CONGUISTI Nicolas

ESPINASSE Virgile

JULLIARD Timéo

LAVERGNE Elsa

SALLENAVE XAN

TD2 Parcours D

**Pôle Développement (C1&C2)**  
**Projet n°21 - BonPlan&Co**  
**Identification du problème algorithmique**



**Table des matières**

[1- Pitch de l’application 3](#_Toc1292171782)

[2- Pitch du problème 4](#_Toc775255960)

[3- Maquettes et/ou Schémas Illustratifs 5](#_Toc1539455168)

[4- Informations manipulées 5](#_Toc1771648700)

# 1- Pitch de l’application

BonPlan&Co est une application web développée par des étudiants pour les étudiants du BAB. Nous vous proposons une plateforme simple d'utilisation offrant une multitude de fonctionnalités.

Tout d’abord, vous pourrez rechercher des restaurants et des activités de divertissement, tourisme, shopping, culture et sport à travers tout le BAB. Cela vous permettra de planifier à l’avance toutes vos futures sorties.

Il vous sera aussi possible de participer à des évènements à durée limitée, c’est l’idéal pour faire de nouvelles rencontres autour d’une passion commune.

Mais nous ne proposons pas que cela, BonPlan&Co permet aussi de vous aider à développer votre vie active en recherchant des offres d’emplois, de covoiturage ou de colocation !

Bien évidemment, chaque bon plan sera noté par les utilisateurs ayant déjà participé, ce qui permet de distinguer les plus belles pépites du BAB. Postez un commentaire et donnez votre avis pour informer les autres utilisateurs des meilleurs plans.

Pour vous permettre de vous repérer parmi toutes les offres disponibles, nous mettons en place une carte interactive répertoriant tous les bons plans proposés !

# 2- Pitch du problème

Le problème algorithmique complexe que nous avons choisi est un algorithme de recommandation. Il consiste à proposer des offres personnalisées aux utilisateurs de BonPlan&Co selon leur localisation et leurs dernières interactions avec l’application.  
En effet, si un étudiant effectue beaucoup de recherches pour trouver des restaurants et en visite souvent, il serait intéressant de lui en proposer de nouveau qui se situent près de chez lui.

Nous avons envisagé plusieurs solutions après avoir consulté le fonctionnement de divers algorithmes déjà existants.

**[Insérer des exemples d’algorithmes]**

Sérendipité > [Lien](https://medium.com/@alexis.anzieu/la-s%C3%A9rendipit%C3%A9-dans-les-algorithmes-de-recommandation-82131cb4e079) | Il nous a intéressés pour la part de hasard qu’il offre. En effet, offrir quelque chose susceptible d’intéresser l’utilisateur mais en même temps sortant un peu de ce qu’il a l’habitude de consulter peut-être intéressant.

User-based

Item-Based

Notre algorithme se baserait sur l’approche “Item-based” (car il regardera les activités/événements/pratiques de l’utilisateur) avec des stéréotypes d’utilisateurs fictifs regroupant plusieurs caractéristiques.

Tout d’abord, l’algorithme analyse tous les individus de la base de données pour essayer de remarquer des liens entre toutes les activités pratiquées par les individus.   
Il va ensuite créer des profils types auquel rattacher les individus.  
Après, il va déterminer à quel profil l’individu ressemble le plus et le classer dans ce groupe.

Ainsi, l'appli peut déduire quelles activités l'utilisateur est le plus susceptible d’apprécier.   
Il se basera sur les caractéristiques suivantes :

