

## S.I. Plateforme de gestion de projets collaboratifs

### Objectifs du projet

Le projet vise à concevoir, avec UML, une application de gestion de projets collaboratifs et agiles basés sur la méthode SCRUM. Cette application doit permettre :

- De créer son compte
- De créer une équipe avec attribution de rôle :
  - o Membres
  - o Propriétaires
- De créer un projet pour une équipe donnée avec attribution de rôles :
  - o Scrum master
  - o Product owner
  - o défaut
- D'inviter une personne à rejoindre une équipe
- De créer les sprints, les backlogs de sprint, les tâches
- Planifier les événements (demo, retrospective)
- Décrire les tests fonctionnels pour un use case donné.

### Contraintes d'accès

Cette plateforme est une application web accessible via un navigateur.

### Contraintes techniques

Utilisation de Java 17+

Frameworks et librairies :

- **Spring Boot**,
- **JPA / Hibernate**
- **OpenAPI** (ex Swagger) pour documenter l'API
- **Facultatif : Spring HATEOAS** sur au moins un contrôleur pour mettre en place une vraie architecture REST.

Base de données :

- MySQL

## Fonctionnalités application

- **Authentification**
- **Gestion des utilisateurs**

Un utilisateur inconnu peut créer son compte utilisateur (nom, prénom, pseudo, email, email confirmé : O/N) sur la plateforme

L'administrateur de la plateforme peut suspendre temporairement ou définitivement un utilisateur.

Un utilisateur connecté, qu'on appellera simplement profil utilisateur, peut modifier son compte.

- **Gestion des équipes**

Un utilisateur peut :

- créer une équipe avec un nom, un contact (email), une description.
  - o Il en devient alors le propriétaire.
  - o Un utilisateur appartenant à une équipe n'a que 2 rôles possibles : membre ou propriétaire.
- demander à rejoindre une équipe existante.
- quitter une équipe.

Un utilisateur propriétaire peut :

- modifier les informations d'une de ses équipes
- Inviter un autre utilisateur à rejoindre une de ses équipes (via un pseudo ou email)
- supprimer un membre d'une de ses équipes
- changer le rôle d'un membre au sein de son équipe : membre ou propriétaire

- **Gestion des projets**

Un projet a un nom, une date de début et date de fin, une description, un contact.

Un projet est découpées en epics, une epic en fonctionnalités, une fonctionnalité en user stories et une user story en tâches et/ou fiches d'anomalies. Globalement on appelle cela un item. Le découpage en autant de couches est facultatif. Seules les couches user stories et tâches sont obligatoires.

La liste des user stories s'appelle le backlog du projet.

Un projet peut être associé à une équipe ou à un utilisateur seul.

Un utilisateur, ou un propriétaire d'équipe, peut :

- créer un projet avec un nom, une date de début (date du jour par défaut), une date de fin, une description et un contact.
- mettre des liens vers un ou plusieurs dépôts distants (ex : GitHub)
- Rattacher un document à un projet
- Modifier un document rattaché à un projet
- Supprimer un document rattaché à un projet
- Saisir un commentaire sur le projet
- Répondre à un commentaire
- Attacher un emoji à un commentaire
- Gérer les utilisateurs associés au projet.

- Les utilisateurs doivent faire partie de l'équipe. Si un utilisateur n'a pas créé d'équipe cette fonction ne lui est pas accessible.
- définir un rôle pour chaque utilisateur du projet :
  - MEMBRE :
    - Rôle par défaut
    - peut créer modifier ou supprimer des tâches associées à une user story existante.
    - Peut créer modifier ou supprimer une fiche de bug associée à une user story
  - SCRUM MASTER :
    - peut créer modifier ou supprimer des tâches associées à une user story existante.
    - Peut créer modifier ou supprimer une fiche de bug associée à une user story
    - Peut créer/modifier/supprimer les sprints et les backlogs de sprints
    - Peut planifier des évènements pour un sprint : planning poker, demo, retrospective, autre
    - Peut planifier des évènements pour un projet : revue de projet, etc.
    - Renseigne durant les daily meetings l'estimation de l'effort restant sur chaque tâche en cours.
  - PRODUCT OWNER
    - Peut visualiser le backlog des sprints et l'avancée des développements.
    - Peut créer modifier ou supprimer une fiche de bug associée à une user story
    - Peut visualiser le planning des évènements prévus pour le sprint ou le projet
  - VISITEUR :
    - Personne autorisée, comme un responsable, à consulter certaines données du sprint : burndown chart, tableau d'avancée des tâches, compte-rendu de démo, planning des évènements.

