Projet CSC4102 : « Suivi d'activité de projet »

Nom Prénom Étudiant1 et Nom Prénom Étudiants2

Année 2022–2023 — 1er février 2023

Table des matières

1 Spécification

1.1 Diagrammes de cas d'utilisation

Le diagramme suivant est à compléter. Ce commentaire est à retirer ensuite. Comme il y aura de nombreux cas d'utilisation, nous demandons que vous fassiez plusieurs diagrammes de cas d'utilisation afin d'avoir des diagrammes lisibles au format A4/portrait.

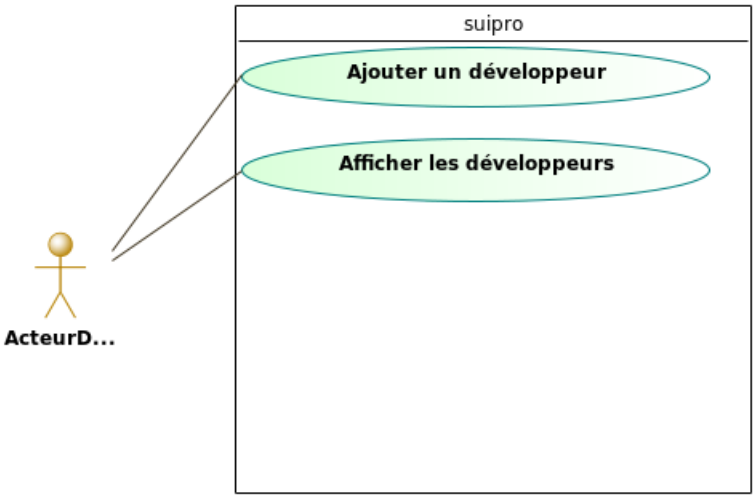

FIGURE 1 – Diagramme de cas d'utilisation — partie développeur.

1.2 Priorités, et préconditions et postconditions des cas d'utilisation

La liste de préconditions et postconditions suivante est à modifier et compléter. Ce commentaire est à retirer ensuite.

Voici les préconditions et postconditions des cas d'utilisation du premier sprint :

HAUTE n° 1
- Ajouter un développeur
  - précondition :
    ∧ alias du développeur bien formé (non `null` et non vide)
    ∧ nom bien formé (non `null` et non vide)
    ∧ prénom bien formé (non `null` et non vide)
    ∧ développeur avec cet alias inexistant
  - postcondition : développeur avec cet alias existant

Moyenne
- Lister les développeurs

2 Préparation des tests de validation

2.1 Tables de décision des tests de validation

La fiche programme du module CSC4102 ne permettant pas de développer des tests de validation couvrant l'ensemble des cas d'utilisation de l'application, les cas d'utilisation choisis sont de priorité HAUTE.

La section est à compléter avec les tables de décision d'autres cas d'utilisation. Ce commentaire est à retirer ensuite.

Numéro de test	1	2	3	4	5
Alias du développeur bien formé (non null et non vide)	F	T	T	T	T
Nom bien formé (non null et non vide)		F	T	T	T
Prénom bien formé (non null et non vide)			F	T	T
Développeur avec cet alias inexistant				F	T
Création acceptée	F	F	F	F	T
Nombre de jeux de test	2	2	2	1	1

TABLE 1 – Cas d'utilisation « ajouter un développeur »

3 Conception

3.1 Diagramme de classes

Le diagramme de classes suivant est à compléter. Ce commentaire est à retirer ensuite.

