République Algérienne Démocratique et Populaire

Ministère de l’Enseignement Supérieur et de la Recherche Scientifique

Université Constantine 2 – Abdel Hamid Mehri



Faculté des Nouvelles Technologies de l’Information et de la Communication

Département des Technologies des Logiciels et Systèmes d’Information

Etude de Cas

Sujet :

***Concéption et réalisation d’un système de gestion d’absences***

**Dirigé par :**

* Dr. Zakaria Lakhdara

**Réalisé par :**

* Djebbes Zakaria
* Ouaden Aymen

-2019/2020

# Table des matières

[Table des matières 1](#_Toc36901857)

[Table des figures 4](#_Toc36901858)

[Table des tableaux 5](#_Toc36901859)

[Introduction 6](#_Toc36901860)

[Chapitre 1 : Analyse des besoins 7](#_Toc36901861)

[1. Introduction : 7](#_Toc36901862)

[2. Expression des besoins 7](#_Toc36901863)

[2.1. Introduction : 7](#_Toc36901864)

[2.2. Cahier des charges : 7](#_Toc36901865)

[2.2.1. Présentation de la faculté NTIC : 7](#_Toc36901866)

[2.2.2. Définition de la problématique : 8](#_Toc36901867)

[2.2.3. Définition des objectifs : 8](#_Toc36901868)

[2.2.4. Cible du projet : 8](#_Toc36901869)

[2.2.5. Périmètre du projet : 8](#_Toc36901870)

[2.2.6. Contraintes techniques : 9](#_Toc36901871)

[2.2.7. Méthodologie de conception : 9](#_Toc36901872)

[2.3. Identification des acteurs et des besoins : 9](#_Toc36901873)

[2.3.1. Introduction : 9](#_Toc36901874)

[2.3.2. Indentification des acteurs : 10](#_Toc36901875)

[2.3.2.1. Diagramme de contexte statique : 10](#_Toc36901876)

[2.3.2.2. Définition des acteurs et de leurs tâches : 10](#_Toc36901877)

[2.3.3. Définition des besoins fonctionnels : 11](#_Toc36901878)

[2.3.4. Définition des besoins non-fonctionnels : 12](#_Toc36901879)

[3. Spécification des besoins : 12](#_Toc36901880)

[3.1. Introduction : 12](#_Toc36901881)

[3.2. Diagramme de cas d’utilisation : 12](#_Toc36901882)

[3.2.1. Définitions : 12](#_Toc36901883)

[3.2.2. Diagramme de cas d’utilisation globale : 13](#_Toc36901884)

[3.2.3. Diagramme de cas d’utilisation structuré en packages : 14](#_Toc36901885)

[3.2.3.1. Package Etudiant : 14](#_Toc36901886)

[3.2.3.2. Package Enseignant : 15](#_Toc36901887)

[3.2.3.3. Package Administrateur : 16](#_Toc36901888)

[3.2.3.4. Package Responsable de formation : 16](#_Toc36901889)

[3.2.3.5. Package Chef de département : 17](#_Toc36901890)

[4. Descriptions textuelles et diagrammes de séquence système : 18](#_Toc36901891)

[4.1. Introduction : 18](#_Toc36901892)

[4.2. Descriptions textuelles et graphiques : 19](#_Toc36901893)

[4.2.1. Cas : « S’authentifier » 19](#_Toc36901894)

[4.2.1.1. Fiche descriptive : 19](#_Toc36901895)

[4.2.1.2. Diagramme de séquence système : 20](#_Toc36901896)

[4.2.2. Cas : « Consulter relever d’absence » 21](#_Toc36901897)

[4.2.2.1. Fiche descriptive : 21](#_Toc36901898)

[4.2.2.2. Diagramme de séquence système : 22](#_Toc36901899)

[4.2.3. Cas : « Marquer la présence » 23](#_Toc36901900)

[4.2.3.1. Fiche descriptive : 23](#_Toc36901901)

[4.2.3.2. Diagramme de séquence système : 25](#_Toc36901902)

[4.2.4. Cas : « Justifier une absence » : 26](#_Toc36901903)

[4.2.4.1. Fiche descriptive : 26](#_Toc36901904)

[4.2.4.2. Diagramme de séquence système : 27](#_Toc36901905)

[4.2.5. Cas : « Consulter une justification » 28](#_Toc36901906)

[4.2.5.1. Fiche descriptive : 28](#_Toc36901907)

[4.2.5.2. Diagramme de séquence système : 29](#_Toc36901908)

[4.2.6. Cas : « Établir la liste des étudiants exclus » 30](#_Toc36901909)

[4.2.6.1. Fiche descriptive : 30](#_Toc36901910)

[4.2.6.2. Diagramme de séquence système : 32](#_Toc36901911)

[5. Conclusion : 33](#_Toc36901912)

[Ce qui a été réalisé dans ce chapitre : 33](#_Toc36901913)

[À quoi cela servira-t-il pour les chapitres suivants ? 33](#_Toc36901914)

[Chapitre 2 : Conception 34](#_Toc36901915)

[1. Introduction : 34](#_Toc36901916)

[2. Diagramme de classe conceptuel : 34](#_Toc36901917)

[2.1. Introduction : 34](#_Toc36901918)

[2.2. Définitions : 34](#_Toc36901919)

[2.3. Diagramme de classe conceptuel : 34](#_Toc36901920)

[2.3.1. Tableau des classes : 36](#_Toc36901921)

[2.3.2. Tableau des énumérations : 42](#_Toc36901922)

[2.3.3. Tableau des classes associations : 43](#_Toc36901923)

[2.3.4. Les relations entre les classes : 44](#_Toc36901924)

[3. Schéma de la base de données : 45](#_Toc36901925)

[3.1. Introduction : 45](#_Toc36901926)

[3.2. Règle de passage vers le modèle relationnel : 45](#_Toc36901927)

[3.2.1. Association un a plusieurs : 45](#_Toc36901928)

[3.2.1.1. Définition : 45](#_Toc36901929)

[3.2.1.2. Exemples : 46](#_Toc36901930)

[3.2.2. Association plusieurs à plusieurs : 49](#_Toc36901931)

[3.2.2.1. Définition : 49](#_Toc36901932)

[3.2.2.2. Exemple : 49](#_Toc36901933)

[3.2.3. En cas de généralisation : 50](#_Toc36901934)

[3.2.3.1. Définition : 50](#_Toc36901935)

[3.2.3.2. Exemple : 51](#_Toc36901936)

[3.2.4. En cas d’énumération : 52](#_Toc36901937)

[3.2.4.1. Définition : 52](#_Toc36901938)

[3.2.4.2. Exemple : 52](#_Toc36901939)

[3.3. Schéma relationnel de la base de données : 53](#_Toc36901940)

[4. Scénarios d’exceptions : 54](#_Toc36901941)

[4.1. Introduction : 54](#_Toc36901942)

[4.2. Diagrammes d’activité : 54](#_Toc36901943)

[4.2.1. Définitions : 54](#_Toc36901944)

[4.2.2. Diagrammes d’activité du cas « S’authentifier » : 55](#_Toc36901945)

[4.2.3. Diagramme d'activité du cas « Consulter relever d’absence » : 56](#_Toc36901946)

[4.2.4. Diagramme d’activité du cas « Marquer la présence » : 57](#_Toc36901947)

[4.2.5. Diagramme d’activité du cas « Justifier absence » : 58](#_Toc36901948)

[4.2.6. Diagramme d’activité du cas « Consulter justification » : 59](#_Toc36901949)

[4.2.7. Diagramme d’activité du cas « Établir la liste des étudiants exclus » : 60](#_Toc36901950)

[5. Conclusion : 61](#_Toc36901951)

# ***Table des figures***

[Figure 1: Diagramme de contexte statique. 10](#_Toc36901953)

[Figure 2: Diagramme de cas d'utilisation global. 13](#_Toc36901954)

[Figure 3: Diagramme de cas d'utilisation « Package Etudiant ». 14](#_Toc36901955)

[Figure 4 : Diagramme de cas d’utilisation « Package Enseignant ». 15](#_Toc36901956)

[Figure 5: Diagramme de cas d’utilisation « Package Administrateur ». 16](#_Toc36901957)

[Figure 6: Diagramme de cas d’utilisation « Package Responsable de formation ». 16](#_Toc36901958)

[Figure 7: Diagramme de cas d’utilisation « Package Chef département ». 17](#_Toc36901959)

[Figure 8: Diagramme de séquence système du cas « S’authentifier ». 20](#_Toc36901960)

[Figure 9: Diagramme de séquence système du cas « Consulter relever d'absence ». 22](#_Toc36901961)

[Figure 10: Diagramme de séquence système du cas « Marquer la présence ». 25](#_Toc36901962)

[Figure 11: Diagramme de séquence système du cas « Justifier une absence ». 27](#_Toc36901963)

[Figure 12: Diagramme de séquence système du cas « Consulter justification ». 29](#_Toc36901964)

[Figure 13: Diagramme de séquence système du cas « Etablir la liste des étudiants exclus ». 32](#_Toc36901965)

[Figure 14 Diagramme de classe conceptuel 35](#_Toc36901966)

[Figure 15: Passage du DCL au modèle relationnel de l'association « Demande » entre « Etudiant » et « CongéAcadémique ». 46](#_Toc36901967)

[Figure 16: Passage du DCL au modèle relationnel de la composition entre « Absence » et « Justification ». 47](#_Toc36901968)

[Figure 17: Passage du DCL au modèle relationnel de l'association « Concerne » entre « Notification » et « SéanceSupp ». 48](#_Toc36901969)

[Figure 18: Passage du DCL au modèle relationnel de l'association « Enseigne » entre « Enseignant » et « Module ». 49](#_Toc36901970)

[Figure 19: Passage du DCL au modèle relationnel de la généralisation entre « Utilisateur » et « Etudiant ». 51](#_Toc36901971)

[Figure 20: Passage du DCL au modèle relationnel de l'énumération « CodeDépartement ». 52](#_Toc36901972)

[Figure 21: Schéma relationnel de la base de données. 53](#_Toc36901973)

[Figure 22: Diagramme d'activité du cas « S'authentifier ». 55](#_Toc36901974)

[Figure 23: Diagramme d'activité du cas « Consulter relever d'absence ». 56](#_Toc36901975)

[Figure 24: Diagramme d'activité du cas « Marquer la présence ». 57](#_Toc36901976)

[Figure 25: Diagramme d'activité du cas « Justifier absence ». 58](#_Toc36901977)

[Figure 26: Diagramme d'activité du cas « Consulter justification ». 59](#_Toc36901978)

[Figure 27: Diagramme d'activité du cas « Établir la liste des étudiants exclus ». 60](#_Toc36901979)

# Table des tableaux

[Tableau 1: Acteurs du système et leurs tâches. 11](#_Toc36902003)

[Tableau 2: Fiche descriptive du cas « S’authentifier ». 19](#_Toc36902004)

[Tableau 3: Fiche descriptive du cas « Consulter relever d'absence ». 21](#_Toc36902005)

[Tableau 4: Fiche descriptive du cas « Marquer la présence ». 24](#_Toc36902006)

[Tableau 5: Fiche descriptive du cas « Justifier une absence ». 26](#_Toc36902007)

[Tableau 6: Fiche descriptive du cas « Consulter justification ». 28](#_Toc36902008)

[Tableau 7: Fiche descriptive du cas « Établir la liste des étudiants exclus ». 31](#_Toc36902009)

[Tableau 8: Description des classes du diagramme de classes conceptuel. 41](#_Toc36902010)

[Tableau 9: Description des énumérations du diagramme de classe conceptuel. 42](#_Toc36902011)

[Tableau 10 Description des classes associations du diagramme de classes conceptuel. 43](#_Toc36902012)

# Introduction

Un logiciel informatique est un ensemble de programmes et de procédures nécessaires au fonctionnement d’un système informatique. Il existe plusieurs catégories de logicielles informatiques spécifiques à la résolution des problèmes de l’utilisateur. Le projet que nous allons réaliser s’inscrit dans la catégorie des logiciels de gestion administratif.

La création d’un logiciel informatique est une tâche intellectuelle et pratique qui s’effectue le plus souvent en équipe suivant une démarche et un processus de développement bien définie et encadré afin d’assurer le bon déroulement du projet tout au long de sa création. C’est ce que nous enseigne la discipline du ***Génie Logiciel.***

Depuis l'apparition des nouvelles technologies de l'information et de la communication, et leur entrée dans le domaine administratif, la gestion des tâches administrative est devenue plus facile, rapide et fiable. Les processus et systèmes les plus compliqués peuvent être réalisés via des systèmes informatiques qui assurent la rapidité, fiabilité et l’intégrité des données manipuler.

Dans la plupart des facultés algériennes actuelles voir leurs totalités et plus particulièrement dans la faculté des nouvelle technologies de l’information et de la communication de Constantine (NTIC) la gestion des absences des étudiants dans les séances de travaux dirigés et pratiques est un élément essentiel qui assure que les étudiants ont bien suivi leur formation et ainsi qu’ils sont qualifiés dans leurs disciplines respectives, gérer manuellement par les enseignants cette tâche engendre plusieurs problèmes, problèmes qui pourraient être éventuellement résolus par un système informatique qui gère automatiquement et dynamiquement les listes des étudiants et leurs absences dans les différentes matières.

Dans ce projet, nous allons modéliser et implémenter ce système permettant la gestion dynamique et automatique des absences au niveau d’une faculté (plus précisément la faculté des nouvelles technologies de l’information et de la communication NTIC).

Ce projet se décomposera en trois grands chapitres :

* Chapitre 1 : analyse des besoins.
* Chapitre 2 : conception.
* Chapitre 3 : implémentation.

# Chapitre 1 : Analyse des besoins

## Introduction :

Dans ce chapitre, nous allons présenter l’étape d’identification des besoins et spécification des fonctionnalités propre au processus de développement Unified Process (UP) par lequel notre projet sera piloté. Cette étape consiste à identifier avec précision les besoins des utilisateurs de l’application et à modéliser ces besoins avec des moyens textuels et graphiques.

Les interactions entre les acteurs et le système (au sein des cas d’utilisation) et entre les cas d’utilisations eux-mêmes seront représentés sous forme textuelle et graphique, ces représentations indiquent avec précision et sans ambiguïté le déroulement séquentiel des différents besoins dans notre système et faciliteront donc la compréhension de ce système pour les futurs chapitres.

À la fin de ce chapitre, nous disposerons des éléments suivants :

* Le cahier des charges.
* Les acteurs de notre système.
* Les besoins de notre système.
* Diagrammes de cas d’utilisations structuré en packages et global.
* Fiches descriptives des cas d’utilisations les plus important.
* Diagrammes de séquence système des cas d’utilisation les plus important.

## Expression des besoins

### Introduction :

Dans cette partie, nous allons définir les acteurs de notre système, c’est-à-dire les personnes ou systèmes externes intervenant dans notre propre système ainsi que les besoins de ce système et de ces acteurs, c’est-à-dire les fonctionnalités de ce système et les contraintes techniques et temporelles que nous devrons respecter.

L’expression des besoins est une étape cruciale du processus de développement Unified Process (UP), celle-ci servira de base au bon déroulement de notre projet.

