**DOSSIER DE MODELISATION D’UNE APPLICATION DE GESTION DES ÉTUDIANTS**

1. VUE DES BESOINS
2. DIAGRAMME DE CONTEXTE
3. Domaine : Gestion des étudiants
4. Acteurs :

+ Admin (A)

+ Responsable Pédagogique (RP)

+ Attaché de Classe (AC)

+ Etudiant (E)

+ Professeur (P)

1. Formalisme du diagramme

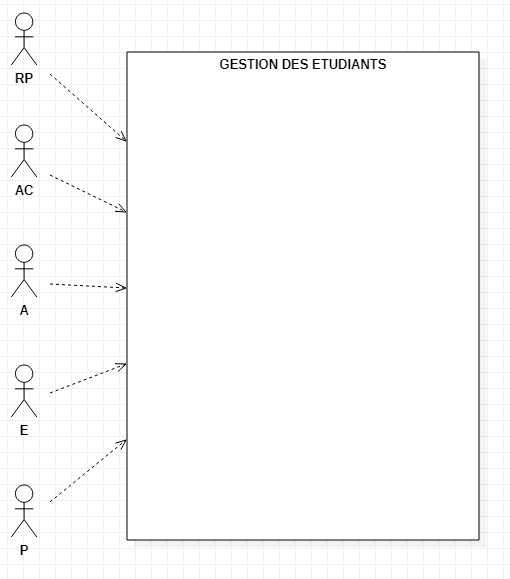


DIAGRAMME DE CONTEXTE 1

1. Liens entre acteurs (Héritage)

Un acteur A hérite des fonctionnalités d’un acteur B lorsque l’acteur A peut faire toutes les fonctionnalités de l’acteur B.

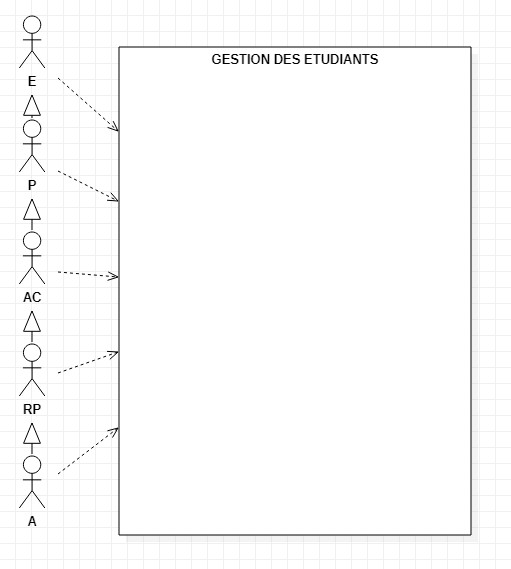


DIAGRAMME DE CONTEXTE 2

1. DIAGRAMME DE PACKAGE
2. Besoins par acteurs
3. En tanque Admin (A), je peux :

* Me connecter
* Me déconnecter
* Modifier mes données après connexion
* Ajouter, modifier, supprimer un RP ou AC

1. En tant que Responsable Pédagogique (RP), je peux :

* Me connecter
* Me déconnecter
* Modifier mes données après connexion
* Lister les étudiants par classe et par année scolaire
* Ajouter un professeur
* Planifier un cours pour un professeur et pour une ou plusieurs classes
* Lister les cours d’un professeur
* Lister les cours d’une classe

1. En tant qu’Attaché de Classe (AC), je peux :

* Me connecter
* Me déconnecter
* Modifier mes données après connexion
* Inscrire un étudiant dans une classe dans une année scolaire
* Lister les étudiants par classe et par année scolaire
* Lister les cours d’un professeur
* Marquer les absences d’un étudiant par rapport à une date
* Lister les absences d’un étudiant
* Lister les absences d’un cours
* Filtrer par classe ou par matricule ou par niveau
* Imprimer les absences d’un cours ou d’un étudiant

1. En tant que Professeur (P), je peux :

* Me connecter
* Me déconnecter
* Modifier mes informations après connexion
* Marquer les absences d’un cours
* Lister ses cours

