

S3.A.D - Pôle Développement

Rist

Projet 23

Spécifications Externes – Pôles C1&C2



BUT Informatique Semestre 3 2023-2024

Parcours D (IAMSI)

Gauthier GOUMEAUX - TD2 / TP3 (Alternant)

Julien LORIDANT - TD2 / TP3

Axel MARRIER - TD2 / TP3

Solène MARTIN - TD2 / TP3

Gabriel VERNIS - TD2 / TP3

Enseignant tuteur: NODENOT Thierry

Table des matières

Spécifications initiales	3
Caractéristiques :	3
Nombre d'utilisateurs	3
Priorité des critères	3
Ressources :	3
Classes	3
Fonctions PHP de récupération des données	3
Sous-programmes PHP de recommandations	4
Formule haversine	4
Fonction JavaScript de vérification du total des pourcentages	4
Scénarios:	4

Spécifications initiales

Caractéristiques:

Nombre d'utilisateurs

La page d'accueil présente 3 profils d'utilisateurs, il est possible d'ajouter, de changer ou de retirer des profils en manipulant la liste utilisateurs avec les pseudonymes de la base de données ('OlivManu', 'JPDubuis', 'Laulau64100', 'PatPat', 'Sparky').

Priorité des critères

Sur la page de recommandations, les valeurs des critères sont par défaut à 50% (Zone géographique), 30% (Prix) et 20% (Intérêts). Il est possible de les changer tant que le total est de 100. Des annotations sont là pour guider l'utilisateur dans la signification des critères.

Ressources:

Classes

Nous avons fait de la programmation orientée objet en réalisant deux classes :

- Utilisateur
- Activité

Elles contiennent toutes les données nécessaires au bon fonctionnement de notre algorithme de recommandations. Toutes les informations sont disponibles sur notre documentation Doxygen.

Fonctions PHP de récupération des données

Pour récupérer des informations de notre base de données et instancier les classes, nous avons développé plusieurs fonctions :

- **recupererCoordGPS**(*objet de type mysqli* lienVersLaBD, *string* pseudonymeUtilisateur) qui retourne un array de string contenant les coordonnées GPS de l'utilisateur;
- recupererGPSHistorique(objet de type mysqli lienVersLaBD, string pseudonymeUtilisateur) qui retourne un array de string contenant les coordonnées GPS des activités les plus récurrentes dans l'historique d'un l'utilisateur;
- **recupererBudgetSaisi**(*objet de type mysqli* lienVersLaBD, *string* pseudonymeUtilisateur) qui retourne un int correspondant au budget saisi par l'utilisateur lors de son inscription;
- recupererBudgetMoyen(objet de type mysqli lienVersLaBD, string pseudonymeUtilisateur)
 qui retourne un int correspondant au budget moyen de l'utilisateur calculé selon son historique d'activités;
- recupererCategoriesPref(objet de type mysqli lienVersLaBD, string pseudonymeUtilisateur)
 qui retourne un array de string contenant les catégories sélectionnées par l'utilisateur lors de son inscription;
- recupererCategoriesHist(objet de type mysqli lienVersLaBD, string pseudonymeUtilisateur)
 qui retourne un array de string contenant les catégories les plus récurrentes dans l'historique d'un l'utilisateur:
- recupererInfosPrincipalesActivite(objet de type mysqli lienVersLaBD) qui retourne un array d'objet Activite correspondant à toutes les activités disponibles;
- recupererCategories(objet de type mysqli lienVersLaBD, Int idActivite) qui retourne un array de string contenant les catégories de l'activité.

Sous-programmes PHP de recommandations

Nous avons développé trois fonctions de calcul de score, une fonction de vérification de similitude entre deux catégories et une procédure de tri des activités par leur score :

- calculScoreDistance(Float distanceUtilisateurActivite, Float distance, Float pourcentageZoneGeo) qui retourne le score calculé pour la distance entre l'utilisateur et l'activité en prenant en compte sa distance moyenne (ou saisie) et le pourcentage attribué;
- calculScorePrix(Float prixActivite, Float budgetUtilisateur, Float pourcentagePrix) qui retourne le score calculé pour le prix entre l'utilisateur et l'activité en prenant en compte sa son budget (moyen ou saisi) et le pourcentage attribué;
- calculScoreCategorie(array de string listeCategorieActivite, array de string listeCategorieUtilisateur, Float pourcentageCategories) qui retourne le score calculé pour la similitude entre les catégories de l'utilisateur et celles de l'activité en prenant en compte sa le pourcentage attribué;
- **categorieSimilaire**(*string* categorie1, *string* categorie2) qui retourne vrai si les deux catégories possèdent un ensemble en commun et faux sinon;
- **tri**(array d'objet Activite activites) qui effectue un tri à bulles décroissant des activités par leur score.

Formule haversine

Pour calculer la distance entre deux points à partir de leurs coordonnées GPS, nous utilisons la formule haversine en PHP :

haversineDistance(Float latitudePoint1, Float longitudePoint1, Float latitudePoint2, Float longitudePoint2) qui retourne la distance en km à vol d'oiseau entre les deux points.

Nous aurions voulu calculer la distance réelle, car évidemment que c'est plus représentatif, cependant toutes les API qui nous permettaient de le faire sont payantes.

Fonction JavaScript de vérification du total des pourcentages

A chaque clic de l'utilisateur pour rafraichir les résultats selon les pourcentages donnés appelle la fonction de vérification :

verifierTotal() qui return vrai si la somme des pourcentages saisis dans les inputs est égale à 100, sinon elle affiche une alerte demandant à la correction avec le résultat.