### Cahier des charges :

#### Présentation de la faculté NTIC :

La faculté des Nouvelles Technologies de l’Information et de la Communication (NTIC) est une faculté disposant de plusieurs départements (IFA, Tronc commun MI, TLSI). Chacun de ces départements dispose de ces propres étudiants, professeurs, Chef de département, Administrateurs et Responsables de formation.

Dans chaque département, les étudiants sont organisés en sections, chaque section est composée de plusieurs groupes et chaque groupe dispose de ces propres travaux dirigés et pratiques.

Chaque travail (pratique ou dirigé) est assuré par un enseignant qui doit faire l’appelle « manuellement ».

L’étudiant peut s’il le souhaite présenter une justification à l’administration pour retirer son absence.

Chaque département dispose de son propre chef de département, ces propres administrateurs et de ces propres responsables de formation.

Un étudiant est considéré comme exclue s’il atteint le seuil de trois absences non justifiées ou cinq absences justifier.

Le système a pour but d’automatiser l’appelle « manuelle » et la gestion globale des absences (justifications, modification de dernière minute…) pour chaque département de la faculté NTIC.

#### Définition de la problématique :

Comment automatiser la tâche jusque-là manuelle du recueil des absences et globaliser la gestion des absences au niveau de la faculté et ainsi facilité et dynamiser cette tâche de gestion ?

#### Définition des objectifs :

* Accélération du processus d’appel des étudiants.
* Facilité, accélérer la justification des absences pour les étudiants.
* Suivi des absences des étudiants.
* Informatisation au niveau administratif des profils, modules et séances.

#### Cible du projet :

Le système ne vise pas le grand public, c’est-à-dire que seuls les intervenants directs dans la faculté peuvent interagir avec celui-ci, c’est-à-dire :

* Les enseignants.
* Les étudiants.
* Le chef du département.
* L’administration de la faculté.
* Responsable de formation.

#### Périmètre du projet :

* L’application doit être disponible en trois langues : Français, English et العربية.
* L’application doit fonctionner dans un environnement web (site internet) et mobile (Android).

#### Contraintes techniques :

* Le système doit être sous forme d’application web.
* Le système doit adopter une architecture Modèle-Vue-Contrôleur (MVC).
* Certaines fonctionnalités du système notamment la partie Étudiant et Enseignant doivent être également disponible sur téléphone mobile.
* Le système devra être hébergé.

#### Méthodologie de conception :

* Le projet devra être réalisé en appliquant la méthode Unified Process (UP).
* Le projet devra être accompagné des différents rapports de conception (Diagrammes UML, Documentation, …).

### Identification des acteurs et des besoins :

#### Introduction :

Cette étape marque effectivement le point de départ de la conception du projet. Nous allons dans cette partie définir les acteurs de notre système, c’est-à-dire les personnes humaines ou les systèmes externes qui vont interagir directement avec notre système.

Une fois les acteurs bien définis, il s’agira de définir leurs tâches (leurs fonctions dans le système), ces tâches représentent les besoins fonctionnels de notre système et plus effectivement les cas d’utilisation de notre système.

Il est impératif que cette partie soit bien réaliser, la suite de notre projet en est totalement dépendante.

#### Indentification des acteurs :

##### Diagramme de contexte statique :

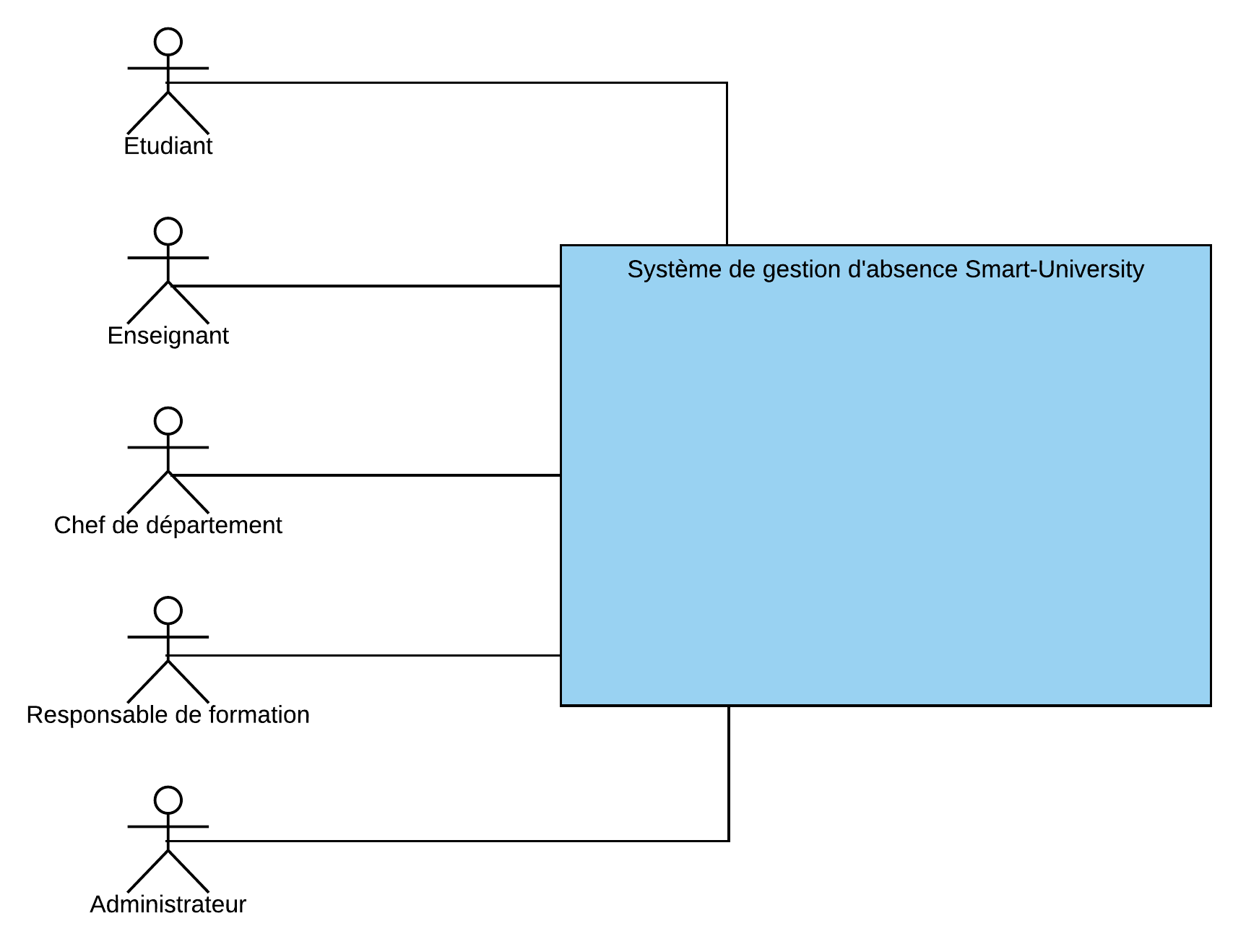


Figure : Diagramme de contexte statique.

##### Définition des acteurs et de leurs tâches :

|  |  |  |
| --- | --- | --- |
| Acteurs | Taches | Utilisabilité |
| Étudiant | * Consulter le relevé des absences. * Justifier son absence. * Gérer son profil utilisateur. * Gérer son emploi du temps. * Demander un congé académique. | * Sur le site internet. * Sur l’application mobile via Internet. * Sur l’application mobile sans Internet. |
| Enseignant | * Modifier son profil. * Supprimer son profil. * Faire l’appelle pour un groupe. * Modification du relevé des absences. * Enregistrement de la Justification d’une absence d’un étudiant. * Consultation du relevé des absences d’un groupe ou d’un étudiant. * Établissement de la liste des étudiants exclus. * Consultation des statistiques pour ses groupes. * Demander la modification d’une séance. * Demander l’ajouter d’une séance supplémentaire. | * Sur le site internet. * Sur l’application mobile via internet. * Sur l’application mobile sans Internet. |
| Administrateur | * Ajout, modification et suppression des profils de :   Modules,  Étudiants,  Enseignants,  Séances. | * Sur le site internet. |
| Responsable de Formation | * Les tâches d’Enseignant. * Consultation des statistiques. | * Sur le site internet. |
| Chef de Département | * Les tâches d’Enseignant. * Affectation des séances aux enseignants. * Gestion des justifications. * Établissement de la liste des étudiants exclus. * Consultation des statistiques. * Gestion des demandes des enseignants et des étudiants. | * Sur le site internet. |

Tableau 1: Acteurs du système et leurs tâches.

#### Définition des besoins fonctionnels :

* Gestion des comptes des étudiants et des enseignants.
* Gestion des modules, séances et justifications.
* Affectation des séances aux enseignants.
* Établissement de la liste des étudiants exclus.
* Consultation des statistiques.
* Marquer la présence (faire l’appel).
* Modification du relevé d’absences d’un étudiant.
* Enregistrement de la Justification d’une absence d’un étudiant.
* Consultation du relevé des absences d’un groupe ou d’un étudiant.
* Établissement de la liste des étudiants exclus.
* La demande et l’ajout de séance supplémentaire.
* La demande et la gestion des modifications de séance supplémentaires.

#### Définition des besoins non-fonctionnels :

* Authentification des utilisateurs.
* Sécurité : il faut garantir la sécurité d’accès aux différents comptes.
* Ergonomie : l'application web et l’application mobile offrent une interface conviviale et facile à utiliser.
* Fiabilité : l’application est fiable, les données sont correctement traitées et sauvegarder et leurs intégrités n’est pas compromis.
* Extensibilité : le code est clair, bien commenter et structurer pour permettre des futures évolutions du logiciel.

## Spécification des besoins :

### Introduction :

Dans cette partie nous allons modéliser visuellement les besoins définis dans l’expression des besoins à l’aide d’un langage de modélisation (le langage UML). Ainsi nous disposerons de diagrammes de cas d’utilisations qui représenterons nos besoins d’une façon abstraite (non détailler). Ces diagrammes serviront par la suite à la réalisation de la description textuelle détailler des différents cas d’utilisation.

### Diagramme de cas d’utilisation :

#### Définitions :

Un diagramme de cas d’utilisation est un diagramme défini dans le langage UML, il est utilisé pour donner une vision globale du comportement fonctionnel de notre système et ainsi les interactions entre les acteurs du système, les acteurs et les cas d’utilisation et les cas d’utilisation entre eux même.

Un acteur est une entité externe au système, une personne humaine, une machine ou un autre système qui appliquera une ou plusieurs de ces fonctionnalités (cas d’utilisations).

Un cas d’utilisation représente une interaction distincte entre l’un des acteurs (humain ou machine) et le système et est donc une unité significative de travail que devra accomplir le système.

#### Diagramme de cas d’utilisation globale :

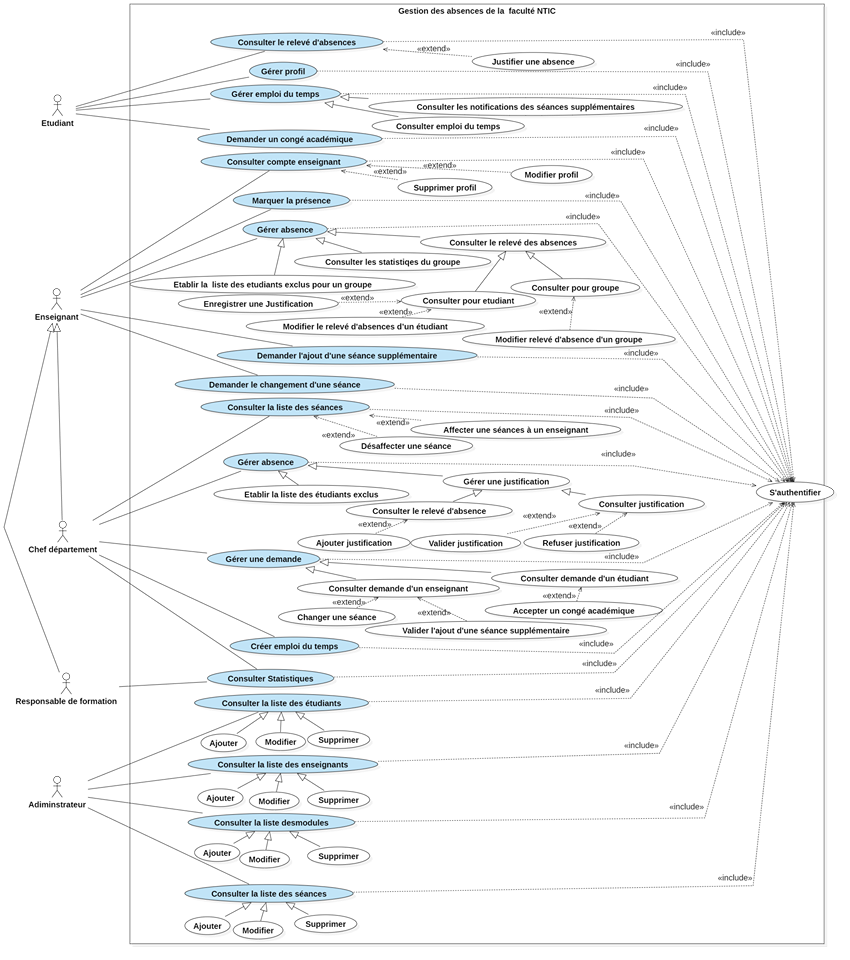


Figure : Diagramme de cas d'utilisation global.

#### Diagramme de cas d’utilisation structuré en packages :

Les diagrammes suivants représentent une version du diagramme de cas d’utilisation présenté plus haut décomposée selon les acteurs du système.

##### Package Etudiant :

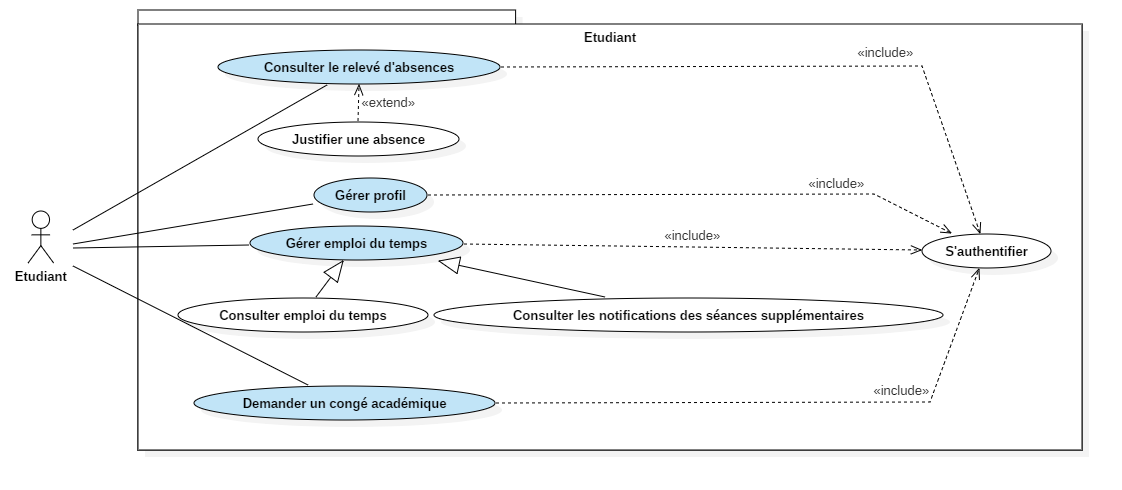
****

Figure : Diagramme de cas d'utilisation « Package Etudiant ».