1. En tant qu’ Etudiant (E), je peux :

* Me connecter
* Me déconnecter
* Modifier mes données après connexion
* Lister ses cours suivant une période entre deux dates
* Lister ses absences par rapport à un cours
* Lister ses absences du semestre

1. Tableau récapitulatif des use case

|  |  |  |
| --- | --- | --- |
| Use Cases | Acteurs | Package |
| Créer un compte pour AC, RP | A | SÉCURITÉ |
| Créer un compte pour Etudiant | AC |
| Se connecter | A, RP, AC, P, E |
| Se déconnecter | A, RP, AC, P, E |
| Modifier données après connexion | A, RP, AC, P, E |
| Lister les étudiants par classe et par année scolaire | AC, RP | RESSOURCES |
| Ajouter un professeur | RP |
| Planifier un cours pour un professeur et pour une ou plusieurs classes | RP |
| Lister les cours d’un professeur | RP, AC, P |
| Lister les cours d’une classe | RP |
| Filtrer par classe ou par matricule ou par niveau | AC |
| Lister ses cours suivant une période entre deux dates | E |
| Marquer les absences d’un étudiant par rapport à une date | AC | Gestion des Absences |
| Lister les absences d’un étudiant | AC |
| Lister les absences d’un cours | AC |
| Imprimer les absences d’un cours ou d’un étudiant | AC |
| Marquer les absences d’un cours | P |
| Lister ses absences par rapport à un cours | E |
| Lister ses absences du semestre | E |

