

# User stories

## Cas 1: consultation des projets pour un étudiant

Arthur, étudiant, souhaite consulter ses projets sur l'application « Student project dashboard ». Il s'identifie via son compte universitaire. Une fois connecté, il peut voir la liste de ses projets universitaires.

Une barre de recherche lui permet de rechercher un projet en particulier. Ainsi, il passe sa recherche. Il choisit de cliquer sur son projet « gl1\_compil\_tp1 », ce qui le dirige vers la page du projet.

De nombreuses informations sont affichées, comme les membres du projet (dont le professeur) avec leur e-mail (pour tous les membres) et leur groupe (pour les étudiants). Arthur voit également la fréquence, la répartition et la pertinence des commits. Enfin, il voit le résultat de l'analyse du code du projet, qui se focalise sur la qualité du code et le respect des normes. Il se rend compte qu'il y a de la duplication de code dans son projet et quelques bugs.

Arthur revient à la liste des projets en cliquant sur le bouton « Home » et s'intéresse maintenant au projet « gl1\_ase\_tp1 ». Une fois sur la page de ce projet, il s'aperçoit que tout va bien, et replie donc ce projet.

Il a fini ses consultations, il décide de corriger les erreurs présentes dans son projet de Compil. Une fois corrigées, il commit et push ses modifications et actualise l'application « Student project dashboard ».

Arthur déroule les détails du projet « gl1\_compil\_tp1 » et se réjouit de voir que les problèmes observés auparavant ont été corrigés.

## Cas 2: consultation des projets pour un professeur

M. Dupont, professeur à l'université, souhaite consulter les projets de ses étudiants sur l'application « Student project dashboard ». Il s'identifie via son compte universitaire. Une fois connecté, il peut voir la liste des projets universitaires auxquels il est affilié.

Une barre de recherche lui permet de rechercher les projets d'un étudiant en particulier. Il veut voir l'avancement des projets de Julie Medard. Ainsi, il passe sa recherche.

Il clique sur le premier projet de la liste (« gl1\_compil\_tp1 »), ce qui le dirige vers la page du projet. De nombreuses informations sont affichées, notamment la fréquence, la répartition et la pertinence des commits. Également, il voit le résultat de l'analyse du code du projet, qui se focalise sur la qualité du code et le respect des normes ou encore la présence de documentation.

M. Dupont se fait alors une idée de l'investissement dans le projet de chacune des étudiantes. La répartition des commits observée est de 9 commits pour Julie et 7 pour Anna. Toutefois, il voit que

certain commits de Julie ne sont que de légères modifications. Il estime donc que les deux étudiantes ont contribué raisonnablement au projet.

M. Dupont s'intéresse ensuite à l'analyse du code. Elle montre un manque de documentation et la présence de deux bugs dans le code. Cela lui permet de savoir plus précisément où regarder dans le code.

Une fois sa correction finie, il attribue une note au projet. Il peut maintenant s'occuper d'autres projets de la liste.

## **Cas 3: Se connecter**

Corentin Delpierre souhaite utiliser l'application « Student project dashboard ». En allant à l'adresse du site, il arrive sur une page de connexion. Il doit remplir deux champs : le login et le mot de passe, ceux-ci correspondant aux identifiants du LDAP de l'université. Il entre donc ses identifiants et arrive sur une page contenant la liste de ses projets.

## **Cas 4: Initialisation de l'application**

Pour pouvoir assurer le bon fonctionnement de l'application, il faut commencer par initialiser un certain nombre de données.

Tout d'abord, on s'intéresse aux identifiants de connexion. Nous utilisons le LDAP de l'université de Lille.

Ensuite, les projets git. Quand on professeur ou un étudiant se connecte pour la première fois, un processus de récupération des projets liés à cette personne se lance. Il vérifie pour chaque projet qu'il n'est pas déjà dans la base de données.

# Structure de la base de données

La base de données est utilisée pour stocker les informations des projets git. Un certain nombre de classes sont créées pour les représenter.

## Classes nécessaires

- Project (id, name, course, professor\_id, student1\_id, student2\_id, url, duplication, documentation, bugs)
- Person (id, name, surname, mail)
  - Student (group)
  - Professor (?)
- Project\_Student (project\_id, student\_id, commits)
- Commit (id, additions, deletions, date, message, student\_id)