

Application “CookingBooking”

La boîte à cuisine, Nancy

Présentation

Contexte

La « boîte à cuisine » est un des services de l'AARS (Association Accueil et Réinsertion sociale). Elle se veut « un lieu participatif d'ateliers culinaires solidaires, entièrement gratuit, destiné aux personnes isolées et touchées par les précarités alimentaires et/ou sociales ». Un des objectifs de ces ateliers est d'apprendre à préparer un repas quand on n'en a pas trop les moyens ou d'apprendre à cuisiner avec ce que l'on trouve dans son frigo. L'association a besoin d'un outil informatique pour recueillir les candidatures, avec critères adaptés, pour optimiser leur répartition dans les ateliers à thème (recette ou liste d'ingrédients). Le développement couvre 3 aspects : une application Web de mise en relation, une optimisation du placement des participants, un déploiement sur un serveur public.

Objectifs et décomposition de l'application

Il s'agit de proposer un outil répondant au besoin complet : développement, optimisation et déploiement. Naturellement, vos parcours respectifs favorisent vos contributions à l'un de ces aspects. Cela étant, vous êtes encouragés à ne pas hésiter à dépasser les thématiques qui sont au cœur de vos parcours :

- les différentes tâches ne sont pas forcément de même importance,
- des fonctionnalités et des comportements attendus sont décrits ; vous avez cependant des choix à réaliser, des technologies à sélectionner et il est préférable de les envisager en concertation,
- les tailles d'équipes sont assez conséquentes : la communication, la collaboration et le transfert de compétences seront certainement décisifs pour garantir une bonne progression de votre projet !

Dernières recommandations

Même si chaque partie fera l'objet d'une évaluation, l'appréciation se fera également sur la chaîne complète de votre projet. Dans cette perspective, veillez à ne pas trop déséquilibrer

ces parties, entraidez-vous. Le temps est limité, faites des choix pragmatiques et efficaces en utilisant des principes et technologies que vous connaissez et que vous êtes tous capables d'appréhender au sein de l'équipe. Arrangez-vous pour produire un prototype d'une chaîne complète !

Partie « Application Web »

Présentation générale des fonctionnalités

Les fonctionnalités concernent d'une part les utilisateurs (front-office) et les gestionnaires (back-office). On y retrouve les fonctionnalités classiques d'une application web : interfaces et expérience utilisateur, gestion de données, privilèges.

Les ateliers sont décrits sommairement : une thématique, une capacité (nombre de personnes maximum), un créneau (qui peut être « mardi 5 mars midi » ou « 5/3/24 12:00 », on peut se restreindre à une simple chaîne de caractères). Une table de correspondance entre des codes atelier et leur libellé est fournie. Pour les membres, il est nécessaire de considérer les nom, prénom, adresse mail. Enfin, les candidatures correspondront aux vœux ordonnés d'un membre sur une liste d'ateliers.

L'esprit de cet événement est de produire un démonstrateur, un prototype. Pour éviter de devoir gérer l'authentification, on pourra, dans un premier temps, disposer d'une URL pour le back-office, destinée aux administrateurs (cette adresse ne nécessitera pas d'authentification). Il n'est pas utile de s'intéresser à la création de compte pour les utilisateurs : ils peuvent simplement saisir les quelques données les concernant à chaque nouvelle requête.

Description détaillée des fonctionnalités

On conseille de ne s'intéresser aux fonctionnalités étiquetées comme « avancées » qu'à partir du moment où l'application développée comporte déjà un ensemble de fonctionnalités de base cohérentes. Les fonctionnalités étiquetées « admin » ne peuvent être réalisées que les par les utilisateurs disposant de ce privilège.

Gestion des ateliers

1. Affichage de la liste des ateliers. Présentation des ateliers existants avec leurs caractéristiques. On souhaite afficher l'ensemble des ateliers à raison de 5 par page.
2. Création d'un atelier (*admin*). Permet la création d'un nouvel atelier avec toutes ses caractéristiques.
3. Filtrage (*avancé*) : recherche dans le catalogue par filtrage des ateliers sur une thématique.