**Commentaire :**

* Le cas **demander un congé académique** offre la possibilité à l’étudiant de « bloquer l’année en cours » afin de ne plus être considéré comme absent par les autres acteurs du système et le système lui-même, et ainsi ne plus apparaître inutilement dans les listes d’étudiants exclus.
* La **gestion de l’emploi du temps** offre la possibilité à l’étudiant de consulter l’emploi du temps de son groupe, mais également de connaître les éventuelles séances supplémentaires ajoutées par ces enseignants et ainsi être au courant des changements sur les horaires de ces activités éducatifs directement sur l’application.

##### Package Enseignant :

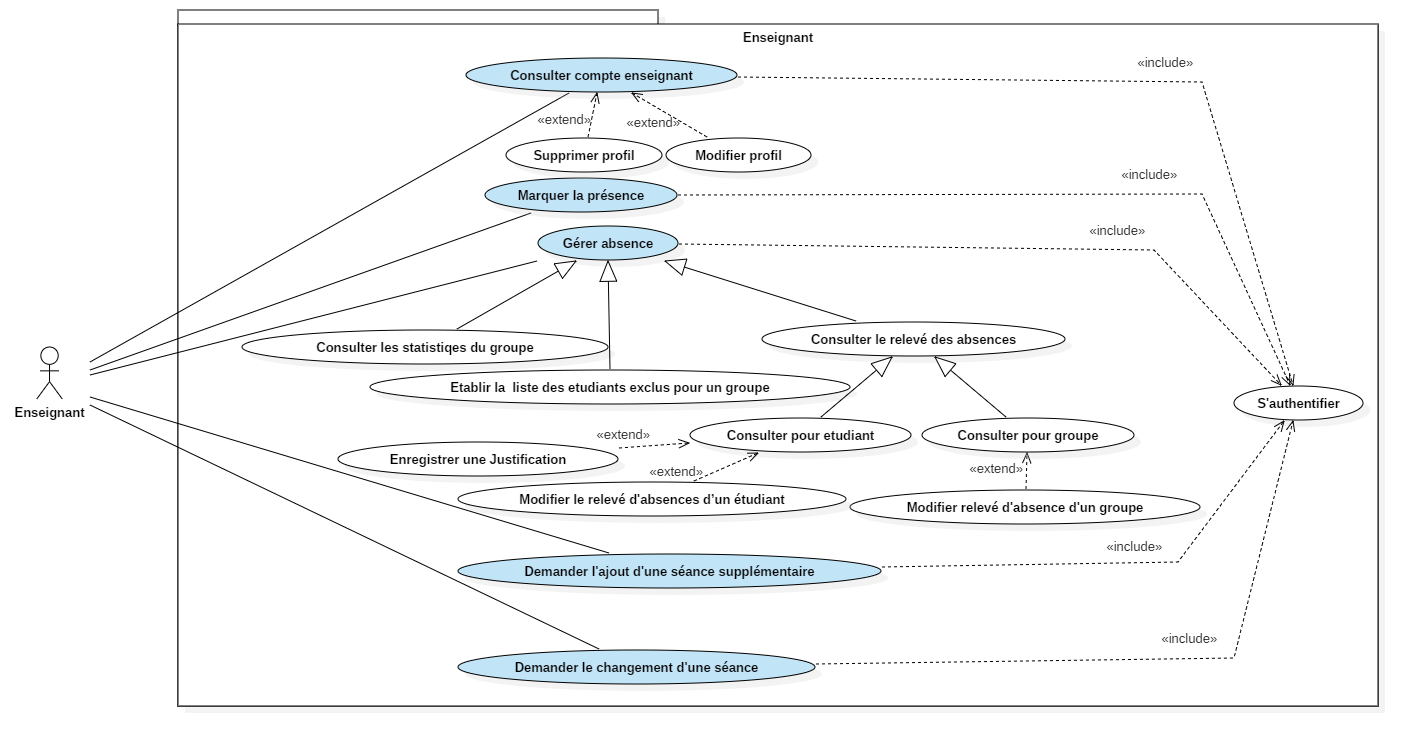
****

Figure  : Diagramme de cas d’utilisation « Package Enseignant ».

**Commentaire :**

* Le cas **demander une séance supplémentaire** à l’enseignant la possibilité d’envoyer directement au chef de département une demande d’ajout de séance pour son module pour un groupe précis. Les demandes de **changement de séances** fonctionnent de la même façon sauf qu’il ne s’agît pas d’ajouter, mais de modifier une séance.
* **L’ajout de justification** dans ce diagramme signifie qu’un enseignant a la possibilité d’ajouter des justifications pour ces étudiants (si davantage, ils ne pouvaient pas le faire eux même pour une raison quelconque par exemple).

##### Package Administrateur :

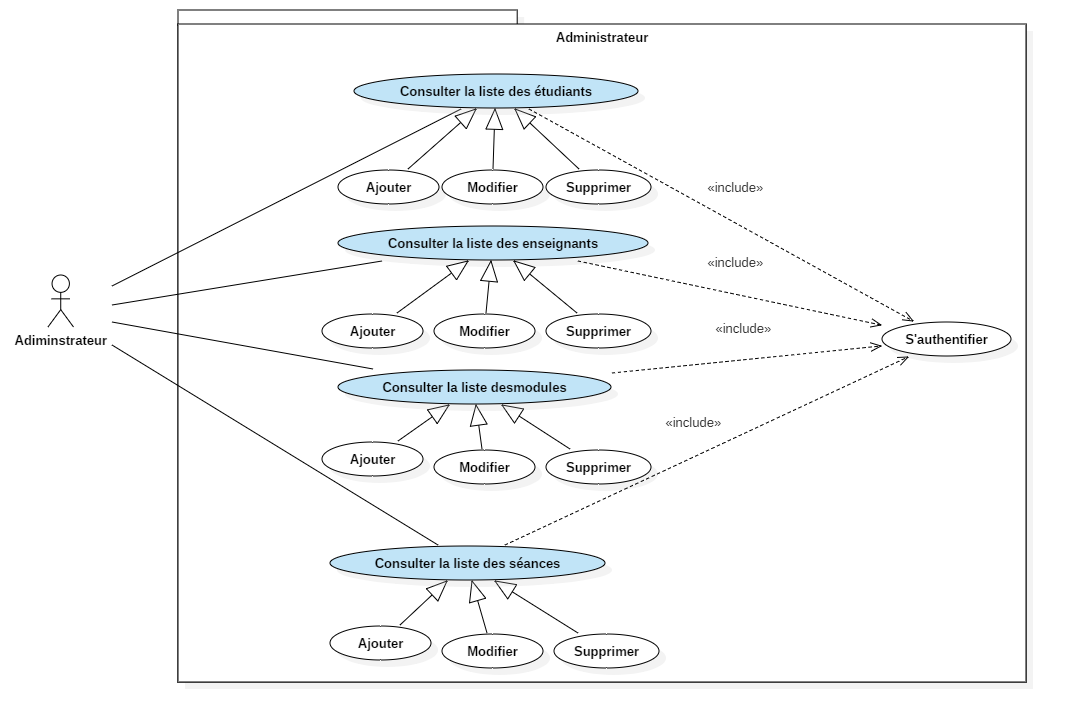


Figure : Diagramme de cas d’utilisation « Package Administrateur ».

##### Package Responsable de formation :

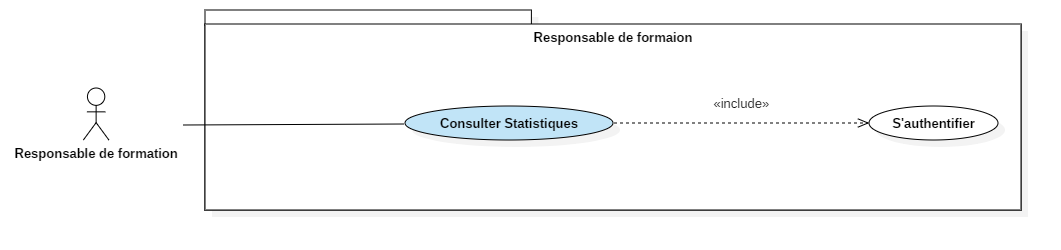


Figure : Diagramme de cas d’utilisation « Package Responsable de formation ».

##### Package Chef de département :

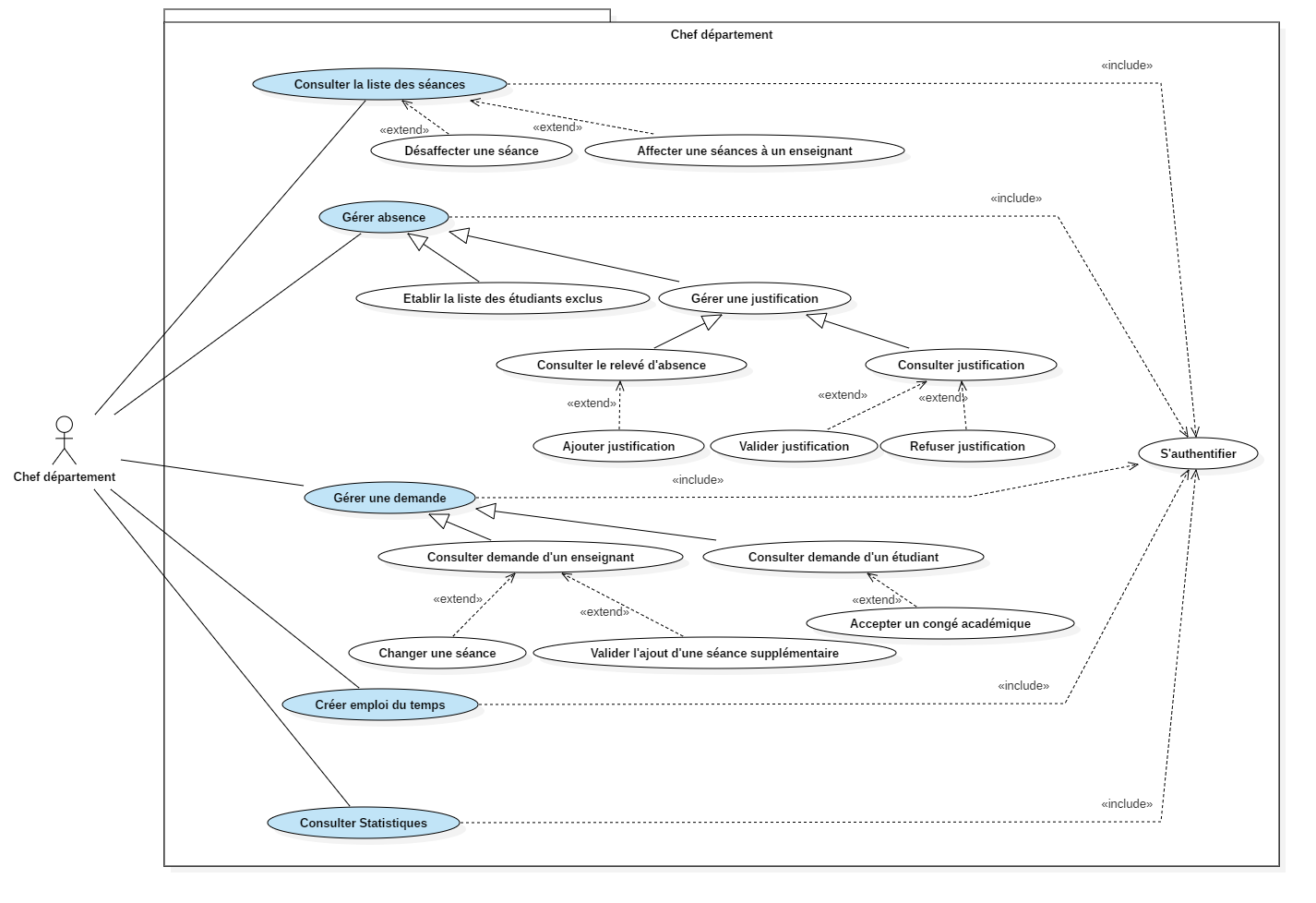


Figure : Diagramme de cas d’utilisation « Package Chef département ».

**Commentaire :**

* La **création d’un emploi du temps** permet au chef de département la création l’emploi du temps pour chacune des spécialités de son département, emploi du temps que l’administrateur consultera et auquel il ajoutera les séances pédagogiques.
* **La gestion des demandes** offres la possibilité au chef de département de gérer les demandes des étudiants (congé académique par exemple) ou enseignants (modification de séances, ajouter de séances supplémentaires) et d’ainsi les accepter ou les refuser.

## Descriptions textuelles et diagrammes de séquence système :

### Introduction :

Cette partie est consacré à détailler les cas d’utilisations de notre système à la fois textuellement et graphiquement.

Textuellement ces cas d’utilisations seront représentés par une fiche descriptive servant à analyser les acteurs principaux et secondaires, les objectifs et les scénarios de ces cas d’utilisation.

Quant à elle la description graphique elle représentera un diagramme de séquence système défini dans le langage UML, il montrera très exactement les interactions entre le système et l’acteur tout au long du cas d’utilisation et ainsi donner une idée sur la structure générale du programme réalisant ce cas d’utilisation.

Dans cette partie nous ne traiterons que les cas d’utilisations jugé important pour notre système, c’est-à-dire la gestion des absences des étudiants de la faculté NTIC.

Les cas que nous traiterons sont les suivants :

* **S’authentifier :** ce cas est important d’un point de vue non-fonctionnel, seuls les acteurs du système ont la possibilité d’utiliser l’application et donc ils doivent être authentifiés avant d’avoir accès au système.
* **Consulter le relevé des absences :** chaque étudiant doit avoir la possibilité de consulter ses absences, l’un des points fondamentaux du système est qu’un étudiant qui a au moins trois absences non justifier ou cinq absences justifier est exclu, il faut donc offrir la possibilité aux étudiants de consulter leurs relevés d’absences.
* **Justifier une absence :** la possibilité de justifier une absence est également fondamentale au système. Elle permet à l’étudiant de retirer l’une de ces absences en la justifiant.
* **Marquer la présence :** le point clé de notre système, l’objectif principal du projet est de faciliter la tâche lente et peut fiable qu’est de marquer la présence manuellement. Il est indispensable donc de détailler ce point.
* **Etablir la liste des étudiants exclus :** un point fondamental du système.
* **Consulter une justification :** permettre au chef de d’un département de valider les justifications est impératif au bon fonctionnement du système, sinon n’importe qui pourrait justifier son absence.

