilisation de l'opérateur « loop ».

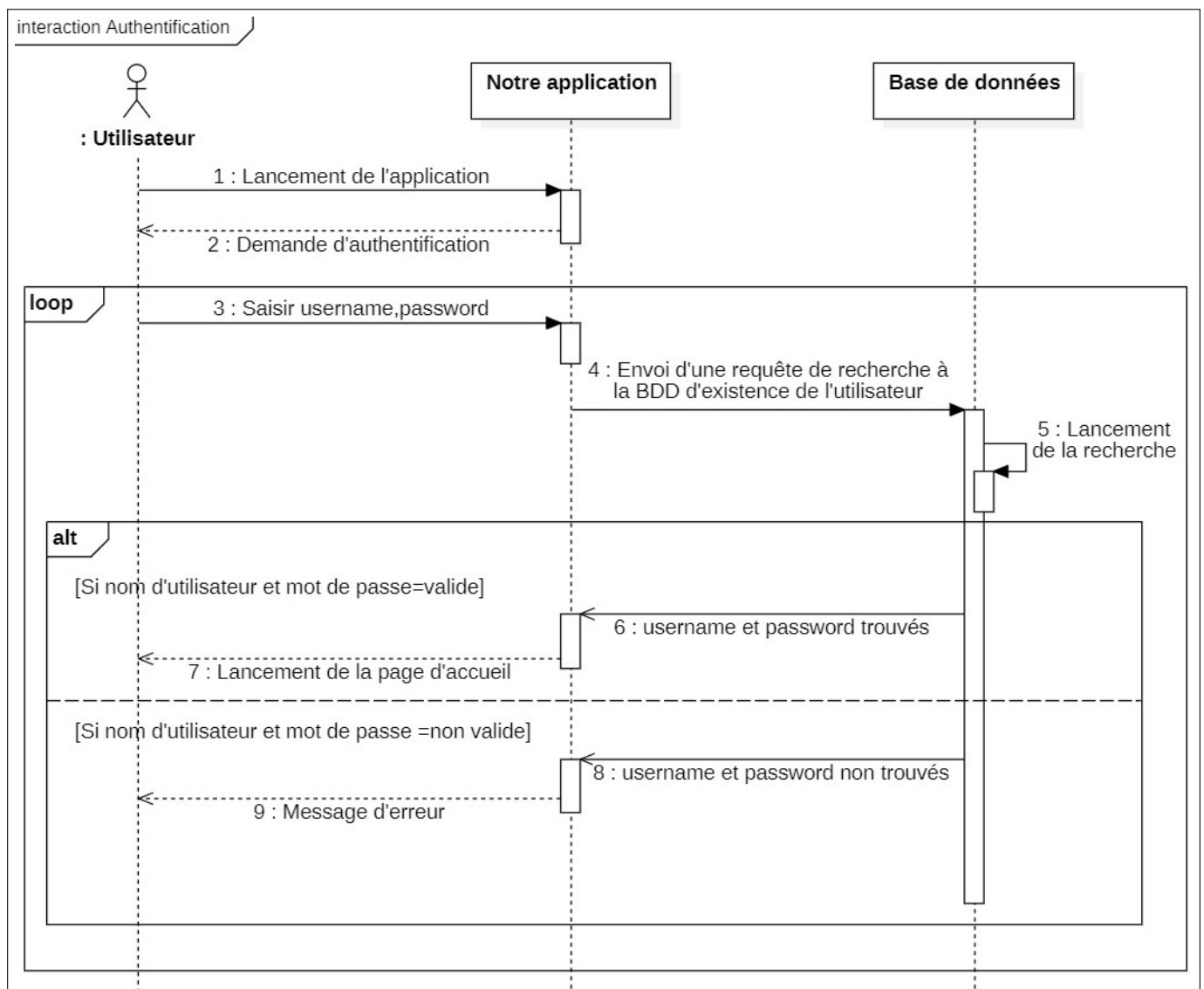


FIGURE 3.5 – Diagramme de séquence « Authentification »

Diagramme de séquence de Recommandation d'activité pédagogique :

Le schéma de séquence nommé « Recommandation d'activité pédagogique » dépeint le mécanisme par lequel l'application suggère des contenus personnalisés à l'apprenant, en fonction de son parcours et de ses performances. L'utilisateur débute en se rendant sur l'interface d'accueil, examine les cours suggérés, choisit un cours spécifique puis navigue parmi les ressources mises à disposition. Par la suite, il peut solliciter pour passer un test recommandé en rapport avec ce cours. L'application conserve son historique de navigation, et suggère un test personnalisé en fonction du moment de la journée. Après avoir effectué le test, l'application procède à l'analyse de la réponse. Si la réponse est juste, elle montre le prochain cours. Si l'application échoue, elle propose des ressources sur mesure : suite à un premier échec, les recommandations s'appuient sur l'historique de navigation, et après un second échec, elles s'appuient sur les réussites d'autres utilisateurs.

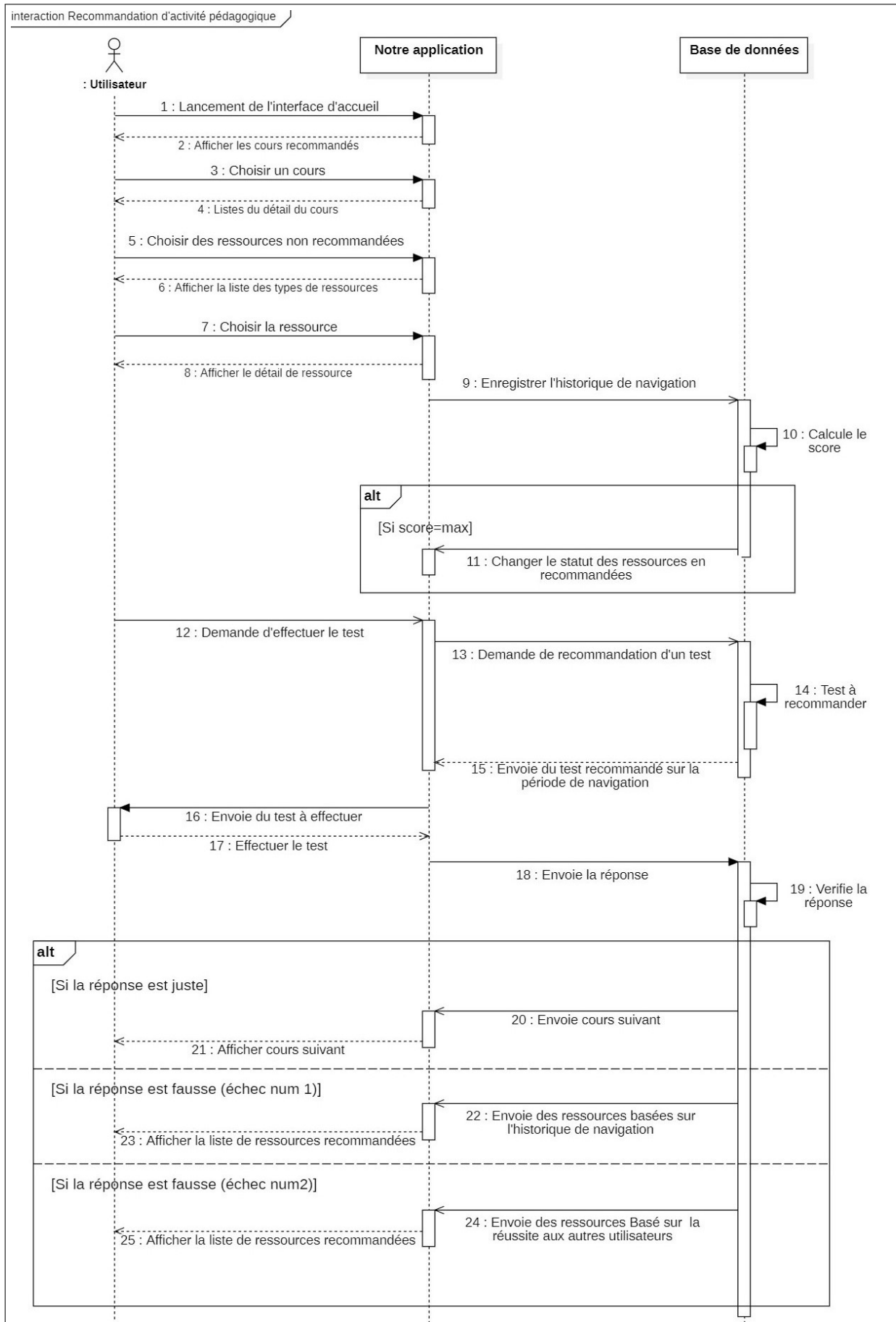


FIGURE 3.6 – Diagramme de séquence « Recommandation d'activité pédagogique »

3.3 Architecture du système proposé

Dans cette section, nous allons présenter l'architecture globale de notre système de recommandation d'activités pédagogiques. La figure 3.7 illustre cette architecture. Deux parties principales sont à distinguer :

- **La partie backend (serveur, logique métier) :** cette partie implémente toutes les fonctionnalités relatives à l'accès, au stockage et au processus de recommandation. Cet ensemble est exposé sous la forme de services WEB de type REST.
- **La partie frontend (coté client) :** cette partie est implémentée sous la forme d'une application mobile éducative destinée aux enfants. Elle propose diverses activités d'apprentissage personnalisées (cours, ressources, tests) et s'appuie principalement sur les services proposés par le backend pour récupérer et afficher les contenus recommandés.

Notre choix des services Web REST peut être attribué principalement aux nombreux avantages qu'ils offrent. En particulier, leur simplicité d'utilisation, la capacité à les invoquer indépendamment du langage de programmation utilisé par le client, ainsi que la facilité de mise à jour de la partie backend sans impacter la partie frontend.

Les détails spécifiques à chaque partie seront abordés dans les sections suivantes.