1. Diagramme de package

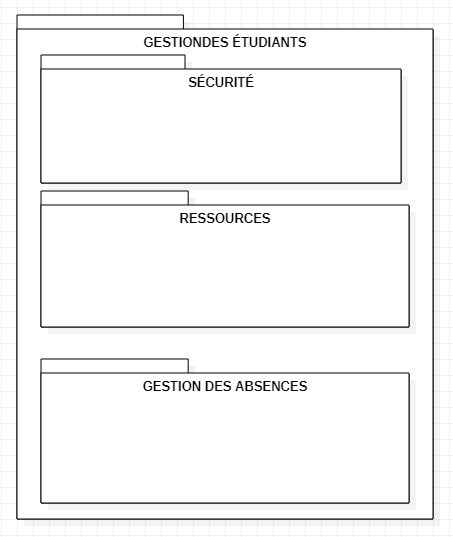


Diagramme de package version 1

1. DIAGRAMME DE USE CASE
2. Diagramme de use case du package Sécurité

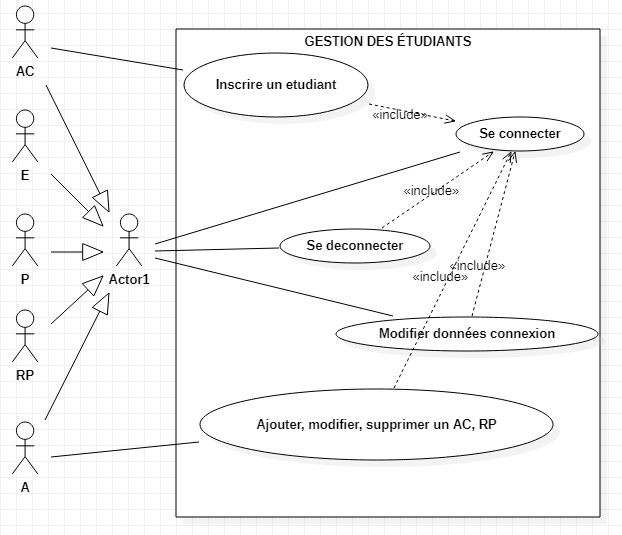


Diagramme de Use Case du package sécurité

1. Diagramme de use case du package ressource

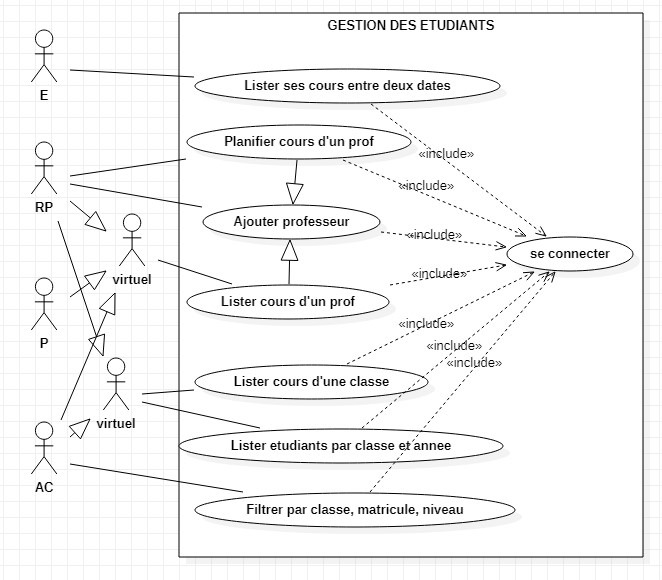


Diagramme de use case du package ressource

1. Diagramme de use case du package gestion des absences

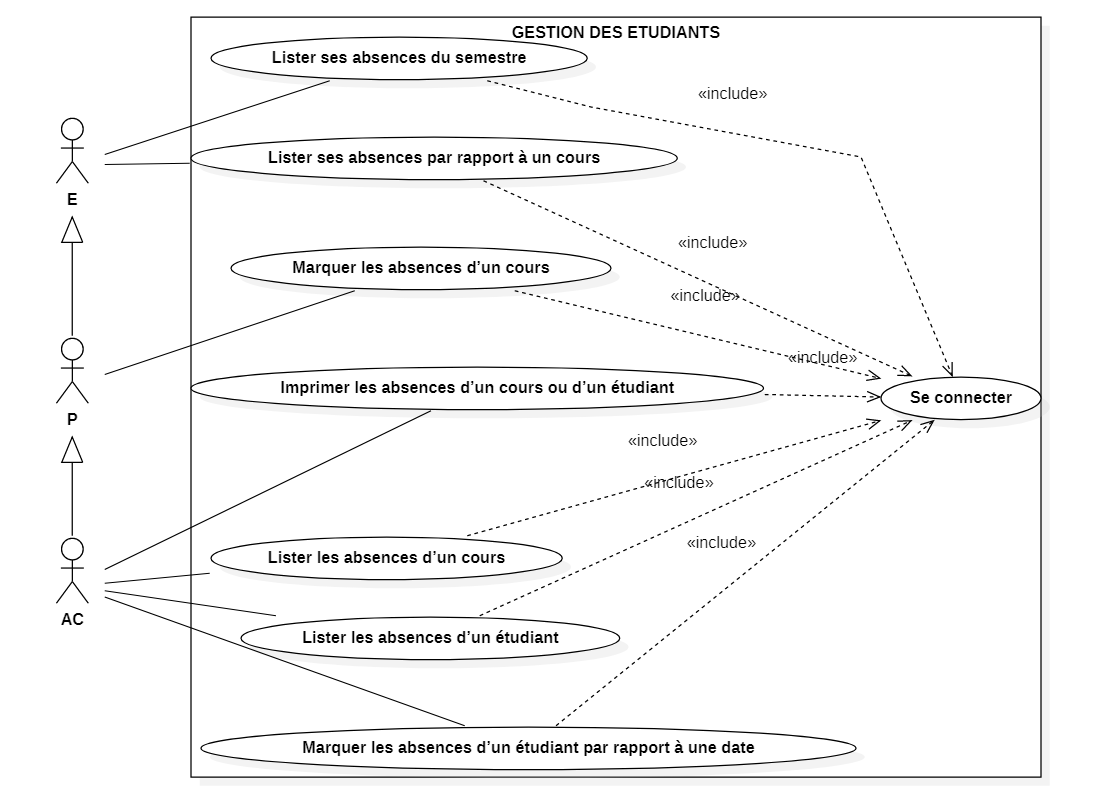
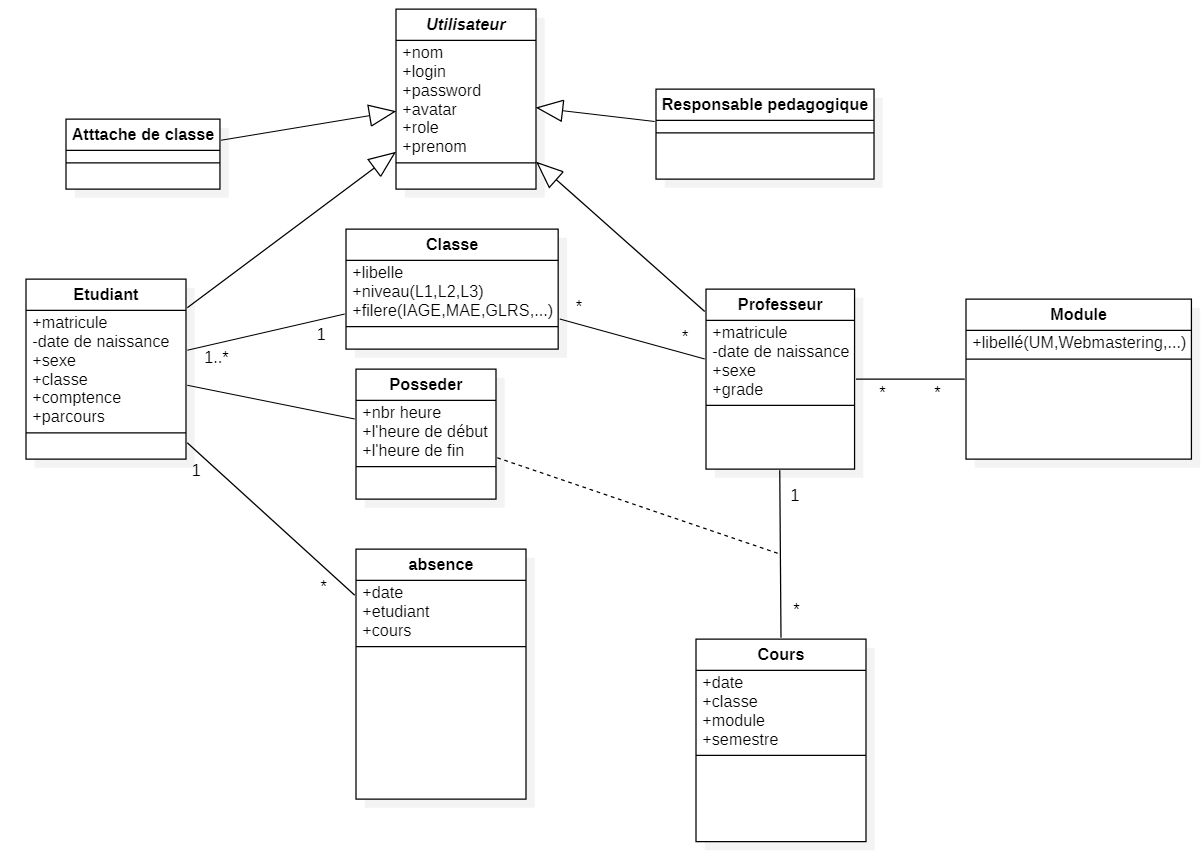


Diagramme de use case du package gestion des absences

1. Diagramme de classe de la gestion des étudiants



**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

1. **DESCRIPTION TEXTUELLE DES uses cases**

Description textuelle du use case « inscrire étudiants »

|  |  |
| --- | --- |
| **Nom UC** | Inscrire étudiants |
| **Acteur** | AC |
| **Objet** | Ce cas permet d’inscrire les étudiants dans le système. |
| **Précondition** | Utilisateur connecté |
| **Postconditions** | Etudiant inscrit |
| **Démarrage** | Demande de la page d’inscription |
| **Scénario** | **Nominal**   1. Le système affiche la page d’inscription 2. L’utilisateur saisit les données de l’étudiant puis valide 3. Le système vérifie et valide les données 4. Le système affiche inscription réussie |
| **Alternatif**  3.a le système affiche un message d’erreur et retourne à 2 |
| **Exceptions**  1.a le système affiche « page non-disponible » |
| **Contraintes** | Données de connexion obligatoire |