### Descriptions textuelles et graphiques :

#### Cas : « S’authentifier »

##### Fiche descriptive :

|  |  |
| --- | --- |
| Nom du cas | S’authentifier |
| Type | Interne. |
| Acteurs principaux | * Etudiant. * Enseignant. * Chef de département. * Responsable de formation. * Administrateur. |
| Acteur secondaire | / |
| Objectif | Permettre à l’utilisateur de se connecter à l’application web ou à l’application mobile. |
| Préconditions | * L’Utilisateur doit disposer d’un compte. |
| Scénario nominal | 1. L’utilisateur clique sur le bouton « S’authentifier ». 2. Le système affiche un formulaire d’authentification. 3. L’utilisateur remplit le formulaire (nom d’utilisateur et mot de passe). 4. Le système vérifie les données saisies. 5. Le système renvois l’utilisateur vers une page d’accueil. |
| Scénarios alternatifs | Scénario Alternatif 1 (A1) :  Démarre au point numéro 4 du scénario nominal (SN).  Les données saisies sont incorrectes.   1. Le système affiche un message d’erreur.   L’enchaînement reprend au point numéro 2 du scénario nominal (SN).  Scénario Alternatif 2 (A2) :  Démarre au point numéro 4 du scénario nominal (SN)  Le compte utilisateur est sécurisé par une confirmation avec un code.   1. Le système envoi un code dans la boite mail de l’utilisateur. 2. Le système affiche une barre de saisie. 3. L’utilisateur saisi le code confirmation. 4. Le système vérifie le code saisi.   L’enchaînement reprend au point numéro 5 du scénario nominal. |
| Scénarios d’exceptions | / |
| Post conditions | * Une session utilisateur est actif sur l’application web ou l’application mobile. |

Tableau : Fiche descriptive du cas « S’authentifier ».

##### Diagramme de séquence système :

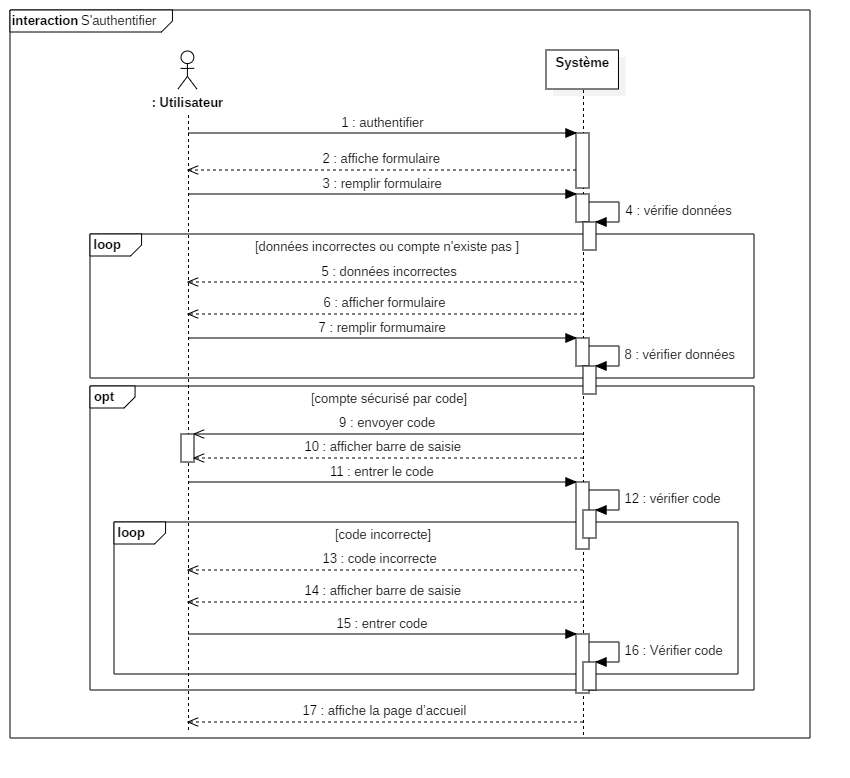


Figure : Diagramme de séquence système du cas « S’authentifier ».

#### Cas : « Consulter relever d’absence »

##### Fiche descriptive :

|  |  |
| --- | --- |
| Nom du cas | Consulter relever d’absence |
| Type | Principal. |
| Acteur principal | Étudiant |
| Acteur secondaire | / |
| Objectif | Permettre à l’étudiant de consulter son relevé d’absence. |
| Préconditions | * L’Étudiant doit être authentifié. |
| Scénario nominal | 1. L’étudiant clique sur le bouton consulter le relevé d’absence. 2. Le système vérifie l’état de l’étudiant. 3. Le système affiche une liste des absences de l’étudiant. |
| Scénarios alternatifs | / |
| Scénarios d’exceptions | Scénario d’exception 1 (E1)  Démarre au point numéro 1 du scénario nominal (SN).  L’étudiant est bloqué à cause d’un congé académique.   1. Le système affiche un message d’erreur. |
| Post conditions | * Le relevé d’absence est affiché à l’étudiant. |

Tableau : Fiche descriptive du cas « Consulter relever d'absence ».

##### Diagramme de séquence système :

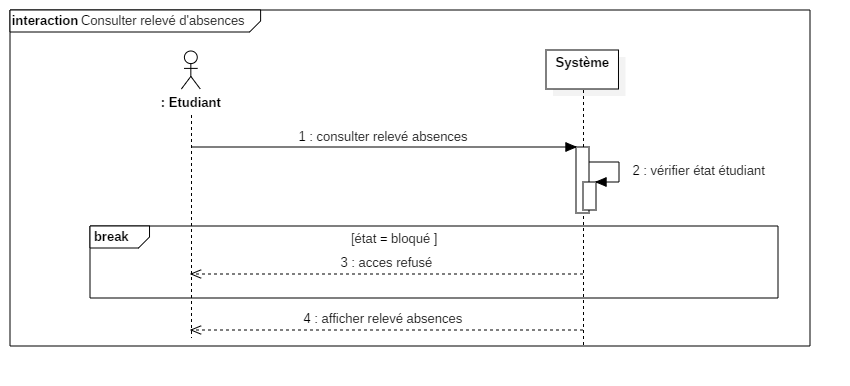


Figure : Diagramme de séquence système du cas « Consulter relever d'absence ».

#### Cas : « Marquer la présence »

##### Fiche descriptive :

|  |  |
| --- | --- |
| Nom du cas | Marquer la présence |
| Type | Principal. |
| Acteur principal | Enseignant. |
| Acteur secondaire | / |
| Objectif | Permettre à l’enseignant d’enregistrer les étudiants présents dans l’une de ces séances. |
| Préconditions | * L’Enseignant doit être authentifié. * L’Enseignant a sélectionné un de ces modules. |
| Scénario nominal | 1. L’enseignant clique sur un bouton « Marquer la présence ». 2. Le système affiche une liste qui contient les différentes méthodes de collecte de présence. 3. L’enseignant choisit l’option « Prise de présence manuelle ». 4. Le système affiche une liste des différents groupes de l’enseignant. 5. L’enseignant choisit l’un de ces groupes. 6. Le système affiche la liste des étudiants du groupe. 7. L’enseignant clique sur un bouton « Présent » devant l’étudiant. 8. L’enseignant clique le bouton « valider ». 9. Le système enregistre l’opération dans la base de données |
| Scénarios alternatifs | Scénario alternatif 1 (A1)  Démarre au point numéro 2 du scénario nominal (SN).  L’enseignant choisi de marquer la présence via un code QR.   1. L’enseignant choisi l’option « Prise de présence via un code QR ». 2. L’enseignant clique le bouton « Crée un code QR ». 3. Le système affiche le code QR.   L’enchaînement reprend au point numéro 8 du scénario nominal (SN).  Scénario alternatif 2 (A2)  Démarre au point numéro 2 du scénario nominal (SN).  L’enseignant choisi la prise de présence via des cartes.   1. L’enseignant choisi l’option « Prise de présence via des cartes ». 2. Le système active le lecteur de cartes.   L’enchaînement reprend au point numéro 8 du scénario nominal (SN).  Scénario alternatif 3 (A3)  Démarre au point numéro 2 du scénario nominal (SN).  L’enseignant choisi d’importer une liste d’étudiants.   1. L’enseignant choisi l’option « Importer un groupe d’étudiants ». 2. L’enseignant sélectionne un groupe. 3. L’enseignant choisi un fichier et l’uploade.   L’enchaînement reprend au point numéro 8 du scénario nominal. |
| Scénarios d’exceptions | / |
| Post conditions | * Une mise à jour de la liste des étudiants présents est enregistrée dans le système. * Une mise à jour de la liste des étudiants absents est enregistrée dans le système. |

Tableau : Fiche descriptive du cas « Marquer la présence ».

##### Diagramme de séquence système :

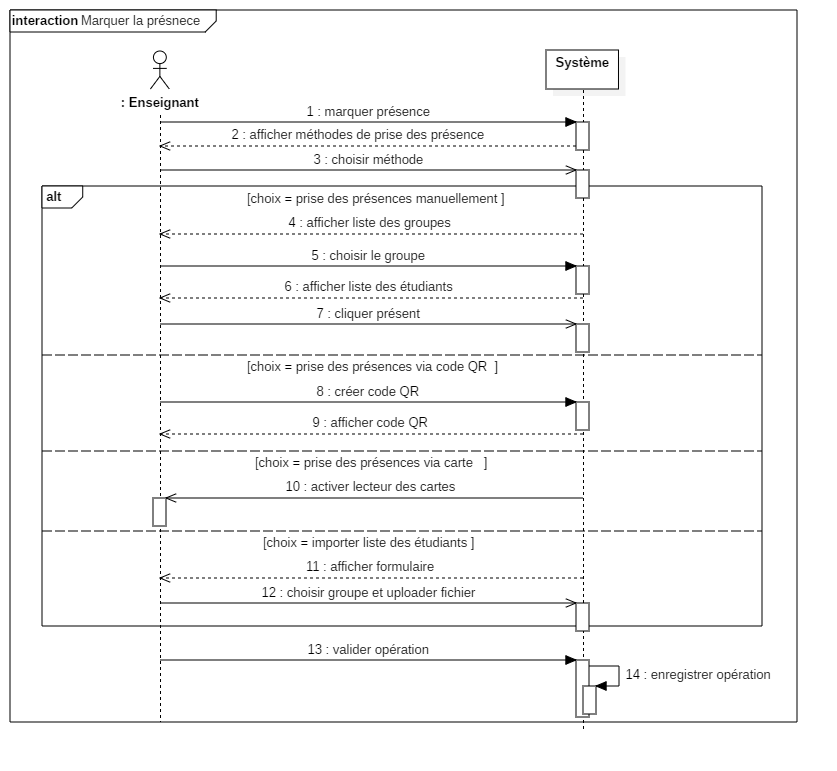


Figure : Diagramme de séquence système du cas « Marquer la présence ».

#### Cas : « Justifier une absence » :

##### Fiche descriptive :

|  |  |
| --- | --- |
| Nom du cas | Justifier une absence. |
| Type | Interne. |
| Acteur principal | Étudiant. |
| Acteur secondaire | / |
| Objectif | Permettre à l’étudiant de justifier l’une de ces absences pour l’un des modules qu’il étudie. |
| Préconditions | * L’Étudiant doit être authentifié. * L’Étudiant a consulté son relevé d’absence et a sélectionné l’une de ces absences |
| Scénario nominal | 1. L’étudiant clique sur un bouton « justifier l’absence ». 2. Le système demande à l’étudiant d’entrée la donnée nécessaire à la création d’un justificatif d’absence. 3. L’étudiant entre les données (certificat médical, convocation, …). 4. Le système valide les données. 5. Le système crée un justificatif d’absence. 6. Le système transmet le justificatif au chef de département. 7. Le système affiche un message de succès à l’étudiant. |
| Scénarios alternatifs | Scénario Alternatif 1 (A1) :  Démarre au point numéro 3 du scénario nominal (SN)  Le système ne valide pas les données saisies par l’Etudiant   1. Le système affiche un message d’échec à l’étudiant (Causes du refus).   L’enchaînement reprend au point numéro 2 du scénario nominal (SN). |
| Scénarios d’exceptions | / |
| Post conditions | * Le justificatif est sauvegardé. * Le justificatif est transmis au chef de département. |

Tableau : Fiche descriptive du cas « Justifier une absence ».

##### Diagramme de séquence système :

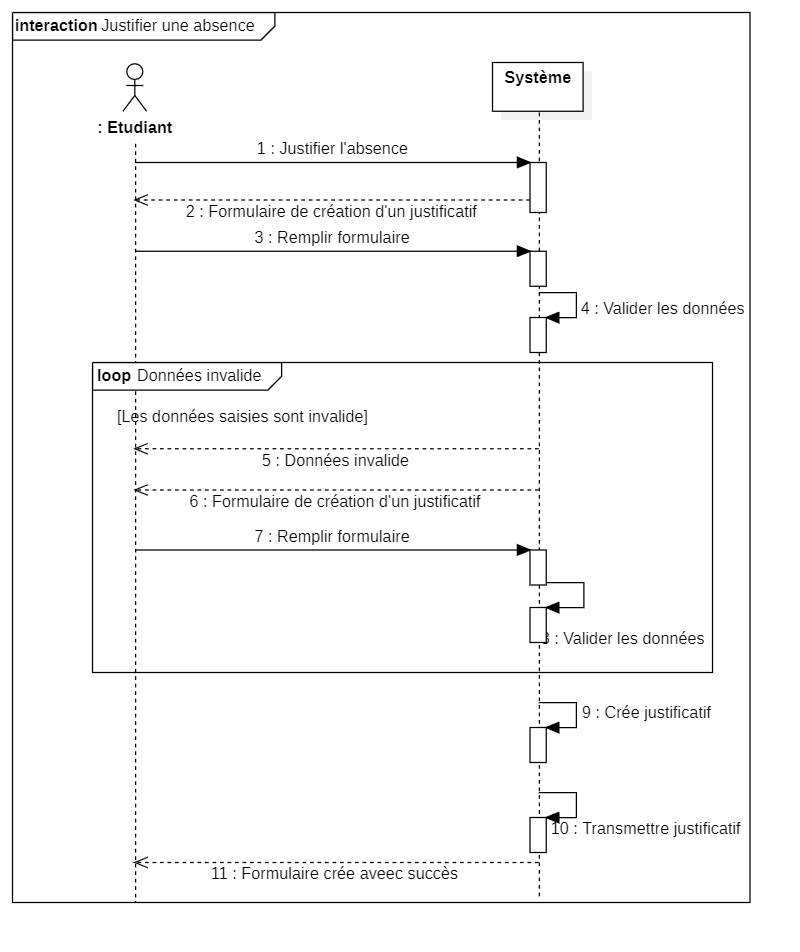


Figure : Diagramme de séquence système du cas « Justifier une absence ».

#### Cas : « Consulter une justification »

##### Fiche descriptive :

|  |  |
| --- | --- |
| Nom du cas | Consulter une justification |
| Type | Interne. |
| Acteur principal | Chef de département. |
| Acteur secondaire | / |
| Objectif | Permettre au chef de département de voir les justificatifs d’absences des étudiants et de les valider ou de les refuser. |
| Préconditions | * Le Chef de département doit être authentifié. |
| Scénario nominal | 1. Le chef de département clique sur un bouton « Consulter la liste des justifications ». 2. Le système affiche la liste des justifications au chef de département. 3. Le chef département choisi une justification et clique dessus. 4. Le système affiche les détails de cette justification d’absence (Nom de l’étudiant/e, groupe, module…). |
| Scénarios alternatifs | Scénario Alternatif 1 (A1) :  Démarre au point numéro 4 du scénario nominal (SN).  Le chef de département valide une justification :   1. Le chef de département clique sur un bouton « Valider ». 2. Appel au cas « Valider justification ».   Scénario Alternatif 2 (A2) :  Démarre au point numéro 4 du scénario nominal (SN).  Le chef de département refuse une justification :   1. Le chef de département clique sur un bouton « Refuser ». 2. Appel au cas « Refuser justification ». |
| Scénarios d’exceptions | / |
| Post conditions | * Le chef de département consulte la liste des justifications. * Les refus/acceptations sont enregistré dans le système. |

Tableau : Fiche descriptive du cas « Consulter justification ».