Gestion des candidatures à un atelier

4. Formulation des vœux : après saisie des informations de l'utilisateur (nom, prénom, adresse mail), présentation des possibilités d'ateliers. Les utilisateurs choisissent et ordonnent les **thèmes** des ateliers auxquels ils souhaitent participer.
5. Calcul des affectations (*admin*) : lancement du calcul par sous-traitement à une brique logicielle (*cf.* partie « optimisation » décrite plus loin dans le sujet). Il sera nécessaire d'être attentif à la communication mise en place (modalité, protocole) autant pour envoyer les données (ou mettre à disposition) que pour récupérer les résultats. Il n'est pas interdit d'envisager un lancement de l'optimisation directement depuis la brique optimisation.
6. Validation des affectations (*avancé*, *admin*) : validation visuelle des affectations automatiques décidées par la brique d'optimisation. Une fois qu'une affectation est validée, le vœux correspondant est supprimé et ne peut plus participer à un prochain calcul d'affectation.
7. Affichage des affectations : affichage (public) des prochains ateliers avec les affectations d'utilisateurs (uniquement noms et prénoms).
8. Acceptation d'une affectation (*avancé*) : à partir de la liste des affectations précédente, un utilisateur peut accepter ou refuser une affectation qui lui est relative. S'il clique sur le lien, son adresse mail lui est demandée : si elle concorde avec celle fournie à l'inscription, l'action demandée est réalisée.

Gestion des ateliers (admin)

9. Affichage des plannings des ateliers. Présente toutes les informations de cet atelier (avant son déroulement). Mentionne la liste des participants.
10. Pointage des présents à un atelier. Permet le pointage des personnes inscrites à un atelier donné, qui vient de se dérouler.
11. État récapitulatif d'un atelier : présente toutes les informations de cet atelier (une fois qu'il s'est déroulé). Mentionne la liste des participants et leur présence effective ou non.

Partie « Optimisation »

Présentation générale du sujet

L'application d'optimisation récupère la liste des candidatures des participants et la liste des ateliers (via un répertoire partagé, un service web ou autre...). Elle a pour objectif d'optimiser la répartition des participants pour satisfaire le plus grand nombre, puis redonne un fichier avec la répartition.

Les ateliers sont décrits à travers des critères qui serviront aux affectations. Dans un premier temps, vous vous limiterez aux deux critères suivants supposés renseignés lors de la création des ateliers :

- (a) le thème de l'atelier choisi parmi les six thèmes suivants :

- (1) IT - cuisine italienne ;
- (2) FR - cuisine française ;
- (3) MEX - cuisine Amérique du Sud ;
- (4) JP - cuisine japonaise ;
- (5) GR - cuisine grecque ;
- (6) OR - cuisine orientale ;
- (b) le nombre de places disponibles dans l'atelier.

Les candidatures se font via l'application (cf interface Web). Lors de leur demande d'inscription à un atelier, les candidats aux ateliers précisent :

- un ordre de préférence sur les thèmes proposés (sous la forme d'une liste des thèmes dans l'ordre de préférence) ;
- le nombre d'ateliers auxquels ils veulent participer (dans la mesure des places disponibles).

L'application d'optimisation va ensuite affecter les candidatures en raisonnant sur les aspects suivants :

- elle va essayer de respecter du mieux possible les préférences de chaque candidat (en fonction des places disponibles) ;
- elle va chercher à répartir les places disponibles entre les candidats. Par souci d'équité, tous les candidats doivent être affectés au même nombre d'ateliers (à 1 près). Bien entendu, les affectations seront aussi limitées par les demandes du candidat (si un candidat a été affecté au nombre d'ateliers demandés, cette règle n'a plus besoin d'être respectée pour ce candidat).

Mesure de la qualité d'une affectation

Afin d'optimiser le procédé d'affectation, on propose une première mesure de la qualité d'une affectation en fonction des préférences des candidats et du nombre et des types d'ateliers disponibles. L'objectif de l'optimisation sera donc de construire l'affectation qui maximise le score proposé (ce qui permet d'être non ambigu sur ce qu'on appelle une bonne/meilleure affectation).