3.3.1 Partie frontend

3.3.1.1 Interface Utilisateur

L'application propose une interface unique et intuitive, adaptée aux enfants et aux parents, et propose des fonctionnalités qui leur sont propres.

Du côté de l'enfant, L'interface permet une navigation aisée entre différents types de ressources éducatives, telles que : des vidéos, des audios et des textes, regroupés par leur nature. L'enfant peut consulter les cours de formation en fonction de son niveau et de passer des tests selon les moments de la journée, progressant ainsi dans l'apprentissage de l'anglais de manière interactive et ludique, ouvrir une liste personnalisée de recommandations en fonction de ses préférences et de son engagement. En outre, l'enfant est stimulé par un design coloré et stimulant.

Du côté des parents, l'interface leur offre la possibilité de suivre les progrès de leur enfant. En outre, l'interaction personnelle peut être lire l'historique des ressources qu'ils regardaient, voir ce que l'enfant aime, et avoir une idée de ses préférences. De plus, le parent a la possibilité de modifier les réglages ou d'adapter l'expérience d'apprentissage de l'enfant selon ses besoins et ses progrès.

3.3.1.2 Appel au Serveur Backend

Notre application mobile éducative fait appel à un serveur backend pour obtenir des recommandations de ressources pédagogiques et réaliser diverses opérations telles que l'inscription, la connexion, la consultation des cours, la mise à jour du profil, ou encore le suivi de l'historique d'apprentissage des utilisateurs, etc. Par exemple, les utilisateurs sont identifiés en tant que ressources par un schéma d'URI du type (en localhost) : **http :/127.0.0.1 :8081/monapi/utilisateur/getAllUtilisateurs** où l'ensemble des utilisateurs est référencé.

Ce qui est intéressant dans notre architecture, c'est que les utilisateurs finaux (enfants et parents) n'ont pas d'accès direct à la base de données ou aux ressources serveur. Toutes leurs actions (GET, POST, PUT, DELETE) sont effectuées via l'interface mobile (frontend Flutter), qui communique avec le backend (Eclipse EE) à travers des requêtes HTTP. Cela garantit une meilleure sécurité, une séparation claire des responsabilités et une facilité de maintenance du système.

3.3.2 Partie backend

La partie backend est chargée de plusieurs tâches essentielles, notamment le stockage des données des utilisateurs, la mise en œuvre de la logique métier pour la recommandation de ressources pédagogiques, etc. Les différentes fonctionnalités développées côté serveur sont exposées sous forme de services web REST.

3.3.2.1 Stockage des Données Utilisateurs

le backend persiste les données liées aux utilisateurs, telles que leurs profils, leur niveau d'apprentissage, leurs interactions avec les ressources (clics, temps passé), les résultats des tests effectués ainsi que leur historique de consultation. Il prend en charge les opérations de création, de récupération, de mise à jour des données. Ces informations sont stockées de manière

structurée et sécurisée, permettant au système de fournir des recommandations personnalisées et de suivre la progression des utilisateurs de façon efficace.

3.3.2.2 Logique Métier de Recommandation des Ressources pédagogiques

La partie serveur met en œuvre la logique métier qui régit le processus de recommandation des ressources pédagogiques. Elle analyse les données des enfants, en particulier leur niveau d'anglais, leurs préférences, les types de contenus les plus consultés, ainsi que leur temps d'engagement. Ces informations permettent de proposer des recommandations personnalisées adaptées au profil et au comportement d'apprentissage de chaque utilisateur.

La logique métier s'appuie sur plusieurs mécanismes de filtrage, notamment le filtrage basé sur le contenu, le filtrage collaboratif, le filtrage par type de ressource, le filtrage par engagement temporel, ainsi que des ajustements selon les moments de concentration de la journée.

Elle peut également intégrer des algorithmes de recommandation avancés et des modèles statistiques pour améliorer la pertinence et l'efficacité des recommandations, assurant ainsi un apprentissage progressif, motivant et adapté aux besoins spécifiques de chaque enfant.

3.3.2.3 Exposition des fonctionnalités en tant que Services Web

Le serveur expose ses fonctionnalités sous forme de services Web REST afin de permettre à l'application mobile d'interagir avec lui de manière fluide et sécurisée. Ces services offrent des points de terminaison (endpoints) que l'application peut interroger pour effectuer diverses opérations telles que : la création et la connexion des comptes utilisateurs, la récupération des cours selon le questionnaire de profil de l'enfant, la soumission des interactions avec les ressources, ou encore l'obtention de recommandations personnalisées.

Les échanges entre le client (l'application) et le serveur se font au format JSON, garantissant une compatibilité optimale avec les différentes plateformes. Cette architecture client-serveur permet de centraliser la logique métier tout en assurant une expérience utilisateur interactive, personnalisée et enrichissante, tant pour l'enfant que pour le parent, avec une intégration possible de services externes pour améliorer la qualité des contenus recommandés.

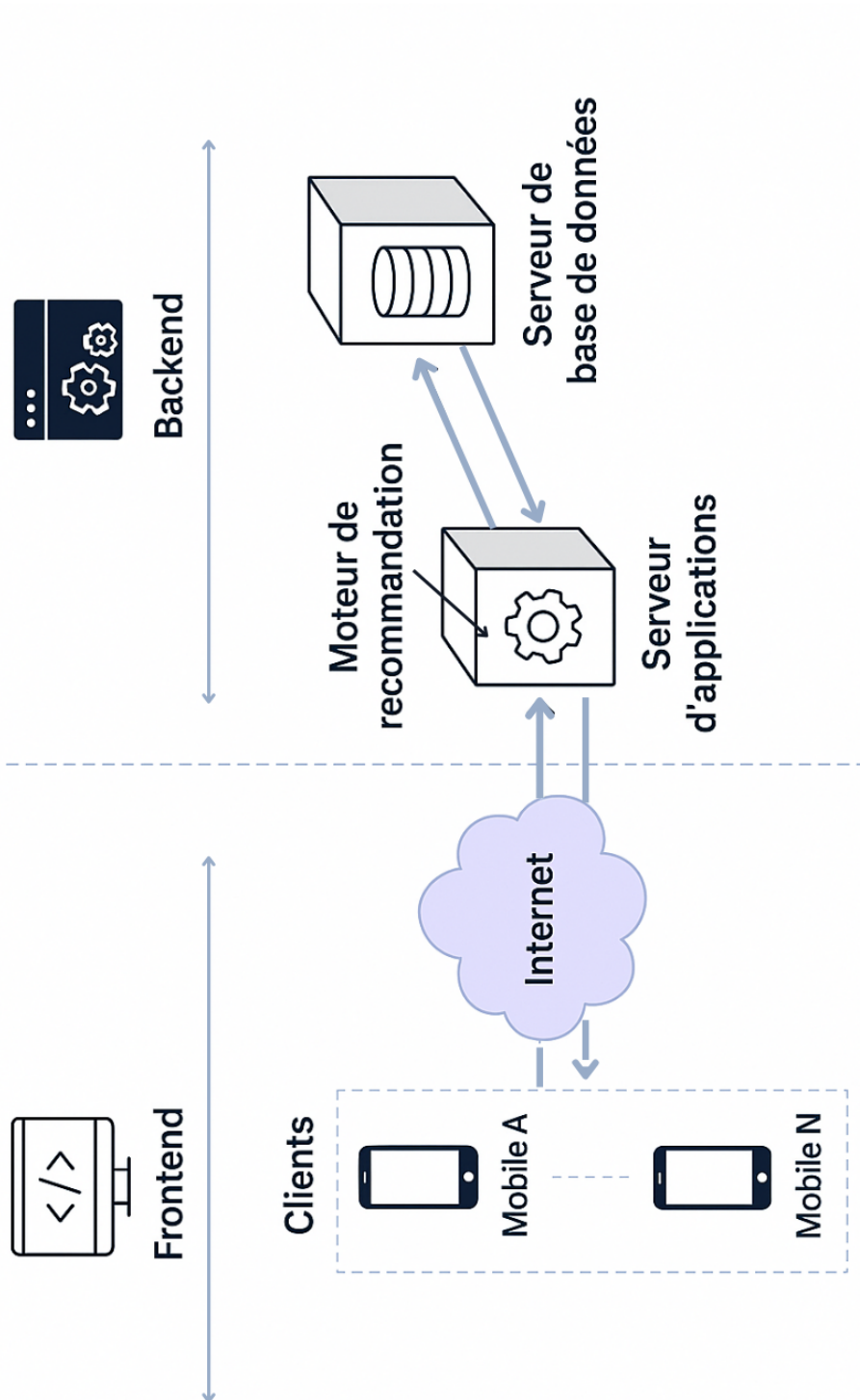


FIGURE 3.7 – Architecteur de notre système

3.4 Conclusion

La phase de conception de notre application est essentielle pour assurer son succès. En utilisant la modélisation UML et en adoptant une architecture client-serveur, nous posons les bases solides pour la mise en œuvre et le développement futurs de notre SR d'activités pédagogiques. Nous avons d'abord défini les cas d'utilisation en surlignant les divers acteurs et leurs relation avec le système. Ensuite, nous avons montré la structure statique de l'application à l'aide du diagramme de classes. Enfin, nous avons exploré les scénarios fonctionnels à travers les diagrammes de séquence, ainsi que l'architecture globale du système. Cette partie fournit une base solide pour la prochaine phase du projet qui est la phase de développement.

Chapitre 4

Implémentation et Réalisation

4.1 Introduction

Dans ce chapitre, nous aborderons plusieurs aspects essentiels liés à l'implémentation pratique de notre application mobile de recommandation d'activités pédagogiques. Nous commencerons par présenter les services web de type REST et soulignerons leur rôle central dans notre application. Nous expliquerons ensuite l'environnement de développement utilisé, en mettant l'accent sur les deux côtés backend et frontend, ainsi que sur les outils et technologies associés. Dans un second temps, nous détaillerons le schéma relationnel de notre base de données, qui forme la base solide de notre application. Nous décrirons ensuite et de manière détaillée notre API proposée, en mettant en évidence les différentes méthodes proposées ainsi que leurs URL d'accès. En dernier lieu, nous présenterons notre application mobile réalisée. Nous décrirons le diagramme de flux utilisateur et mettrons en évidence certaines interfaces graphiques clés.

