|  |
| --- |
| 17 décembre 2017 |

|  |
| --- |
| **Dossier d’analyse et d’architecture** |
| Ingénieur Généraliste  SLR F23B  Encadrant : Jacques SIMONIN |

1, rue de l’Institut

75000 PARIS – France

Tél. +33 (0)1 00 00 00 00

Siret : 000 000 000 000 00

APE : 0000 A

[**www.mines-telecom.fr**](http://www.mines-telecom.fr)

Mines ParisTech

Mines Albi

Mines Alès

Mines Douai

Mines Nantes

Mines Saint Étienne

Télécom ParisTech

Télécom Bretagne

Télécom SudParis

Télécom Ecole de Management

Télécom Lille1

Eurecom

Chao CHEN

Junshuai ZHU

Sommaire

[1. Analyse Fonctionnelle 2](#_Toc501297607)

[1.1 Contexte 2](#_Toc501297608)

[1.2 Rôles 2](#_Toc501297609)

[1.3 Cas d’utilisation 3](#_Toc501297610)

[1.3.2 Cas d’utilisation UcCréerProjet 4](#_Toc501297611)

[1.3.3 Cas d’utilisation UcArrêterProjet 6](#_Toc501297612)

[1.3.4 Cas d’utilisation UcChangerContrat 8](#_Toc501297613)

[1.3.5 Cas d’utilisation UcAjouterContributeur 10](#_Toc501297614)

[1.3.6 Cas d’utilisation UcSupprimerContributeur 11](#_Toc501297615)

[1.3.7 Cas d’utilisation UcConsulterFicheProjet 13](#_Toc501297616)

[1.4 Entités participantes aux cas d’utilisation 15](#_Toc501297617)

[2. Architecture Fonctionnelle 16](#_Toc501297618)

[2.1 îlots fonctionnels 16](#_Toc501297619)

[2.2 Dépendance des îlot fonctionnels 17](#_Toc501297620)

[2.3 Données Fonctionnelles 45](#_Toc501297621)

[3. Architecture Applicative 47](#_Toc501297622)

[3.1 Composant applicatives 47](#_Toc501297623)

[3.2 Dépendance des composants applicatifs 48](#_Toc501297624)

[3.3 Données applicatives 50](#_Toc501297625)

# Analyse Fonctionnelle

Ce chapitre présente l'analyse des exigences fonctionnelles du système:

* Le contexte du logiciel définissant le système
* Les rôles interagissant avec le système
* Les cas d’utilisation du système et les scénarios qui les illustrent
* Le modèle d’entité associé aux cas d' utilisation

L'analyse fonctionnelle proposée ici complète une description d'acteur de la soumission d'un article et un modèle UML de données.

## Contexte

Figure 1 Contexte

## Rôles

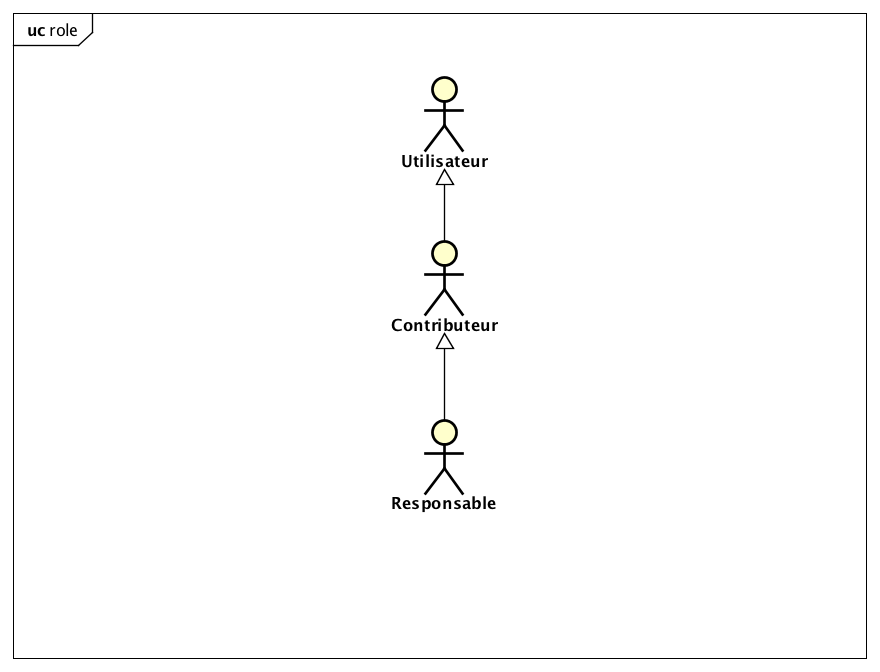
Les rôles sont définis dans le tableau Tableau 1 et modélisés avec UML dans la Figure 2.

Figure 2 Rôle

|  |  |
| --- | --- |
| Nom du rôle | Description du rôle |
| Utilisateur | Il peut être authentifié par son login et son mot de passe. Il a un champ de rôle. |
| Contributeur | Il contribue au projet. Il peut consulter le projet. |
| Responsable | Il est unique. Il gère tous les projets. |

Tableau 1

## Cas d’utilisation

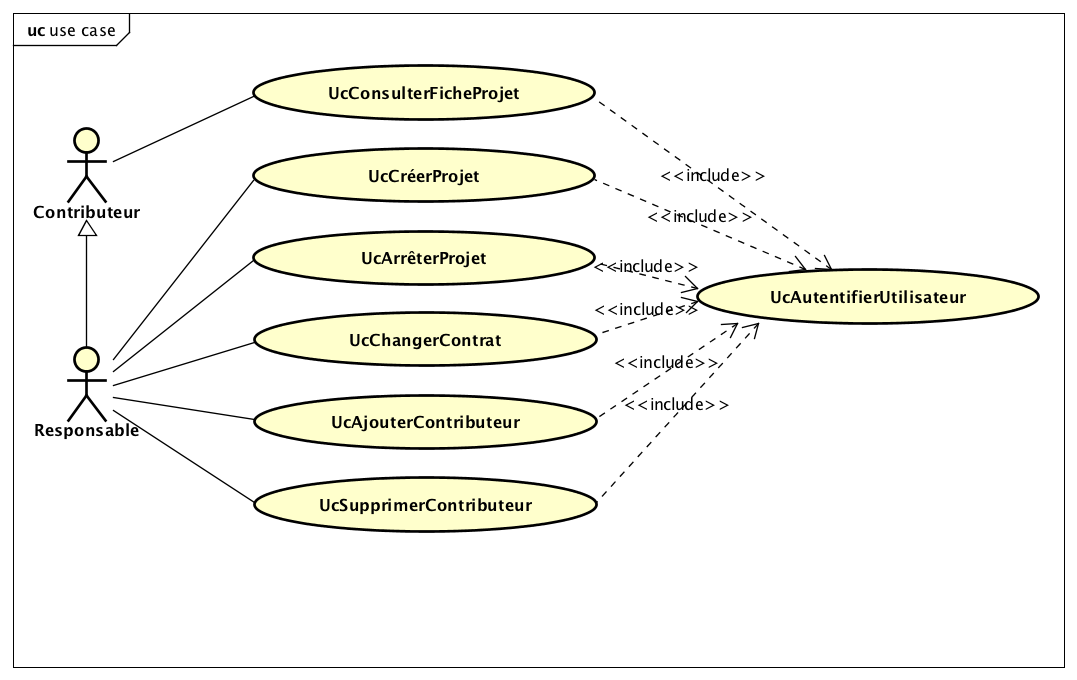
La Figure 3 représente le diagramme UML des cas d'utilisation comprenant les rôles qui interagissant avec cas d’utilisation et leurs relations d'inclusion et d’extension.

Figure 3 Cas d’utilisation

* + 1. **Cas d’utilisation UcAuthentifierUtilisateur**

**Résumé**

Ce cas d'utilisation permet d’authentifier le rôle d’un utilisateur du système.

**Contexte de déclenchement**

Le contexte est d’un cas d’utilisation secondaire utile à tous les cas d'utilisation de gestion de direction scientifique pour des projet.

**Rôle**

Utilisateur

**Pré-condition**

Aucune

**Description**

L’utilisateur s'authentifie avec son login et son mot de passe.

Le système renvoie son rôle (contributeur, responsable) ou refuse l'accès si l’utilisateur n’est pas authentifié.

**Post-condition**

Le rôle de l’utilisateur est connu.

**Exception**

L’utilisateur n’est pas authentifié.

**Scénarios**

1. Scénario nominal d’authentification d’un utilisateur

|  |
| --- |
| **ScAuthentifierUtilisateur** |
| 1. Contrôle du login et du mot de passe de l’utilisateur 2. Le système renvoi le rôle de l’utilisateur |

1. Scénario Exception d’authentification d’un utilisateur inexistant.

|  |
| --- |
| **ScExceptionAuthentifierUtilisateurInexistant** |
| 1. Contrôle du login et du mot de passe de l’utilisateur 2. Le système refuse l'accès |

### Cas d’utilisation UcCréerProjet

**Résumé**

Ce cas d'utilisation permet de créer un projet.

**Contexte de déclenchement**

Ce cas d’utilisation est exécuté à la demande de responsable.

**Rôle**

Responsable

**Pré-condition**

Aucune

**Description**

L’utilisateur s’authentifie son login et son mot de passe(cf. UcAuthentifierUtilisatuer).

S’il n’a pas le rôle de responsable alors le cas d’utilisation est abandonné.

Le responsable peut demander à créer un nouveau projet.

Le système vérifie que le projet n'existe pas déjà. Si ce n'est pas le cas, le cas d’utilisation est abandonné.

Le responsable doit donner un date de début et un data de fin pour un projet.

Le responsable ajoute des contributeurs avec ses charges dans ce projet.

Le système vérifie que le somme de charge est au moins 3 hommes par an.

Le responsable associer un contrat existant sur ce projet.

Le projet est identifié par le nom du projet.

Le système enregistre ce projet.

**Post-condition**

Un nouveau projet est créé

**Exception**

Un non-responsable demande à créer un projet

Un projet a été créé précédemment avec le même nom

La création d’un projet sans le donne la quantité de travail suffisante

La création d’un projet sans le donne le contrat existant

**Scénarios**

1. Scénario nominal de création d’un projet

|  |
| --- |
| **ScCréerProjetAvecResponsable** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Test positif de son rôle de responsable 3. Identification du projet par le nom du projet 4. Contrôle négatif de l'existence d’un projet ayant le même titre 5. Le responsable donne la date de début, la date de fin du projet et le contrat 6. Le responsable ajoute des contributeurs avec ses charges dans ce projet 7. Contrôle positif de la somme de charge est au moins 3 hommes par an 8. Contrôle positif de l’existence du contrat. 9. Enregistrement du projet |

2. Scénario d’exception signifiant l'impossibilité de créer un projet utilisant le rôle non-responsable

|  |
| --- |
| **ScCréerProjetAvecNonResponsable** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Test négatif de son rôle de responsable 3. Abandon de la création du projet |

1. Scénario d’exception signifiant l'impossibilité de créer un projet déjà existé

|  |
| --- |
| **ScExceptionCréerProjetExistant** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Identification du projet par le nom du projet 3. Contrôle positif de l'existence d’un projet ayant le même titre 4. Abandon de la création du projet |

1. Scénario d’exception signifiant l'impossibilité de créer un projet sans le donne la quantité de travail suffisante

|  |
| --- |
| **ScExceptionCréerProjetQuantitéNonSuffisant** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Identification du projet par le nom du projet 3. Contrôle négatif de l'existence d’un projet ayant le même titre 4. Le responsable ajoute des contributeurs avec ses charges dans ce projet 5. Le système vérifie que le somme de charge est au moins 3 hommes par an 6. Contrôle négatif de la somme de charge est au moins 3 hommes par a 7. Abandon de la création du projet |

1. Scénario d’exception signifiant l'impossibilité de créer un projet sans le donne le contrat

|  |
| --- |
| **ScExceptionCréerProjetSansContratExistant** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Identification du projet par le nom du projet 3. Contrôle négatif de l'existence d’un projet ayant le même titre 4. Le responsable ajoute des contributeurs avec ses charges dans ce projet 5. Le système vérifie que le somme de charge est au moins 3 hommes par an 6. Contrôle positif de la somme de charge est au moins 3 hommes par an 7. Contrôle négatif de l’existence du contrat. 8. Abandon de la création du projet |

### Cas d’utilisation UcArrêterProjet

**Résumé**

Ce cas d’utilisation permet d’arrêter un projet.

**Contexte de déclenchement**

Le cas d’utilisation est exécuté à la demande d’un responsable qu’il souhaite arrêter un projet.

**Rôle**

Responsable

**Pré-condition**

Aucune

**Description**

L’utilisateur s’authentifie son login et son mot de passe(cf. UcAuthentifierUtilisatuer).

S’il n’a pas le rôle de responsable alors le cas d’utilisation est abandonné.

Le responsable sélectionne un projet dans l’application par son nom.

Le responsable modifie la date finale de ce projet.

Le système enregistre la modification.

**Post-condition**

Le date finale de projet sélectionné est modifié à la date de le jour de la modification.

**Exception**

L’utilisateur n’a pas été authentifié comme responsable

Le projet est déjà arrêté.

Le projet n’existe pas.

**Scénarios**

1. Scénario nominal d'arrêt d’un projet

|  |
| --- |
| **ScArrêterProjetCourantAvecResponsable** |
| 1. Demande d'arrêt de projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable 4. Identification du projet 5. Vérification de la validation de date final de projet 6. Modification de la date de fin de projet 7. Enregistrement de la modification |

1. Scénario d’exception signifiant un utilisateur n’ayant pas le rôle de responsable

|  |
| --- |
| **ScExceptionArrêterProjetAvecNonResponsable** |
| 1. Demande d'arrêt de projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test négatif de son rôle de responsable 4. Abandon de la modification de la date de fin de projet |

1. Scénario d’exception d'arrêt d’un projet expiré

|  |
| --- |
| **ScExceptionArrêterProjetExpiré** |
| 1. Demande d'arrêt de projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable 4. Identification du projet 5. Vérification de la validation de date final de projet 6. Abandon de la modification de la date de fin de projet |

1. Scénario d’exception d'arrêt d’un projet inexistant

|  |
| --- |
| **ScExceptionArrêterProjetInexistant** |
| 1. Demande d'arrêt de projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable 4. Identification du projet 5. Abandon de la modification de la date final de projet |

### Cas d’utilisation UcChangerContrat

**Résumé**

Ce cas d’utilisation permet d’ajouter un contrat sur un projet.

**Contexte de déclenchement**

Le cas d’utilisation est exécuté à la demande d’un responsable qu’il souhaite ajouter un contrat sur un projet.

**Rôle**

Responsable

**Pré-condition**

Aucune

**Description**

L’utilisateur s’authentifie son login et son mot de passe(cf. UcAuthentifierUtilisatuer).

S’il n’a pas le rôle de responsable alors le cas d’utilisation est abandonné.

Le responsable sélectionne un projet par son nom.

Le responsable sélectionne un contrat par son nom.

Le responsable remplace le contrat de ce projet par le contrat sélectionné.

Le contrat est identifié par le nom du contrat.

Le système enregistre la modification de contrat sur ce projet.

**Post-condition**

Un contrat est changé sur le projet

**Exception**

Un non-responsable demande à ajouté un contrat.

Le projet n’est pas existant.