Le score d'une affectation repose sur plusieurs règles qui permettent de contraindre les affectations et de donner un score élevé aux affectations qui respectent certains principes.

Règle 1 (taille des ateliers limités) : un atelier ne peut pas avoir plus de participants que le nombre de places proposées dans l'atelier.

Règle 2 (pas de dédoublement) : une même personne ne peut pas participer deux fois au même atelier (c'est à dire au même atelier sur le même créneau, une personne peut participer deux fois à un atelier abordant le même thème si ce sont deux ateliers différents - par exemple l'atelier de cuisine japonaise du mardi et l'atelier de cuisine japonaise du mercredi).

Règle 3 (Gain par préférence) : Chaque affectation d'une personne à un atelier rapporte un certain nombre de points à la qualité de l'affectation (c'est-à-dire à son score). Ce nombre

de points dépend de la position de l'atelier dans l'ordre des préférences de la personne affectée. Ces points sont attribués selon le tableau suivant :

Ordre de préférence	1 ^{er}	2 nd	3 ^{ème}	4 ^{ème}	5 ^{ème}	6 ^{ème}
Points octroyés	10	8	6	4	2	1

Ainsi, si Hugues a mis le thème cuisine japonaise en premier, affecter Hugues à un atelier de cuisine japonaise rapporte 10 points.

Règle 4 (Utilités dégressives) : Pour éviter qu'une seule personne ne soit affectée à tous les ateliers, si une personne est affectée à plusieurs ateliers, le nombre de points reçus est dégressif.

Pour cela, on trie les ateliers de la personne selon son ordre de préférence :

- le premier atelier dans l'ordre de préférence rapporte un nombre de points correspondant au tableau de la règle 3 ;
- le second atelier rapporte un nombre de points correspondant au même tableau mais en ajoutant un rang de pénalité ;
- et ainsi de suite (le troisième atelier, deux rangs de pénalité) ;
- les rangs après le 6^{ème} rang rapportent tous un point.

Par exemple, Hugues a été affecté à trois ateliers : un atelier "cuisine orientale", un atelier "cuisine italienne" et un second atelier "cuisine italienne". Dans ses préférences, Hugues a placé "cuisine orientale" en second et "cuisine italienne" en troisième :

- "cuisine orientale" rapporte donc 8 points selon le tableau ci-dessus (car il est 2nd dans l'ordre de préférence de Hugues) ;
- "cuisine italienne" rapporte 4 point (il est au 3^{ème} rang de préférence d'Hugues, mais, comme c'est le second atelier de Hugues, il rapporte donc les points du 4^{ème} rang) ;
- le second atelier "cuisine italienne" rapporte uniquement 2 points (il est au 3^{ème} rang, mais comme c'est le troisième atelier de Hugues, il rapporte donc les points du 5^{ème} rang).

Règle 5 (tout le monde est servi) : Lorsqu'une personne n'est affectée à aucun atelier, l'affectation subit un malus de -10. Cela oblige à trouver des ateliers pour tout le monde (quitte à ce qu'une personne n'ait pas son premier choix).

Règle 6 (règle d'équité) : Lorsqu'une personne voit qu'elle a deux ateliers de moins qu'une autre personne alors que son nombre d'ateliers souhaité n'est pas atteint, l'affectation subit aussi un malus global de -10.

Par exemple, Hugues a demandé 4 ateliers et n'en a eu que 1 alors que Albert et Bertrand en ont eu 3 chacun. Cette affectation est globalement pénalisée de -10 (même si Hugues a moins d'ateliers que 2 autres personnes).