##### Diagramme de séquence système :

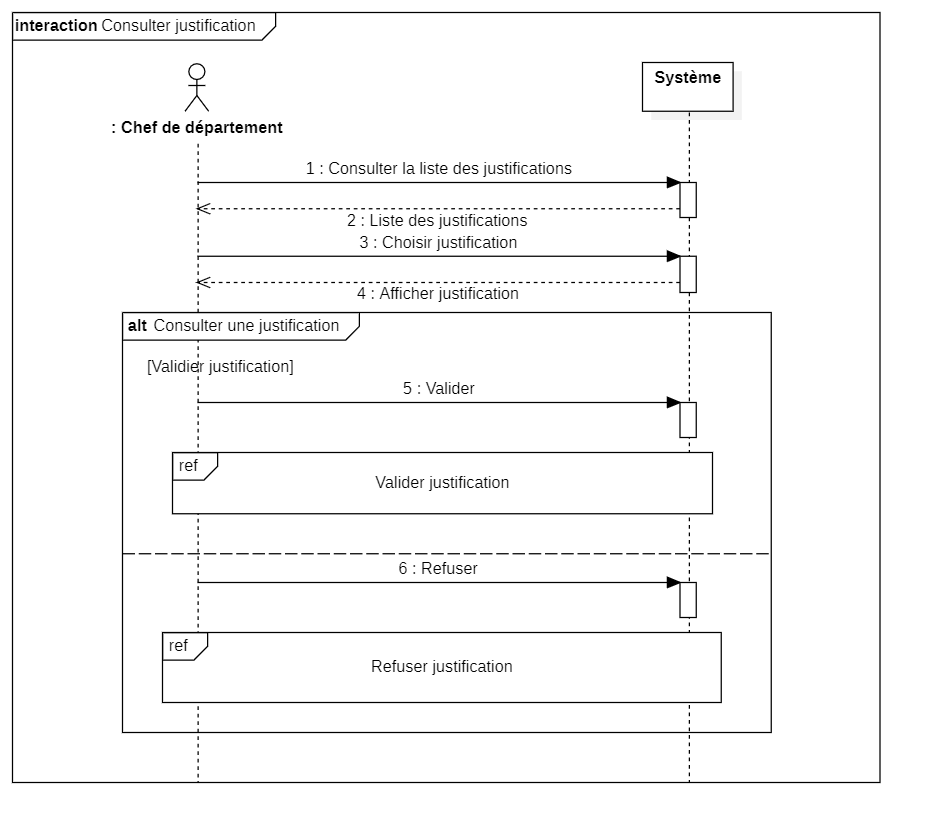


Figure : Diagramme de séquence système du cas « Consulter justification ».

#### Cas : « Établir la liste des étudiants exclus »

##### Fiche descriptive :

|  |  |
| --- | --- |
| Nom du cas | Établir la liste des étudiants exclus. |
| Type | Interne. |
| Acteur principal | Chef de département. |
| Acteur secondaire | / |
| Objectif | Permettre au chef de département d’établir la liste des étudiants exclus. |
| Préconditions | * Le Chef de département doit être authentifié. |
| Scénario nominal | 1. Le chef de département clique sur un bouton « Établir la liste des étudiants exclus ». 2. Le système affiche une page dans laquelle le chef de département peut établir une liste des étudiants exclus pour le département entier, une spécialité particulière ou un groupe particulier. 3. Le chef de département choisi d’établir la liste des étudiants exclu pour le département. 4. Le système génère la liste des étudiants exclus selon le choix du chef de département. 5. Le système affiche la liste des étudiants au chef de département. |
| Scénarios alternatifs | Scénario Alternatif 1 (A1) :  Démarre au point numéro 2 du Scénario nominal (SN).  Le chef de département choisi d’établir la liste des étudiants exclus pour une spécialité.   1. Le chef de département choisi d’établir la liste des étudiants exclus pour une spécialité particulière. 2. Le système demande au chef de département de choisir l’une des spécialités de son département. 3. Le chef de département choisi une spécialité.   L’enchaînement reprend au point numéro 4 du Scénario nominal (SN).  Scénario Alternatif 2 (A2) :  Démarre au point numéro 2 du Scénario nominal (SN).  Le chef de département choisi d’établir la liste des étudiants exclus pour un groupe particulier.   1. Le chef de département choisi d’établir la liste des étudiants exclus pour un groupe particulier. 2. Le système demande au chef de département de choisir l’une des spécialités de son département. 3. Le chef de département choisi une spécialité. 4. Le système demande au chef de département de choisir l’un des groupes pour ce département pour cette spécialité. 5. Le chef de département choisi un groupe.   L’enchaînement reprends au point numéro 4 du Scénario nominal (SN). |
| Scénarios d’exceptions | / |
| Post conditions | * Le chef de département dispose d’une liste d’étudiants exclus. |

Tableau : Fiche descriptive du cas « Établir la liste des étudiants exclus ».

##### Diagramme de séquence système :

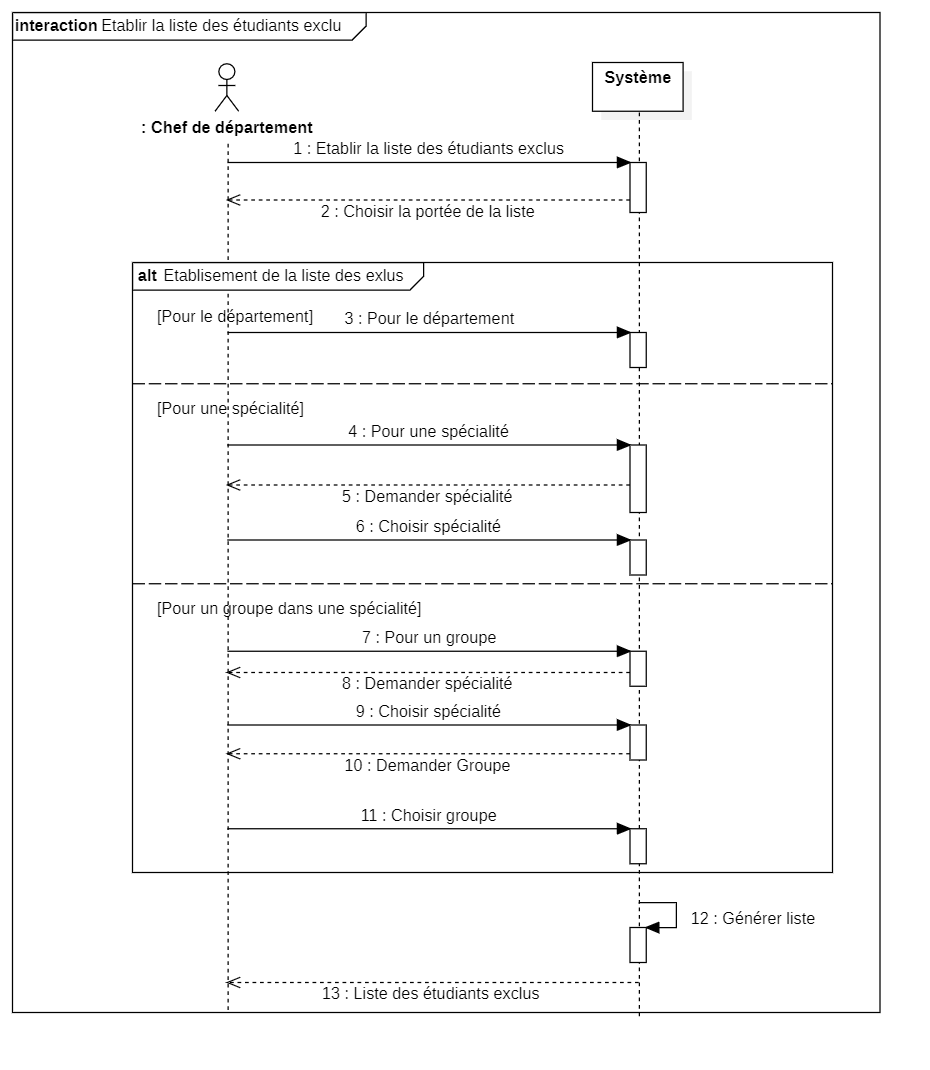


Figure : Diagramme de séquence système du cas « Etablir la liste des étudiants exclus ».

## Conclusion :

##### Ce qui a été réalisé dans ce chapitre :

Dans ce chapitre, nous avons dégagé les éléments principaux de la première phase du processus de développement Unified Process (UP) c’est-à-dire, l’analyse des besoins qui permettent de comprendre le domaine du projet, ces intervenants principaux et secondaires et les principales fonctionnalités ainsi que leur déroulement séquentiel.

Nous avons donc :

* Défini le domaine sur lequel nous allons travailler, la faculté NTIC.
* Défini les acteurs de notre système.
* Défini les besoins fonctionnels et non-fonctionnels du système.
* Défini les cas d’utilisations.
* Détaillé textuellement et graphiquement les cas d’utilisation les plus importants.

##### À quoi cela servira-t-il pour les chapitres suivants ?

Les différents éléments présentés jusque-là serviront de base essentielle aux éléments des chapitres suivants, la suite donc du processus de développement Unified Process (UP) par exemple :

* Le diagramme de classe.
* La conception de la base de données.
* Implémentation concrète du projet.

# Chapitre 2 : Conception

## Introduction :

Ce chapitre représente la deuxième phase du processus de développement Unified Process (UP), la conception. Dans ce chapitre nous allons mettre en évidence le coté conceptuel de notre projet, cette phase permettra de détailler les scénarios présentés dans le chapitre précèdent du point de vue dynamique.

Nous allons également modéliser du point de vue statique les attributs et les actions (méthodes) des différents acteurs et éléments de notre système via un diagramme de classes et les stocker dans une base de données relationnelle, pour cela nous allons appliquer les règles de passage vers le modèle relationnel sur le diagramme de classe obtenu.

À la fin de ce chapitre nous disposerons des éléments suivants :

* Diagramme de classe conceptuel.
* Schéma de la base de données.
* Diagrammes d’activités des scénarios présenté dans le chapitre précédent.

## Diagramme de classe conceptuel :

### Introduction :

Dans cette partie nous allons à l’aide des différentes descriptions textuelles et graphiques réaliser lors du premier chapitre définir et réaliser un diagramme de classe conceptuel pour notre application et ceci à l’aide d’un langage de modélisation (UML). Ce diagramme représentera les différents attributs extraits des descriptions textuelles et les différentes méthodes de nos acteurs (tel que l’étudiant) et des éléments du projet (par exemple une absence) extraits des diagrammes de séquence et les relations entre ces classes.

Ce diagramme servira ensuite à la réalisation d’un schéma d’une base de données pour notre application.

### Définitions :

Un diagramme de classe est un diagramme défini dans le langage de modélisation UML. Il permet de modéliser visuellement les systèmes en classes et interfaces.

Une classe décrit le type, le comportement et les attributs de chaque élément.

Une classe peut être en relation avec une autre classe via différents types de mécanisme tel que l’héritage.

Le diagramme de classe conceptuel ne vise qu’a modéliser les éléments réels du système et non pas le système en entier.