### ➤ **Gestion du backlog du projet et des user stories**

Le scrum master, ou le product owner, d'un projet peuvent :

- Créer une user story avec un libellé, un code, une description et éventuellement des documents rattachés (facultatifs)

Le scrum master peut :

- Gérer les user stories rattachées à un projet, ce qu'on appelle le backlog d'un projet.

Les autres utilisateurs du projet peuvent :

- Consulter le backlog du projet avec user stories et découpage en tâches.

Le scrum master ou un membre peuvent découper une user story en tâches. Une tâche est constituée de :

- Un libellé, une description et éventuellement des documents rattachés
- Une estimation de sa difficulté exprimée en points d'effort (cf. & suivant)
- Le temps en heures consacré à sa réalisation complète

Le scrum master ou un membre peuvent rattacher une fiche de bug à une user story. Une fiche de bug a :

- Un libellé, une description et éventuellement des documents rattachés
- Un niveau de gravité (Cosmétique, mineur, majeur, bloquant)
- Une date de détection
- Un utilisateur à l'origine de la détection (cela peut être un externe)
- Une date de correction
- Un correcteur (membre du projet)
- Le temps exprimé en heures passé à corriger.

**Focus sur les points d'efforts** : on utilise une échelle non linéaire comme la suite de Fibonacci pour éviter les estimations trop précises et favoriser la réflexion :

- **1, 2, 3, 5, 8, 13, 20, 40, 100**

Chaque valeur représente un effort relatif :

- **1-2** : Très simple, trivial.
- **3-5** : Moyennement complexe.
- **8-13** : Complexe ou risqué.
- **20+** : Très complexe ou mal compris, à redécouper.

### ➤ Gestion des sprints

Le scrum master, ou le product owner, d'un projet peuvent :

- Créer un sprint avec un numéro (automatique), une date de début, une date de fin, une vélocité exprimée en nombre de points d'efforts/jour : estimation du nombre de points d'efforts que peut réaliser l'équipe par jour. Cet effort peut être estimé si le numéro de sprint est >1 sur la base du temps passé à réaliser les tâches lors des sprints précédents.
- Attacher des user stories au backlog du sprint
- Faire évoluer le workflow d'une tâche, ou fiche de bug, pour le sprint (quand toutes les tâches sont terminées, la user story est terminée)
- Modifier le workflow par défaut des tâches et fiches de bugs du sprint : A FAIRE, EN COURS, TERMINE
- Planifier un événement pour le sprint : avec une description et une date
  - Exemples d'événements : planning poker, demo, retrospective, autre
- Rattacher un document à un événement: comme un compte-rendu par exemple.

Les membres du projet peuvent :

- Consulter le backlog du sprint et les indicateurs comme la vélocité, la date de début et la date de fin.
- Consulter le burndown chart
- Consulter le tableau des tâches.
- Consulter la liste des événements dans le planning

## Travail à réaliser :

Réaliser un document de spécifications fonctionnelles contenant :

- Le diagramme de cas d'utilisation
- La description des user stories avec la liste des règles métier (i.e. contrôles à mettre en place côté API)

Réaliser un document de conception contenant :

- Le diagramme de classes
- Le diagramme entités relations.
- La liste des endpoints de l'API avec l'exemple de formalisme suivant :
  - POST /projects/:projectid/sprints : Créer un nouveau sprint pour un projet donné.
  - GET /projects/:projectid/sprints : Lister les sprints d'un projet.
  - GET /sprints/:id : Récupérer les détails d'un sprint.
  - PUT /sprints/:id : Modifier les informations d'un sprint.
  - DELETE /sprints/:id : Supprimer un sprint et tout ce qui lui est associé.