Le nouveau contrat n’existe pas.

**Scénarios**

1. Scénario nominal d’ajouter un contrat

|  |
| --- |
| **ScChangerContratAvecResponsable** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Test positif de son rôle de responsable 3. Identification du projet par le nom du projet 4. Identification du contrat par le nom du contrat 5. Remplacement de contrat sur le projet 6. Enregistrement de la modification |

1. Scénario d’exception signifiant l'impossibilité d’ajouter un contrat utilisant un rôle non-responsable

|  |
| --- |
| **ScExceptionChangerContratAvecNonResponsable** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Test négatif de son rôle de responsable 3. Abandon de la modification d'un contrat |

1. Scénario d’exception de changement du contrat sur un projet inexistant

|  |
| --- |
| **ScExceptionChangerContratSurProjetInexistant** |
| 1. Demande de changer le contrat sur projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable 4. Identification négative du projet 5. Abandon de la modification du contrat sur le projet |

1. Scénario d’exception de changement du contrat inexistant

|  |
| --- |
| **ScExceptionChangerContratInexistant** |
| 1. Demande de changer le contrat sur projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable 4. Identification positive du projet 5. Identification négative du projet 6. Abandon de la modification du contrat sur le projet |

### Cas d’utilisation UcAjouterContributeur

**Résumé**

Ce cas d’utilisation permet d’ajouter un contributeur sur un projet.

**Contexte de déclenchement**

Le cas d’utilisation est exécuté à la demande d’un responsable qu’il souhaite ajouter un contributeur dans un projet.

**Rôle**

Responsable

**Pré-condition**

Aucune

**Description**

L’utilisateur s’authentifie son login et son mot de passe(cf. UcAuthentifierUtilisatuer).

S’il n’a pas le rôle de responsable alors le cas d’utilisation est abandonné.

Le responsable peut demander à ajouter un contributeur dans un projet.

Le responsable choisit le contributeur et le projet.

Le système vérifie que le contributeur n'existe pas déjà dans ce projet. Si ce n'est pas le cas, le cas d’utilisation est abandonné.

Le contributeur est identifié par le nom du contributeur.

Le système enregistre ce contributeur dans le projet.

**Post-condition**

Un contributeur est ajouté

**Exception**

Un non-responsable demande à ajouter un contributeur.

Un contributeur a été ajouté précédemment avec le même nom.

Un contributeur est ajouté dans un projet inexistant.

Un contributeur n’existe pas.

**Scénarios**

1. Scénario nominal d’ajouter contributeur dans un projet

|  |
| --- |
| **ScAjouterContributeurAvecResponsable** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Test positif de son rôle de responsable 3. Identification du projet par le nom 4. Identification du contributeur par le nom du contributeur 5. Contrôle négatif de l'existence d’un contributeur existant dans le projet 6. Enregistrement du contributeur |

1. Scénario d’exception signifiant l'impossibilité d’ajouter un contributeur utilisant un rôle non-responsable

|  |
| --- |
| **ScExceptionAjouterContributeurAvecNonResponsable** |
| 1. Demande d’ajouter un contributeur sur le projet 2. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test négatif de son rôle de responsable 4. Abandon de l'ajout d'un contributeur |

1. Scénario d’exception signifiant l'impossibilité d’ajouter un contributeur déjà lié par le projet.

|  |
| --- |
| **ScExceptionAjouterContributeurDejadansProjet** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Test positif de son rôle de responsable 3. Identification du projet par le nom 4. Identification du contributeur par le nom du contributeur 5. Contrôle positif de l'existence d’un contributeur existant dans le projet 6. Abandon de l'ajout d'un contributeur |

1. Scénario d’exception d’ajouter un contributeur dans un projet inexistant

|  |
| --- |
| **ScExceptionAjouterContributeurDansProjetInexistant** |
| 1. Demande d'ajouter un contributeur sur un projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable 4. Identification négative du projet 5. Abandon de l’ajoute du contributeur sur le projet |

1. Scénario d’exception d’ajouter un contributeur Inexistant

|  |
| --- |
| **ScExceptionAjouterContributeurInexistant** |
| 1. Authentification d’utilisateur (cf. UcAuthentifierUtilisatuer) 2. Test positif de son rôle de responsable 3. Identification négative du contributeur par le nom du contributeur 4. Abandon de l'ajout d'un contributeur |

### Cas d’utilisation UcSupprimerContributeur

**Résumé**

Ce cas d’utilisation permet de supprimer un contributeur sur un projet .

**Contexte de déclenchement**

Le cas d’utilisation est exécuté à la demande d’un responsable qu’il souhaite ajouter un contributeur dans un projet.

**Rôle**

Responsable

**Pré-condition**

Le contributeur sélectionné travaille sur le projet sélectionné.

**Description**

L’utilisateur s’authentifie son login et son mot de passe(cf. UcAuthentifierUtilisatuer).

S’il n’a pas le rôle de responsable alors le cas d’utilisation est abandonné.

Le responsable sélectionne un projet par son nom.

Le responsable supprime un contributeur sur le projet.

Le système vérifie que ce contributeur existe dans ce projet. Si ce contributeur n'existe pas, alors le cas d’utilisation est abandonné.

Le système vérifie que le somme de charge est au moins 3 hommes par an. Si le somme de charge est moins que 3 hommes par an, alors le cas d’utilisation est abandonné.

Le système enregistre la suppression dans le projet.

**Post-condition**

Le contributeur sélectionné est supprimé sur le projet.

**Exception**

L'utilisateur n’a pas été authentifié comme responsable.

Le projet sélectionné n'existe pas.

Le somme de charge est moins que 3 hommes par an après avoir supprimé le contributeur.

**Scénarios**

1. Scénario nominal de supprimer un contributeur sur un projet.

|  |
| --- |
| **ScSupprimerContributeurAvecResponsable** |
| 1. Demande de supprimer un contributeur sur un projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable 4. Identification du projet 5. Suppression un contributeur sur ce projet 6. Vérification de la somme de charge est au moins que 3 hommes par an 7. Enregistrement de la suppression |

1. Scénario d’exception de supprimer un contributeur dans un projet inexistant

|  |
| --- |
| **ScExceptionSupprimerContributeurDansProjetInexistant** |
| 1. Demande de supprimer un contributeur de projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable 4. Identification négative du projet 5. Abandon de la suppression du contributeur sur le projet |

1. Scénario d’exception signifiant un utilisateur n’ayant pas le rôle de responsable

|  |
| --- |
| **ScExceptionSupprimerContributeurAvecNonResponsable** |
| 1. Demande de supprimer un contributeur sur un projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test négatif de son rôle de responsable 4. Abandon de la suppression de contributeur sur le projet |

1. Scénario d’exception de la somme de charge insuffisant

|  |
| --- |
| **ScExceptionChargeInsuffisant** |
| 1. Demande d'arrêt de projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable 4. Identification du projet 5. Suppression un contributeur sur ce projet 6. Vérification de la somme de charge est moins que 3 hommes par an 7. Abandon de la modification de la date de fin de projet |

### Cas d’utilisation UcConsulterFicheProjet

**Résumé**

Ce cas d’utilisation permet de consulter la fiche du projet.

**Contexte de déclenchement**

Le cas d’utilisation est exécuté à la demande d’un Contributeur qu’il souhait consulter la fiche d'un projet.

**Rôle**

Contributeur

**Pré-condition**

Aucune

**Description**

L’utilisateur s’authentifie son login et son mot de passe(cf. UcAuthentifierUtilisatuer).

S’il n’a pas le rôle de responsable ou contributeur alors le cas d’utilisation est abandonné.

Le contributeur peut consulter l’information un projet.

Le système le renvoie la liste du projet

Le contributeur choisit le projet qu’il veut consulter.

Le système vérifier l'existence du projet, si le projet existe le système le renvoie les informations correspond au projet. Sinon, le processus est abandonné.

**Post-condition**

L’information de la fiche est renvoyé

**Exception**

Le projet n’existe pas.

**Scénarios**

1. Scénario nominal de consulter un projet.

|  |
| --- |
| **ScConsulterFicheProjet** |
| 1. Demande d'arrêt de projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable ou contributeur 4. Identification du projet par le nom du projet 5. Contrôle positif de l'existence d’un projet 6. Renvoie tous les informations correspondent au projet |

1. Scénario d’exception signifiant l'impossibilité de consulter un projet qui n'existe pas.

|  |
| --- |
| **ScExceptionConsulterFicheProjetNonExistant** |
| 1. Demande d'arrêt de projet dans l'application DS 2. Authentification de l’utilisateur (cf. UcAuthentifierUtilisatuer) 3. Test positif de son rôle de responsable ou contributeur 4. Identification du projet par le nom du projet 5. Contrôle négatif de l'existence d’un projet 6. Abandon de la consultation du projet |

## Entités participantes aux cas d’utilisation

Les entités participantes aux cas d'utilisation sont définies dans le Tableau 2 et modélisées avec leur relation, conformément à UML dans la Figure 4.

|  |  |
| --- | --- |
| Nom de l’entité | Définition de l’entité |
| Utilisateur | Personne ayant un accès au système d’un rôle et identifiée par son login et son mot de passe |
| Contributeur | Personne travaillant sur les projets qui peux consulter les fiches de projet, donc il est identifié comme contributeur |
| Responsable | Personne gérant les projets dans ce système, donc il est identifié comme responsable |
| Charge | Relation entre des contributeurs et un projet ayant la quantité de charge sur un projet |
| Projet | Projet ayant les informations de dates, de contributeurs, de contrat relatifs |
| Contrat | Règle de projet |

Tableau 2

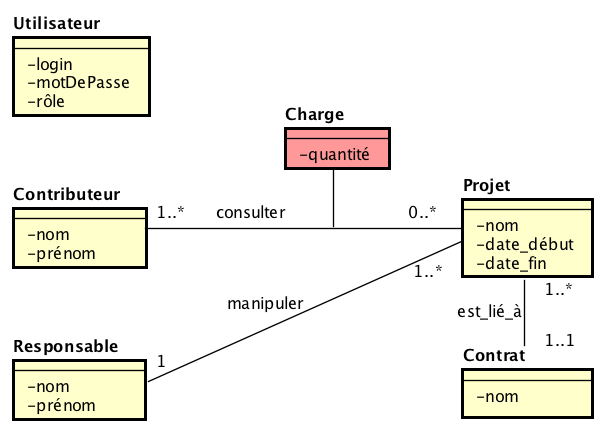


Figure 4 Diagramme des entités participantes aux cas d’utilisation

# Architecture Fonctionnelle

## îlots fonctionnels

La conception des îlots fonctionnels du système est déduite des entités participant aux cas d’utilisation(Tableau 3) et de leurs relations(Tableau 4). Elle aurait pu aussi déduite des cas d'utilisation.

|  |  |  |
| --- | --- | --- |
| Entité participante | Attribute | îlot fonctionnel |
| Contributeur | -nom | IF Gestion Personne |
| -prénom |
| Responsable | -nom | IF Gestion Personne |
| -prénom |
| Projet | -data\_debut | IF Gestion Projet |
| -date\_fin |
| -nom |
| Contrat | -nom | IF Gestion Contrat |
| Utilisateur | -login | IF Gestion Authentification |
| -password |
| -role |

Tableau 3

|  |  |  |
| --- | --- | --- |
| Relation entre entités  participantes | Entités reliées | îlot fonctionnel |
| est\_lié\_à | Projet | IF Gestion Contrat |
| Contrat |
| manipule | Responsable | IF Gestion Projet |
| Project |
| consulte | Contributeur | IF Gestion Projet |
| Projet |

Tableau 4

## Dépendance des îlot fonctionnels

La dépendance des îlots fonctionnels répertoriés précédemment est conçu à partir des scénarios spécifiés en analyse fonctionnelle. C'est une approche dynamique de l'architecture fonctionnelle. La traçabilité des scénarios dans les diagrammes de séquences fonctionnelles est reportée dans le tableau ci-dessous:

|  |  |
| --- | --- |
| Scénario | Diagramme de séquences fonctionnelles |
| ScAuthentifierUtilisateur | Figure 5 |
| ScExceptionAuthentifierUtilisateurInexistant | Figure 6 |
| ScCréerProjetAvecResponsable | Figure 7 |
| ScExceptionCréerProjetAvecNonResponsable | Figure 8 |
| ScExceptionCréerProjetExistant | Figure 9 |
| ScExceptionCréerProjetQuantitéNonSuffisant | Figure 10 |
| ScExceptionCréerProjeSansContratExistant | Figure 11 |
| ScArrêterProjetCourantAvecResponsable | Figure 12 |
| ScExceptionArrêterProjetAvecNonResponsable | Figure 13 |
| ScExceptionArrêterProjetExpiré | Figure 14 |
| ScExceptionArrêterProjetInexistant | Figure 15 |
| ScChangerContratAvecResponsable | Figure 16 |
| ScExceptionChangerContratAvecNonResponsable | Figure 17 |
| ScExceptionChangerContratSurProjetInexistant | Figure 18 |
| ScExceptionChangerContratInexistant | Figure 19 |
| ScAjouterContributeurAvecResponsable | Figure 20 |
| ScExceptionAjouterContributeurAvecNonResponsable | Figure 21 |
| ScExceptionAjouterContributeurDansProjetInexistant | Figure 22 |
| ScExceptionAjouterContributeurDejadansProjet | Figure 23 |
| ScExceptionAjouterContributeurInexistant | Figure 24 |
| ScSupprimerContributeurAvecResponsable | Figure 25 |
| ScExceptionSupprimerContributeurAvecNonResponsable | Figure 26 |
| ScExceptionSupprimerContributeurDansProjetInexistant | Figure 27 |
| ScExceptionChargeInsuffisant | Figure 28 |
| ScConsulterFicheProjet | Figure 29 |
| ScExceptionConsulterFicheProjetNonExistant | Figure 30 |

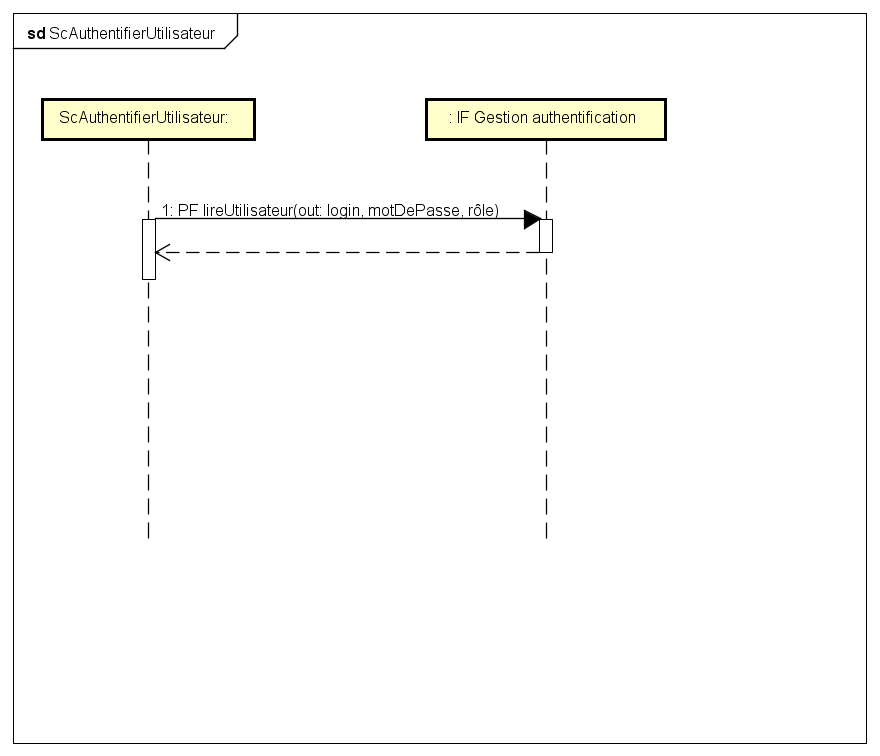


Figure 5 Diagramme de séquences fonctionnelles du scénario ScAuthentifierUtilisateur