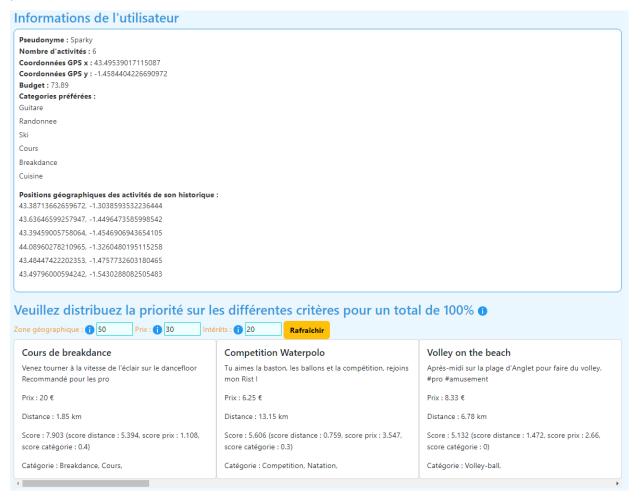
Scénarios:

Scénario nominal : L'utilisateur choisit un profil d'utilisateur dont il souhaite voir la page de recommandations

Sur la page d'accueil contenant 3 profils d'utilisateur variés, l'utilisateur choisit un profil.

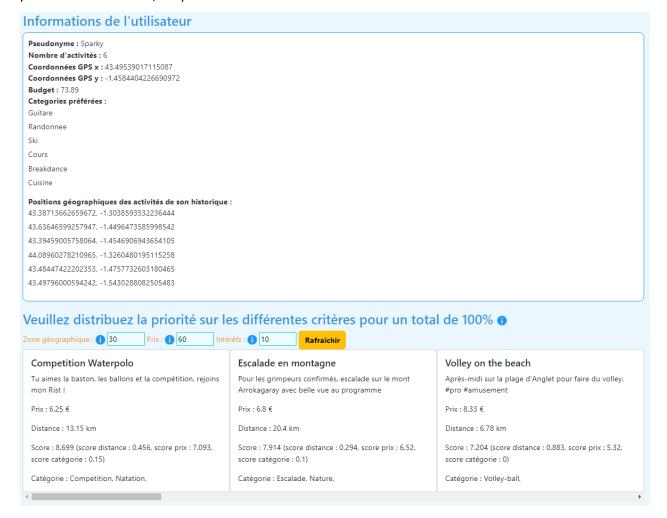


Il accède alors à la page de recommandation avec les activités triées selon les valeurs des priorités pré-saisies.



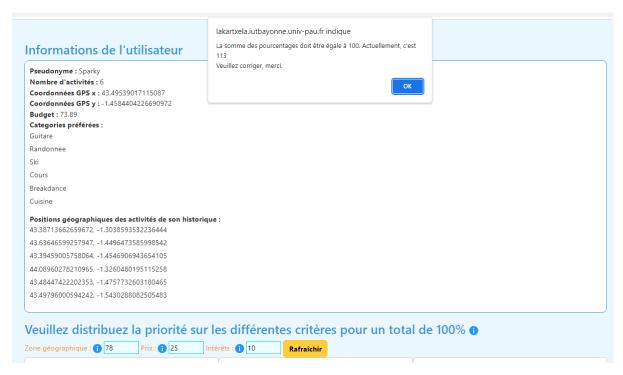
Scénario où l'utilisateur choisit un profil et modifie la répartition des pourcentages d'importance pour un total de 100%

L'utilisateur choisit un profil, accède à la page de recommandation et change les valeurs des priorités pour un total de 100%, ce qui affiche une liste d'activité recommandées nouvellement triée.



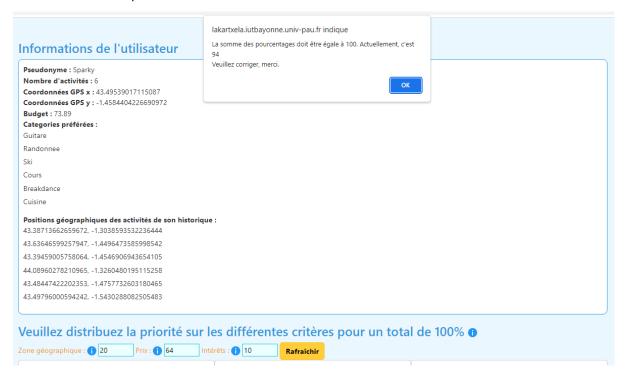
Scénario où l'utilisateur choisit un profil et modifie la répartition des pourcentages d'importance pour un total supérieur à 100%

L'utilisateur choisit un profil, accède à la page de recommandation et change les valeurs des priorités pour un total supérieur à 100%, un message d'erreur apparait avec la somme des valeurs saisies pour avertir l'utilisateur que son total est incorrect.



Scénario où l'utilisateur choisit un profil et modifie la répartition des pourcentages d'importance pour un total inférieur à 100%

L'utilisateur choisit un profil, accède à la page de recommandation et change les valeurs des priorités pour un total inférieur à 100%, un message d'erreur apparait avec la somme des valeurs saisies pour avertir l'utilisateur que son total est incorrect.



Scénario où l'utilisateur choisit un profil et modifie la répartition des pourcentages d'importance pour que l'un soit à 0%

L'utilisateur choisit un profil, accède à la page de recommandation et change les valeurs des priorités en en mettant une à 0% (donc qui ne compte pas), ce qui affiche une liste d'activité recommandées nouvellement triée, avec dans les détails, un critère de score à 0.

