

## Gestion d'une campagne de tests

### Objectifs du projet :

Nous souhaitons mettre à disposition de nos utilisateurs une plateforme permettant de gérer des campagnes de tests, des cas de tests, et consulter des statistiques sur les campagnes de tests.

### Contraintes d'accès:

Cette plateforme devra être accessible depuis un navigateur web.

IHM Responsive : fonctionnalités s'adaptent à différentes tailles d'écran mais pas aux tablettes et smartphones.

### Contraintes techniques :

Utilisation d'une version 17 du JDK a minima et des versions récentes de Spring et Angular.

### Fonctionnalités application web:

- Création d'un compte utilisateur (user, nom, prénom, adresse mail, adresse postale).
- Création d'une organisation
- Possibilité d'inviter d'autres personnes à rejoindre une organisation.
  - Le clique sur le lien d'invitation mène au formulaire de création de compte
- Seul le propriétaire de l'organisation a le droit d'inviter d'autres personnes, sauf s'il donne les droits PROPRIETAIRE à un membre.
- Un gestionnaire peut créer des projets dans l'organisation. On peut relier le projet à un projet GitHub.
- Un gestionnaire peut créer une suite de tests pour une organisation donnée
- Un gestionnaire peut versionner une suite de tests
- Un testeur peut saisir un cas de test associé à une suite de tests.
- Un cas de test a :
  - Un nom
  - Une description détaillée
  - Un lien avec une tâche (item) du backlog de projet GitHub
  - Des préconditions :
    - Autre test qui doit être passant
    - Un jeu de tests présent en base de données
    - Un utilisateur de test avec le bon profil qui existe
  - Des étapes :
    - Description des étapes à suivre (1., 2. Etc...)
  - Un résultat attendu
  - Un résultat obtenu
  - Un statut : succès, échec, en cours
  - Commentaires

- Un niveau de priorité : haut, moyen, bas
- Un type de test : fonctionnel, performance
- Auteur de la personne qui a créé le test
- Date de création
- Date de dernière modification
- Auteur de la personne qui a exécuté le test
- Date de l'exécution du test.
- Dans le cas où un cas de test est en échec une issue est automatiquement créée et associée au projet correspondant sur GitHub
- Un gestionnaire peut lancer une campagne de tests à partir d'une suite de tests.
  - Une campagne de tests a une date de début, une date de fin prévue, des testeurs associés qui recevront alors une notification de participation à une campagne de tests.
- Il est possible de lancer une campagne de tests à partir de n'importe quelle version d'une suite de tests.
- Un gestionnaire peut clore une campagne de tests avant la date de fin prévue en saisissant un motif : application indisponible, application instable, etc..
- Un profil quelconque peut consulter le rapport d'une campagne de tests d'une organisation à laquelle il appartient.
  - Un camembert présente le nombre de tests réalisés par testeur
  - Un autre camembert présente le nombre de tests passants, en échecs, en cours
  - Un dernier camembert présente le nombre d'issues créées / le nombre d'issues corrigées.

### Travail à réaliser :

- Réaliser les maquettes
- Définir avec le client toutes les règles métier
- Prioriser les divers cas d'utilisation
- Rédiger les spécifications fonctionnelles.
  - Réaliser le diagramme de cas d'utilisation avec les acteurs associés
  - Pour chaque fonctionnalité :
    - Reformuler le besoin du Client en précisant toutes les règles de gestion identifiées.
    - Afficher la copie d'écran de la maquette
    - Détailler l'ensemble des actions réalisables par le client et ce que cela produit
    - Détailler les exceptions à afficher lorsque l'utilisateur réalise une action interdite.
- Rédiger le document de conception
  - Réaliser un diagramme de classe métier
  - Réaliser un diagramme technique présentant l'architecture en couches.
  - Réaliser un diagramme entités-relations.
  - Architecture technique
    - liste des composants utilisés pour le BACK (librairies, frameworks)
    - liste des composants utilisés pour le FRONT
    - liste des logiciels utilisés sur les postes de développement :
      - version de langage
      - base de données utilisée
      - serveur de déploiement
    - Quelle architecture pour le développement : hébergement du code source ?  
Exemple : GitHub
  - Diagrammes de séquences pour le traitement de nuit
- Commencer à rédiger les scénarios de tests que vous saisissez dans votre outil de tests.
- L'intégration continue (théorie et mise en place sur le projet) sera étudiée ultérieurement avec un formateur spécialisé.