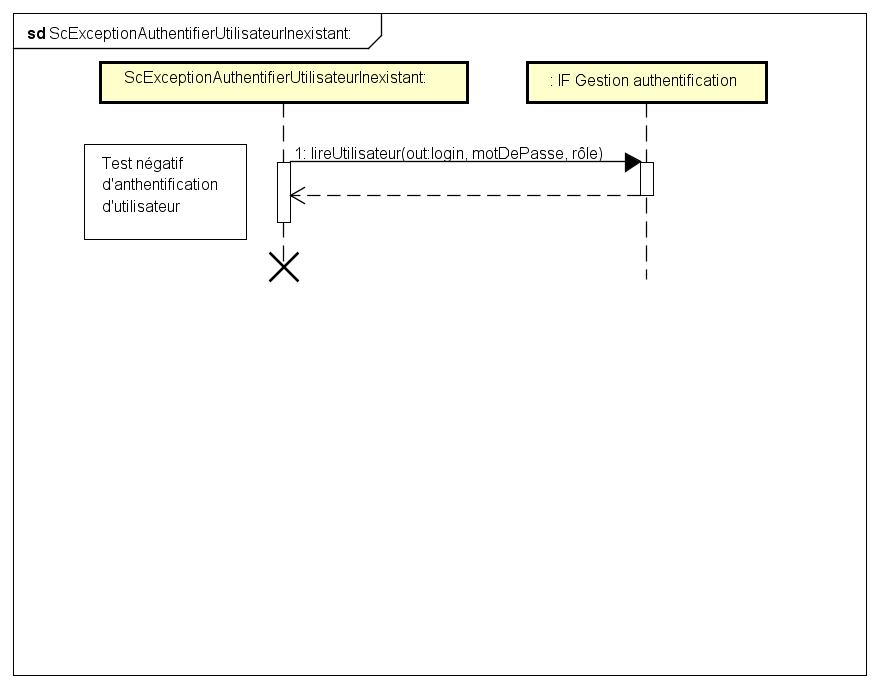


Figure 6 Diagramme de séquences fonctionnelles du scénario ScExceptionAuthentifierUtilisateurInexistant

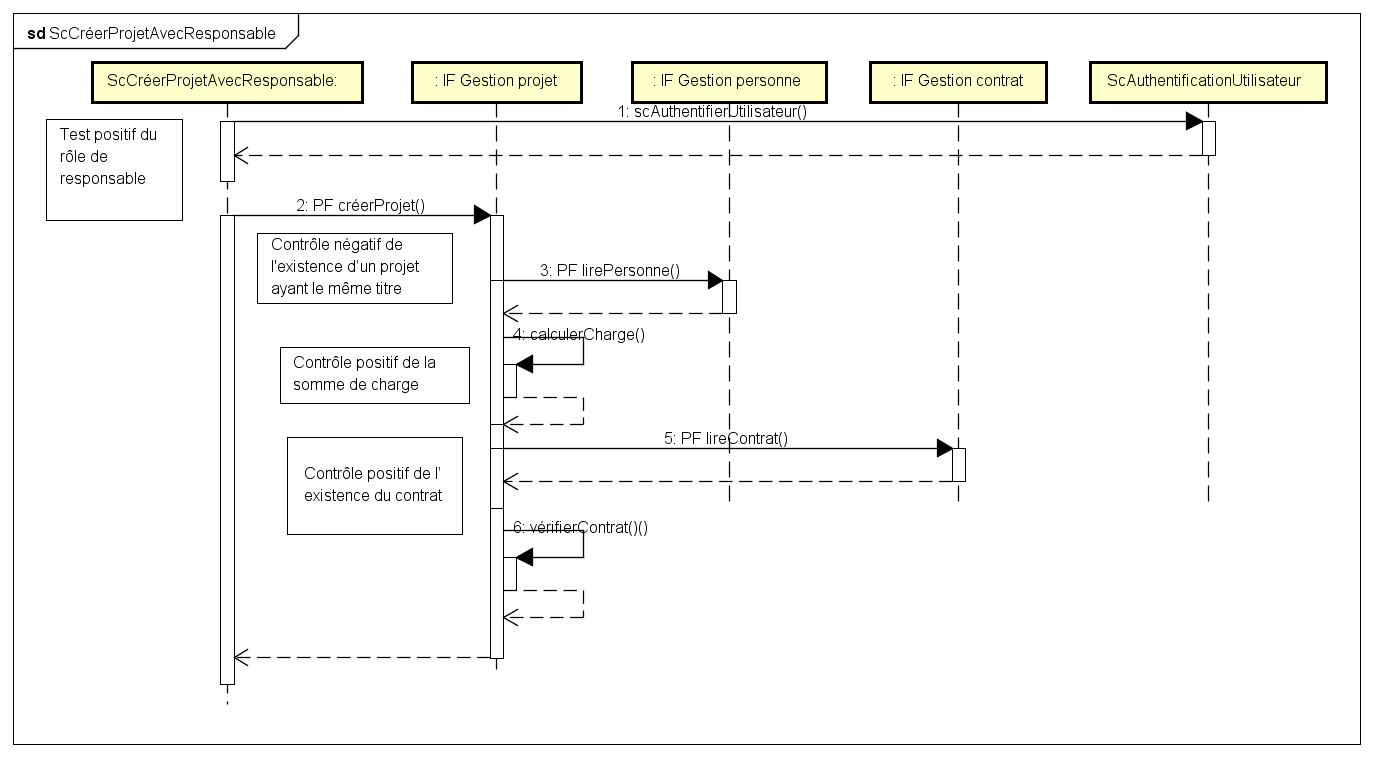
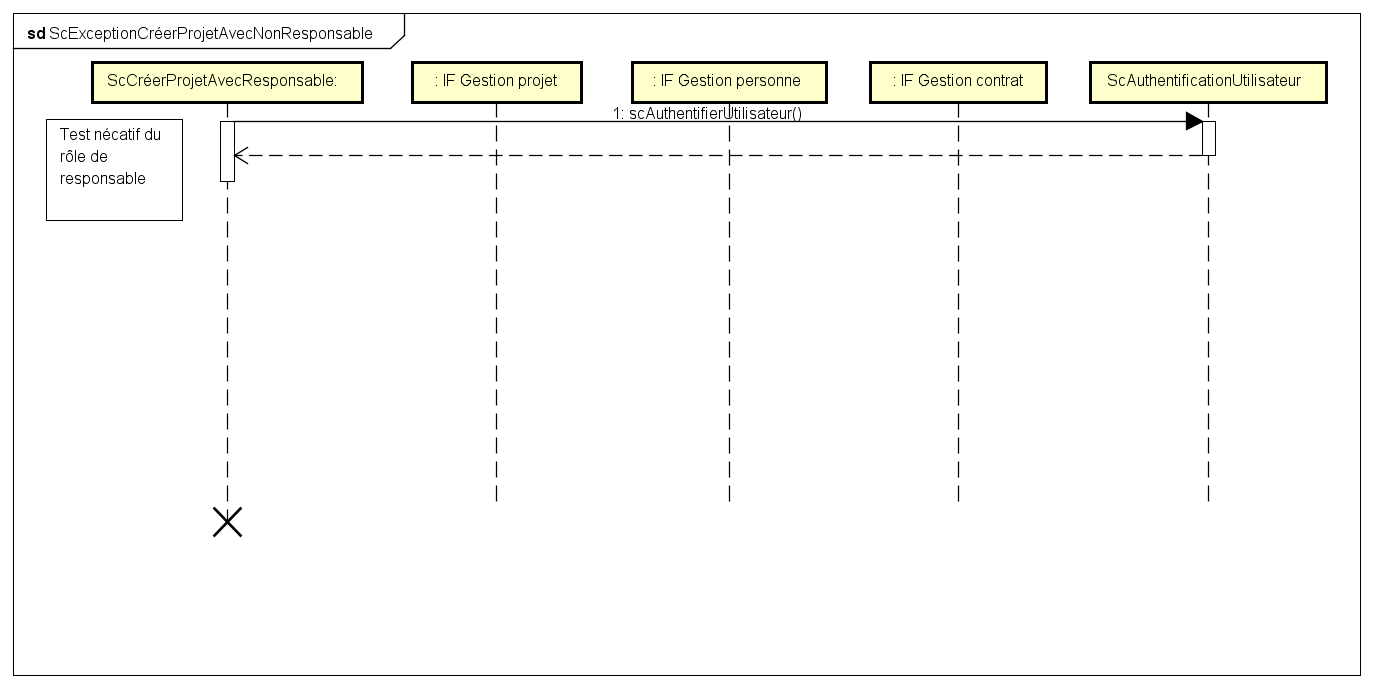
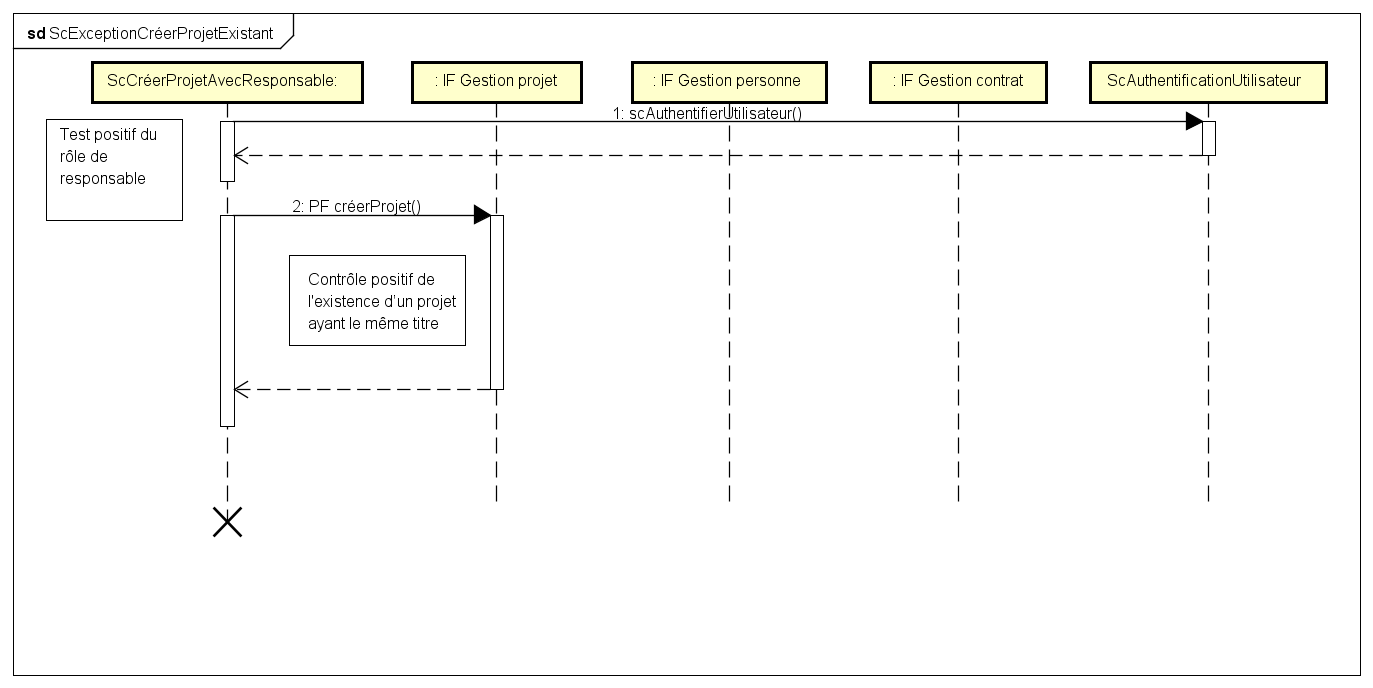
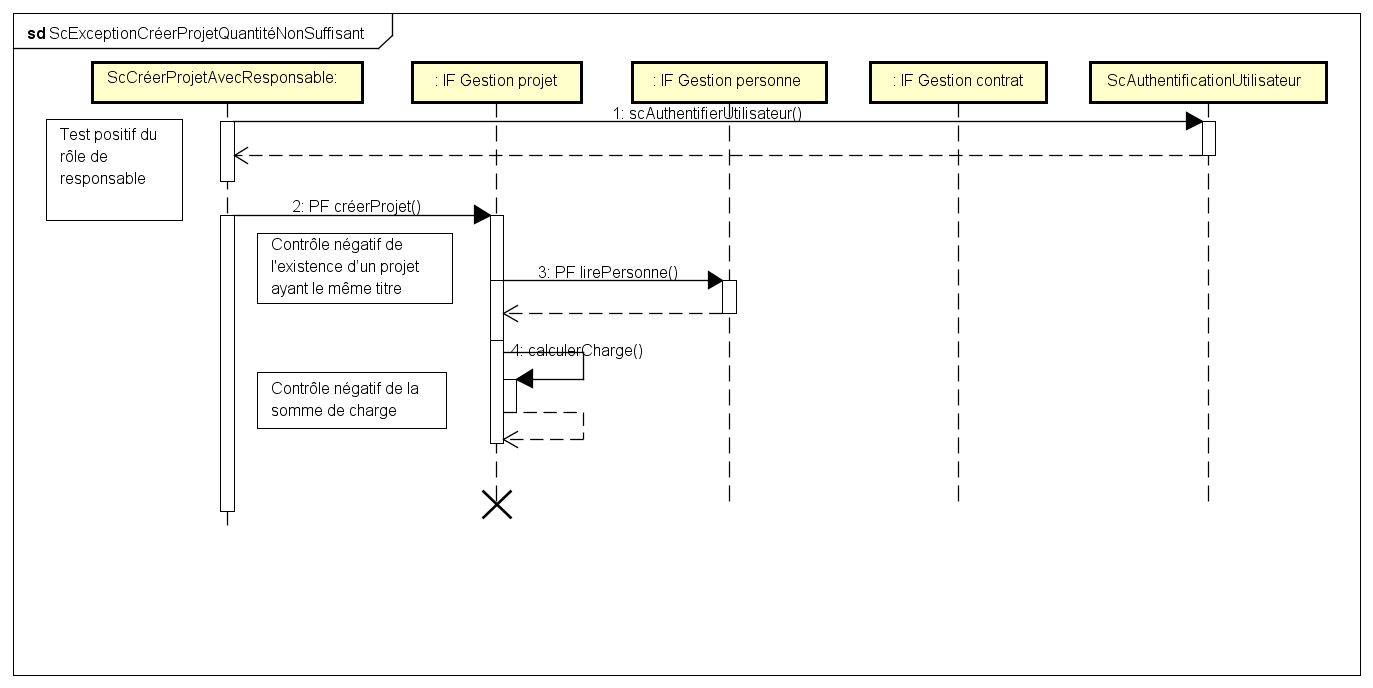
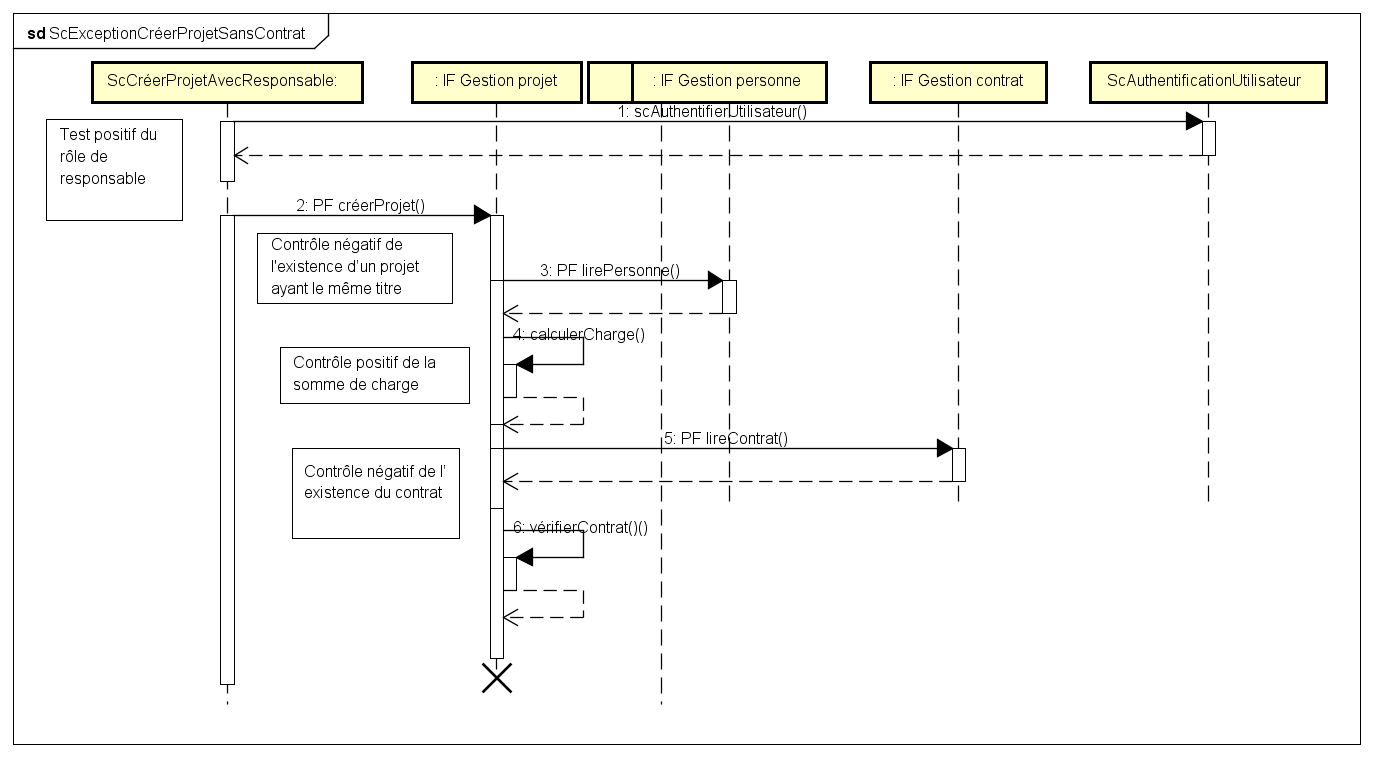
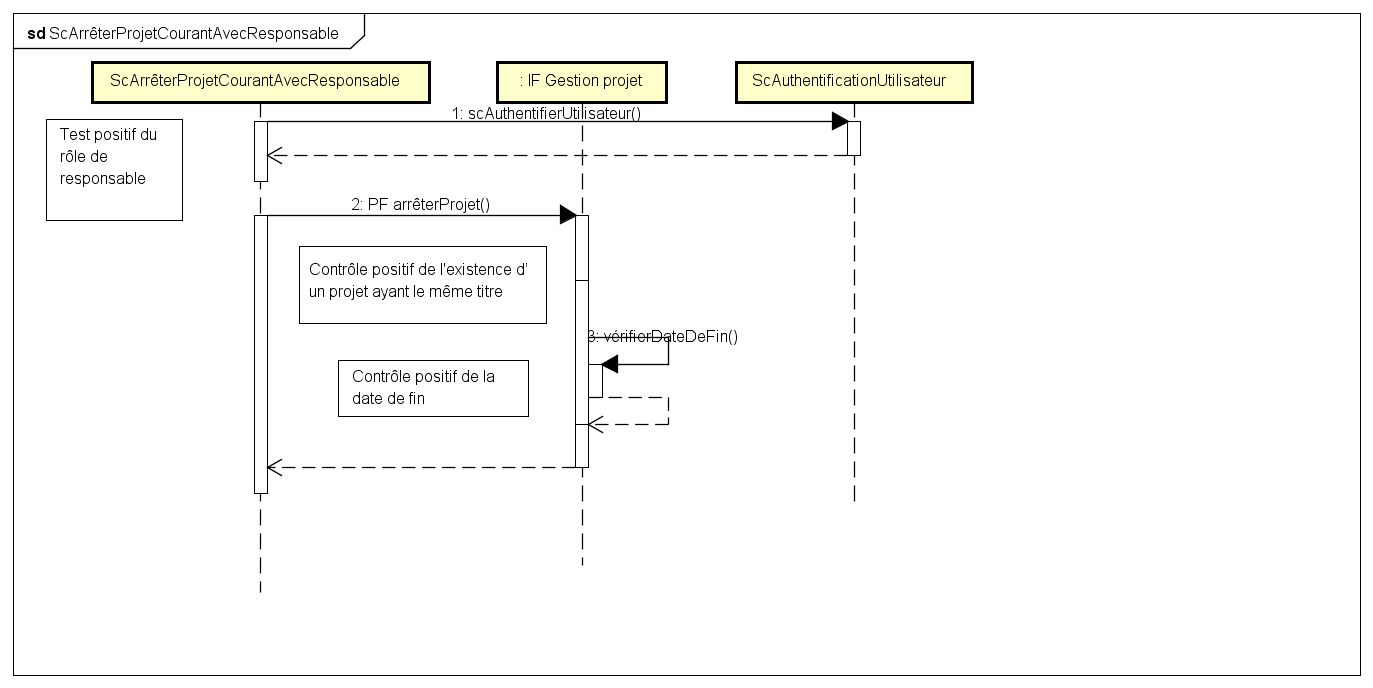
Figure 7 Diagramme de séquences fonctionnelles du scénario ScCréerProjetAvecResponsable