Description textuelle du use case « marquer absence »

|  |  |
| --- | --- |
| **Nom UC** | Marquer abscence |
| **Acteur** | AC P |
| **Objet** | Ce cas permet de marquer les absences des étudiants dans le système |
| **Précondition** | * Utilisateur connecté |
| **Postconditions** | * Absence marquée |
| **Démarrage** | Demande la page de la liste des étudiants d’une classe |
| **Scénario** | **Nominal**   1. Le système affiche la liste des étudiants d’une classe 2. L’utilisateur marque absent les étudiants puis valide 3. Le système enregistre les données |
| **Alternatif** |
| **Exceptions**  1.a le système affiche « page non-disponible » |
| **Contraintes** | * Données de connexion unique obligatoire |

Description textuelle du use case «Planifier cours »

|  |  |
| --- | --- |
| **Nom UC** | Planifier cours |
| **Acteur** | RP |
| **Objet** | Ce cas permet de planifier un cours |
| **Précondition** | * Utilisateur connecté |
| **Postconditions** | * Cours planifié |
| **Démarrage** | Demande de la page de la planification de cours |
| **Scénario** | **Nominal**   1. Le système affiche la page de la planification de cours 2. L’utilisateur planifie le cours puis valide 3. Le système enregistre la planification |
| **Alternatif** |
| **Exceptions**  1.a le système affiche « page non-disponible » |
| **Contraintes** | Données de connexion obligatoire |

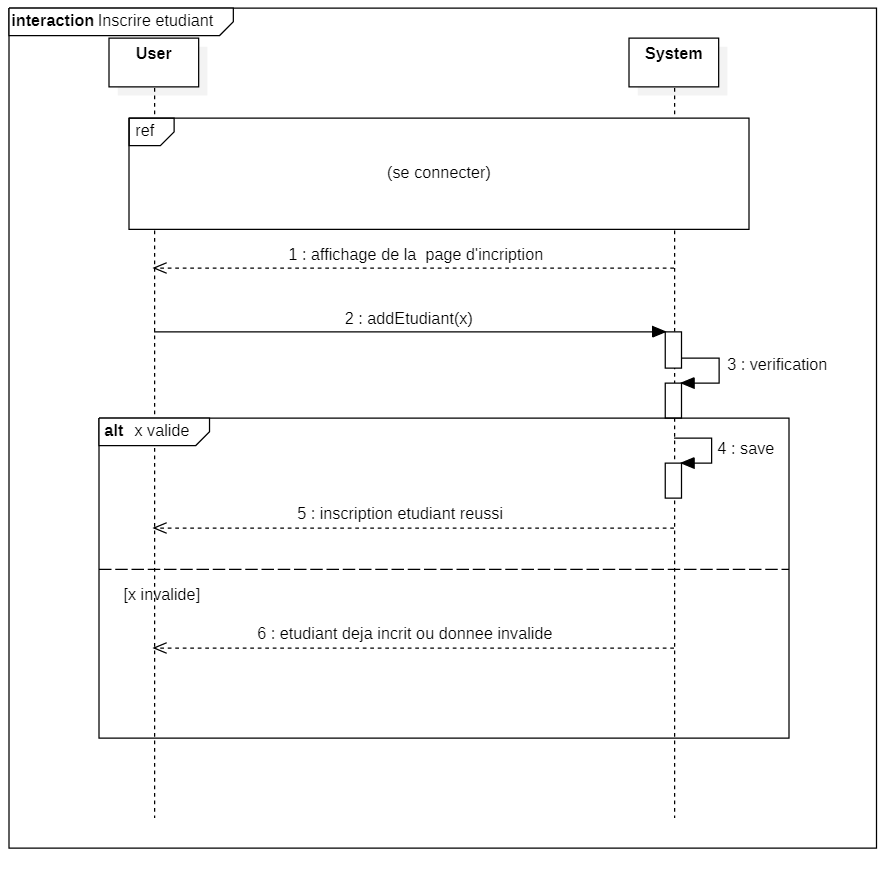
Description textuelle du use case «Lister des étudiants par classe»

