



Application de recommandation adaptative sur mobile

SAÉ BUT3 5.Real.01 – 2025 – Situation d'Apprentissage et d'Évaluation – Développement avancé

[L'application : recommandation adaptative indépendante sur mobile](#)

[Bibliographie](#)

[💡 Exemples d'applications pour vous alimenter en idées](#)

[❓ Voici quelques questions non-exhaustives à vous poser pour bien démarrer le projet](#)

[Fonctionnalités obligatoires](#)

[Livrables obligatoires](#)

[Livrables et fonctionnalités supplémentaires](#)

[Contraintes organisationnelles](#)

[Évaluation et compétences ciblées](#)

[Dates butoirs : développement](#)

[Dates butoirs : optimisation](#)

[Dates butoirs : communication](#)

L'application : recommandation adaptative indépendante sur mobile

Les applications mobiles sont au cœur des préoccupations de nombreuses entreprises du tertiaire, en particulier dans le domaine du service¹. Les smartphones étant désormais quasiment devenus une extension de notre corps (Ross *et al.*, 2024), il est important de considérer leur usage comme étant de facto l'usage principal de toute application. Ceci a notamment amené l'émergence de stratégies de développement comme le mobile-first (Wroblewski, 2012). Cependant, ces applications sont bien souvent uniquement des coquilles vides, des vues (dans le cas d'un paradigme MVC) interagissant avec une application backend déportée. De telles applications sont très utilisées et populaires, comme Discord, Teams, Visual Studio Code, pour ne citer qu'eux. C'est également le cas des applications dans lesquelles la recommandation est au cœur de l'expérience utilisateur, comme TikTok, Instagram, Netflix, par exemple. Mais alors, quid d'une recommandation dans une application indépendante ?

Notre entreprise, l'IUT Villetaneuse Corp., a besoin d'une application mobile permettant la recommandation d'éléments à l'utilisateur sans qu'il n'ait besoin de se connecter. En effet, nous souhaitons prendre en compte l'utilisation directe sur smartphone de l'utilisateur et adapter la recommandation d'éléments en fonction de l'évolution de son usage de cette

¹ <https://www.robinwaite.com/blog/focuses-on-mobile-apps-as-a-strategic-advantage-for-businesses>

dernière. Cela a pour avantage d'isoler les informations d'utilisation (i.e. les traces numériques) de l'utilisateur qui pourra alors décider de les supprimer ou non, de les partager ou non et gardera ainsi la souveraineté sur ses données. Cependant, plusieurs problématiques émergent alors :

- Comment représenter ces données pour pouvoir recommander efficacement ?
 - Sous forme de graphe ? Sous forme d'espace vectoriel ? Sous forme de distance paramétrique ?
- Comment avoir une recommandation efficiente qui puisse ne pas trop drainer la batterie du téléphone ?
 - Doit-on créer des groupes de données similaires ? Traiter toutes les données ? À quel moment faire les calculs ?
- Comment adapter et faire évoluer la recommandation à l'utilisation ?
- Quelle technologie utiliser et pourquoi ?
 - Android Studio ? Javascript (React Native ? VueJS ? Cordova ?) ? Flutter ? etc.

L'entreprise, dans sa grande bonté, vous laisse le champ libre quant à la thématique de l'application et donc, aux données à considérer, aux technologies à utiliser (tant que cela finit en application mobile). L'objectif étant pour l'entreprise de montrer aux investisseurs sa capacité à créer une application mobile efficiente dont la recommandation est adaptative.

L'IUT villetaneuse Corp. a donc besoin d'une équipe de développeurs compétents, qui fourniront une application développée proprement en équipe (compétence 1 – développement), optimisée et pas trop gourmande en ressources (compétence 2 – algorithmique) et avec un aspect commercial travaillé (compétence 3 – communication). Cette équipe... c'est VOUS ! 😊

Bibliographie

Ross, M. Q., & Kushlev, K. (2024). Antecedents and consequences of smartphone self-extension. *Psychology of Popular Media*.

Wroblewski, L. (2012). *Mobile first: Preface de jeffrey zeldmann*. Editions Eyrolles.



Exemples d'applications pour vous alimenter en idées

- recommandation de livres avec évolution des suggestions de livres en fonction des genres littéraires et habitudes (heure de lecture, durée de lecture, etc.)
- recommandation de cours et formations en informatique en fonction du niveau de l'étudiant, de son parcours, etc. (i.e. parcoursup en mieux 🤔)
- recommandation d'activité sportive en fonction du lieu et de l'heure actuels
- etc. (Surprenez-nous !)

? Voici quelques questions non-exhaustives à vous poser pour bien démarrer le projet

- Comment représenter les données ?



- Comment obtenir une bonne performance pour un tel système ?
- Comment monitorer la performance de l'application ?
- Quels composants techniques ? (langage, framework, etc.)
- Quel nom / branding / logo pour l'application ?
- Comment échanger / communiquer entre vous ?