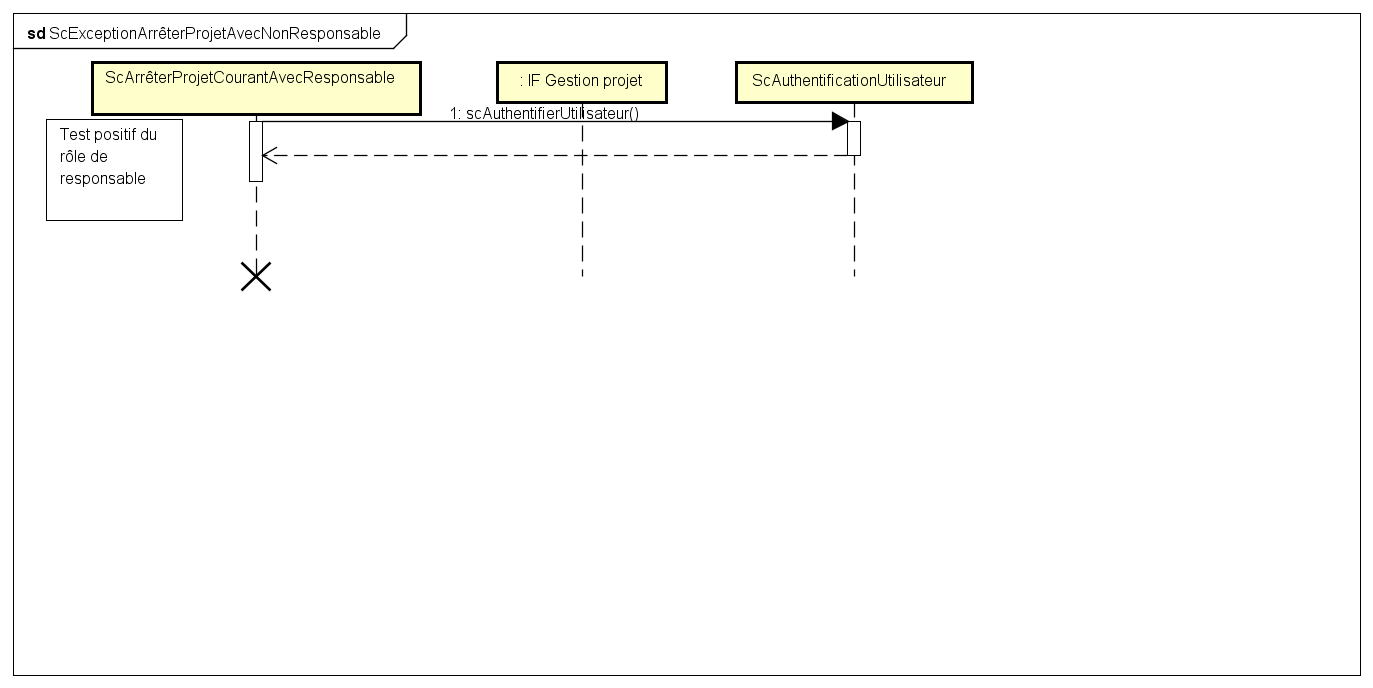
Figure 8 Diagramme de séquences fonctionnelles du scénario ScExceptionCréerProjetAvecNonResponsable

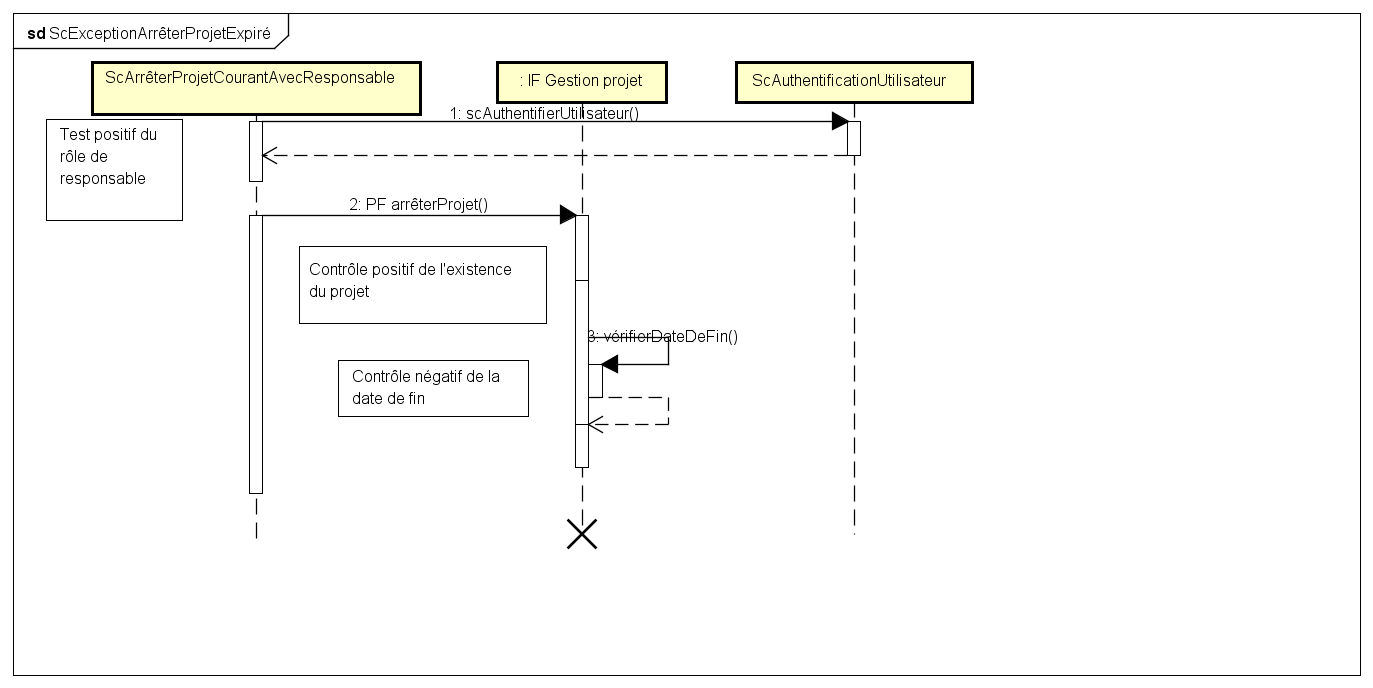
Figure 9 Diagramme de séquences fonctionnelles du scénario ScExceptionCréerProjetExistant

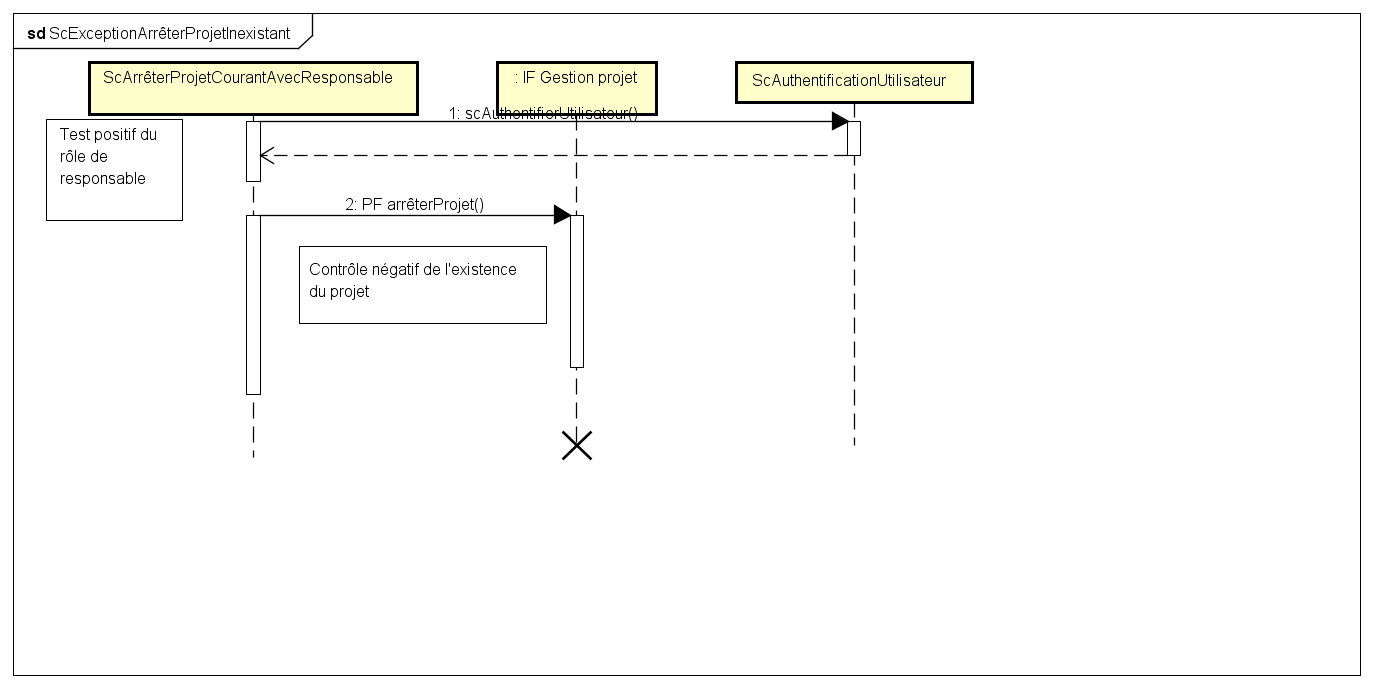
Figure 10 Diagramme de séquences fonctionnelles du scénario ScExceptionCréerProjetQuantitéNonSuffisante