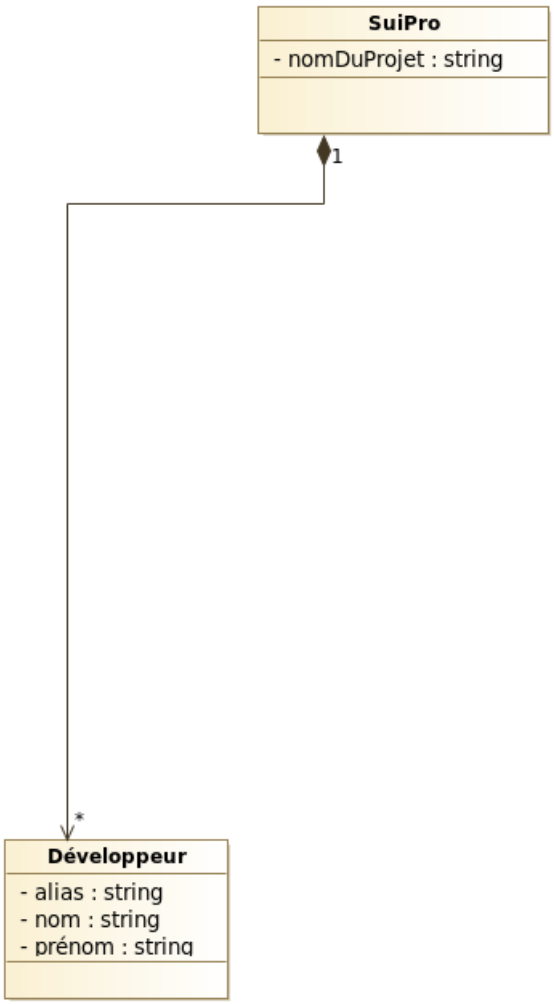

FIGURE 2 – Diagramme de classes.

3.2 Diagrammes de séquence

La section est à compléter avec les diagrammes de séquence de vos cas d'utilisation les plus importants, c'est-à-dire avec ceux de priorité haute. Ce commentaire est à retirer ensuite.

- arguments en entrée : l'alias du développeur, et le nom et le prénom du développeur
- rappel de la précondition : alias bien formé (non null et non vide) ∧ nom bien formé (non null et non vide) ∧ prénom bien formé (non null et non vide) ∧ développeur avec cet identifiant inexistant
- algorithme :
  1. vérifier les arguments
  2. chercher un développeur avec cet alias
  3. vérifier que le développeur est inexistant
  4. instancier le développeur
  5. ajouter le développeur dans la collection des développeurs

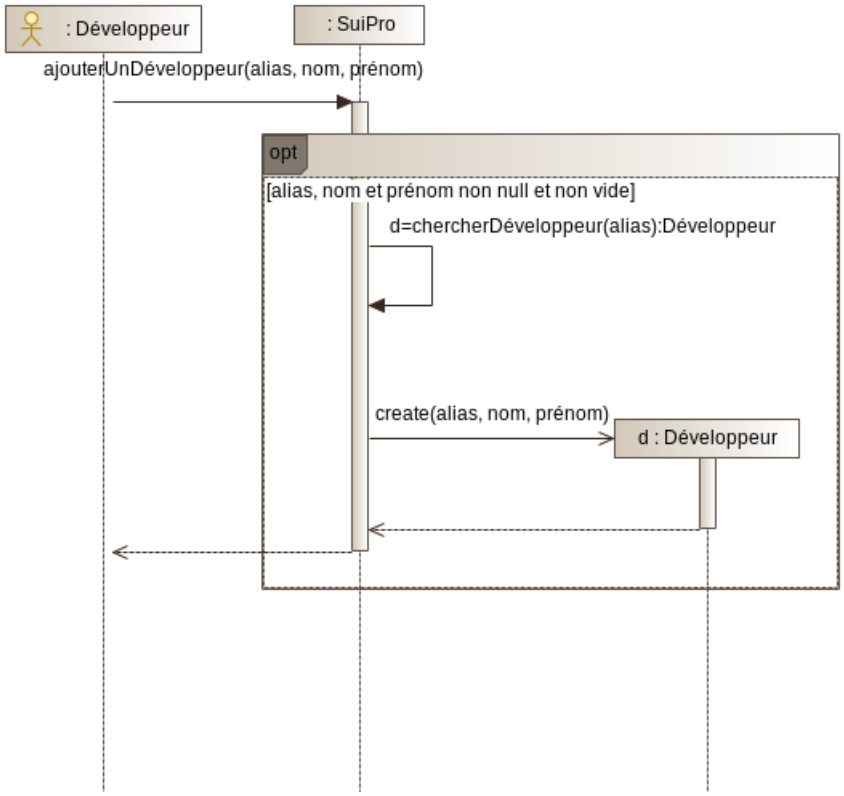

FIGURE 3 – Diagramme de séquence du cas d'utilisation « ajouter un développeur »

4 Diagrammes de machine à états et invariants

4.1 Classe Développeur

La section est à mettre à jour. Ce commentaire est à retirer ensuite.