### Diagramme de classe conceptuel :

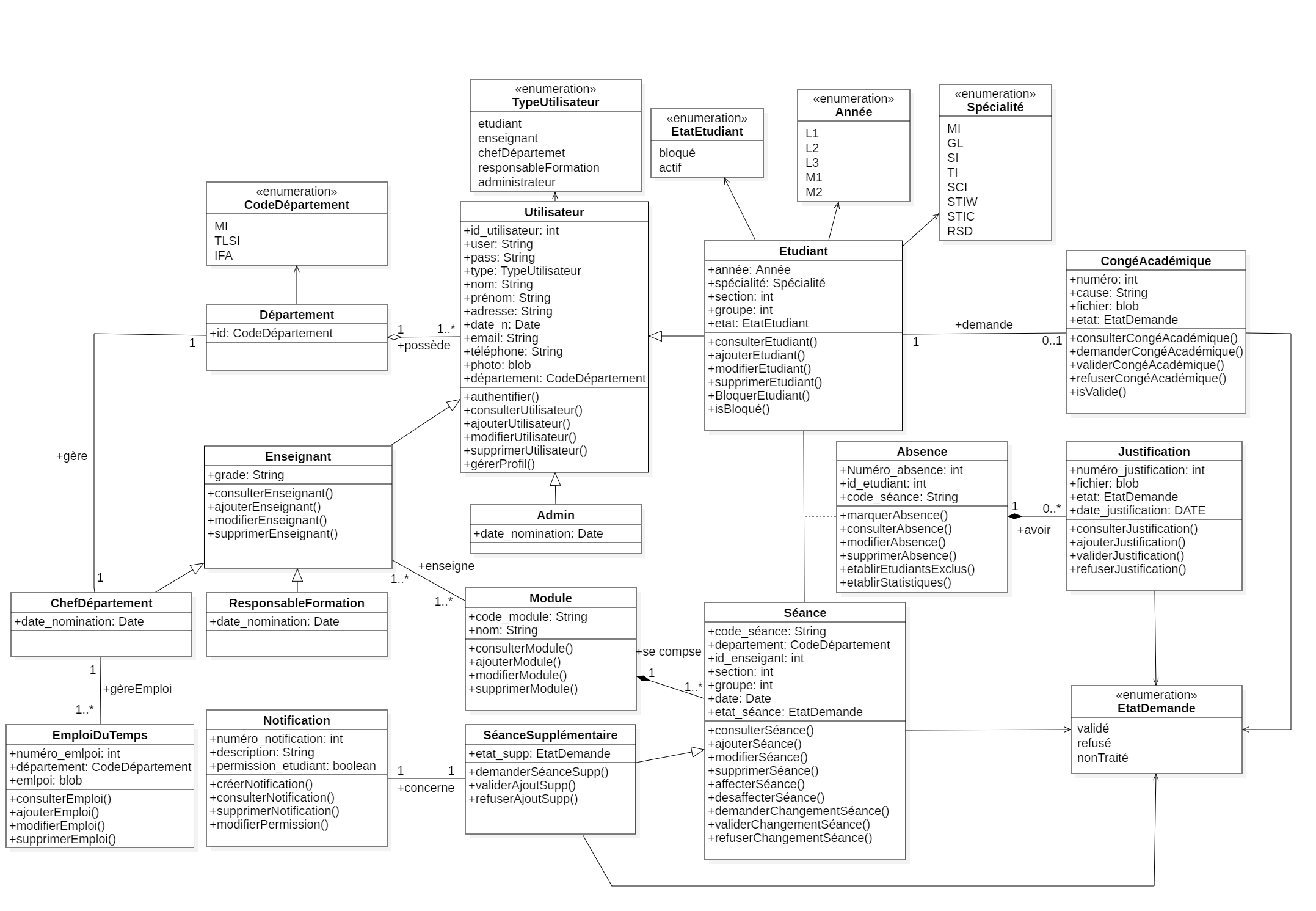


Figure Diagramme de classe conceptuel

#### Tableau des classes :

|  |  |  |
| --- | --- | --- |
| Classe | Description | Attributs et les méthodes |
| Département | Les différents départements de la faculté « NTIC ». | 1. Les attributs :  * **Id** : identifiant de la classe de type « CodeDépartement ». |
| Utilisateur | Les différents utilisateurs du système (les cinq acteurs définis dans le chapitre précédent) avec les attributs et les méthodes communes entre eux. | 1. Les attributs :  * **Id\_utilisateur** : identifiant de l’utilisateur, de type « Entier ». * **Nom**, **prénom**, **date\_n**, **email**, **adresse**, **téléphone**, **photo**, **département**, **user** et **pass** : représentent le nom, le prénom, la date de naissance, l’adresse email, le numéro de téléphone, la photo, le département, le nom d’utilisateur et le mot de passe de l’utilisateur en question dans l’ordre. * **Type\_utilisateur** : le type de l’utilisateur, de type « TypeUtilisateur ».  1. Les méthodes :  * **Authentifier** : permet à l’utilisateur de se connecter au site ou à l’application mobile après vérification du user et de la pass. * **ConsulterUtilisateur**, **AjouterUtilisateur**, **ModifierUtilisateur**, **SupprimerUtilisateur** : permettent à l’admin la gestion des comptes utilisateur. * **GérerProfil** : permet à l’utilisateur de modifier quelques attributs tel que : pass, email… |
| Etudiant | Les étudiants de la faculté « NTIC », elle hérite de la classe Utilisateur, elle représente les attributs et les méthodes spécifiques à l’étudiant. | 1. Les attributs :  * **Année** : année d’inscription de l’étudiant, de type « TIME ». * **Spécialité**: l’actuelle spécialité de l’étudiant, de type « Spécialité ». * **Section**, **groupe**: respectivement la section et le groupe de l’étudiant, de type « Entier ». * **Etat**: l’état actuel de l’étudiant, de type « EtatEtudiant »  1. Les méthodes :  * **ConsulterEtudiant, ajouterEtudiant, modifierEtudiant, supprimerEtudiant**: permettent à l’admin la gestion des comptes des différents étudiants(ces méthodes sont appliquées seulement sur les attributs de la classe). * **IsBloqué**: retourne un résultat de type booléen, si l’état d’étudiant est « bloqué » le résultat est « true ». Si l’état d’étudiant est « actif » le résultat est « false ». * **BloquerEtudiant**: permet au chef de département de bloquer un étudiant (modifier son état de « actif » à « bloqué »). |
| CongéAcadémique | Représente les demandes de congés académiques des étudiants. | 1. Les attributs :  * **Numéro**: identifiant de chaque demande d’un congé académique, de type « Entier ». * **Cause**: définit une justification textuelle de la cause de la demande d’un congé académique, de type « Chaîne de caractère ». * **Fichier**: contient un fichier de justification comme un certificat médical. * **Etat**: il définit l’état de congé académique, de type « EtatDemande ».  1. Les méthodes :  * **DemanderCongéAcadémique** : permet à un étudiant de faire une demande d’un congé académique. * **ConsulterCongéAcadémique**, **validerCongéAcadémique**, **refuserCongéAcadémique**:ces méthodes permettent au chef de département la gestion des demandes des congés académiques. |
| Enseignant | Représente les enseignants de la faculté, elle hérite de la classe « Utilisateur » et définit les attributs et les méthodes spécifiant aux enseignants. | 1. Les attributs :  * **Grade :** représente le grade actuel de l’enseignant, de type « Chaîne de caractère ».  1. Les méthodes :  * **ConsulterEnseignant**, **ajouterEnseignant**, **modifierEnseignan**t, **supprimerEnseignant :** ces méthodes permettent à l’admin la gestion des comptes des différents enseignants(ces méthodes sont appliquées seulement sur les attributs de la classe). |
| Module | Représente les modules existant dans les différentesspécialités de la faculté**.** | 1. Les attributs :  * **Code\_module**: **l**’identifiant du module de type « Chaîne de caractère »**.** * **Nom**: le nom complet du module.  1. Les méthodes :  * **ConsulterModule**, **ajouterModule**, **modifierModule**, **supprimerModule** :ces méthodes permettent à l’admin la gestion des Modules. |
| Séance | Représente les séances de chaque module. | 1. Les attributs :  * **Code\_séance**: l’identifiant de la séance de type chaîne de caractère**.** * **Id\_enseignant :** identifie l’enseignant chargé de la séance. * **Spécialité, section, groupe, date**: la spécialité, la section, le groupe et la date de la séance en question. * **Etat**: les séances qui sont ajoutées par l’admin sont toujours validées. si un enseignant demande le changement d’une séance, cette demande est enregistrée dans cette classe, mais avec un état « non traité », elle n’est pas prise en considération jusqu'elle soit validé par le chef de département.  1. Les méthodes :  * **ConsulterSéance**, **ajouterSéance**, **modifierSéance**, **supprimerSéance**:ces méthodes permettent à l’admin la gestion des séances. * **AffecterSéance** : permet au chef département d’affecter une séance à un enseignant. * **DesaffecterSéance**: permet au chef département de désaffecter une séance d’un enseignant. * **DemanderChangementSéance**: permet à un enseignant de faire une demande de changement d’une séance. Cette séance est stockée dans la classe avec un état « non traité ». * **ValiderChangementSéance**: permet au chef département d’accepter la demande de changement d’une séance, la méthode modifie l’état de la séance demandé de « non traité » à « validé » ou change l’état de la séance à « refusé ». * **RefuserChangementSéance**: permetau chef département de refuser la demande du changement d’une séance, la méthode modifie l’état de la séance demandé de « non traité » à « refusé ». |
| SéanceSupp | Représente les demandes des séances supplémentaires, elle hérite de la classe séance. | 1. Les attributs :  * **Etat\_supp**: l’état de la séance supplémentaire fait la différence entre les séances supplémentaire qui sont validées ou refusées par le chef département, de type « EtatDemande ».  1. Les méthodes :  * **DemanderChangementSéance :** elle permet à un enseignant de faire une demande d’ajout d’une séance supplémentaire.   Cette séance est stockée dans la classe avec un état « non traité ».   * **ValiderAjoutSupp :** cette méthode permet au chef département d’accepter la demande d’ajout d’une séance supplémentaire, La méthode modifie l’état de la séance demandé de « non traité » à « validé ». * **RefuserAjoutSupp :** cette méthode permet au chef département de refuser la demande d’ajout d’une séance supplémentaire,   La méthode modifie l’état de la séance demandé de « non traité » à « refusé ». |
| Notification | Cconcerne les séances supplémentaires, si un enseignant fait une demande d’une séance supplémentaire, une notification dédiée au chef département sera créé automatiquement. S’il accepte la séance, la notification sera transférée à l’enseignant qui a fait la demande et aux étudiants concernés. S’il n’accepte pas la séance, la notification sera transférée seulement à l’enseignant. | 1. Les attributs :  * **Numéro\_notification**:un entier unique qui identifie une notification**.** * **Description**:description textuelle de la notification. * **Permission\_etudiant**: s’il est affecté à « true » l’étudiant pourra consulter cette notification, de type « Boolean ».  1. Les méthodes :  * **CréerNotification, SupprimerNotification, ModifierNotification**: crée, supprimer ou modifier une notification. |
| ResponsableFormation | Représente les responsables de formations dans la faculté « NTIC », et elle hérite de la classe enseignant. | 1. Les attributs :  * **Date\_nomination**: la date de nomination de l’enseignant au poste de responsable de formation. |
| ChefDépartement | Représente les chefs des départements de la faculté, elle hérite de la classe enseignant**.** | 1. Les attributs :  * **Date\_nomination**: la date de nomination de l’enseignant au poste de chef de département. |
| EmploiDuTemps | Représente les emplois du temps en vigueur dans la faculté « NTIC ». | 1. Les attributs :  * **Numéro\_emploi**: identifiant de l’emploi du temps. * **Département** : le département, le groupe et la spécialité concerné par l’emploi du temps. * **Emploi**: un fichier (PNG, JPEG …) qui représente l’emploi du temps.  1. Les méthodes :  * **ConsulterEmploi, ajouterEmploi, supprimerEmploi**: permettent l’ajout, la suppression et la consultation d’un emploi du temps. |
| Admin | Représente les administrateurs de la faculté, elle hérite de la classe utilisateur | 1. Les attributs :  * **Date\_nomination**: la date de nomination de l’enseignant au poste d’administrateur. |
| Justification | Représente les justifications des absences des étudiants. | 1. Les attributs :  * **Numéro\_justification**: identifiant de la justification. * **Fichier**: de type « blob », représente la justification au format PNG, JPEG… (Image d’un certificat médical…) * **Etat\_demande**: l’état de la demande de type « Etat demande ». * **Date\_justification**: la date de remise de la justification.  1. Les méthodes :  * **ConsulterJustification, ajouterJustification, validerJustification, refuserJustification**: permettent la consultation, l’ajout, la validation ou le refus d’une justification |

Tableau : Description des classes du diagramme de classes conceptuel.

#### Tableau des énumérations :

|  |  |  |
| --- | --- | --- |
| Énumération | Description | Valeurs |
| CodeDépartement | Représente les trois départements de la faculté « NTIC ». | * MI. * TLSI. * IFA. |
| TypeUtilisateur | Représente les cinq acteurs de notre système définit dans le chapitre 1. | * Étudiant. * Enseignant. * ChefDépartement. * ResponsableFormation. * Administrateur. |
| Année | Représente les années universitaire du système LMD. | * L1. * L2. * L3. * M1. * M2. |
| Spécialité | Représente les différentes spécialités présente dans la faculté « NTIC ». | * GL. * SI. * SCI. * TI. |
| EtatEtudiant | Représente l’état possible d’un étudiant. | * Bloqué. * Actif. |
| EtatDemande | Représente les états possibles d’une demande selon son traitement. | * NonTraité. * Validé. * Refusé. |

Tableau : Description des énumérations du diagramme de classe conceptuel.

#### Tableau des classes associations :

|  |  |  |
| --- | --- | --- |
| Classe association | Description | Attributs et méthodes |
| Absence | Elle représente les relevés des absences des étudiants (on n’a pas besoin d’une autre classe pour stocker le relevé des présences, car tous les étudiants qui sont actifs et ne sont pas dans la liste des absences sont automatiquement présents). | 1. Les attributs :  * **Numéro\_absence**: identifiant de l’absence. * **Id\_etudiant**: identifiant de l’étudiant absent. * **Code\_séance**: identifiant de la séance qui concerne l’absence.  1. Les méthodes :  * **MarquerAbsence**:permet à l’enseignant de stocker les étudiants absents dans une séance**.** * **ConsulterAbsence**:permet à un étudiant de consulter son relevé d’absence**.** * **ModifierAbsence**:permet à l’enseignant de modifier le relevé d’absence d’un étudiant ou d’ajouter une absence**.** * **SupprimerAbsence**: permet à l’enseignant de supprimer une absence pour un étudiant ou pour un groupe. * **EtablirEtudiantExclus**:permet à un enseignant ou au chef département de consulter la liste des étudiants exclus**.** * **EtablirStatistiques**:permet au chef département et au responsable de formation de consulter les statistiques concernant les absences des étudiants |

Tableau Description des classes associations du diagramme de classes conceptuel.

#### Les relations entre les classes :

* **Agrégation « possède » entre la classe « Département » et la classe « Utilisateur »** : indique qu’un département possède plusieurs utilisateurs. Et plusieurs utilisateurs appartiennent à un seul département.
* **Généralisation entre la classe « Utilisateur » et les classes « Etudiant », « Enseignant » et « Admin » :** indique que les étudiants, les enseignants et les administrateurs ont quelques attributs et méthodes communs entre eux (les attributs dans la classe « Utilisateur »).
* **Généralisation entre la classe « Enseignant » et les classes « ChefDépartement », « ResponsableFormation » :** indique que les Chefs des Départements, et les Responsables des Formations ont quelques attributs et méthodes communs entre eux (les attributs dans la classe « Enseignant »).Tous les chefs de département et tous les responsables de formation sont des enseignant.
* **Association « demande » entre la classe « Etudiant » et « CongéAcadémique » :** indique qu’un étudiant peut demander au maximum un seul congé académique. Et un congé académique est demandé par un seul étudiant.
* **Association « enseigne » entre la classe « Enseignant » et « Module » :** indique qu’un enseignant peut enseigner un ou plusieurs modules. Et un module est enseigné par un ou plusieurs enseignants.
* **Association « gère » entre la classe « ChefDépartement » et « Département » :** indique qu’un chef département gère exactement un seul département. Et un département est géré par un seul chef département.
* **Association « gèreEmploi » entre la classe « ChefDépartement » et « EmploiDuTemps » :** indique qu’un chef département gère un ou plusieurs emplois du temps (le chef de département **MI** gère l’emploi du temps de L1 et L2). Et plusieurs emplois du temps sont gérés par un seul chef département.
* **Association « concerne » entre la classe « SéanceSupp » et « Notification »** : indique qu’une notification concerne une seule séance supplémentaire. Et une séance supplémentaire a exactement une seule notification.
* **Composition « avoir » entre la classe « Absence » et la classe « Justification »** : indique qu’une absence peut avoir zéro à plusieurs justifications. Et zéro à plusieurs justifications peuvent justifiés une seule absence.
* **Composition « se compose » entre la classe « Module » et la classe « Séance »** : indique qu’un module se compose d’une séance ou de plusieurs séances.
* **Classe association « Absence » entre la classe « Etudiant » et « Séance » :** indique que plusieurs étudiants peuvent s’absenter dans plusieurs séances. Et plusieurs séances peuvent avoir plusieurs absences. Et donc si l’étudiant est absent dans une séance, la classe « Absence » stocke l’identifiant de l’étudiant absent et l’identifiant de la séance**.**

## Schéma de la base de données :

### Introduction :

L’utilisation d’un système de gestion de base de données relationnelle (SGBDR) dans notre cas **MYSQL** impose un changement dans la structure de classes représenté par le diagramme de classe vers le modèle relationnel. Ce passage est possible en suivant un ensemble de règles. Ces règles seront présentées dans le point suivant.

Une fois les règles appliquées sur le diagramme de classe, nous disposerons de l’intention de notre base données, intention (schéma) qui servira par la suite à la création concrète de la base de données.

### Règle de passage vers le modèle relationnel :

Le **modèle relationnel** est une manière de modéliser les relations existantes entre plusieurs informations ou éléments, et de les ordonner entre elles.