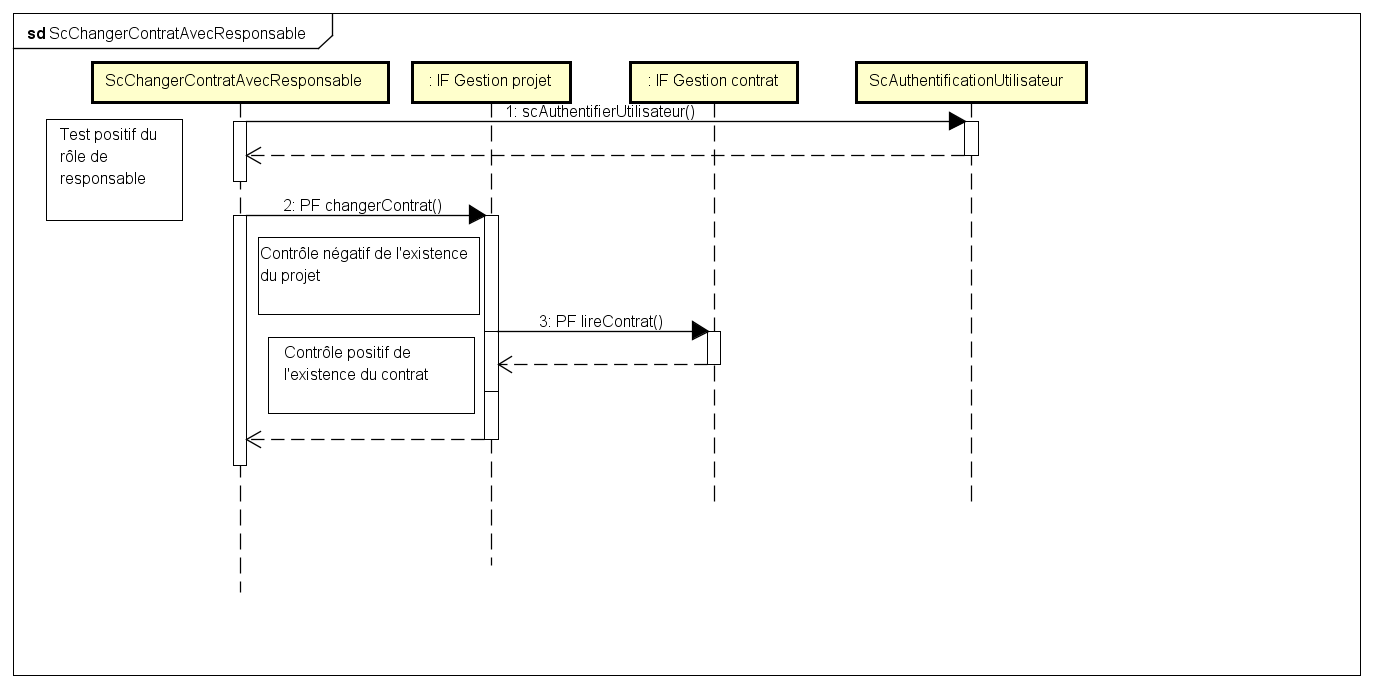
Figure 11 Diagramme de séquences fonctionnelles du scénario ScExceptionCréerProjetQuantitéSansContrat

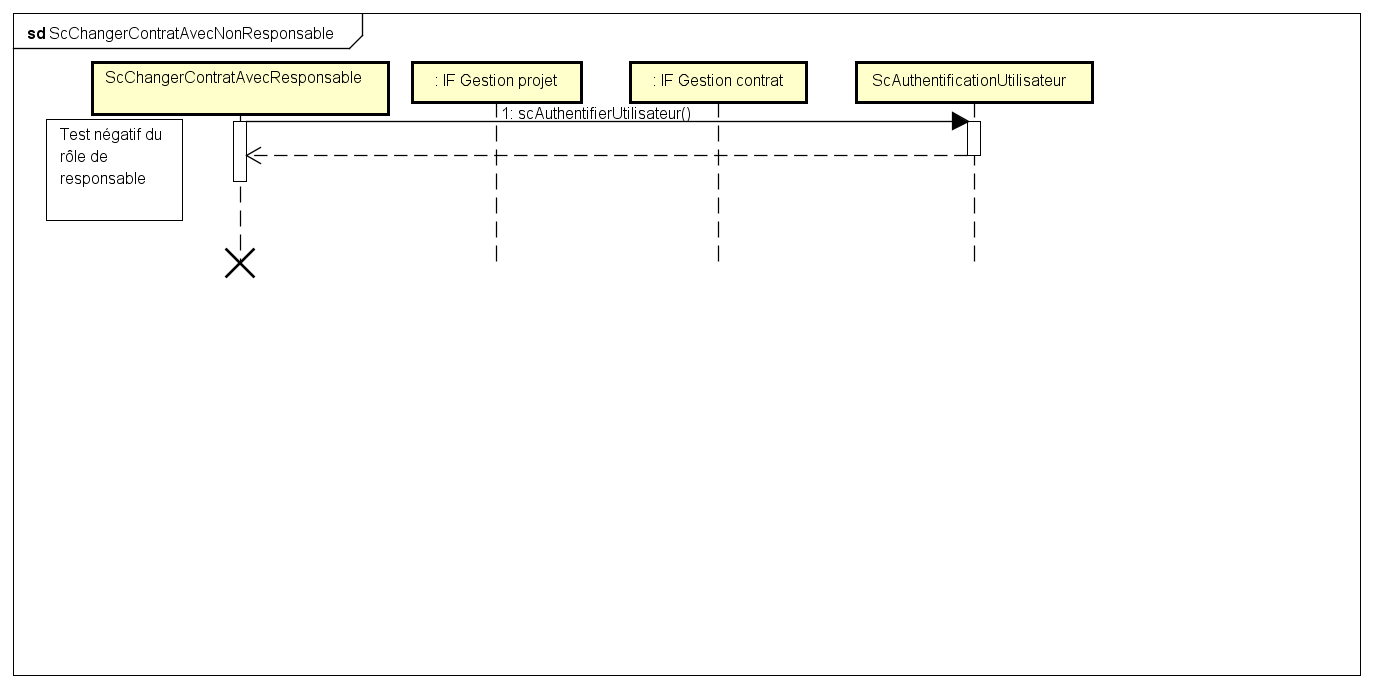
Figure 12 Diagramme de séquences fonctionnelles du scénario ScArrêterProjetCourrantAvecResponsable

Figure 13 Diagramme de séquences fonctionnelles du scénario ScExceptionArrêterProjetAvecNonResponsable

Figure 14 Diagramme de séquences fonctionnelles du scénario ScExceptionArrêterProjetAvecExpiré

Figure 15 Diagramme de séquences fonctionnelles du scénario ScExceptionArrêterProjetAvecNonInexistant

Figure 16 Diagramme de séquences fonctionnelles du scénario ScChangerContratAvecResponsable

Figure 17 Diagramme de séquences fonctionnelles du scénario ScExceptionChangerContratAvecNonResponsable