|  |  |
| --- | --- |
| **Nom UC** | Lister des étudiants par classe |
| **Acteur** | RP , AC |
| **Objet** | Ce cas permet de Lister des étudiants par classe |
| **Précondition** | * Être connecté * Avoir des étudiants inscrits |
| **Postconditions** | * Affichage de la liste des étudiants |
| **Démarrage** | Demande la page des classes |
| **Scénario** | **Nominal**   1. Le système affiche la liste des classes 2. L’utilisateur choisit une classe et affiche la liste des étudiants 3. Le system valide |
| **Alternatif**  2.a le système envoie message d’erreur << liste indisponible>> et retourne à l’action 1 |
| **Exceptions**  1.a le système affiche « page non-disponible » |
| **Contraintes** |  |

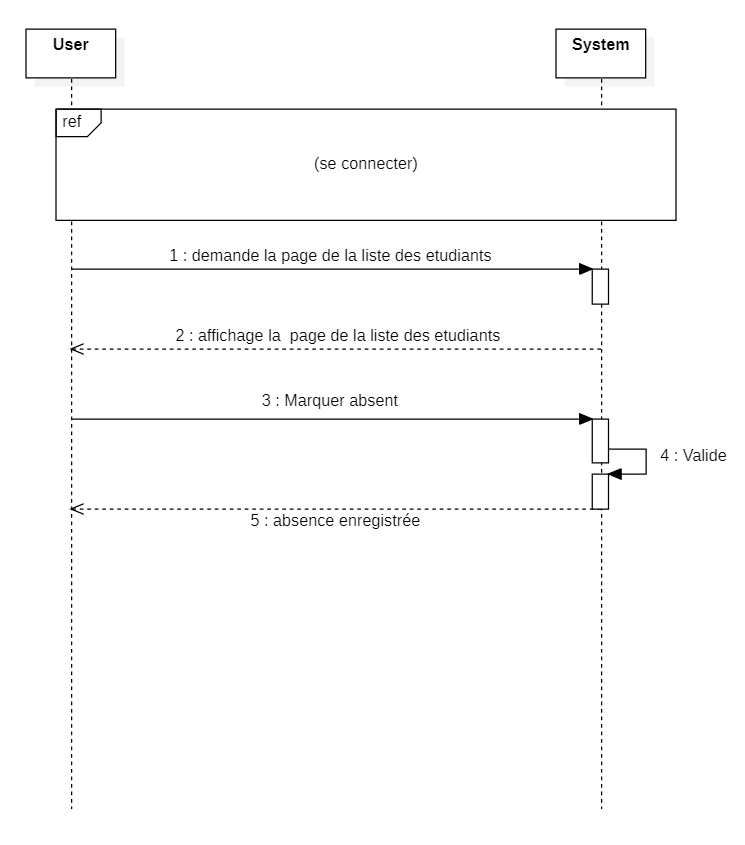
Description textuelle du use case «Lister des cours d’un professeur»

|  |  |
| --- | --- |
| **Nom UC** | Lister des cours d’un professeur |
| **Acteur** | RP |
| **Objet** | Ce cas permet de Lister des cours d’un professeur |
| **Précondition** | * Être connecté |
| **Postconditions** | * Affichage de la liste des cours d’un professeur |
| **Démarrage** | Demande la page des professeurs |
| **Scénario** | **Nominal**   * + - 1. Le système affiche la page des professeurs       2. L’utilisateur choisit un professeur et affiche la liste de ces cours       3. Le system valide |
| **Alternatif**  2. a le système envoie message d’erreur << liste indisponible>> et retourne à l’action 1 |
| **Exceptions**  1. a le système affiche « page non-disponible » |
| **Contraintes** |  |