Les règles de passage vers le modèle relationnel sont les suivantes :

#### Association un a plusieurs :

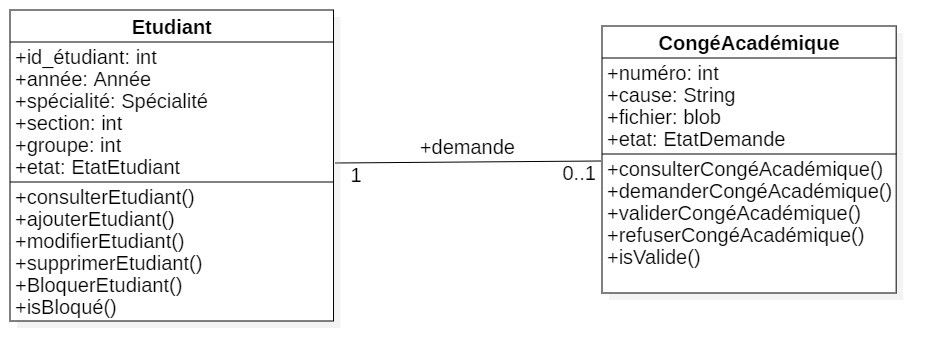
##### Définition :

En présence d’une cardinalité **« 1..1 »** d’un côté de l’association (association, agrégation ou composition) et l’autre côté **« a..n »** :

* Chaque classe se transformera en une table.
* Chaque attribut de classe se transforme en un champ de table.
* L’identifiant de la classe qui est associée à la cardinalité**« 1..1 »** devient la clé étrangère de l’autre table.
* Si on a une cardinalité **« 1..1 »**sur les **deux côtés de l’association**, on fait le même traitement que la cardinalité **« 1..1 »** sauf que cette fois on applique cette règle sur les deux classes de l’association (on obtient la clé primaire de chaque table comme une clé étrangère dans la deuxième table).

##### Exemples :

* Association « Demande » entre « Etudiant » et « CongéAcadémique » :



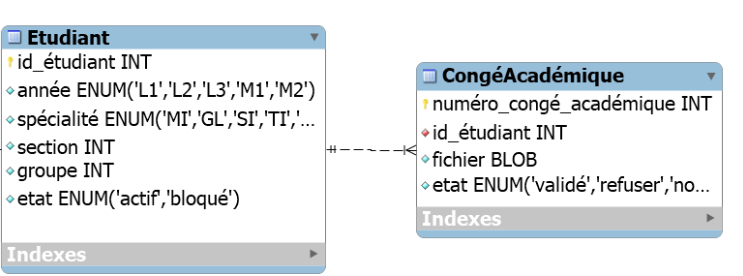
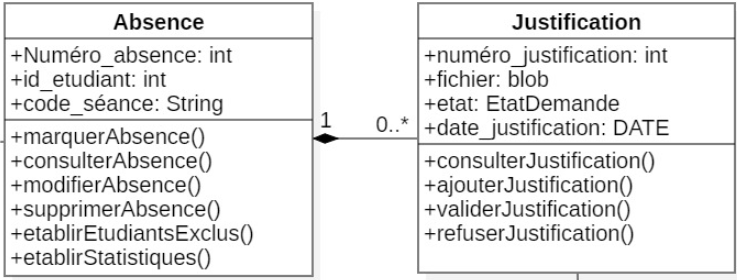


Figure : Passage du DCL au modèle relationnel de l'association « Demande » entre « Etudiant » et « CongéAcadémique ».

* Composition entre « Absence » et « Justification » :



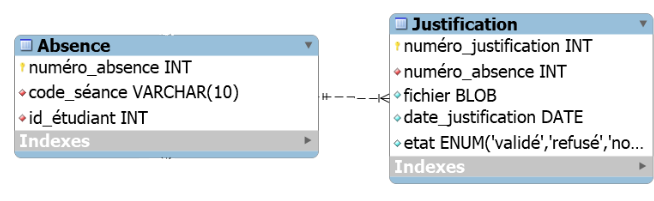
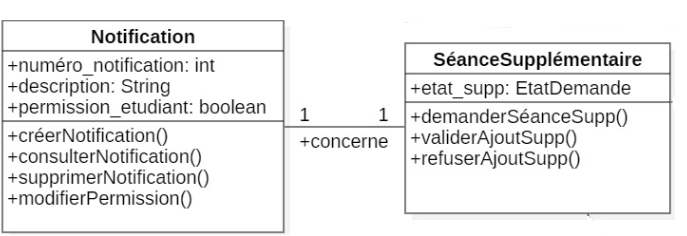


Figure : Passage du DCL au modèle relationnel de la composition entre « Absence » et « Justification ».

* Association « Concerne » entre « Notification » et « SéanceSupp » :



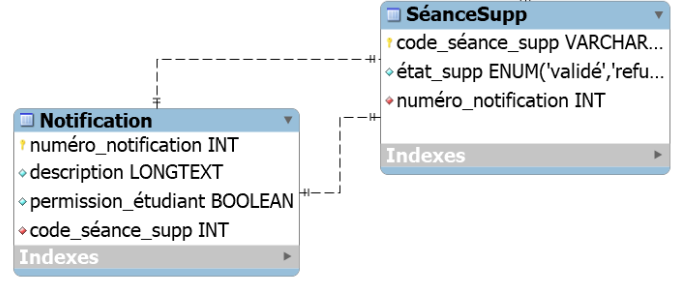


Figure : Passage du DCL au modèle relationnel de l'association « Concerne » entre « Notification » et « SéanceSupp ».

#### Association plusieurs à plusieurs :

##### Définition :

En présence d’une cardinalité « \*..\* » des deux côtes de l’association :

* + - Chaque classe se transforme en une table.
    - Chaque attribut de classe se transforme en un champ de table.
    - L’association se transforme en une table. Cette table a comme champs l’identifiant de chacune des deux classes, plus d’éventuels autres attributs.

##### Exemple :

* Association « Enseigne » entre « Enseignant et « Module » :

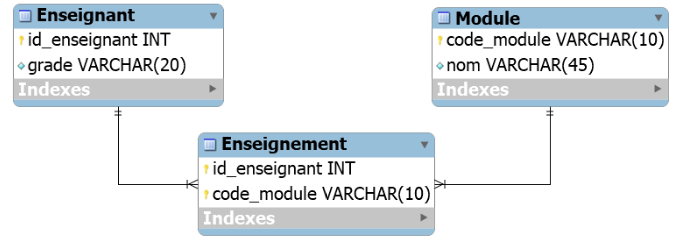
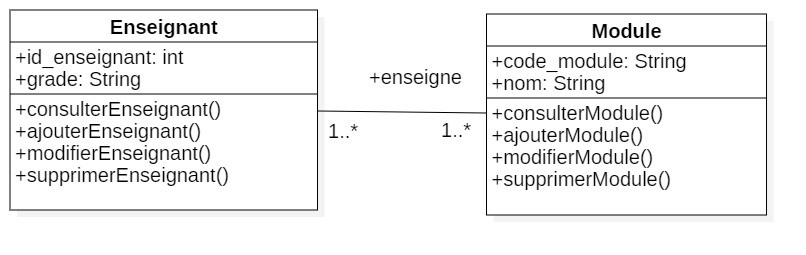


Figure : Passage du DCL au modèle relationnel de l'association « Enseigne » entre « Enseignant » et « Module ».

#### En cas de généralisation :

##### Définition :

En présence d’une généralisation entre deux classes il faut appliquer l’une des méthodes suivantes :

* **Méthode 1 :**
  + 1. Ajouter Une table avec tous les attributs des classes.
    2. Ajouter un attribut pour distinguer les types des objets.
* **Méthode 2 :**

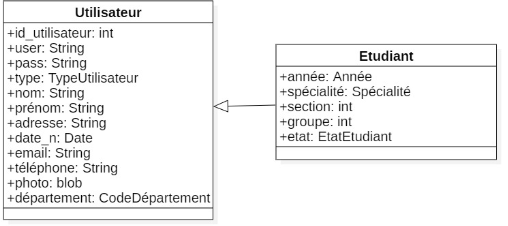
1. Créées, une table pour chaque sous-type, chaque table se compose des attributs génériques et d’attributs spécifiques.

* **Méthode 3 :**

1. Créer une table par classe et des associations.

##### Exemple :

* Généralisation entre « Utilisateur » et « Etudiant » :



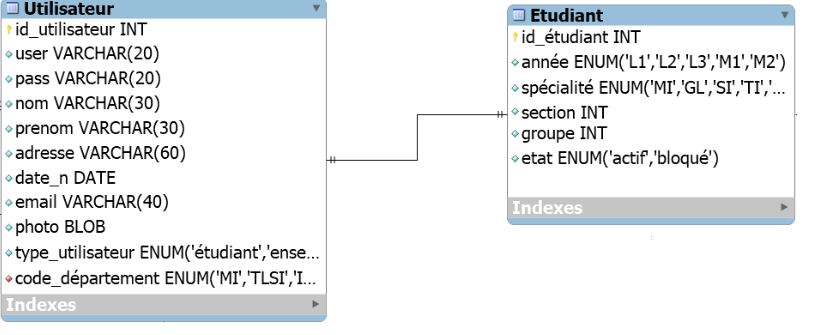


Figure : Passage du DCL au modèle relationnel de la généralisation entre « Utilisateur » et « Etudiant ».

***NB :*** *Id\_utilisateur = Id\_étudiant.*

#### En cas d’énumération :

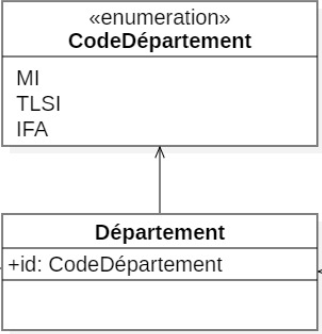
##### Définition :

En présence d’une énumération il faut :

1. Créer une table pour la classe associée à l’énumération avec tous ses attributs.
2. L’énumération devient un type ENUM(…).

##### Exemple :

* Énumération « CodeDépartement » :



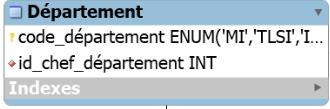


Figure : Passage du DCL au modèle relationnel de l'énumération « CodeDépartement ».

### Schéma relationnel de la base de données :

Suivant ce qui a été expliqué dans les points précédents, nous pouvons grâce au digramme de classe et en appliquant les règles de passage vers le modèle relationnel disposer du schéma relationnel suivant pour notre application :

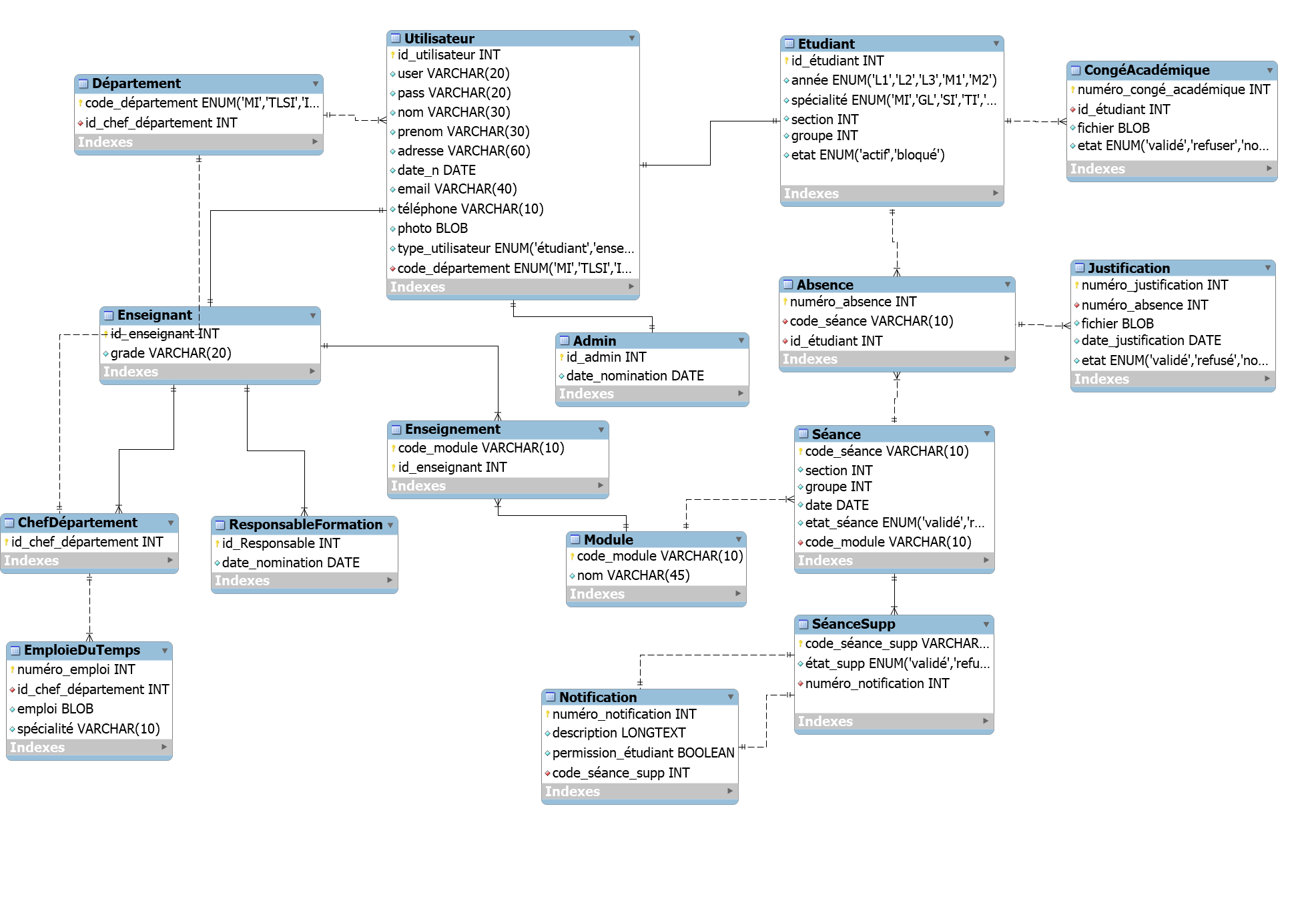


Figure : Schéma relationnel de la base de données.

***Nb :*** *les relations doivent être en troisième forme normale.*

## Scénarios d’exécutions :

### Introduction :

Dans cette partie nous allons mettre en évidence de manière plus détaillées les scénarios d’exécutions des six cas d’utilisation traité lors du chapitre 1. Cela sera fait à l’aide des diagrammes d’activité.

### Diagrammes d’activité :

#### Définitions :

Le diagramme d’activité est un diagramme comportemental, c’est-à-dire qu’il permet de modéliser le comportement des acteurs et du système (à l’image du Diagramme d’interaction) et de représenter le déclenchement d’événements en fonction des états du système que ceux-là soient en parallèle (multithreading), séquentiel ou en branches (conditionnel).