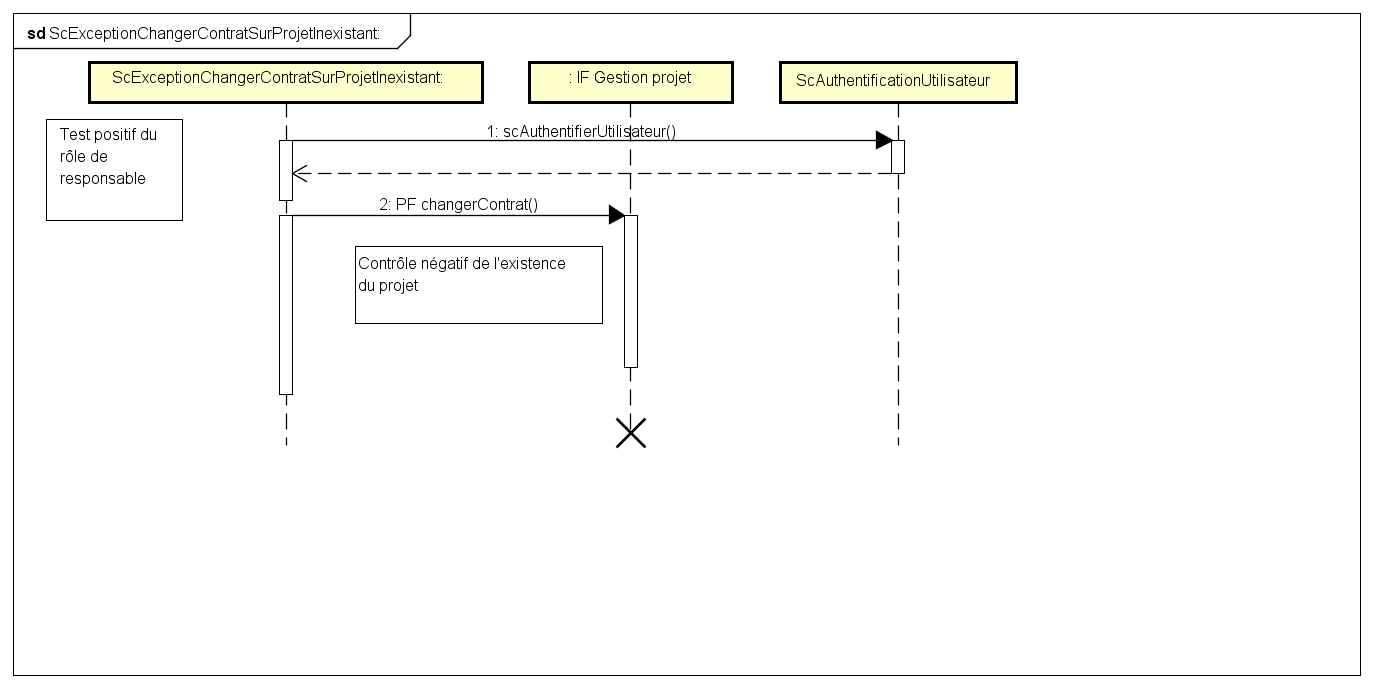
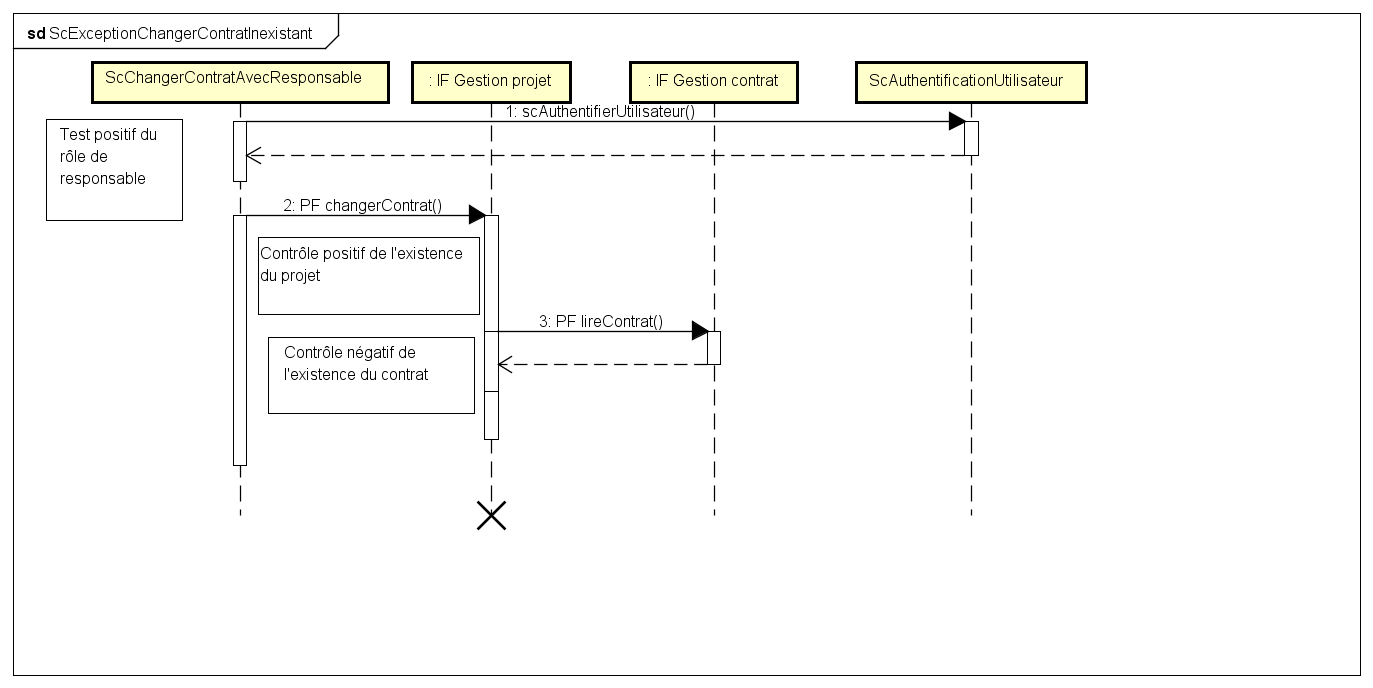
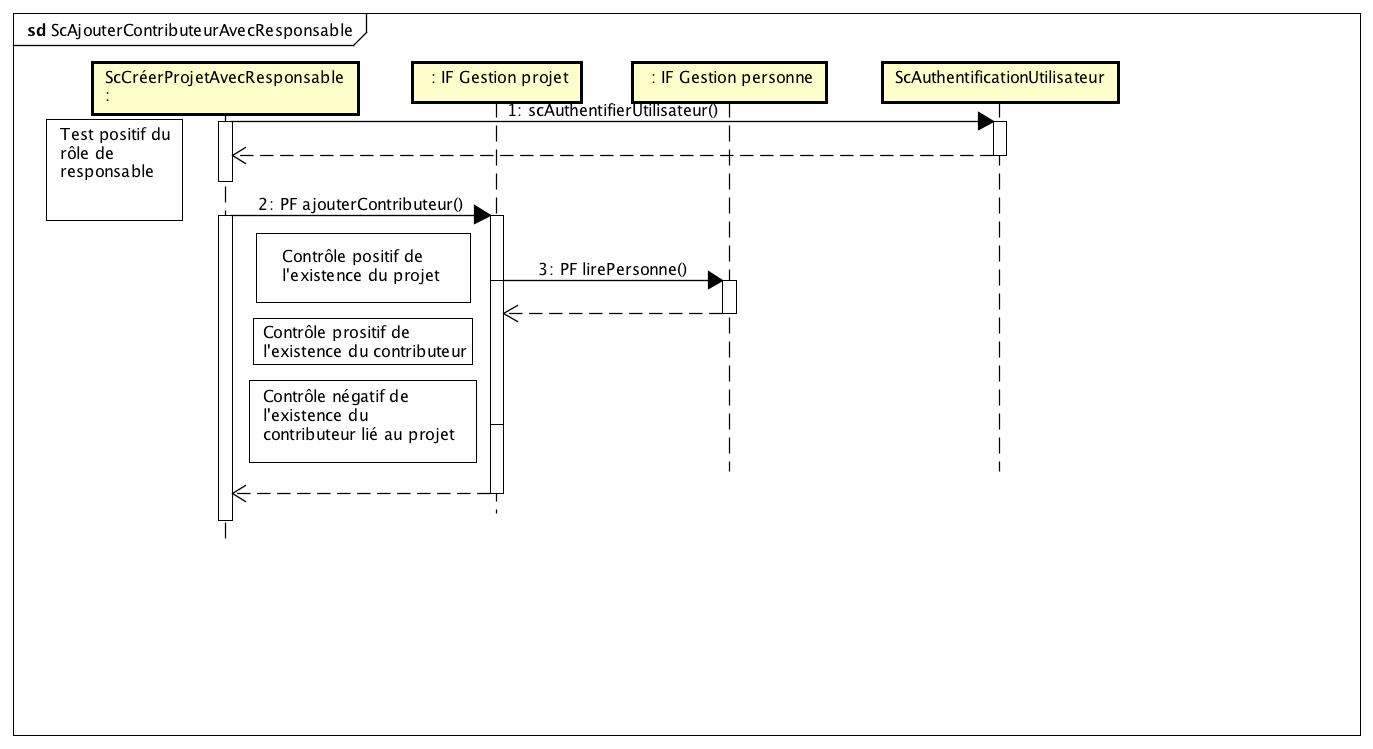
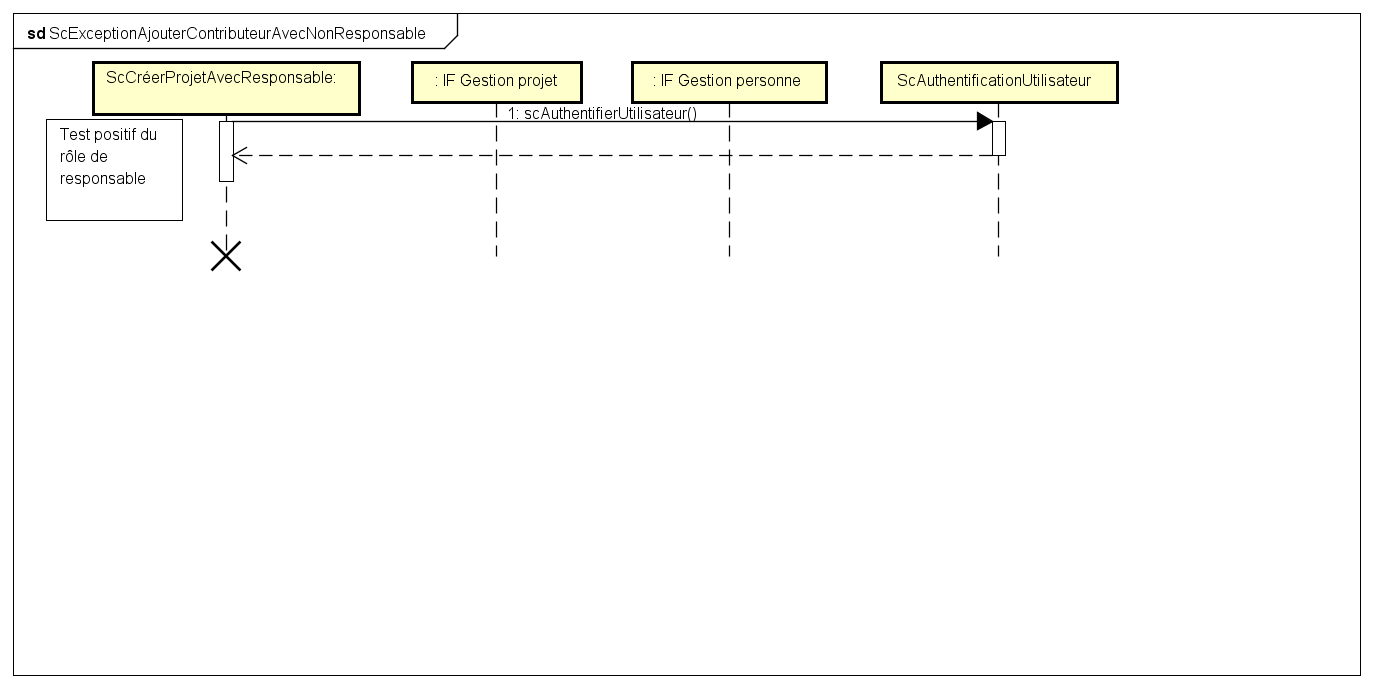
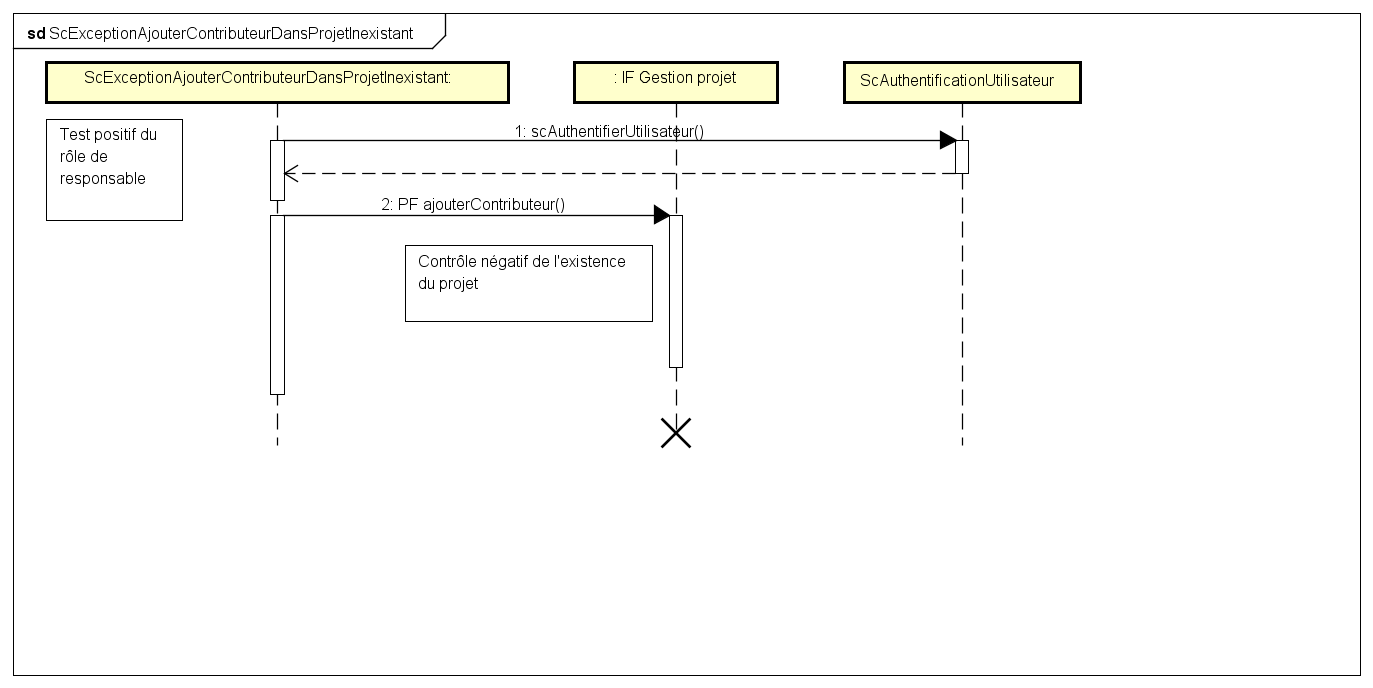


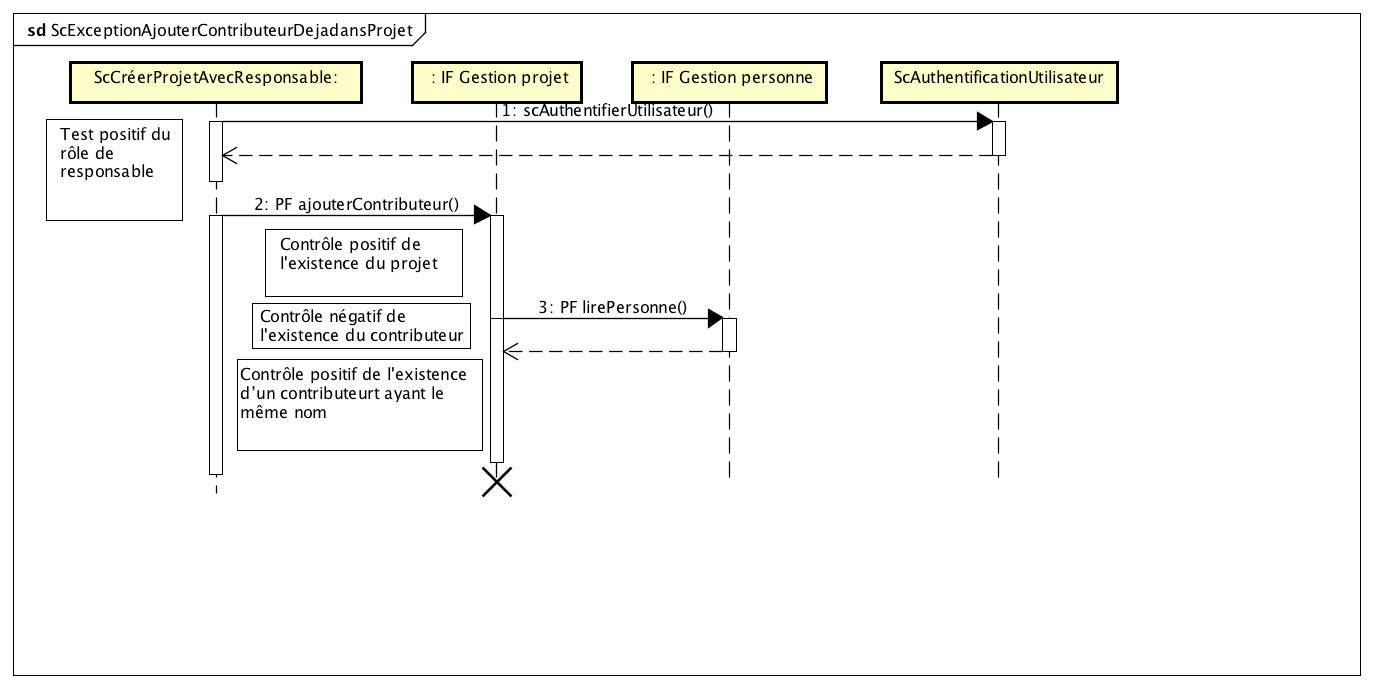
Figure 18 Diagramme de séquences fonctionnelles du scénario ScExceptionChangerContratSurProjetInexistant

Figure 19 Diagramme de séquences fonctionnelles du scénario ScExceptionChangerContratInexistant

Figure 20 Diagramme de séquences fonctionnelles du scénario ScAjouterContributeurAvecResponsable

Figure 21 Diagramme de séquences fonctionnelles du scénario ScExceptionAjouterContributeurAvecNonResponsable

Figure 22 Diagramme de séquences fonctionnelles du scénario ScExceptionAjouterContributeurDansProjetInexistant

Figure 23 Diagramme de séquences fonctionnelles du scénario ScExceptionAjouterContributeurDejadansProjet

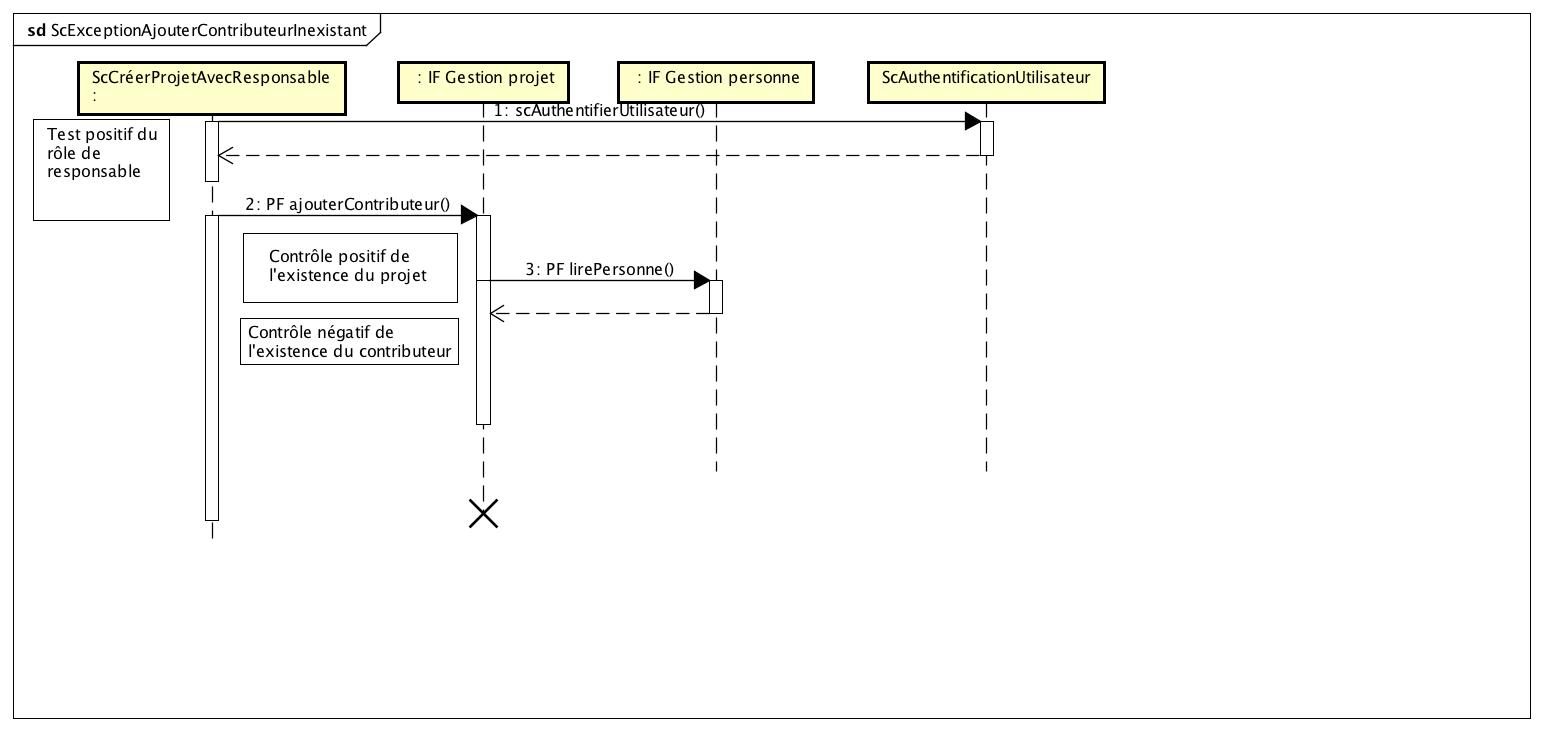
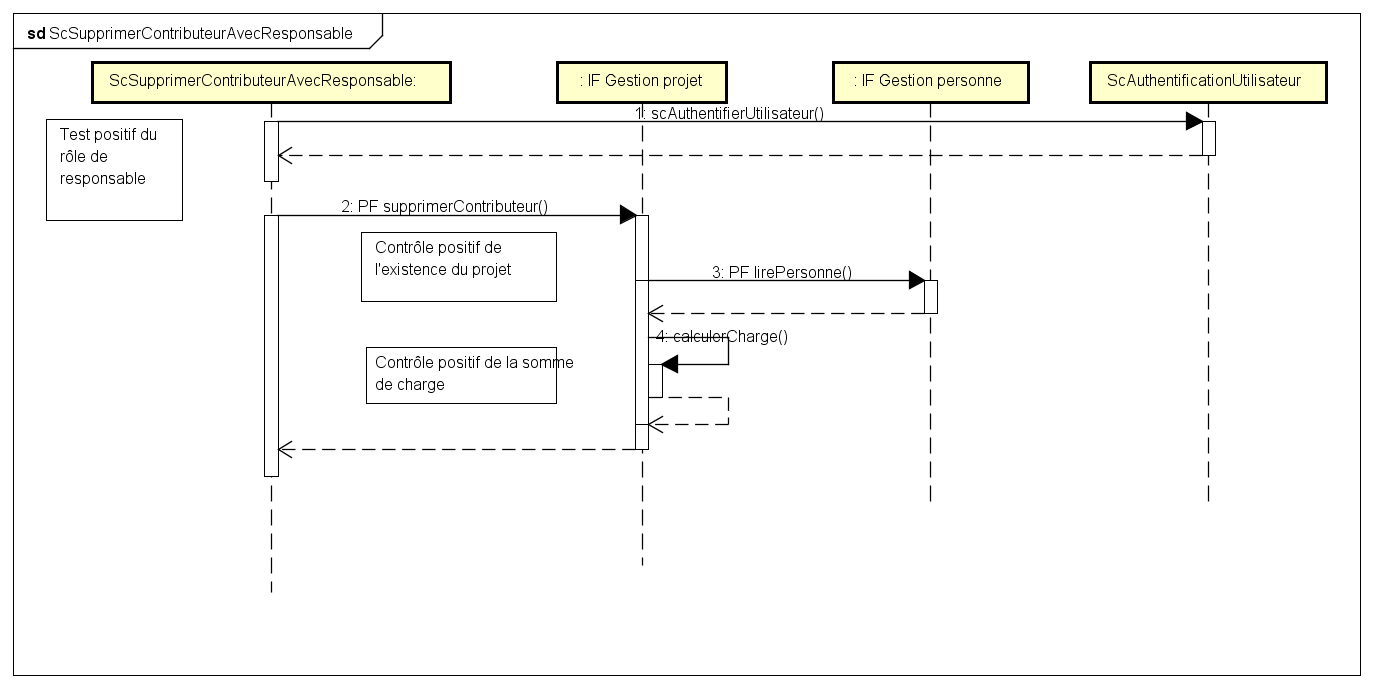