Exemple détaillé

Supposons la présence de deux ateliers :

- un atelier "cuisine japonaise" (JP) avec deux places disponibles

- un atelier “cuisine orientale” (OR) avec deux places disponibles

Trois candidats sont inscrits sur la plateforme:

- Albert
 - qui souhaite participer à 3 ateliers
 - qui a pour ordre de préférence JP, FR, OR, MEX, IT, GR
- Bertrand
 - qui souhaite participer à 3 ateliers
 - qui a pour ordre de préférence OR, JP, FR, IT, MEX, GR
- Celine
 - qui souhaite participer à 2 ateliers
 - qui a pour ordre de préférence JP, OR, FR, IT, MEX, GR

Si l'affectation des ateliers est la suivante

- Atelier JP
 - Albert
 - Céline
- Atelier OR
 - Albert
 - Céline

l'affectation rapporte

- pour Albert : +10 (JP - ordre 1) + 4 (OR - 2^{ème} atelier ordre 3 équivalent à ordre 4)
- pour Bertrand : -10 (pas d'atelier) -10 (car d'autres ont deux ateliers de plus que lui)
- pour Céline : + 10 (JP - ordre 1) + 6 (OR - 2^{ème} atelier ordre 2 équivalent à ordre 3)

Soit un total de 10 points.

Une changement d'affectation où Albert ne fait que l'atelier de cuisine orientale (OR) et Bertrand prend sa place à l'atelier de cuisine japonaise (JP) donnerait

- pour Albert : +6 (OR - ordre 3)
- pour Bertrand : +8 (JP - ordre 2)
- pour Céline : + 10 (JP - ordre 1) + 6 (OR - 2^{ème} atelier ordre 2 équivalent à ordre 3)

pour un total de 30 points.

Échanger les ateliers d'Albert et Bertrand permettrait même d'atteindre un score de 36 points.

Données fournies

Les données sont fournies sous la forme de fichier texte CSV (comma-separated values) respectant le format suivant :

- la première ligne contient simplement l'intitulé “ateliers”;
- les lignes suivantes contiennent les descriptifs des ateliers avec (1) un id, (2) un thème parmi les thèmes possibles et (3) le nombre de places disponibles;
- une ligne vide sépare la suite;
- une ligne contenant “candidats” introduit la liste des candidatures
- chaque candidature est ensuite décrite sur une ligne avec (1) l'id de la personne; (2) son prénom (supposé unique); (3) le nombre d'ateliers souhaités et (4) son ordre de préférence listant les thèmes dans l'ordre (du préféré au moins souhaité)

Les intitulés utilisés pour les thèmes sont les suivants:

- **IT** pour cuisine italienne ;
- **FR** pour cuisine française ;
- **MEX** pour cuisine d'Amérique du Sud ;
- **JP** pour cuisine japonaise ;
- **OR** pour cuisine orientale ;
- **GR** pour cuisine grecque.

Travail proposé

Le travail proposé est ouvert. Nous vous conseillons cependant de suivre la démarche ci-dessous pour avancer progressivement. Bien entendu, vous pouvez vous organiser comme vous le souhaitez pour travailler en parallèle.

(1) Mise en place du problème d'optimisation

Proposez dans un premier temps les structures de données utiles pour représenter le problème et une affectation des candidats aux ateliers. Proposer et implémenter ce qu'il faut pour mesurer la qualité d'une affectation.

Pour simplifier cette partie, vous pouvez débiter par implémenter une partie des règles de score (1 point par affectation d'une personne) puis implémenter de plus en plus de détails du calcul de score. Par exemple, utiliser les points par ordre de préférence, puis, ajouter les malus des personnes sans atelier, puis considérer les décalages de rang en fonction du numéro d'atelier, etc ...

Cela permet de disposer d'une première fonction de scoring pour travailler en parallèle sur l'algorithme d'optimisation.

(2) Algorithme d'optimisation

Proposez un algorithme cherchant à maximiser cette mesure de qualité. On vous conseille de commencer par un algorithme simple qui constituera une base de comparaison avec les algorithmes plus évolués que vous pourrez proposer dans un second temps.