Ce chapitre offre ainsi une vue d'ensemble complète de notre projet, mettant en évidence les choix technologiques et les fonctionnalités implémentées à chaque étape.

4.2 Les services web de type REST

Dans le monde numérique d'aujourd'hui, les services web (SW) jouent un rôle essentiel dans la communication et l'échange de données entre les applications. Ils permettent à des systèmes hétérogènes de partager et d'accéder à des ressources de manière standardisée.

Les SW reposent sur des protocoles connus de tous tels que HTTP¹, garantissant une communication fiable et sécurisée entre les systèmes et applications [54].

Parmi les différentes méthodes de mise en œuvre, l'architecture REST se distingue par sa simplicité, sa flexibilité et sa généralisation [54]. REST est un style architectural logiciel largement utilisé pour construire des applications web qui articule un ensemble de principes et de contraintes pour la conception des APIs [54]².

Les principes clés de REST, tels que définis par Fielding [54], incluent les éléments suivants :

- **Architecture orientée ressources** : chaque ressource est identifiée de manière unique par une URI. Cela permet de structurer les données de manière logique et hiérarchique, offrant une meilleure organisation et une navigation simplifiée pour les clients.
- **Stateless** : chaque requête vers le SW doit être autonome, incluant toutes les informations nécessaires. L'approche stateless favorise la scalabilité et simplifie la gestion des sessions et transactions, car chaque requête est indépendante.
- **Utilisation des opérations standards du protocole HTTP** : les opérations standards du protocole HTTP, telles que GET, POST, PUT et DELETE, sont utilisées pour manipuler les ressources de manière cohérente.

Les ressources peuvent être représentées dans différents formats, tels que **JSON** ou **XML**, selon les besoins du client.

En conclusion, les services web REST offrent une approche simple, flexible et largement adoptée pour la conception d'API. En respectant les principes de l'architecture REST, les développeurs peuvent créer des SW évolutifs, et adaptés à une variété de domaines d'application.

1. HTTP est un protocole qui garantit une transmission efficace des données et la gestion des requêtes et des réponses.

2. Les APIs sont des ensembles de règles et de protocoles qui facilitent la communication et l'interaction entre différents logiciels et applications.

4.3 Environnement de développement

Cette section présente les différentes technologies que nous avons utilisées pour mettre en pratique notre solution de recommandation proposée dans les deux côtés : backend et frontend

4.3.1 Côté backend

4.3.1.1 Jakarta EE

Jakarta EE (JAVA EE) est une plateforme de développement d'applications d'entreprise basée sur Java. Elle fournit un ensemble de spécifications et de technologies pour la création d'applications robustes, évolutives et sécurisées. Jakarta EE offre une architecture modulaire et permet l'intégration de nombreux frameworks et outils pour répondre aux besoins spécifiques des applications d'entreprise [55].

4.3.1.2 TomEE

TomEE est un serveur d'applications open source, spécialement conçu pour exécuter de manière fiable et efficace des applications web basées sur Jakarta EE. Il agit comme un conteneur de servlets léger et performant, prenant en charge les spécifications Java EE telles que les Servlets, les JSP, les JSF et les technologies de persistance comme JPA. Sa flexibilité et sa configuration simple en font un choix très prisé par les développeurs Java lorsqu'ils déploient des applications web [56].

4.3.1.3 Eclipse Enterprise Java and Web Developers

L'IDE Eclipse est un environnement de développement intégré populaire pour le développement Java. En particulier, l'édition Eclipse Enterprise Java and Web Developers est tout spécialement dédiée aux développeurs d'applications d'entreprise et de projets web. Elle offre des fonctionnalités avancées pour le développement, le débogage et le déploiement de telles applications Java [57].

4.3.1.4 L'API Jakarta RESTful Web Services

L'API Jakarta RESTful Web Services JAX-RS est une spécification standard pour la création de services web RESTful en utilisant Java. Elle fournit un ensemble de fonctionnalités

pour la création, la publication et la consommation de services web RESTful, en utilisant les principes de l'architecture REST [58].

4.3.1.5 SGBDR MySQL

MySQL est un logiciel SGBDR (Système de Gestion de Bases de Données Relationnelles) open source et très populaire, qui permet de stocker, gérer et manipuler des données pour de nombreuses applications Web ou Systèmes d'Information. MySQL offre des fonctionnalités avancées, une grande performance et une stabilité éprouvée.

Pour utiliser ce SGBDR, nous avons utilisé le logiciel WampServer. Ce dernier (Windows, Apache, MySQL, PHP Server), est un logiciel tout-en-un qui facilite le développement d'applications web sur un environnement Windows. Il intègre, en plus du SGBDR MySQL, particulièrement, un serveur web Apache et un interpréteur du langage de script PHP [59].

4.3.2 Côté front-end

4.3.2.1 Android

Android est un système d'exploitation mobile développé par Google, conçu principalement pour les appareils mobiles. Il permet également le développement d'applications multiplateformes grâce à Flutter, un framework moderne utilisant le langage Dart, permettant la création d'applications mobiles avec des fonctionnalités avancées, des interfaces utilisateur interactives, la gestion de données et la communication avec d'autres appareils [60].

4.3.3 Android Studio

Android Studio est l'IDE officiel pour le développement d'applications Android. Il est spécifiquement conçu pour offrir des fonctionnalités avancées et faciliter le processus de développement. Basé sur IntelliJ IDEA, Android Studio comprend un éditeur de code, un débogueur, un émulateur Android et d'autres outils essentiels. Il offre une intégration étroite avec les SDKs et les outils Android, permettant aux développeurs de créer plus facilement des applications mobiles puissantes [61].

4.3.4 La bibliothèque HTTP

La bibliothèque http est un package officiel de Flutter qui permet de simplifier l'appel d'API REST. Elle fournit des méthodes prêtes à l'emploi pour envoyer des requêtes comme GET, POST, PUT et DELETE pour envoyer des requêtes HTTP et recevoir des réponses depuis un serveur. La bibliothèque permet l'encodage et le décodage des données JSON, ce qui permet de transformer facilement les réponses en objets exploitables. Grâce à sa gestion des appels asynchrones avec Flutter, elle s'intègre naturellement dans le fonctionnement de Flutter. Facile à utiliser et bien adaptée à l'écosystème Dart, la bibliothèque http est largement utilisée pour connecter les applications Flutter à des services web [62].

4.4 Schéma relationnel de notre base de données

Le modèle relationnel, représenté par la figure 4.1, contient la liste des tables qui constituent la base de données de notre système de recommandation d'activités pédagogiques.

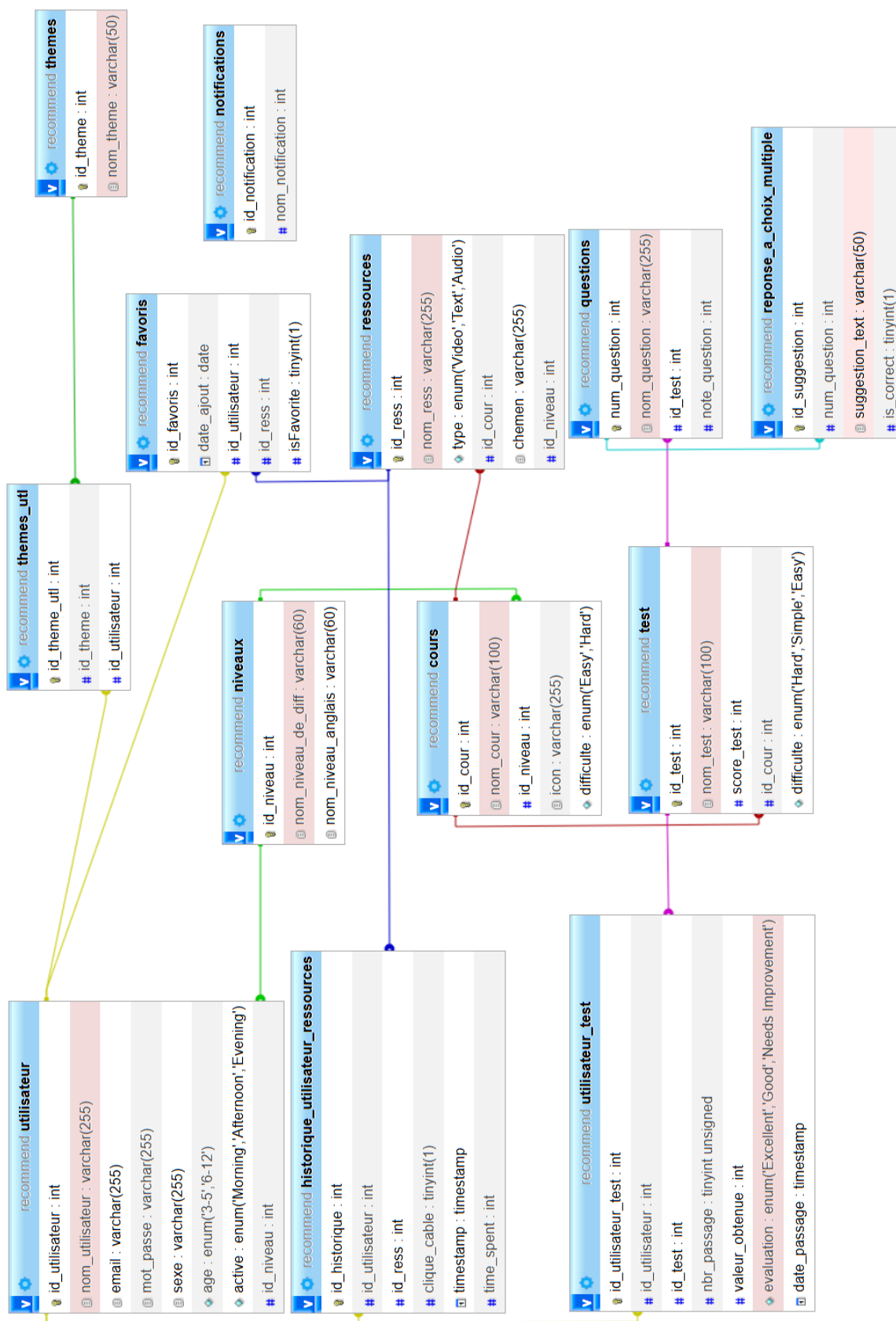


FIGURE 4.1 – Schéma relationnel de notre base de données

Voici une brève description des tables de votre base de données relationnelle, adaptée pour

des activités pédagogiques :