Figure 24 Diagramme de séquences fonctionnelles du scénario ScExceptionAjouterContributeurInexistant

Figure 25 Diagramme de séquences fonctionnelles du scénario ScSupprimerContributeurAvecResponsable

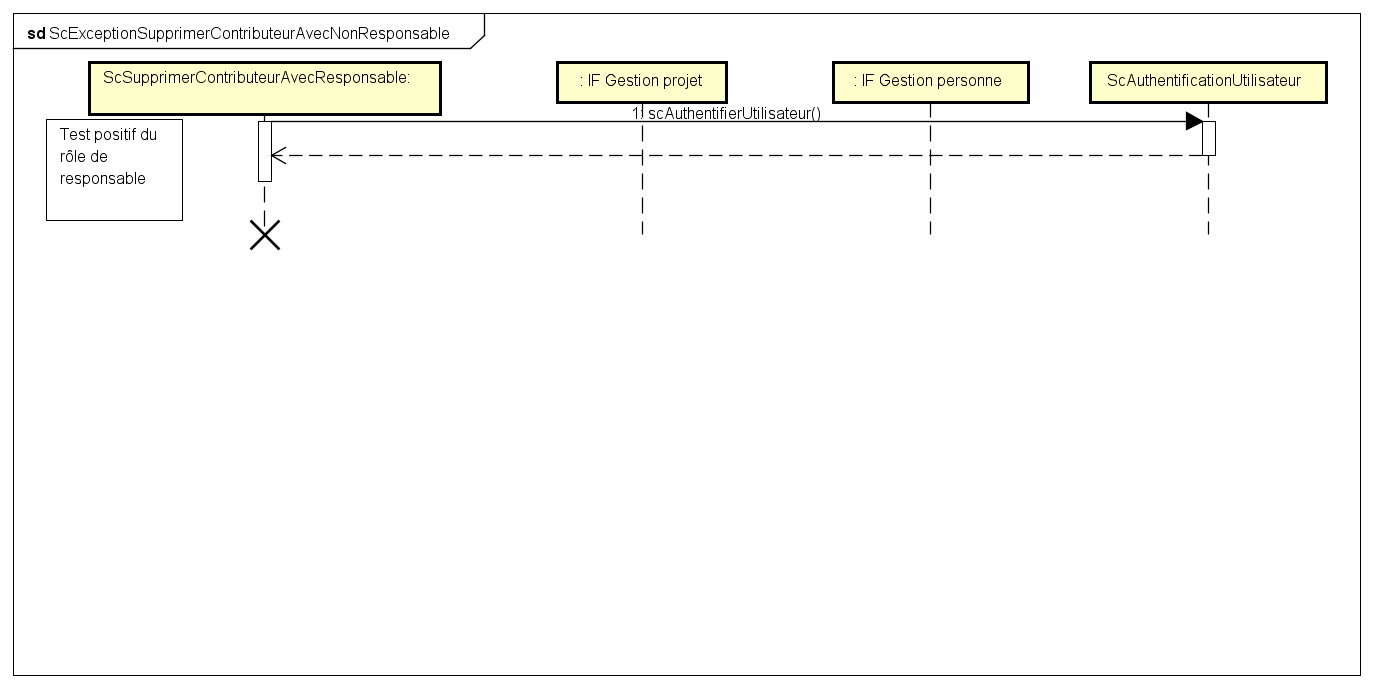
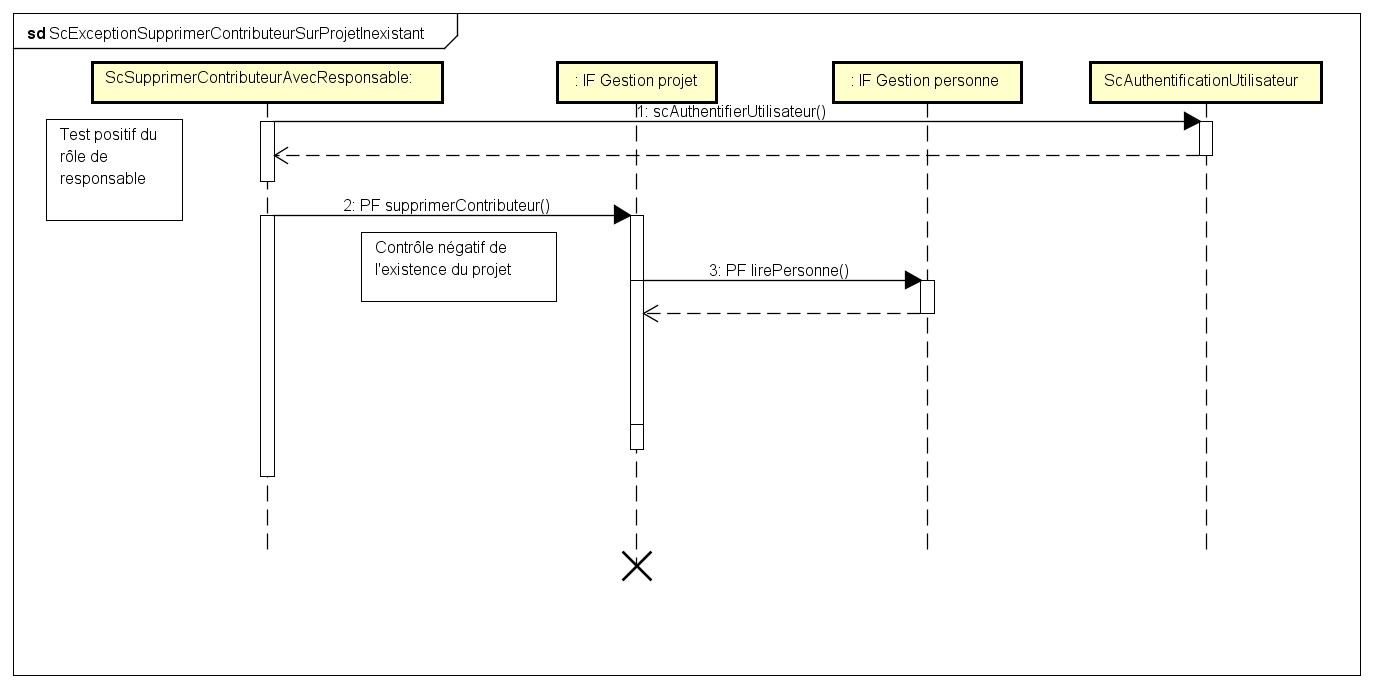
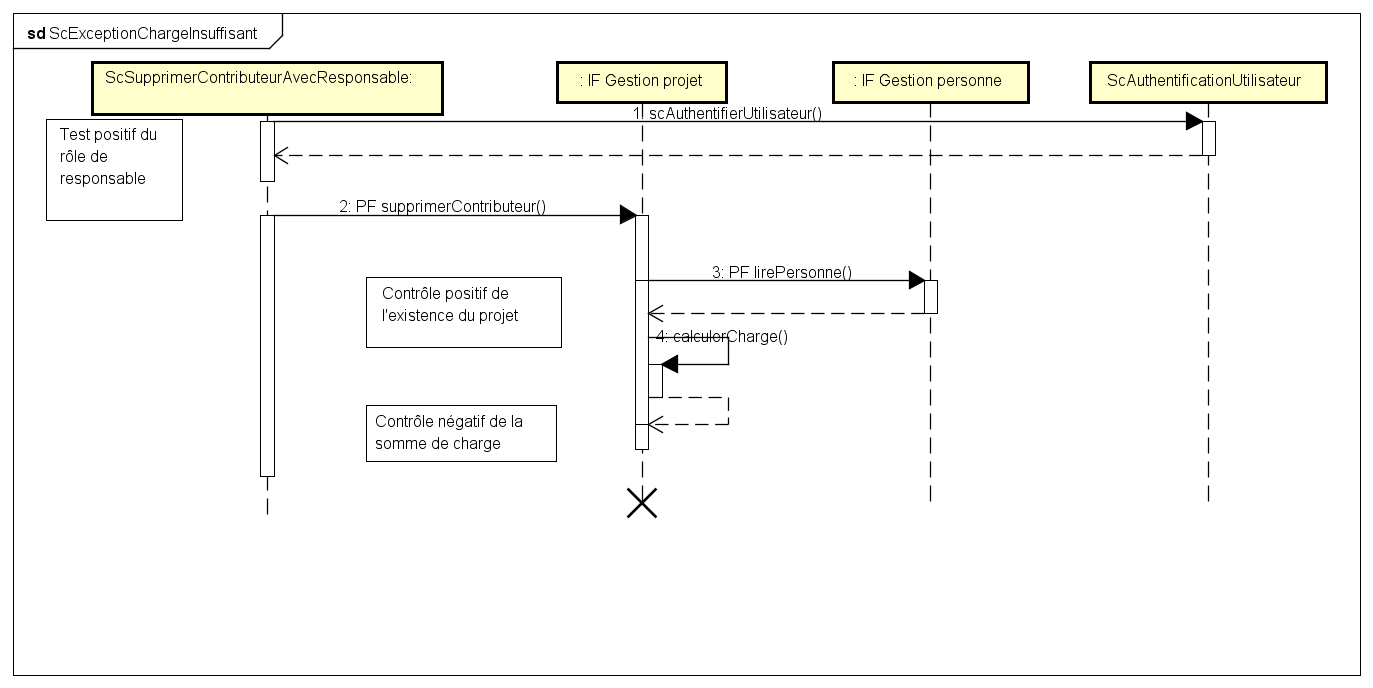
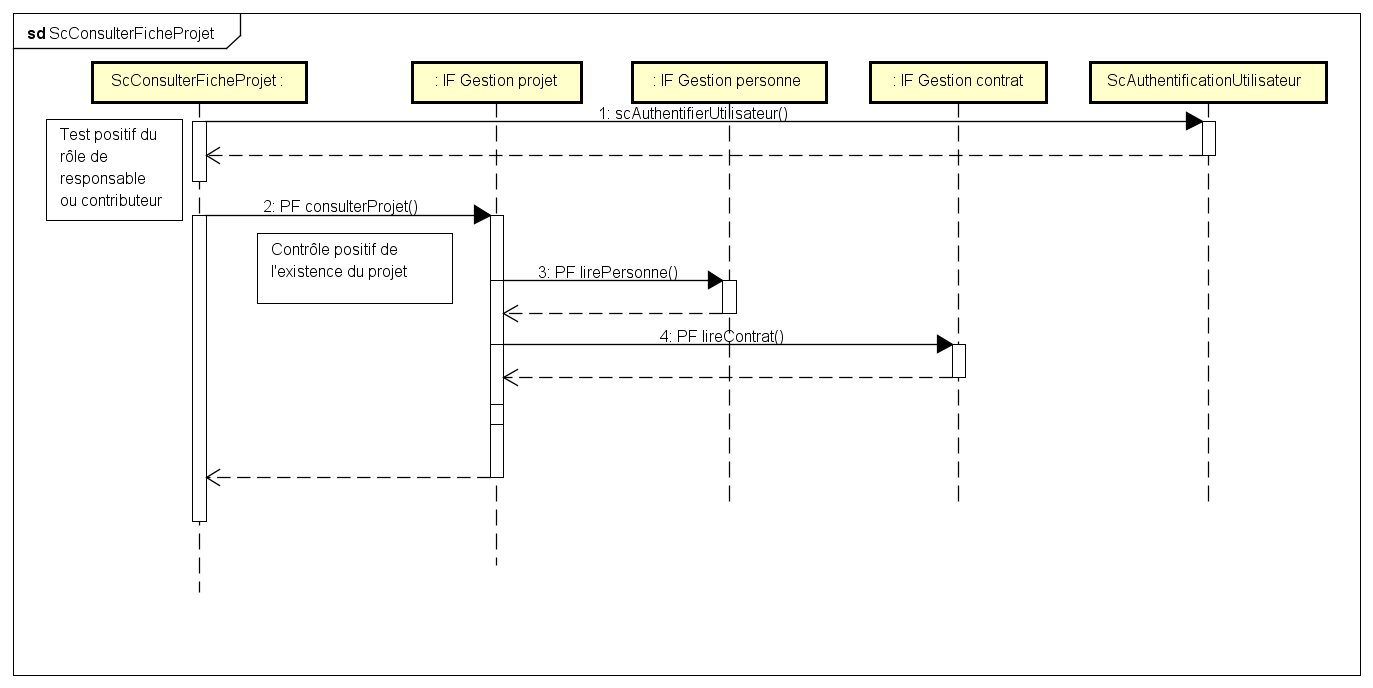
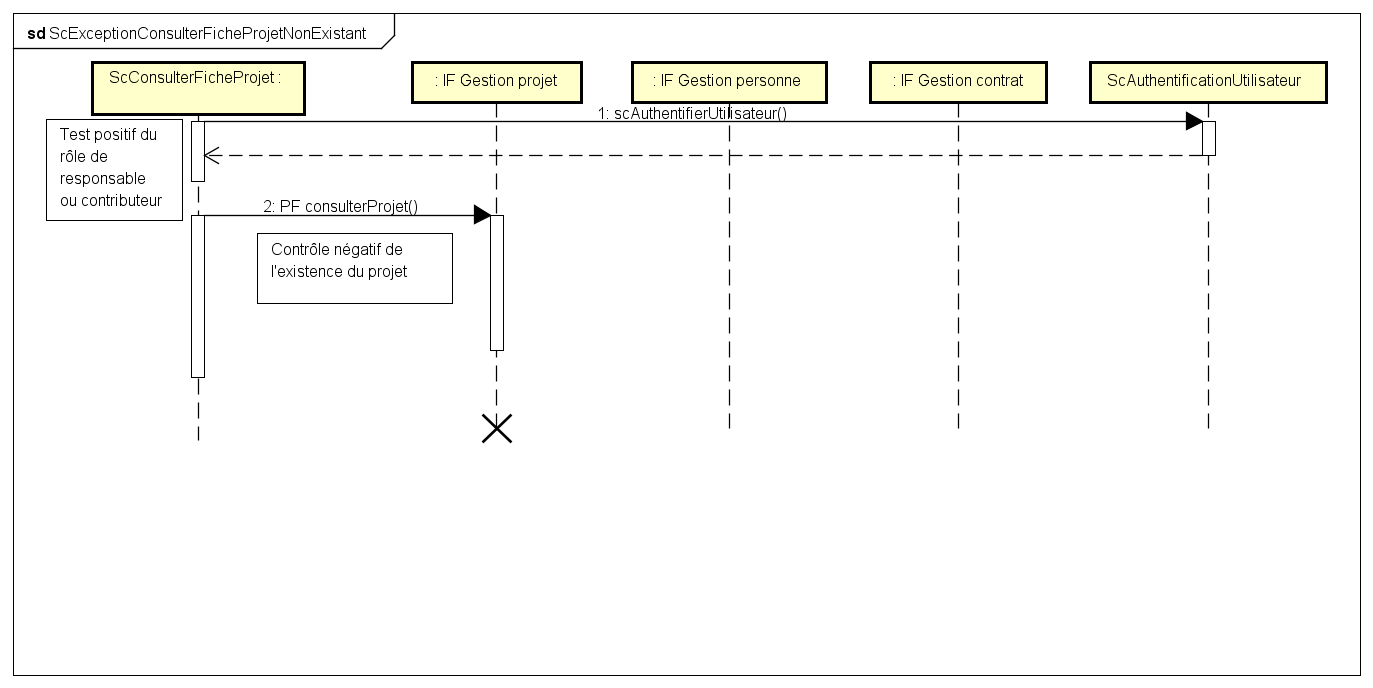