Fonctionnalités obligatoires

- Métriques de performance avec historique : consommation des ressources systèmes ; temps de réponse max, min, moyen ; statistiques d'utilisation de la batterie en utilisation active VS en utilisation passive (tâche de fond) ; complexité de l'algorithme de recommandation, etc.
- Interface utilisateur travaillée, justifiée et motivée (pourquoi avoir une telle disposition, en quoi est-ce bénéfique pour la recommandation souhaitée, etc.).
- Adaptation de la recommandation au fur et à mesure de l'utilisation : quels éléments et à quelle intensité la recommandation doit évoluer. Justifier la stratégie choisie.
- Licence open-source au choix : <https://opensource.org/licenses> Attention à bien justifier son choix

Livrables obligatoires

- Description du projet :
 - a. membres de l'équipe
 - b. thématique de l'application + plan pour les données
 - c. stack technique choisie
- Version fonctionnelle de l'application :
 - a. application fonctionnelle mais pas encore finalisée
 - b. fonctionnalités principales disponibles
 - c. interface sommaire mais fonctionnelle
- Version finale peaufinée :
 - a. interface pensée et motivée par des choix précis
 - b. branding, logo, catch phrase, slogan
- Un rapport avec présentation de l'application, du branding, justifications, etc. Ce rapport est à mettre dans un fichier markdown.
 - a. stratégie de présentation de vente de l'application : quel public cible ? quel intérêt ? quel besoin utilisateur est couvert ?
 - b. stratégie commerciale : quel argument de vente ?
 - c. Un test utilisateur final doit être prévu dans ce rapport : par un camarade de classe, ou une autre personne
- Le code source, ou plus précisément, le dépôt git du code source.
- Une simple capture vidéo montrant l'utilisation de l'application (avec commentaires audios si vous le souhaitez, mais recommandé) : cette vidéo n'a pas besoin d'un montage particulier, elle sert surtout à ce que chaque enseignant évaluateur puisse aisément voir les applications de chaque groupe.

Livrables et fonctionnalités supplémentaires

Si le cœur ❤️ vous en dit, vous pouvez ajouter toute fonctionnalité ou éléments à livrer. N'oubliez pas de justifier succinctement ces ajouts.

Contraintes organisationnelles

- Utiliser git pour échanger au sein des équipes
- Prévoir des jalons pour la création de l'application mais surtout pour son intégration, jusqu'à la mise en production (en local sur le téléphone dans les faits – par exemple un .apk sous android)
- Utilisation des conventions de nommage des versions :
https://en.wikipedia.org/wiki/Software_versioning

Évaluation et compétences ciblées

L'évaluation se fait par compétence. Chaque compétence est évaluée indépendamment des autres. Toutes les compétences ont une valeur égale dans l'évaluation.

Il apparaît évident que la compétence **Développement** concerne la création de l'application, la compétence **Optimisation** concerne l'algorithme de recommandation, son optimisation, ainsi que celle de l'usage de la batterie, et que la compétence **Communication** concerne notamment l'interface, le branding, et le rapport de communication.

Voici les responsables de l'évaluation pour chaque compétence :

- **Développement** : Développer — c'est-à-dire concevoir, coder, tester et intégrer — une solution informatique pour un client.
 - Gaël Guibon : gael.guibon@sorbonne-paris-nord.fr
- **Optimisation** : Proposer des applications informatiques optimisées en fonction de critères spécifiques : temps d'exécution, précision, consommation de ressources.
 - Pierrick Siest : pierrick.siest@sorbonne-paris-nord.fr
- **Communication** : Acquérir, développer et exploiter les aptitudes nécessaires pour travailler efficacement dans une équipe informatique.
 - Marie-Eva Lesaunier : marieevalesaunier@gmail.com

Dates butoirs : développement

- **06 octobre** : Description du projet :
 - membres de l'équipe (5 à 6 personnes par groupe)
 - thématique de l'application + plan pour les données
 - stack technique choisie
- **03 décembre** : *dépôt git* : Première version de l'application
 - Interface complète (mais pas forcément jolie)
 - Backend mis en place (Controllers et Services)
 - Model (Persistence, BDD) pas obligatoirement finalisé

- Recommandation et application pas encore optimisée
- **20 janvier** : Version finale avec interface finale, performance et fonctionnalités
 - Vidéo d'utilisation de l'application
 - Dépôt git final

Dates butoirs : optimisation

- **09 décembre** : *fichier PDF* : Rapport sur les axes d'amélioration pour la Première version de l'application
 - section vulgarisation de la partie dev pour les non dev / mathématiciens
 - identification des points à optimiser pour une recommandation rapide et efficace
 - identification des points à améliorer pour économiser la batterie
 - propositions de stratégies possibles
- **09 janvier** : *fichier PDF* : rapports de performance et calculs de coûts de la recommandation
- **20 janvier** :
 - rapport sur l'optimisation de l'application en général : complexité optimisée et expliquée, identification des points bloquants pour la réactivité de l'application, explication des stratégies d'optimisation de l'utilisation de la batterie, etc.

Dates butoirs : communication

- **17 décembre** : branding + logo avec explications et motivation
- **15 janvier** : rapport de la communication intra-équipe
 - Section sur l'usage du git et tout autres outils (trello, jira, etc.)
- **20 janvier** : rapport de mise en avant / présentation de l'application : ses fonctionnalités discriminantes, etc.