(3) Directions possibles de travail

Parmi les pistes que vous pouvez utiliser pour votre programme d'optimisation, vous pouvez

- utiliser des approches gloutonnes comme point de départ (ou des algorithmes de type recuit simulé) ;
 - https://fr.wikipedia.org/wiki/Algorithme_glouton
 - https://fr.wikipedia.org/wiki/Recuit_simulé
- raisonner sur des approches de type backtracking ou branch and bound en remettant en cause des affectations déjà effectuées ;
 - https://fr.wikipedia.org/wiki/Séparation_et_évaluation
- utiliser les algorithmes de type mariage stable spécifiquement dédiés aux problèmes d'affectation ;

- https://fr.wikipedia.org/wiki/Problème_des_mariages_stables
- utiliser des algorithmes évolutionnaires.
 - https://fr.wikipedia.org/wiki/Algorithme_évolutionniste

(4) Livrables attendus

Le rendu attendu consistera :

- à décrire l'algorithme que vous avez utilisé (écriture de l'algorithme général, détails éventuels) en justifiant vos choix ;
- à programmer l'application correspondante qui sera sollicitée par l'application web (cf autres sujets) ;
- à donner quelques les affectations obtenues sur les données exemples fournies ;
- à donner le descriptif éventuel des différentes extensions que vous avez proposées en détaillant votre raisonnement.

Ces éléments pourront être présentés dans un rapport d'une page (ou deux si besoin) pour donner une vision synthétique de votre travail.

Extensions possibles

Le sujet proposé initialement est un sujet ouvert à de nombreux algorithmes et variantes. Néanmoins, si vous avez des résultats qui vous semblent pertinents sur les critères proposés, vous pouvez étendre le travail dans différentes directions:

- **Besoin en nouveaux ateliers** : à l'issue du processus d'optimisation, retourner en plus des affectations une liste des ateliers les plus utiles classés par ordre d'importance pour mieux répondre aux besoins.
- **Modification de la qualité d'une affectation** : la fonction de qualité d'une affectation fournie est un premier exemple. Si vous le jugez utile, n'hésitez pas à proposer d'autres mesures.
- **Régimes alimentaires** : pour le moment, la notion de régime alimentaire n'est pas prise en compte. Il est cependant possible d'ajouter ce critère dans les affectations. Il existe plusieurs régimes alimentaires possibles (par exemple, sans contrainte / sans gluten / végétarien / végan) et les ateliers et les candidats précisent cette information. Un candidat ne peut être affecté que dans un atelier correspondant à exactement à son régime alimentaire déclaré.
- **Hiérarchie de régimes alimentaires** : les régimes alimentaires ne sont pas stricts. Par exemple, si une place est disponible pour un atelier vegan, un végétarien peut y participer.
- **Créneaux et disponibilités** : ajouter des contraintes de disponibilités en croisant les créneaux des ateliers et les disponibilités des candidats indiquées lorsqu'ils s'inscrivent.
- **Critère supplémentaire** : ajouter un autre critère à considérer dans l'affectation.

Partie « Déploiement »

Présentation

Il s'agit de mettre en production la solution globale (application Web, brique logicielle d'optimisation, dépendances...) sur un serveur public. Le serveur peut être physique, virtuel ou dans le cloud (AWS, GoogleCloud, docketu...). Le déploiement doit être optimisé pour une mise en production (« poids » des images, sécurité). Il accueillera :

- Le composant de l'application web de gestion des candidatures et des ateliers,
- Le composant de l'application d'optimisation qui échange ou partage des informations avec l'application web (exploitation de la liste des candidats et de celle des ateliers, génération de répartition des candidats dans les ateliers).

Les 2 conteneurs doivent donc pouvoir s'échanger des données (web-service ou autre...) Par ailleurs, dans l'optique d'un déploiement automatisé, il est demandé de mettre en place une chaîne d'intégration continue (et ainsi d'exploiter les tests unitaires qui ont pu être définis) et de déploiement continu.

La solution globale doit être accessible avec une simple URL publique.

On pourra, après avoir traité les points précédents, envisager :

- l'ajout d'un load balancer sur le serveur de production,
- l'ajout d'un serveur SMTP s'il est intégré à certaines des fonctionnalités de l'application web (qui pourront donc être agrémentées). Un outil de type MailCatcher peut être intéressant à exploiter dans ce cadre.

Livrable

Le rendu attendu consistera :

- à décrire les choix que vous avez effectués en les justifiant ;
- à présenter l'architecture globale de ce que vous avez mis en œuvre (un schéma peut permettre de synthétiser cette information) ;
- à présenter les résultats obtenus, de façon qualitative et quantitative.