- **Utilisateur** : stocke les cours proposés, avec leur identifiant, nom, niveau associé, icône visuelle et un niveau de difficulté (Facile, Difficile).
- **Niveaux** : représente les différents niveaux d'anglais de l'apprentissage (débutant, intermédiaire et avancé) en leur associant des domaines de difficulté spécifiques et réciproquement (ex : alphabet, compréhension orale, grammaire, ...).
- **Cours** : stocke l'ensemble des cours proposés, avec une association de l'identifiant du cours, son nom, son niveau associé, une icône visuelle, et un niveau de difficulté (facile ou difficile).
- **Ressources** : stocke les ressources pédagogiques (activités) proposées, en associant l'identifiant, le nom de la ressource, son type (vidéo, texte, audio), le cours lié, le chemin d'accès au fichier, et le niveau de la ressource associée.
- **Historique_Utilisateur_Ressources** : table de liaison que stocke l'historique de navigation de chaque utilisateur avec les ressources pédagogiques, en conservant les clics, le temps passé sur la ressource et un horodatage.
- **Test** : liste des tests disponibles sur la plateforme avec, pour chacun, son nom, son score requis, et son niveau de difficulté (Difficile, simple, facile).
- **Utilisateur_Test** : table de liaison entre les utilisateurs et les tests, qui stock les résultats des utilisateurs aux tests, incluant le nombre de passages, la note obtenue (Excellent, Good, Needs Improvement) et la date de passage.
- **Questions** : stocke les questions associées aux tests, avec les colonnes id de la question, texte de la question, test associé, et ordre de la question.
- **Réponse_A_Choix_Multiple** : liste les propositions de réponse possibles aux questions des tests, avec le texte de la proposition, l'indication si la réponse est correcte, ainsi que l'identifiant de la question correspondante.
- **Thèmes** : liste les thèmes éducatifs auxquels accède la plateforme, avec pour chaque thème son identifiant et son nom (ex : Animaux, Couleurs).
- **Thèmes_Utilisateur** : table de liaison entre les utilisateurs et les thèmes sélectionnés pour connaître les affinités thématiques de chaque utilisateur.
- **Notifications** : liste des notifications accessibles (ex : cours à suivre, rappels), avec pour chaque notification son identifiant et son nom.

4.5 Description de notre API proposée

Le tableau 4.1 présente les différentes méthodes proposées côté back-end à travers la classe ressource "Ressource" et auxquelles notre application mobile fait appel. Ces méthodes permettent de :

- Gérer l'authentification des utilisateurs (connexion, inscription, vérification de login).
- Associer des niveaux aux utilisateurs pour une personnalisation des cours.
- Accéder aux informations utilisateur, y compris leurs favoris, niveaux.
- Ajouter ou supprimer des ressources aux favoris, et vérifier leur statut favori.
- Accéder aux recommandations de cours et de tests basées sur l'historique de navigation, la période de la journée et le niveau.
- Suivre les résultats des tests et enregistrer l'historique de navigation et les préférences de chaque utilisateur.
- Mettre à jour les informations personnelles et pédagogiques des utilisateurs.

Notre API implémentée offre ainsi une gamme complète d'opérations qui peuvent être invoquées à partir du smartphone afin de répondre aux différentes demandes des utilisateurs, permettant ainsi une expérience utilisateur riche et personnalisée.

TABLE 4.1 – Liste des opérations proposées

Signature de l'opération	Méthode	URL d'accès relative : utilisateur/...
	HTTP associés	
boolean ajouterNiveauUtilisateur (int idUtilisateur, int idNiveau)	POST	/ajouterNiveauUtilisateur
boolean doesUserExistWithPassword (String username, String password)	POST	/check-user
List<QuestionDTO> getCorrectAnswersByTest (int idTest)	GET	/correctAnswers/{id_test}
List<Favoris> getFavorisById (int id_utilisateur)	GET	/favoris/{id_utilisateur}
public void ajouterFavoris (Favoris favoris)	POST	/favoris_ajouter
boolean isResourceFavoriteForUser (int id_utilisateur, int id_ress)	GET	/favoris_status/{id_utilisateur}/{id_ress}
void supprimerFavoris (Favoris favoris)	DELETE	/favoris_supprimer
ArrayList<Niveaux> getListeNiveauxUtilisateur(int id_utilisateur)	GET	/get-niveau-utilisateur/{id}
ArrayList<Theme> getThemesUtilisateur(int id_utilisateur)	GET	/get-themes-utilisateur/{id}
Response getAllUtilisateurs ()	GET	/getAllUtilisateurs
public Utilisateur findById (int id_utilisateur)	GET	/getUtilisateur/{id}
List<HistoriqueUtilisateurRessources> getHistoriqueCliqueCable ()	GET	/historique-clique-cable
List<HistoriqueUtilisateurRessources> getHistoriqueByUser (int idUtilisateur)	GET	/historique-utilisateur/{id}
int getLdNiveauByLabels (String nomDiff, String nomAnglais)	PUT	/mettre-a-jour-niveau
void mettreAJourUtilisateur (Utilisateur utilisateur)	PUT	/mettre-a-jour-utilisateur
List<Ressources> getRecommendedResources (int idUtilisateur)	GET	/recommandation/{id_utilisateur}
String getUserActivePeriod (int idUtilisateur)	GET	/recomtest/{id_utilisateur}/{id_cour}
int enregistrerEtRetournerId (Utilisateur utilisateur)	POST	/register-complete
List<UtilisateurTest> getResultsForUser (int idUtilisateur)	GET	/results/{id_utilisateur}
int enregistrerEtRetournerId (Utilisateur utilisateur)	POST	/s-enregistrer
boolean saveTestResult (UtilisateurTest utilisateurTest)	POST	/saveTestResult
boolean saveHistory (HistoriqueUtilisateurRessources history)	POST	/save_history
String valider_utilisateur (String nom_utilisateur, String mot_passe)	GET	/se-connecter
List<Ressources> getSimilarResources(int idUtilisateur, int idCour)	GET	/similar/{id_utilisateur}/{id_cour}
List<Integer> getPassedCoursesForUser(int id_utilisateur)	GET	/userPassedCourses/{id_utilisateur}

4.6 Présentation de notre application mobile

Cette section présente quelques détails relatifs à notre application mobile réalisée. Elle présente dans un premier temps, le diagramme de flux utilisateur qui illustre le processus de navigation dans celle-ci et aussi quelques interfaces qui illustrent les fonctionnalités de base proposées.

4.6.1 Diagramme de flux utilisateur

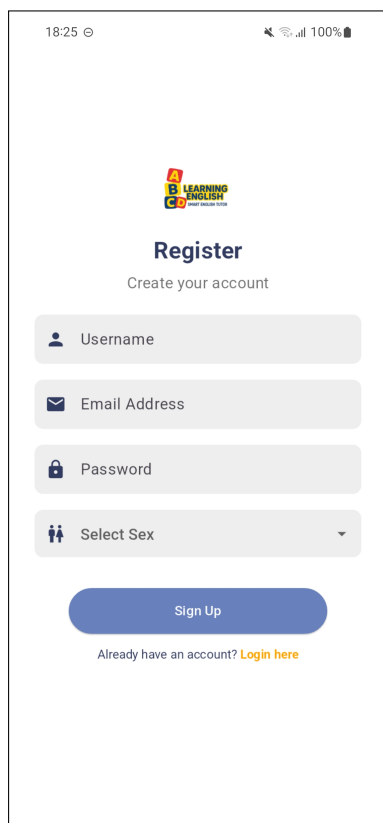
Le diagramme de flux utilisateur présenté dans la Figure 4.2 offre une visualisation des différentes étapes parcourues par l'utilisateur lors de l'utilisation de notre application. Ce diagramme permet de mettre en évidence le cheminement de l'utilisateur à travers les différentes fonctionnalités et actions disponibles, offrant ainsi une représentation claire et cohérente de l'expérience utilisateur.