- Activités pratiquées

- Événement fréquentés

- Destinations de covoiturage

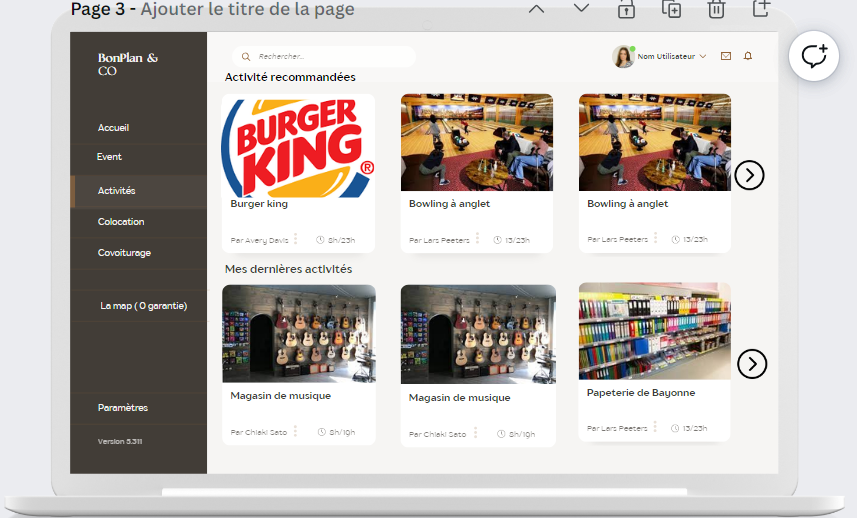
- Offres d’emploi cherchées

- Recherche

- Localisation

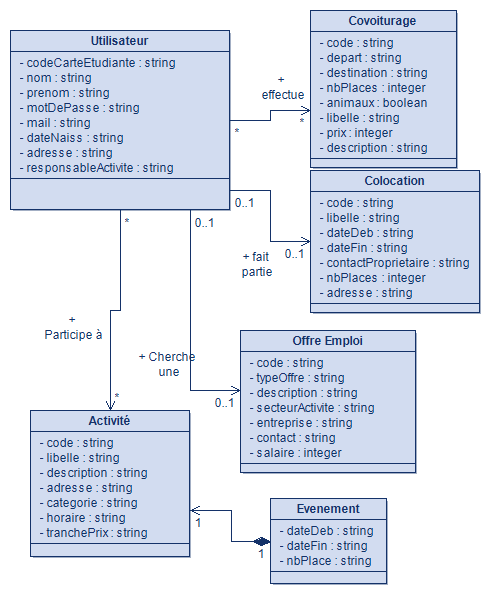
Une fois l’utilisateur classifié dans un des profils donnés, il sera possible de le cibler avec des publicités d’offres qui lui conviennent.

# 3- Maquettes et/ou Schémas Illustratifs



# 4- Informations manipulées

Diagramme de Classe (03/10/23)



Informations manipulées (Données+Résultats).

|  |  |  |
| --- | --- | --- |
| Nom | Description | Type |
| derniereActivite | Activité la plus récente pratiquée par l’utilisateur | Activite |
| dernierCovoiturage | Dernier voyage en covoiturage effectué par l’utilisateur | Covoiturage |
| dernierEvenement | Evénement le plus récent auquel a assisté l’utilisateur | Evenement |
| offreEmploi | Offre d’emploi consultée par l’utilisateur | offreEmploi |
| derniereRecherche | Dernière recherche de l’utilisateur | string |
| localisationUtilisateur | Position de l’utilisateur sur la carte | string / Coordonnées ? |

Réflexion sur les objets et classes :

|  |  |  |  |
| --- | --- | --- | --- |
| CLASSE Activité | | | |
| Nom attribut | Signification | Type | Exemple |
| codeActivite | Code unique à chaque activité | int | 06943 |
| nomActivite | Nom de l’activité | string | LaserGameAnglet |
| adresseActivite | Localisation de l’activité | string | 20 rue Pierre |
| typeActivite | Type de l’activité | enum | restaurant |
| detailActivite | Détails au sujet de l’activité | enum | thaï |
| prixActivite | Tranche de prix approximatif de l’activité | string | ‘20€-60€’ |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASSE Covoiturage | | | |
| Nom attribut | Signification | Type | Exemple |
| codeCovoiturage | Code unique à chaque covoiturage | string | ‘06943’ |
| nombrePlaces | Nombre de places disponibles pour le covoiturage | Unsigned short int | 3 |
| depart | Point de départ du covoiturage | string | 20 rue Pierre |
| destination | Destination du covoiturage | string | 2 avenue Larue |
| Prix | Prix du covoiturage en € | int | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASSE Evénement héritant de Activité | | | |
| Nom attribut | Signification | Type | Exemple |
| codeEvenement | Code unique à chaque événement | string | ‘06943’ |
| activiteLiee | Ce qui est pratiqué durant l’événement | activiteLiee | ... |
| dateDep | Heure de démarrage de l’événement | string | '20h30 21/11’ |
| dateFin | Heure de fin de l’événement | string | '21h30 21/11’ |

|  |  |  |  |
| --- | --- | --- | --- |
| CLASSE OffreEmploi | | | |
| Nom attribut | Signification | Type | Exemple |
| codeOffre | Code unique à chaque offre d'emploi | string | ‘06943’ |
| typeOffre | Catégorie d’offre d’emploi | enum | ‘CDD’ |
| detailOffre | Des détails sur l’offre comme le secteur | enum | ‘Informatique’ |
| nomEntreprise | Entreprise postant l’offre | string | Safran |
| emailEntreprise | Mail de l’entreprise pour la contacter | string | Info@safran.fr |
| salaire | Salaire en € par mois | int | 1200 |

***Algorithme***

Informations manipulées (Données+Résultats).

|  |  |  |
| --- | --- | --- |
| Nom | Description | Type |
| preferencesUtilisateurs | Liste de map les map sont définie par clé : nom de la catégorie (musique, culture) et comme valeur une liste de string représentant les choix de l’utilisateur | liste de map  de liste de  string |
| listePoids | Dernier voyage en covoiturage effectué par l’utilisateur | Liste de paire avec comme first le nom (string) de l’element et en second la valeur (int) en poid qu’il a par rapport au autres |
| Compteur ou compteurN | Sert a compter les différentes itérations de différentes boucle | Entier positive |
| Iterator ou it ou it\_XXX | Sert a parcourir les map et liste utiliser le nom change entre iterator si unique ou it\_ suivi d’un nom plus explicite ou encore juste it | Iterateur de liste ou de map |
| elemPref | On récupère l’élément préférer actuel pour calculer la persona | Liste de string |
| listeUtilisateurElempref | Liste des reponses utilisateur qui on elemenPref dans leur choix | Liste de map |
| poid tota | Un nombre représentant le poid totale | int |
| listCoresp | Liste de nombre quireprésenteront la corespondant que l’utilisateur a avec le persona | Liste d’int |