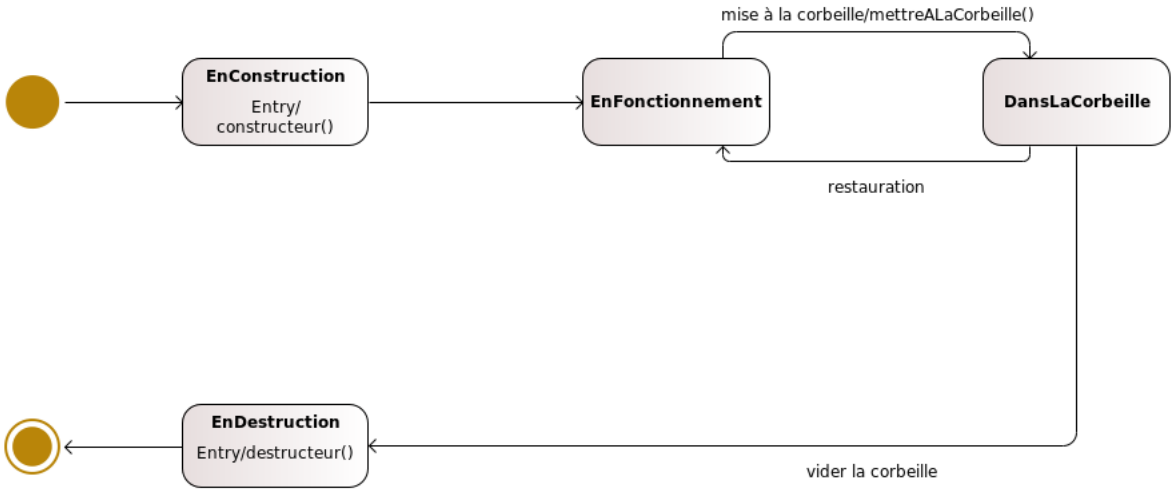

FIGURE 4 – Diagramme de machine à états de la classe `Développeur`

L'invariant de la classe Développeur est le suivant :

$\wedge$ alias $\neq$ null $\wedge$ alias $\neq$ vide
$\wedge$ nom $\neq$ null $\wedge$ nom $\neq$ vide
$\wedge$ prenom $\neq$ null $\wedge$ prenom $\neq$ vide

4.2 Classes Tâche

La section est à compléter. Ce commentaire est à retirer ensuite.

FIGURE 5 – Diagramme de machine à états de la classe `Tâche`.

L'invariant de la classe Tâche est le suivant :

5 Fiche des classes

5.1 Classe Développeur

La section est à mettre à jour. Ce commentaire est à retirer ensuite.

Développeur
<– **attributs « association »** –> – periodesDeTravail : Liste<PeriodeDeTravail> <– **attributs « modifiables »** –> – alias : String – nom : String – prénom : String
<– **opérations** –> + constructeur(String identifiant, String description) + invariant() : boolean + getIdentifiant() : String + getDescription() : String + afficher() : List<String>

5.2 Classe Tâche

La section est à compléter. Ce commentaire est à retirer ensuite.

6 Préparation des tests unitaires

6.1 Classe Développeur

La section est à mettre à jour. Ce commentaire est à retirer ensuite.

Numéro de test	1	2	3	4
`alias` $\neq$ null $\wedge \neg$vide	F	T	T	T
`nom` $\neq$ null $\wedge \neg$vide		F	T	T
`prenom` $\neq$ null $\wedge \neg$vide			F	T
alias$'$ = alias				T
nom$'$ = nom				T
prenom$'$ = prenom				T
invariant				T
Levée d'une exception	OUI	OUI	OUI	NON
Objet créé	F	F	F	T
Nombre de jeux de test	2	2	2	1

TABLE 3 – Méthode `constructeur` de la classe `Développeur`

6.2 Classe Tâche

La section est à compléter. Ce commentaire est à retirer ensuite.