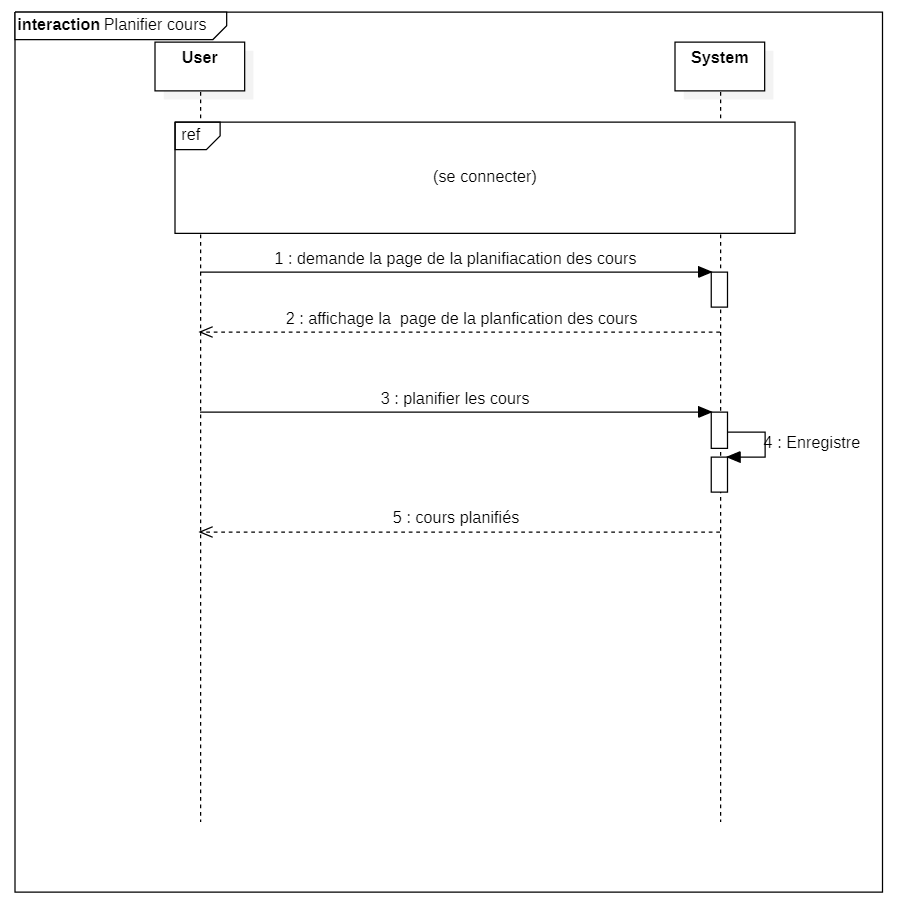
1. VUE DES PROCESSUS



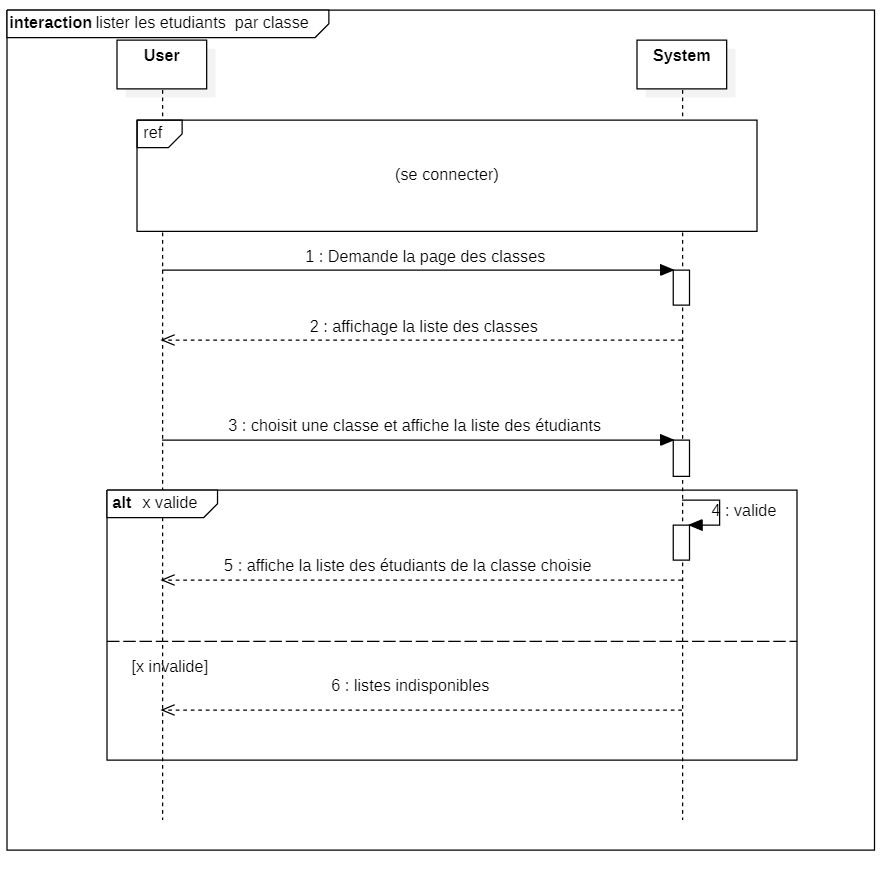
**DIAGRAMME DE SEQUENCE DE « inscrire étudiant »**