Figure 26 Diagramme de séquences fonctionnelles du scénario ScExceptionSupprimerContributeurAvecNonResponsable

Figure 27 Diagramme de séquences fonctionnelles du scénario ScExceptionSupprimerContributeurSurProjetNonExistant

Figure 28 Diagramme de séquences fonctionnelles du scénario ScExceptionChargerInsuffisant

Figure 29 Diagramme de séquences fonctionnelles du scénario ScConsulterFichier

Figure 30 Diagramme de séquences fonctionnelles du scénario ScExceptionConsulterFichierProjetNonExistant

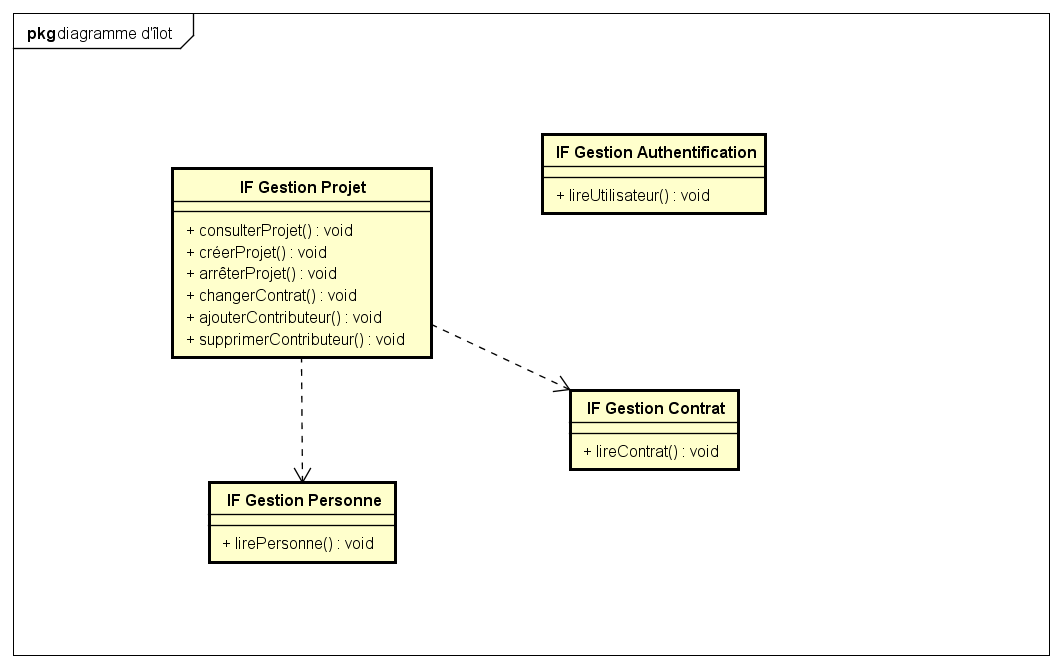
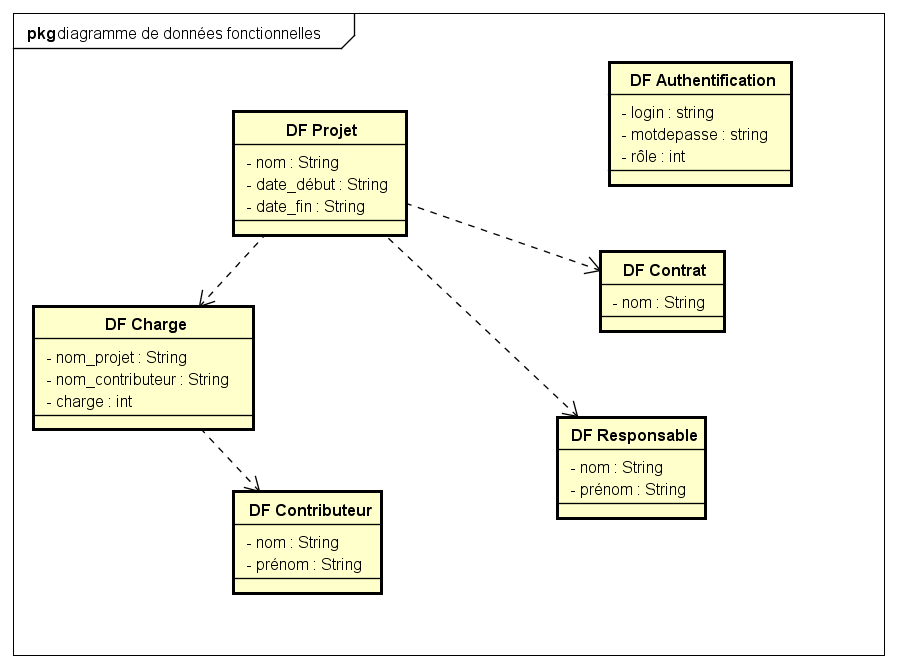
Le modèle des îlots fonctionnels est représenté par un diagramme de classes UML stéréotypées “îlot fonctionnel” dans la figure ci-dessous. Les dépendances entre îlots sont déduites des diagrammes de séquences fonctionnelles.

Figure 31 Modèle des îlots fonctionnels

## Données Fonctionnelles

Le modèle des données fonctionnelles est représenté par un diagramme de class UML stéréotypes “donnée fonctionnelle” dans la figure ci-dessous. Chaque donnée et chaque dépendance entre donnée sont cohérentes avec le modèle des îlots fonctionnels.

Figure 32 Modèle des données fonctionnelles

Le Tableau 5 représenté le lien entre les îlots fonctionnels et les données fonctionnelles. Un îlot fonctionnel peut fournir une donnée fonctionnelle(F) ou utiliser une donnée fonctionnelle(U).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | DF Authentification | DF Projet | DF Contributeur | DF Responsable | DF Contrat | DF Charge |
| IF Gestion Personne |  |  | U | U |  |  |
| IF Gestion Projet |  | U/F |  |  |  | U/F |
| IF Gestion Contrat |  |  |  |  | U |  |
| IF Gestion Authentification | U |  | U | U |  |  |

Tableau 5

La traçabilité des entité participantes aux cas d’utilisation est représentés dans le Tableau 6 :

|  |  |  |
| --- | --- | --- |
| Entité participante | Attribut | Donnée fonctionnelle |
| Utilisateur | nom | DF Authentification |
| prénom |
| rôle |
| Projet | nom | DF Projet |
| date\_début |
| date\_fin |
| Contributeur | nom | DF Contributeur |
| prénom |
| Responsable | nom | DF Responsable |
| prénom |
| Contrat | nom | DF Contrat |
| Charge | quantity | DF Charge |

Tableau 6

# Architecture Applicative

## Composant applicatives

Les composants applicatifs du logiciel sont décrits au niveau logiciel dans le Tableau 7.

|  |  |  |
| --- | --- | --- |
| îlot fonctionnel | Parcelle fonctionnelle | Logiciel |
| IF Gestion Authentification | lireUtilisateur | Direction scientifique |
| IF Gestion Projet | créerProjet |
| consulterProjet |
| arrêterProjet |
| changerContrat |
| ajouterContributeur |
| supprimerContributeur |
| IF Gestion Personne | lirePersonne |
| IF Gestion Contrat | lireContrat |

Tableau 7

La décomposition par protection sur chaque couche est décrite dans le Tableau 8.

|  |  |  |
| --- | --- | --- |
| Logiciel | Couche | Composant applicatif |
| Direction Scientifique | Présentation | CA Test Direction Scientifique |
| Service | CA Authentifier Utilisateur |
| CA Créer Projet |
| CA Arrêter Projet |
| CA Changer Contrat |
| CA Ajouter Contributeur |
| CA Supprimer Contributeur |
| CA Consulter Fiche Projet |
| Données | CA Authentification |
| CA Projet |
| CA Contributeur |
| CA JoinProjetContributeur |
| CA Contrat |
| CA Responsable |

Tableau 8

## Dépendance des composants applicatifs

Les dépendances des composants applicatifs répertoriés précédemment sont conçues à partir des scénarios spécifiés en analyse fonctionnelle et des diagrammes de séquences conçu en architecture fonctionnelle qui sont à réaliser. C'est une approche dynamique de l'architecture applicatives. La traçabilité des scénarios dans les diagrammes de séquences applicatives est reportée dans le Tableau 9.

|  |  |
| --- | --- |
| Scénario | Diagramme de séquences fonctionnelles |
| ScAuthentifierUtilisateur |  |
| ScExceptionAuthentifierUtilisateurInexistant |  |
| ScCréerProjetAvecResponsable |  |
| ScExceptionCréerProjetAvecNonResponsable |  |
| ScExceptionCréerProjetExistant |  |
| ScExceptionCréerProjetQuantitéNonSuffisant |  |
| ScExceptionCréerProjeSansContratExistant |  |
| ScArrêterProjetCourantAvecResponsable |  |
| ScExceptionArrêterProjetAvecNonResponsable |  |
| ScExceptionArrêterProjetExpiré |  |
| ScExceptionArrêterProjetInexistant |  |
| ScChangerContratAvecResponsable |  |
| ScExceptionChangerContratAvecNonResponsable |  |
| ScExceptionChangerContratSurProjetInexistant |  |
| ScExceptionChangerContratInexistant |  |
| ScAjouterContributeurAvecResponsable |  |
| ScExceptionAjouterContributeurAvecNonResponsable |  |
| ScExceptionAjouterContributeurDansProjetInexistant |  |
| ScExceptionAjouterContributeurDejadansProjet |  |
| ScExceptionAjouterContributeurInexistant |  |
| ScSupprimerContributeurAvecResponsable |  |
| ScExceptionSupprimerContributeurAvecNonResponsable |  |
| ScExceptionSupprimerContributeurDansProjetInexistant |  |
| ScExceptionChargeInsuffisant |  |
| ScConsulterFicheProjet |  |
| ScExceptionConsulterFicheProjetNonExistant |  |

Tableau 9

Le modèles des composants applicatifs est représenté par un diagramme de classes UML stéréotypées “composant applicatifs - ‘couche’” dans la figureXXXX. Les dépendances entre composants sont deduites des diagrammes de séquence applicatives.

//TODO: add CA figure here

## Données applicatives

Le modèle des données applicatives est représenté par un diagramme de classes UML stéréotypées “donnée applicative” dans la figureXXXX. Chaque donnée et chaque dépendance entre données sont cohérentes avec le modèle des composants applicatifs.

//TODO: add DA figure here

Le Tableau 10 représente le lien entre les composants applicatifs d’accès aux données et les données applicatives. Un composant applicatif peut fournir une donnée applicative(F) ou utiliser une donner applicative au sens objet(U). La règle d’urbanisme étant qu’une donnée applicative ne peut être fournie que par un seul composant applicatif d’accès aux données.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | DF Authentification | DF Projet | DF Contributeur | DF Responsable | DF Contrat | DF Charge |
| CA Authentification Utilisateur |  |  | U | U |  |  |
| CA Arrêter Projet |  | U/F |  |  |  | U/F |
| IF Gestion Contrat |  |  |  |  | U |  |
| IF Gestion Authentification | U |  | U | U |  |  |

Tableau 10

La traçabilité des données fonctionnelles est représenté dans le Tableau 11.

|  |  |  |
| --- | --- | --- |
| Donnée fonctionnelle | Attribut | Donnée applicative |
| DF Authentification | nom | DA Authentification |
| prénom |
| rôle |
| DF Projet | nom | DA Projet |
| date\_début |
| date\_fin |
| DF Contributeur | nom | DA Contributeur |
| prénom |
| DF Responsable | nom | DA Responsable |
| prénom |
| DF Contrat | nom | DA Contrat |
| DF Charge | quantity | DA JointProjetContributeur |

Tableau 11

Des dépendances des données fonctionnelles réalisant des associations entre entités participantes ayant pour cardinalités \*-\* dans le Tableau 12 :

|  |  |
| --- | --- |
| Dépendance entre données fonctionnelles | Donnée applicative |
| DF Projet vers DF Contributeur | DA JointProjetContributeur |

Tableau 12

|  |  |
| --- | --- |
|  | Technopôle Brest-Iroise CS 83818 29238 Brest Cedex 3 France  +33 (0)2 29 00 11 11  **www.telecom-bretagne.eu** |
| INSTITUT-MINES-TELECOM_Logo_Buro_petit | |

|  |  |
| --- | --- |
|  |  |
|  | |