

Spécifications fonctionnelles de **l'application de réservation régionale**

Le Pourquoi

Nous souhaitons créer une version plus légère de notre système de réservation phare pour les organisateurs de compétitions locales et régionales. Cela devrait permettre à l'organisateur de se concentrer moins sur l'administration et plus sur la logistique de la compétition, en veillant à ce que les événements soient sûrs et efficaces pour tous.

L'application permettra aux clubs d'inscrire des athlètes aux compétitions organisées au sein de la division.

Le comment

Au lieu de demander aux athlètes de réserver directement auprès de l'organisateur de la compétition, les secrétaires des clubs pourront leur réserver des places. Seules les secrétaires auront accès à l'application et pourront voir combien de places il reste.

Actuellement, les clubs gagnent des points via la mise en place et le déroulement des compétitions. Chaque club peut voir son solde actuel et échanger des points pour inscrire des athlètes à de futures compétitions, à raison d'un point par inscription.

Chaque compétition aura un nombre limité d'inscriptions, et chaque club ne peut inscrire qu'un maximum de 12 athlètes.

Exemple de parcours :

1. Le secrétaire se connecte à l'application.
2. Le secrétaire identifie un événement à venir et clique sur celui-ci.
 - a) Il voit le nombre d'inscriptions disponibles et si d'autres peuvent être acceptées ou non.
 - b) Le secrétaire peut alors utiliser les points accumulés pour acheter des places pour le concours.
 - i. S'il y a suffisamment de points et de places disponibles, il devrait voir un message de confirmation. Les points utilisés sont alors déduits du compte du club.
 - ii. Il doit recevoir un message d'erreur si le club n'a pas assez de points, s'il essaie d'acheter plus de 12 places ou si le concours est complet.
3. Il peut consulter la liste des points des autres clubs.

Phase 0

1. Lancement du projet
2. Conférences téléphoniques avec des organisations :
 - a) Besoin d'un moyen pour les secrétaires de se connecter et de visualiser leur solde de points actuel.

- b) Il faut pouvoir utiliser des points pour acheter des places d'accès rapide.
- c) Il faut éviter la surréservation.
- d) Il faut un tableau indiquant les points pour tous les clubs (il y a eu des appels au favoritisme).

3. Utilisation de grilles en niveaux de gris pour la démonstration.

Phase 1

1. Les secrétaires des clubs de l'organisation pourront utiliser leur adresse électronique pour se connecter et consulter la liste des compétitions à venir.
 - a) Ils pourront ensuite sélectionner une compétition et utiliser leurs points pour acheter des places.
 - b) Ils devraient voir un message confirmant le nombre de places achetées, ou un message indiquant que le concours est complet. Les points utilisés doivent être déduits du total précédent.
 - c) Ils ne doivent pas pouvoir réserver plus de places que celles disponibles ou plus de 12 places dans une compétition (afin de garantir l'équité envers les autres clubs).
2. Les secrétaires des clubs pourront se déconnecter.

Phase 2

1. Par souci de transparence, il devrait y avoir un tableau public, en lecture seule, des totaux de points indiquant le nombre de points disponibles pour chaque club. Il ne devrait pas être nécessaire de se connecter au site pour voir la page.
2. Compte tenu du nombre d'utilisateurs potentiels, il faut réduire au minimum les temps de construction et de rendu ; il ne devrait pas falloir plus de 5 secondes pour récupérer une liste de compétitions, et pas plus de 2 secondes pour mettre à jour le total de points

Guide de développement

En standardisant la façon dont nous créons des projets et des branches dans GitHub, il est facile pour d'autres personnes de lire et de comprendre l'avancement des projets, qu'il s'agisse de l'assurance qualité, des utilisateurs ou d'autres développeurs.

1. Lorsque vous créez votre projet, ajoutez toujours un README. Cela vous aidera à expliquer l'objectif de votre projet. N'oubliez pas d'inclure des éléments tels que :
 - a) Des liens vers des ressources externes.
 - b) Des configurations particulières.
 - c) Les conventions de nommage.
 - d) Tout ce qui nécessiterait que quelqu'un vienne vous dire : "J'ai une question sur X."
2. La branche principale (master) est votre source de vérité (code fini). Tout le monde devrait pouvoir cloner à partir de la branche principale et faire fonctionner le projet correctement.

3. Si vous voulez ajouter une fonctionnalité, corriger un bogue ou tout ce qui pourrait enfreindre la règle n°2, créez une branche. Le nom de la branche doit citer le type d'ajout et suivre le format <fonctionnalité/bug/amélioration>/nom-descriptif. Par exemple, si vous trouvez un bogue, vous devez créer une branche appelée bug/nom-de-bogue, et le corriger (évidemment).

4. Le Testing :

- a) Il faut toujours faire des tests. Si vous ne l'avez pas testé, il est cassé.
- b) Nous sommes agnostiques au frameworks - vous pouvez utiliser le pytest, l'unittest ou le Morelia. Peu importe ce que vous utilisez, vous devriez pouvoir le lancer depuis la ligne de commande.
- c) Veuillez mettre tous les tests dans un dossier de test pour que cela soit facile à trouver.
- d) Tests unitaires >> tests d'intégration >> tests fonctionnels. Au moins, écrivez des tests unitaires.
- e) Notre objectif est de couvrir notre code avec au moins 60% de tests. Plus de tests signifie plus de sommeil.
- f) Essayez d'écrire deux fois plus de tests unitaires que de tests d'intégration ou de tests fonctionnels. Les tests unitaires nous permettent de savoir comment les choses fonctionnent et quand elles ne fonctionnent pas.
- g) Pour éviter les douleurs oculaires, veuillez regrouper les tests unitaires, d'intégration et fonctionnels dans des dossiers séparés. Vous en serez heureux.

5. La Performance :

- a) Même lors de la création d'un MVP (minimum viable product), veuillez à aller le plus vite possible et à écrire le meilleur code possible.
 - b) Utilisez Locust pour vous assurer que le temps de chargement ne dépasse jamais 5 secondes, et que les mises à jour ne prennent pas plus de 2 secondes.
 - c) Le nombre d'utilisateurs par défaut pour les tests de performance est de six utilisateurs.
6. Concentrez-vous sur un bogue/une fonction/une amélioration à la fois (un par branche) ! Travailler sur plusieurs choses peut vous faire mal à la tête. De plus, il est difficile pour le reste de l'équipe de savoir sur quoi ils peuvent travailler.
 7. Ne revenez dans la branche principale que lorsque tous vos tests sont réussis. N'oubliez pas que si ça ne marche pas, ça n'a pas sa place dans la branche principale.
 8. Lorsque vous êtes prêt pour la révision du code, créez une branche QA à partir de la branche principale. Cependant, contrairement aux autres branches, celle-ci ne devrait pas être fusionnée dans la branche principale.