4.6.2 Présentation de quelques interfaces graphiques

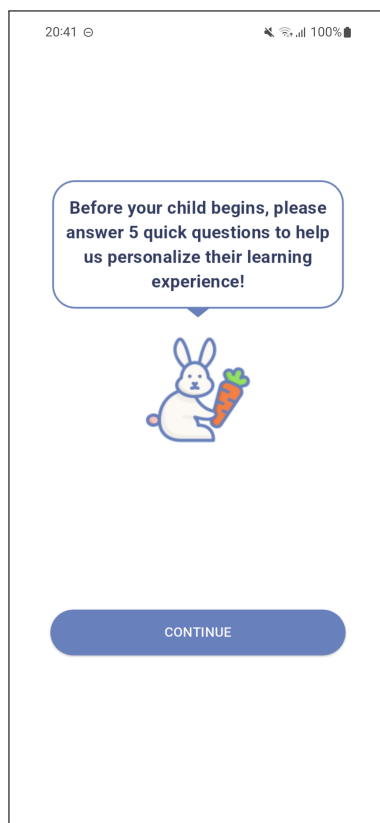
Dans notre application, nous avons développé des interfaces graphiques conviviales qui offrent une expérience utilisateur fluide et intuitive. Elles intègrent diverses fonctionnalités essentielles pour garantir une utilisation optimale. Voici un aperçu de certaines de ces interfaces :

4.6.2.1 Processus d'inscription et création du profil personnalisé

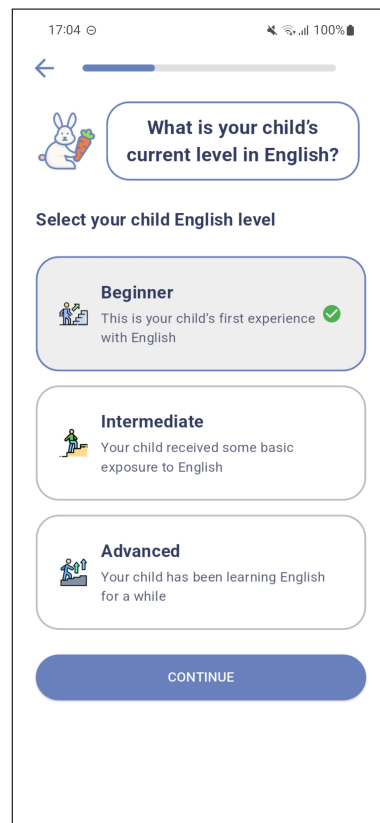
L'interface d'inscription (figure 4.3a) demande au parent de fournir différentes informations (nom, adresse e-mail, mot de passe) et le sexe de l'enfant. puis après validation de ces informations, le parent est dirigé vers une interface d'accueil qui introduit l'expérience à venir, suivie d'une interface de transition l'informant qu'il devra répondre à quelques questions rapides (figure 4.3b). Il accède ensuite au questionnaire de profil permettant de collecter des informations spécifiques sur l'enfant. Ce questionnaire inclut l'âge de l'enfant, son niveau d'anglais (figure 4.3c), ses faiblesses ainsi que ses thèmes préférés, et ses disponibilités horaires (la période de concentration) (figure 4.3d). Ces données permettent de créer un profil personnalisé et adapté, et sont ensuite validées via une dernière interface de confirmation. Le parent est alors redirigé vers la page d'accueil, où il découvrira une sélection de cours recommandés adaptée au profil de son enfant.



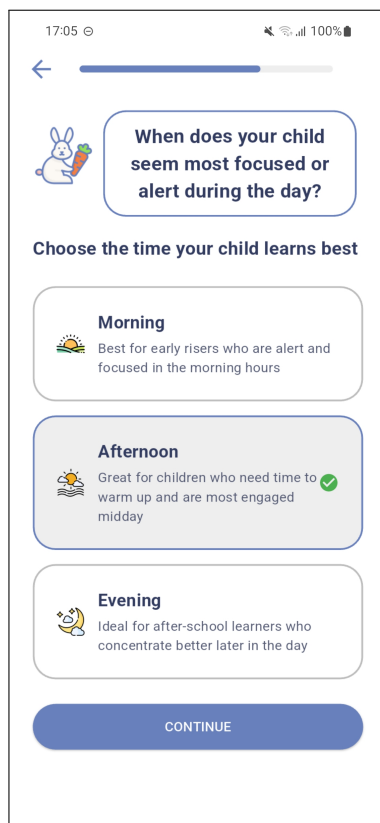
(a) Interface d’inscription avec saisie des informations du parent et de l’enfant.



(b) Interface de transition annonçant le questionnaire de profil.



(c) Interface de sélection du niveau d’anglais de l’enfant.



(d) Interface de choix des disponibilités horaires de l’enfant.

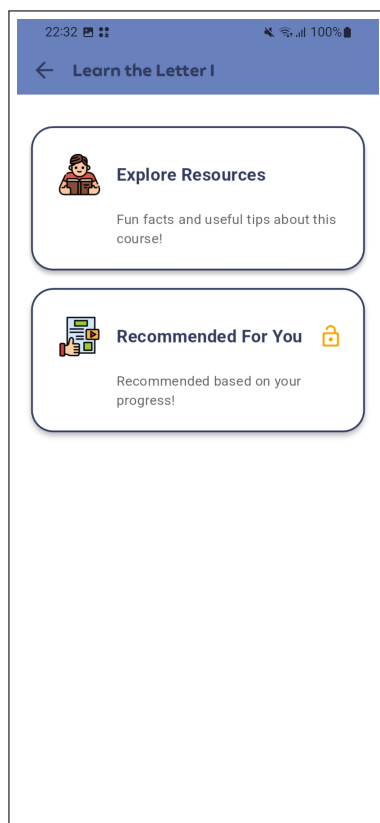
4.6.2.2 Navigation après l'inscription et accès aux cours

Après l'inscription et la création du profil, le parent accède à la page d'accueil (figure 4.4a), où il peut consulter les cours disponibles et en sélectionner un pour commencer l'apprentissage de l'enfant. Une fois un cours sélectionné, l'apprenant accède à une nouvelle interface (figure 4.4b) lui proposant deux options : accéder aux ressources non recommandées ou aux ressources recommandées.

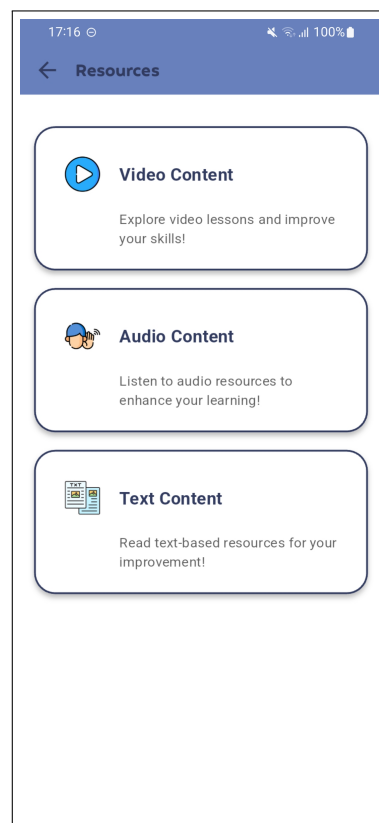
Lorsqu'il choisit l'option ressources non recommandées, toutes les ressources pédagogiques associées au cours sélectionné lui sont présentées (figure 4.4c). L'enfant peut alors naviguer librement parmi ces contenus, et à la fin de chaque ressource, un test recommandé de cours proposé (figure 4.4d).



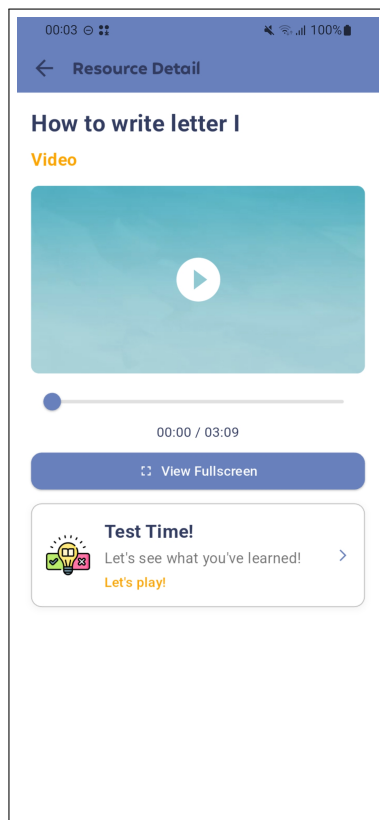
(a) Interface d'accueil des cours disponibles.



(b) Interface des options de ressources.



(c) Interface listant les ressources non recommandées.



(d) Interface de détail de la ressource avec test proposé.

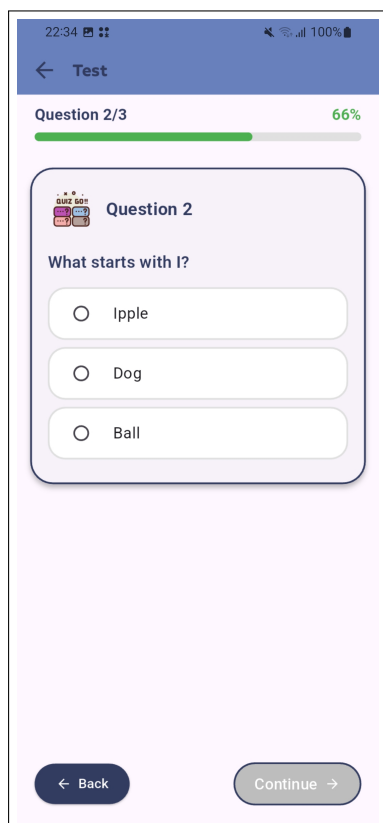
4.6.2.3 Suivi de l'évaluation et adaptation des ressources

En cliquant sur ce test, il accède à un test recommandé selon sa période de concentration estimée (figure 4.5a). Si l'enfant réussit le test (figure 4.5b), il est redirigé vers la page d'accueil (figure 4.4a), où le cours suivant est automatiquement débloqué. En revanche, en cas d'échec (figure 4.5c), plusieurs scénarios sont possibles :

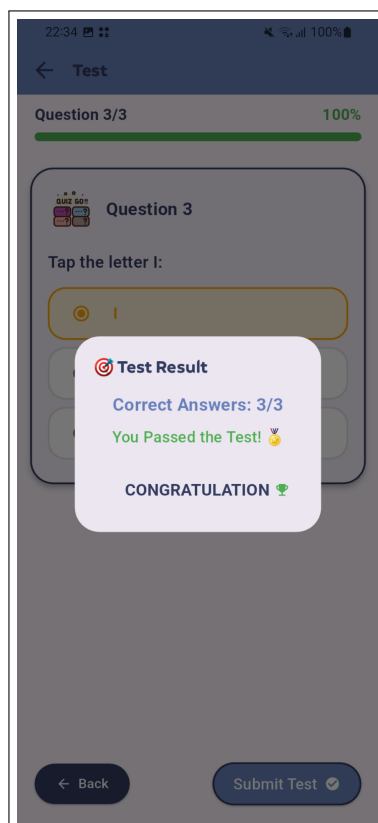
Si les ressources recommandées ne sont pas encore accessibles, l'enfant retourne à la page des ressources non recommandées (figure 4.4c) afin de réviser le contenu du cours échoué.