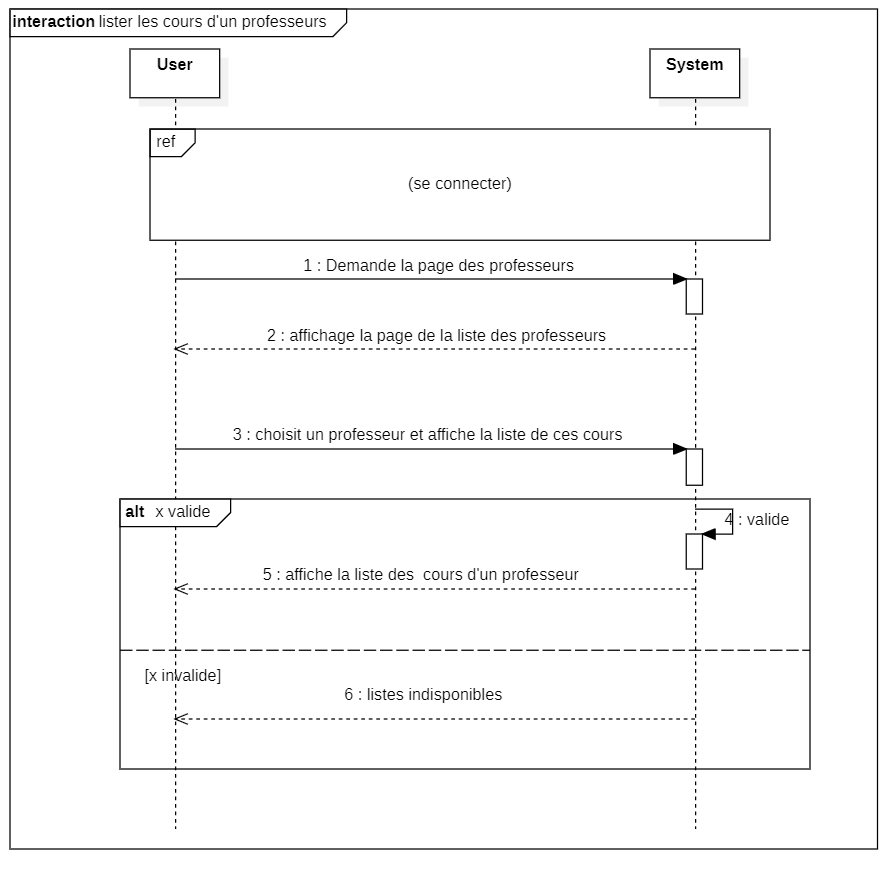
**DIAGRAMME DE SEQUENCE Marquer absence**



**DIAGRAMME DE SEQUENCE Planifier les absences**



**DIAGRAMME DE SEQUENCE Affichage de la Liste des Etudiants par Classe**



**DIAGRAMME DE SEQUENCE Affichage de la Liste des cours d’un professeur**