

# Mission de ce Projet

## Scénario

Vous venez de terminer votre premier mois chez Güdlft, une société qui a créé une plateforme numérique pour coordonner les compétitions de force (deadlifting, strongman) en Amérique du Nord et en Australie.

Güdlft a commencé par organiser des compétitions pour les clubs locaux, a gagné en popularité et accueille désormais exclusivement des compétitions pour les entreprises de vêtements de fitness de marque. Peu après avoir célébré son cinquième anniversaire, la société s'est retrouvée critiquée sur les réseaux sociaux :

Les régionaux veulent bien soutenir @gudlft, mais on n'a pas les moyens ! Vous nous avez oubliés ? #noubliezpasvosracines

En réponse, Güdlft a mis sur pied une équipe appelée Regional Outreach chargée de créer une version plus légère (et moins coûteuse) de leur plateforme actuelle pour les organisateurs régionaux. L'objectif de l'application est de rationaliser la gestion des compétitions entre les clubs (hébergement, inscriptions, frais et administration). L'équipe a dressé une liste de spécifications fonctionnelles pour un prototype, réparties en plusieurs phases. Depuis votre arrivée, vous avez suivi Sam, le développeur principal, qui a mis en œuvre la phase 1 du prototype.

En arrivant à votre bureau un matin, vous recevez le courriel suivant de Sam :

De : Sam Osei

À : vous

Sujet : FW : Construction de QA v0.9.1

Hey,

Je viens de recevoir le rapport de QA pour la phase 1 du projet. Il y a plusieurs bogues, dont un qui fait planter l'application ! Malheureusement, je ne suis pas au bureau pour les prochains jours (un de mes enfants est tombé malade ce week-end). Je ne sais pas encore quand je pourrai être là dans la semaine. Pourriez-vous prendre en charge la mise en œuvre du projet ? Vous devrez régler les bogues de la phase 1 et mettre en œuvre les éléments de la phase 2 (j'ai ajouté le travail de la phase 2 et les bogues de la section "issues" du repo).

Vous devrez cloner et forker le repo et le mettre en place sur votre machine locale (tout ce dont vous avez besoin se trouve dans le fichier README). Ensuite, passez en revue les bogues dans la section des problèmes, puis essayez de reproduire les problèmes sur votre machine locale pour résoudre les bogues et ajouter la gestion des erreurs. Pour gagner du temps de configuration, nous utilisons Flask et JSON pour éviter d'utiliser une base de données. La plupart des outils dont vous aurez besoin se trouvent dans le fichier requirements.txt dans le repo, mais vous devrez installer Flask et notre framework de test préféré, pytest, ainsi que notre outil de test de performance, Locust.

Vous devrez également préparer un rapport de test et un rapport de performances, conformément au guide de développement à la fin des spécifications fonctionnelles ci-jointes. Veillez à suivre toutes les directives, car le QA nous reproche de ne pas respecter les normes. Vous devez tester de manière approfondie les résultats requis (à la fois les happy paths et les sad paths) pour toutes les fonctionnalités de l'application. Je vous encourage également à adopter une approche de TDD, car cela vous aidera à rationaliser votre travail.

Une fois que vous aurez terminé, nous ferons un examen de ce que vous avez dans la branche QA du code. Nous examinerons les rapports et la manière dont vous avez résolu les problèmes, nous examinerons votre code et nous testerons la couverture de la nouvelle fonctionnalité.

Merci ! ...Sam

Pièces-jointes : il faut impérativement bien analyser tout le code existant sur le Repositories  
suivant : [https://github.com/OpenClassrooms-Student-Center/Python\\_Testing/tree/master](https://github.com/OpenClassrooms-Student-Center/Python_Testing/tree/master)