Si les ressources recommandées (figure 4.4b) sont disponibles, il est redirigé automatiquement vers cette section, où il trouvera des ressources personnalisées basées sur son historique de navigation (figure 4.5d).

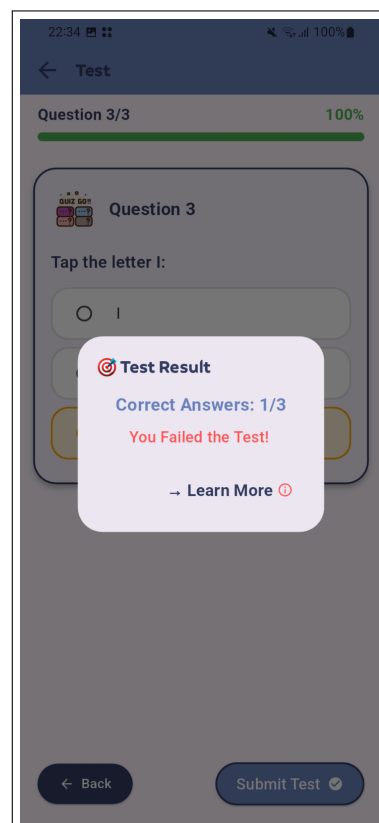
En cas de deux échecs consécutifs au même test, le système ajuste encore davantage les recommandations : les ressources proposées dans la section recommandée (figure 4.5d) sont alors issues des contenus ayant été les plus consultés et validés par d'autres enfants qui ont réussi auparavant. Ces activités sont sélectionnés sur la base de leur fort taux de réussite et de leur niveau d'interaction élevé, garantissant ainsi une meilleure pertinence pédagogique.



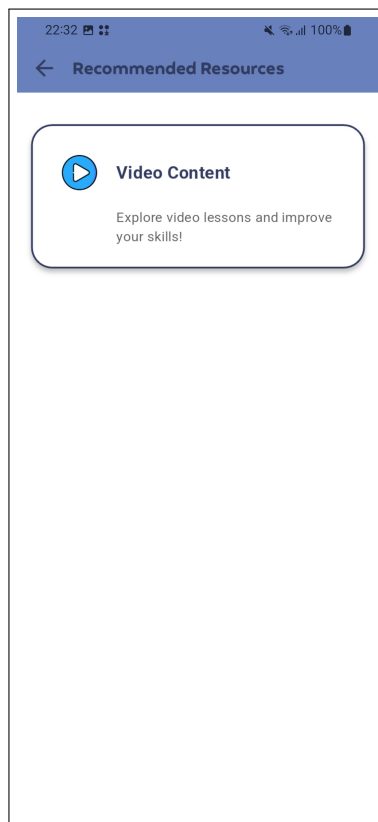
(a) Interface du test recommandé.



(b) Interface de résultat en cas de réussite.



(c) Interface de résultat en cas d'échec.



(d) Interface des ressources recommandées après échec.

4.7 Conclusion

La phase d'implémentation est cruciale dans le cycle de vie de notre application, car elle permet de concrétiser la conception et de mettre en place les fonctionnalités clés. Dans ce dernier chapitre, nous avons détaillé le processus de réalisation de notre application recommandant d'activités pédagogiques aux enfants grâce à l'utilisation de services REST. Nous avons fourni une description approfondie des différentes étapes, en mettant en évidence les choix technologiques et les outils de développement utilisés. Enfin, nous avons conclu en mettant en avant les interfaces graphiques essentielles de notre application. Cette phase d'implémentation constitue une étape fondamentale pour donner vie à notre projet et ouvre la voie à des développements futurs et à des améliorations possibles.

Chapitre 5

Conclusion générale

5.1 Contributions

Ce mémoire porte sur la conception et le développement d'un système intelligent de recommandation d'activités pédagogiques basé sur l'apprentissage adaptatif, destiné à accompagner l'apprentissage de la langue anglaise chez les enfants âgés de 3 à 12 ans. En réponse à l'incapacité des plateformes d'apprentissage classiques à faire face aux besoins des élèves, avec la faible personnalisation des contenus que ces dernières proposent, cette application vise à proposer des contenus éducatifs adaptés aux besoins et au profil de chaque enfant.

Le système a été développé sous forme d'une application mobile Android, afin de garantir une accessibilité optimale et une utilisation intuitive pour les enfants et leurs parents. Le processus de recommandation repose sur plusieurs sources d'information : un questionnaire de profil initial rempli par les parents, l'historique de navigation, les résultats à des tests périodiques, ainsi que les comportements d'apprenants similaires. Pour cela, différentes techniques de filtrage ont été intégrées, notamment le filtrage basé sur le contenu, le filtrage d'activité, le filtrage de l'heure et le filtrage collaboratif.

La modélisation du système a été réalisée à l'aide du langage UML, à travers des diagrammes de cas d'utilisation, de séquence et de classes, ce qui a permis de structurer et de formaliser les composants fonctionnels et les interactions du système.

Nous obtiendrons un système adaptatif faisant des recommandations adaptatives évolutives

d'activités variées (tests éducatifs, vidéos, audios, etc.) en tenant compte des préférences, du niveau et du comportement d'apprentissage de chaque enfant, avec pour objectif de renforcer l'engagement et d'optimiser la progression linguistique.

5.2 Travaux futurs et perspectives

Ce travail en cours offre des opportunités d'expansion et de développement futur. Plusieurs perspectives peuvent être envisagées pour améliorer et étendre les fonctionnalités, notamment :

- Le questionnaire de profil pourra être enrichi par les sujets préférés de l'enfant (ex. : animaux, nourriture, nature), afin de rendre les recommandations plus personnalisées et engageantes dès les premières utilisations.
- L'efficacité des vidéos, textes ou audios sera analysée pour chaque utilisateur. Le système pourra alors recommander le type de ressource ayant donné les meilleurs résultats pour chaque enfant.
- Des protocoles d'évaluation rigoureux seront mis en place pour mesurer l'efficacité des recommandations, en s'appuyant sur des métriques adaptées (engagement, taux de réussite, satisfaction parentale...).
- Les évaluations pourront être proposées en fonction du moment de la journée où l'enfant est le plus réceptif, ou selon son historique de performance, afin d'optimiser son expérience d'apprentissage.
- Incorporation de d'autres informations contextuelles supplémentaires en complément à celles déjà utilisées.

Références

- [1] M. Khaoula, L. Imane, and D. B. Nadjib, “Vers des systèmes de recommandation sensibles au contexte.”
- [2] H. Lhamri, “L’enseignement en ligne et la scénarisation pédagogique,” *IOSR Journal of Research & Method in Education (IOSR-JRME)*, vol. 10, no. 4, pp. 7–10, 2020, <https://www.iosrjournals.org/iosr-jrme/papers/Vol-10%20Issue-4/Series-7/B1004070710.pdf>.
- [3] C. Hélot and M.-N. Rubio, *Développement du langage et plurilinguisme chez le jeune enfant*. Toulouse : Éditions érès, 2014.
- [4] C. A. Tomlinson, *How to Differentiate Instruction in Mixed-Ability Classrooms*. Alexandria, VA : ASCD, 2001.
- [5] A. Talan. (2024) Apprentissage adaptatif et au-delà : Comment l’intelligence artificielle façonne l’éducation. Blog Talan.
- [6] S. Abdullah *et al.*, “Adaptive learning systems : Harnessing ai for customized educational experiences,” *International Journal of Computer Science and Information Technology*, vol. 6, no. 3, pp. 18–22, 2024.
- [7] P. Brusilovsky and E. Millán, “User models for adaptive hypermedia and adaptive educational systems,” in *The Adaptive Web : Methods and Strategies of Web Personalization*, ser. Lecture Notes in Computer Science, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Springer, 2007, vol. 4321, pp. 3–53.
- [8] F. Ricci, L. Rokach, and B. Shapira, “Recommender systems : Introduction and challenges,” in *Recommender Systems Handbook*, 2nd ed., F. Ricci, L. Rokach, and B. Shapira, Eds. Springer, 2015, pp. 1–34.

- [9] “Recommender systems handbook.” [Online]. Available : <https://link.springer.com/10.1007/978-1-0716-2197-4>
- [10] K. Falk, *Practical Recommender Systems*. Manning, 2019. [Online]. Available : https://books.google.dz/books?id=_dbdnAAACAAJ
- [11] F. Isinkaye, Y. Folajimi, and B. Ojokoh, “Recommendation systems : Principles, methods and evaluation,” vol. 16, no. 3, pp. 261–273. [Online]. Available : <https://linkinghub.elsevier.com/retrieve/pii/S1110866515000341>
- [12] F. Aioli, “Recommender systems : From collaborative filtering to deep learning,” in *Recommender Systems Handbook*, 2016, pp. 17–45. [Online]. Available : <https://link.springer.com/10.1007/978-1-0716-2197-4>
- [13] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems : An Introduction*. Cambridge University Press, 2010.
- [14] S. B. Ticha, “Recommandation personnalisée hybride,” publisher : Unpublished. [Online]. Available : <http://rgdoi.net/10.13140/RG.2.2.22484.81287>
- [15] V. R. Kagita, A. K. Pujari, and V. Padmanabhan, “Virtual user approach for group recommender systems using precedence relations,” vol. 294, pp. 15–30. [Online]. Available : <https://linkinghub.elsevier.com/retrieve/pii/S0020025514009086>
- [16] P. Kumar, S. Vairachilai, S. Potluri, and S. Mohanty, *Recommender Systems : Algorithms and Applications*. CRC Press, 2021. [Online]. Available : <https://books.google.dz/books?id=dT4oEAAAQBAJ>
- [17] I. Benouaret, “Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels,” Ph.D. dissertation, Université (nom de l’université, si connu), 2017.
- [18] G. Linden, B. Smith, and J. York, “Amazon.com recommendations : Item-to-item collaborative filtering,” *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
- [19] T. Duricic, E. Lacic, D. Kowald, and E. Lex, “Trust-based collaborative filtering : Tackling the cold start problem using regular equivalence,” *arXiv preprint arXiv :1807.06839*, 2018. [Online]. Available : <https://arxiv.org/abs/1807.06839>

- [20] M. D. Ekstrand, “Collaborative filtering recommender systems,” vol. 4, no. 2, pp. 81–173. [Online]. Available : <http://www.nowpublishers.com/article/Details/HCI-009>
- [21] P. S. Kamble, S. V. Todkari, and S. R. Kolhe, “Comparison of collaborative filtering, content-based filtering and hybrid recommendation systems in a movie recommender system,” 2020.
- [22] B. Smith and G. Linden, “Two decades of recommender systems at amazon.com,” vol. 21, no. 3, pp. 12–18. [Online]. Available : <http://ieeexplore.ieee.org/document/7927889/>
- [23] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” vol. 42, no. 8, pp. 30–37. [Online]. Available : <http://ieeexplore.ieee.org/document/5197422/>
- [24] Celma, *Music Recommendation and Discovery : The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer Berlin Heidelberg. [Online]. Available : <https://link.springer.com/10.1007/978-3-642-13287-2>
- [25] Microsoft, “Que sont les recommandations intelligentes,” <https://learn.microsoft.com>, 2024.
- [26] M. Saraei, “Développement d’un système de recommandation de produits dans le commerce électronique,” <https://publications.polymtl.ca>, 2022.
- [27] U. du Québec à Montréal (UQAM), “Recommandations intelligentes de produits basées sur l’utilisation d’algorithmes d’apprentissage machine,” <https://archipel.uqam.ca>, 2021.
- [28] V. Institute, “Systèmes de recommandation : Quand l’université rencontre l’industrie,” <https://vectorinstitute.ai>, 2024.
- [29] C. C. Aggarwal, *Recommender Systems*. Springer International Publishing. [Online]. Available : <http://link.springer.com/10.1007/978-3-319-29659-3>
- [30] S. Ouyang, L. Li, W. Pan, and Z. Ming, “Asymmetric bayesian personalized ranking for one-class collaborative filtering,” in *Proceedings of the 13th ACM Conference on Recommender Systems*. ACM, pp. 373–377. [Online]. Available : <https://dl.acm.org/doi/10.1145/3298689.3347051>

- [31] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” vol. 46, pp. 109–132. [Online]. Available : <https://linkinghub.elsevier.com/retrieve/pii/S0950705113001044>
- [32] Vorecol. (2023) La personnalisation de l’apprentissage : comment les logiciels s’adaptent aux besoins individuels des apprenants. [Online]. Available : <https://vorecol.com/fr/blogs/blog-la-personnalisation-de-lapprentissage-comment-les-logiciels-sadaptent-aux-besoins-individuels-d>
- [33] D. T. Seaton, Y. Bergner, I. Chuang, P. Mitros, and D. E. Pritchard, “Who does what in a massive open online course ?” *Communications of the ACM*, vol. 57, no. 4, pp. 58–65, 2014.
- [34] B. Settles and B. Meeder, “A trainable spaced repetition model for language learning,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 1848–1858.
- [35] Knewton, “The knewton platform : Adaptive learning infrastructure,” 2013. [Online]. Available : <https://www.knewton.com/resources/knewton-adaptive-learning-platform/>
- [36] S. Sparrow, “The science of adaptive learning,” 2018. [Online]. Available : <https://web.archive.org/web/20201010043434/https://www.smartsparrow.com/the-science-of-adaptive-learning/>
- [37] K. Cherry, “Learning theories in psychology : An overview,” <https://www.verywellmind.com/learning-theories-in-psychology-an-overview-2795082>, 2023. [Online]. Available : <https://www.verywellmind.com/learning-theories-in-psychology-an-overview-2795082>
- [38] Learning Leap Consultants. (2023) The role of cognitive load theory in adaptive learning systems. [Online]. Available : <https://learningleapconsultants.com/role-cognitive-load-theory-adaptive-learning-systems/>
- [39] Wikipédia contributors, “Zone proximale de développement,” https://fr.wikipedia.org/wiki/Zone_proximale_de_d%C3%A9veloppement, 2024. [Online]. Available : https://fr.wikipedia.org/wiki/Zone_proximale_de_d%C3%A9veloppement
- [40] R. Nkambou, J. Bourdeau, and R. Mizoguchi, “Introduction : What are intelligent tutoring systems, and why this book ?” in *Advances in Intelligent Tutoring Systems*. Springer,

- 2010, pp. 1–12.
- [41] P. Brusilovsky, “Adaptive hypermedia,” *User Modeling and User-Adapted Interaction*, vol. 11, no. 1–2, pp. 87–110, 2001.
- [42] J. Kay, “User modeling for adaptation,” *User Modeling and User-Adapted Interaction*, vol. 11, no. 1–2, pp. 101–108, 2001.
- [43] K. Verbert, N. Manouselis, H. Drachsler, and E. Duval, “Dataset-driven research to support learning recommender systems,” *Adaptive Technologies for Training and Education*, pp. 39–63, 2012.
- [44] V. Aleven, B. M. McLaren, I. Roll, and K. R. Koedinger, “Toward meta-cognitive tutoring : A model of help-seeking with a cognitive tutor,” *International Journal of Artificial Intelligence in Education*, vol. 16, no. 2, pp. 101–128, 2006. [Online]. Available : <https://www.ijaied.org/>
- [45] S. Loewen, D. R. Isbell, and Z. Sporn, “The effectiveness of app-based language instruction for developing receptive linguistic knowledge and oral communicative ability,” *Foreign Language Annals*, vol. 53, no. 2, pp. 209–233, 2020.
- [46] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein, “Deep knowledge tracing,” in *Advances in neural information processing systems*, vol. 28, 2015.
- [47] A. Zimmermann, A. Lorenz, and R. Oppermann, “An operational definition of context,” 2007.
- [48] T. Gross and M. Specht, “Awareness in context-aware information systems,” *Mensch Computer 2001 Berichte des German Chapter of the ACM*, 2001.
- [49] R. Klemke, “Modelling context in information brokering processes,” 2002.
- [50] V. Psyché and P. Ruer, “L’apprentissage adaptatif intelligent,” *Le Tableau – Bulletin de la pédagogie de l’Université du Québec*, vol. 8, no. 4, 2019. [Online]. Available : <https://pedagogie.quebec.ca/le-tableau/lapprentissage-adaptatif-intelligent>
- [51] Object Management Group, “OMG Unified Modeling Language (OMG UML), Version

- 2.5.1,” <https://www.omg.org/spec/UML/2.5.1>, December 2017, accessed May 2025.
- [52] M. Fowler, *UML Distilled : A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 2004.
- [53] M. Seidl, M. Scholz, C. Huemer, and G. Kappel, *UML@Classroom : An Introduction to Object-Oriented Modeling*. Springer, 2015.
- [54] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. Dissertation, University of California, Irvine, 2000.
- [55] “Jakarta ee,” <https://jakarta.ee/>.
- [56] “Apache tomcat,” <https://tomcat.apache.org/>.
- [57] “Eclipse foundation,” <https://www.eclipse.org/>.
- [58] “Jakarta restful web services specification,” <https://jakarta.ee/specifications/restful-ws/>.
- [59] “Wampserver,” <https://www.wampserver.com/>.
- [60] “Android,” <https://www.android.com/>.
- [61] “Android studio,” <https://developer.android.com/studio>.
- [62] “http package — flutter package,” <https://pub.dev/packages/http>, 2025.

Annexe A

Exemples de codes pour notre API

Les exemples suivants illustrent quelques-unes des principales méthodes de notre API, telles qu'elles sont utilisées dans notre application mobile. Ils permettent de mieux comprendre l'interaction entre l'application et le back-end à travers des appels REST concrets.

1.1 Exemple : Enregistrement complet d'un enfant

Le code ci-dessous met en perspective comment se met en œuvre l'inscription intégrale de l'enfant dans notre application. Cet enregistrement intervient après que le parent a rempli un formulaire de création de compte ainsi qu'un questionnaire de profil initial, permettant de recueillir des données détaillées sur l'enfant.

L'objectif est d'enregistrer dans la base de données toutes les données utiles à la personnalisation du système de recommandation, à savoir : les données basiques (nom, email, mot de passe), le niveau actuel en anglais, les points faibles de l'enfant, ses préférences de thèmes et le moment de la journée où il est le plus concentré.

Côté back-end (Java – méthode REST) :

L'enregistrement se fait via un endpoint **REST** accessible à l'URL **/utilisateur/register-complete**.

Ce dernier reçoit un objet JSON contenant les données suivantes :

L'enregistrement des enfants se fait via un endpoint REST accessible à l'URL /utilisateur/register-complete. Ce point d'entrée reçoit un objet JSON contenant les données suivantes :

- Données de base de l'utilisateur : username, email, password, sex
- Données issues du questionnaire de profil initial : age_group, levelEnglish, difficultyEnglish, themes, active

Une fois les données reçues, le traitement backend suit les étapes suivantes :

Récupération de l'identifiant du niveau (id_niveau) Une méthode dédiée getIdNiveauByLevelAndDifficulty dans la classe UtilisateurDAO permet de récupérer l'identifiant correspondant à la combinaison du niveau d'anglais et de la difficulté perçue. Cette requête est sécurisée via PreparedStatement.

```

1  \ \  ...
2  public int getIdNiveauByLabels(String nomDiff, String nomAnglais)
   throws SQLException {
3      String sql = "SELECT id_niveau FROM niveaux WHERE
   nom_niveau_de_diff = ? AND nom_niveau_anglais = ?";
4      try (PreparedStatement stmt = conn.prepareStatement(sql)) {
5          stmt.setString(1, nomDiff);
6          stmt.setString(2, nomAnglais);
7          ResultSet rs = stmt.executeQuery();
8              \ \  ...
9
10 }

```

Création de l'objet Utilisateur Un objet de type Utilisateur est alors instancié avec l'ensemble des données JSON reçues, y compris l'id_niveau récupéré précédemment.

```

1  \ \  ...
2  Utilisateur utilisateur = new Utilisateur();
3  utilisateur.setNom_utilisateur(request.getUsername());
4  utilisateur.setEmail(request.getEmail());
5  utilisateur.setMot_passe(request.getPassword());
6  utilisateur.setSexe(request.getSex());
7  utilisateur.setAge(request.getAge_group());

```

```

8 utilisateur.setActive(request.getActive());
9 utilisateur.setId_niveau(niveauId);

```

Insertion de l'enfant en base de données L'objet utilisateur est ensuite inséré dans la base à l'aide d'une requête INSERT via PreparedStatement. L'identifiant généré automatiquement est récupéré.

```

1  \ \ ...
2  public int enregistrerEtRetournerId(Utilisateur utilisateur) throws
   SQLException {
3      String sql = "INSERT INTO utilisateur (nom_utilisateur, email
   , mot_passe, sexe, age, active, id_niveau) VALUES (?, ?, ?, ?, ?, ?,
   ?)";
4
5      try (PreparedStatement stmt = conn.prepareStatement(sql,
   Statement.RETURN_GENERATED_KEYS)) {
6          stmt.setString(1, utilisateur.getNom_utilisateur());
7          stmt.setString(2, utilisateur.getEmail());
8          stmt.setString(3, utilisateur.getMot_passe());
9          stmt.setString(4, utilisateur.getSexe());
10         stmt.setString(5, utilisateur.getAge());
11         stmt.setString(6, utilisateur.getActive());
12         stmt.setInt(7, utilisateur.getId_niveau());
13
14         int affectedRows = stmt.executeUpdate();
15         \ \ ...
16
17     }
18 }

```

Association des thèmes à l'utilisateur Chaque themeId reçu dans la liste est associé à l'enfant via une boucle qui remplit la table de liaison correspondante.

```

1  String insertQuery = "INSERT INTO themes_utl (id_utilisateur, id_theme)
   VALUES (?, ?)";
2      try (PreparedStatement insertStmt = conn.prepareStatement(
   insertQuery)) {
3          insertStmt.setInt(1, id_utilisateur);
4          insertStmt.setInt(2, id_theme);

```

```

5         int rowsAffected = insertStmt.executeUpdate();
6         \\ ...
7     }

```

Retour de la réponse HTTP En fin de traitement, une réponse HTTP de succès est retourné si tout s’est bien

```

1 return Response.status(Response.Status.CREATED).entity("Utilisateur
   enregistr\’e avec succ\’es").build();

```

Côté front-end (Flutter – appel API) :

L’envoi des données enfant se fait à l’aide d’un modèle structuré UserDataModel, qui regroupe les informations de base ainsi que les réponses au questionnaire de profil initial. Ce modèle permet une conversion simple vers le format JSON attendu par l’API.

```

1 class UserDataModel {
2     // basic data
3     final String username;
4     final String email;
5     final String password;
6     final String sex;
7     // questionnaire data
8     String? age_group;
9     String? levelEnglish;
10    String? difficultyEnglish;
11    List<int> themes;
12    String? active;
13    \\ ...
14
15    String toString() {
16        return 'UserDataModel{username: $username, email: $email,
17            password: $password, sex: $sex, age_group: $age_group, levelEnglish:
18                $levelEnglish, difficultyEnglish: $difficultyEnglish, themes:
19                $themes, active: $active}';
20    }

```

Une fois les données regroupées dans un objet UserDataModel, elles sont envoyées vers l’endpoint /utilisateur/register-complete via une requête HTTP POST. L’URL de base

pointe vers le serveur local de développement.

```

1  \\ ...
2  final String baseUrl = "http://192.168.1.103:8081/recommandation/monapi
   ";
3  }

```

La méthode `submitCompleteUserData` s'occupe de l'envoi des données. Elle convertit l'objet en JSON, effectue l'appel réseau, et retourne un booléen selon le succès de l'opération.

```

1  \\ ...
2  Future<bool> submitCompleteUserData(UserDataModel userData) async {
3    try {
4      final Map<String, dynamic> requestBody = userData.toJson();
5      print('Submitting complete user data: $requestBody');
6
7      final response = await http.post(
8        Uri.parse('$baseUrl/utilisateur/register-complete'),
9        headers: {'Content-Type': 'application/json'},
10       body: jsonEncode(requestBody),
11     );
12  \\ ...
13  }

```

Enfin, l'appel à cette méthode est déclenché lorsque le parent valide le formulaire dans l'application. L'objet `userData` contenant toutes les réponses est transmis à la méthode, et le résultat est traité en conséquence.

```

1  final success = await _apiService.submitCompleteUserData(widget.
   userData);
2  \\ ...

```

1.2 Exemple : Recommandation personnalisée basée sur l'historique de navigation

Le code ci-dessous illustre comment l'application propose dynamiquement des ressources pédagogiques à un enfant, en analysant son historique d'interactions (clics et temps passé sur les ressources).

Ce système intervient après que l'enfant a commencé à utiliser l'application. L'objectif est de recommander les ressources les plus adaptées à ses préférences comportementales.

Côté back-end (Java – méthode REST) :

La recommandation est déclenchée via un endpoint REST, accessible à l'URL suivante :

```
1 GET /utilisateur/recommend/{id_utilisateur}
```

Par exemple : /utilisateur/recommend/2 permet de demander les recommandations pour l'utilisateur n°2.

À partir de cet identifiant, le back-end suit les étapes suivantes :

Analyse des données d'historique Les données enregistrées sont issues des interactions précédentes de l'enfant avec l'application. Deux paramètres principaux sont pris en compte :

- Le nombre de clics sur les ressources (clique_cable)
- Le temps passé sur chaque ressource (time_spent)

Ces données sont envoyées via une méthode REST comme suit :

```
1 public Response saveUserHistory(HistoriqueUtilisateurRessources history
2 ) {
3     System.out.println(" Received history from Flutter:");
4     System.out.println(" id_utilisateur: " + history.getIdUtilisateur()
5 );
6     System.out.println(" id_ress: " + history.getIdRes());
7     System.out.println(" clique_cable: " + history.isCliqueCable());
8     System.out.println(" time_spent: " + history.getTimeSpent());
9
10    \\ ...
11    boolean success = userHistoryDAO.saveHistory(history);
12    \\ ...
```

Génération des recommandations Après enregistrement, les données sont traitées pour générer une liste personnalisée. Une pondération entre les clics et le temps moyen passé est utilisée pour déterminer un score de pertinence.

```

1  \ \ ...
2  public Response getRecommendation(@PathParam("id_utilisateur") int
   idUtilisateur) {
3      try {
4          \ \ ...
5          UtilMethods utilMethods = new UtilMethods();
6          List<Ressources> recommandations = utilMethods.
   getRecommendedResources(idUtilisateur);
7
8          return Response.ok(recommandations).build();
9      } catch (Exception e) {
10         \ \ ...
11     }
12 }
13

```

La prise en charge SQL à l'intérieur de `getRecommendedResources` s'organise autour La prise en charge SQL à l'intérieur de `getRecommendedResources` s'organise autour d'éléments de logique algorithmique se traduisant par des requêtes SQL imbriquées à l'aide de CTEs (pour Common Table Expression) et devant permettre :

- Calculer le nombre total de clics par type de ressource (`ClickCounts`)
- Calculer le temps moyen passé par type (`TimeSpent`)
- Fusionner les deux métriques dans un score pondéré (`WeightedScores`)
- Identifier les types les plus pertinents (`TopTypes`)
- Filtrer les ressources par niveau de l'enfant

```

1  SELECT r.*
2  FROM ressources r
3  JOIN (
4      SELECT type,
5             (nb_clicques * 1.0 / max_clicques) * 0.5 +
6             (avg_time_spent / max_time) * 0.5 AS score
7      FROM ...
8  ) scores ON r.type = scores.type
9  WHERE r.id_niveau = ?
10 ORDER BY score DESC
11

```

Côté front-end (Flutter – appel API) :

Côté **front-end**, les interactions de l'enfant avec les ressources sont captées et transmises au **back-end** via des appels HTTP. Deux fonctionnalités principales sont concernées ici :

Enregistrement des données d'usage L'application Flutter utilise une méthode dédiée pour envoyer au serveur les interactions effectuées par l'enfant sur les différentes ressources pédagogiques.

```

1 // Save resource interaction history to backend
2 Future<bool> saveHistory({
3   \\ ...
4 }) async {
5   final url = Uri.parse('$baseUrl/utilisateur/save_history');
6   final Map<String, dynamic> historyData = {
7     "idUtilisateur": int.parse(userId),
8     "idRess": resourceId,
9     "cliqueCable": cliqueCable,
10    "timeSpent": timeSpent,
11  };
12  print("Sending history data: $historyData");
13  \\ ...
14  }
15

```

Récupération des recommandations personnalisées Après avoir collecté l'historique de navigation, la méthode `fetchRecommendations` permet de récupérer dynamiquement les recommandations générées pour l'utilisateur.

```

1 Future<List<String>> fetchRecommendations(String userId) async {
2   final url = Uri.parse('$baseUrl/utilisateur/recommandation/$userId');
3   final response = await http.get(url);
4   \\ ...
5 }

```