#### Diagrammes d’activité du cas « S’authentifier » :

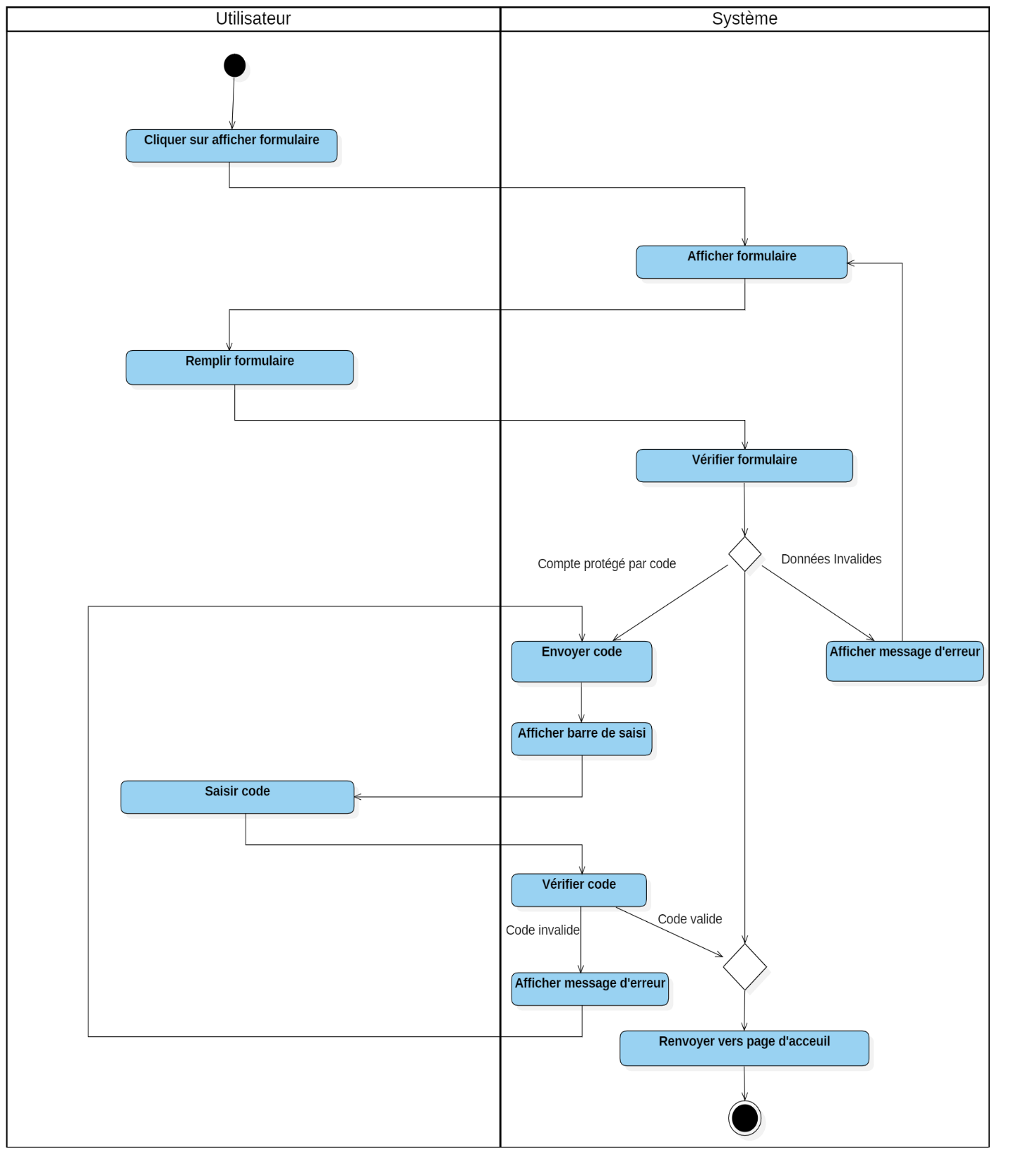


Figure : Diagramme d'activité du cas « S'authentifier ».

#### Diagramme d'activité du cas « Consulter relever d’absence » :

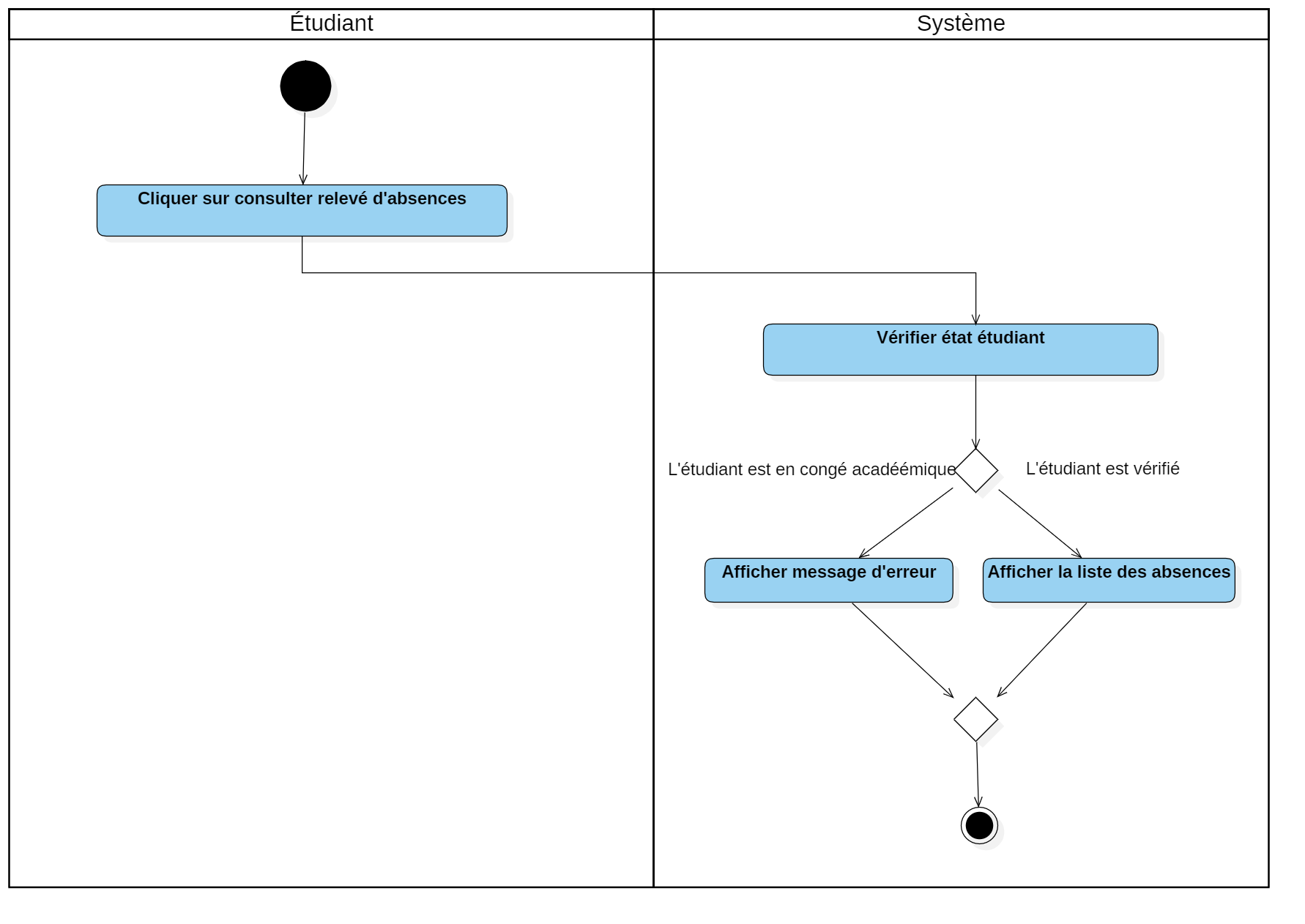


Figure : Diagramme d'activité du cas « Consulter relever d'absence ».

#### Diagramme d’activité du cas « Marquer la présence » :

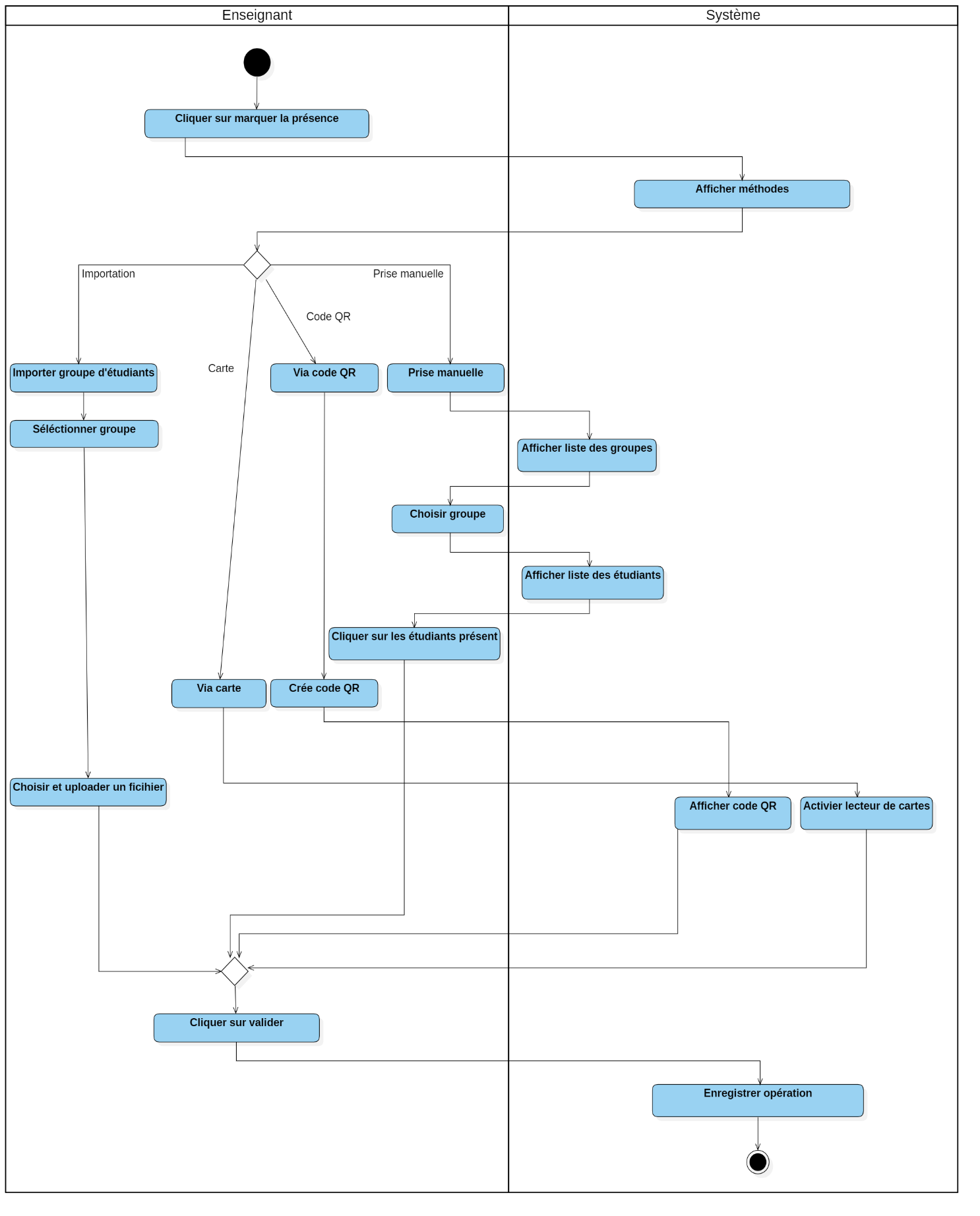


Figure : Diagramme d'activité du cas « Marquer la présence ».

#### Diagramme d’activité du cas « Justifier absence » :

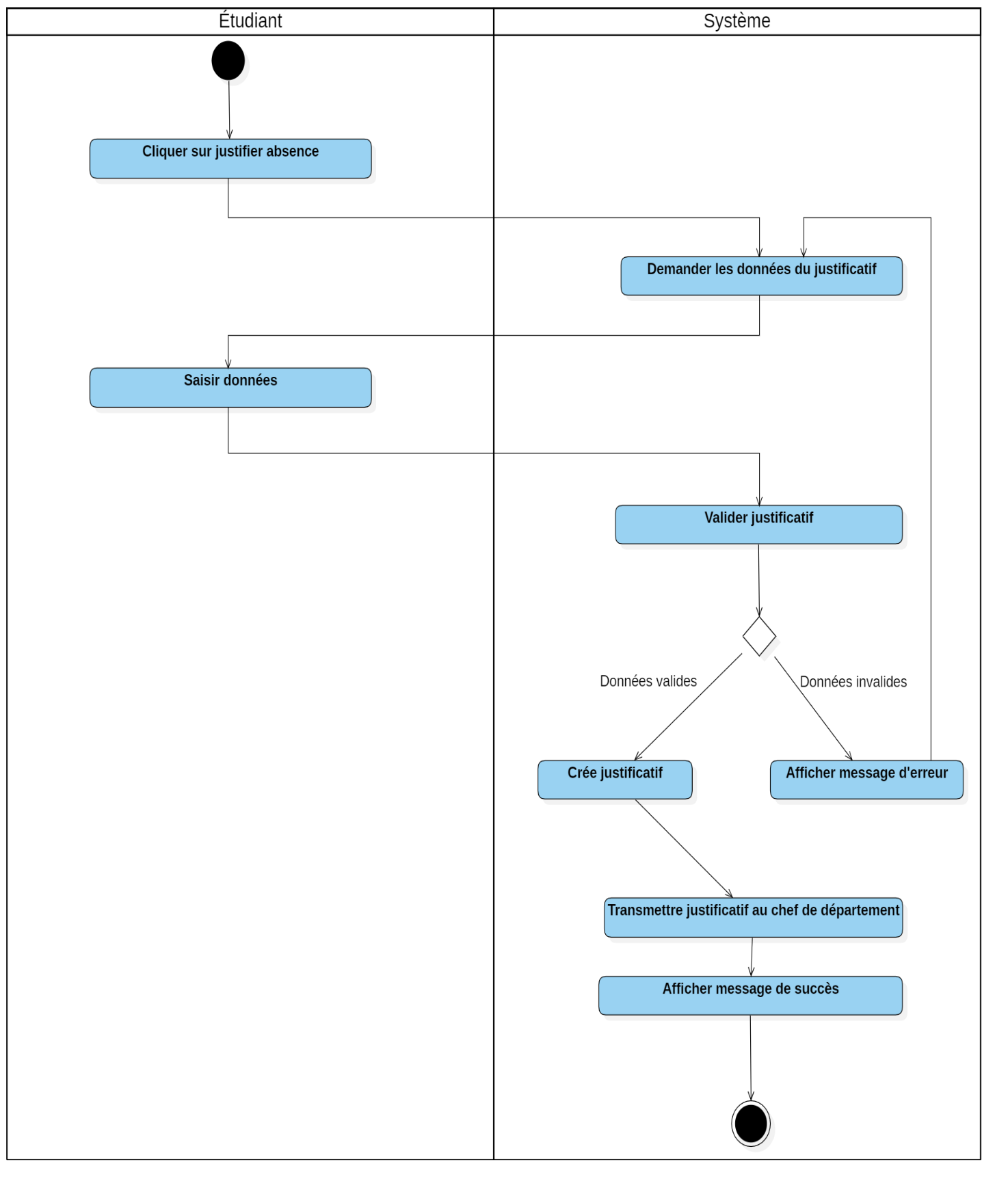


Figure : Diagramme d'activité du cas « Justifier absence ».

#### Diagramme d’activité du cas « Consulter justification » :

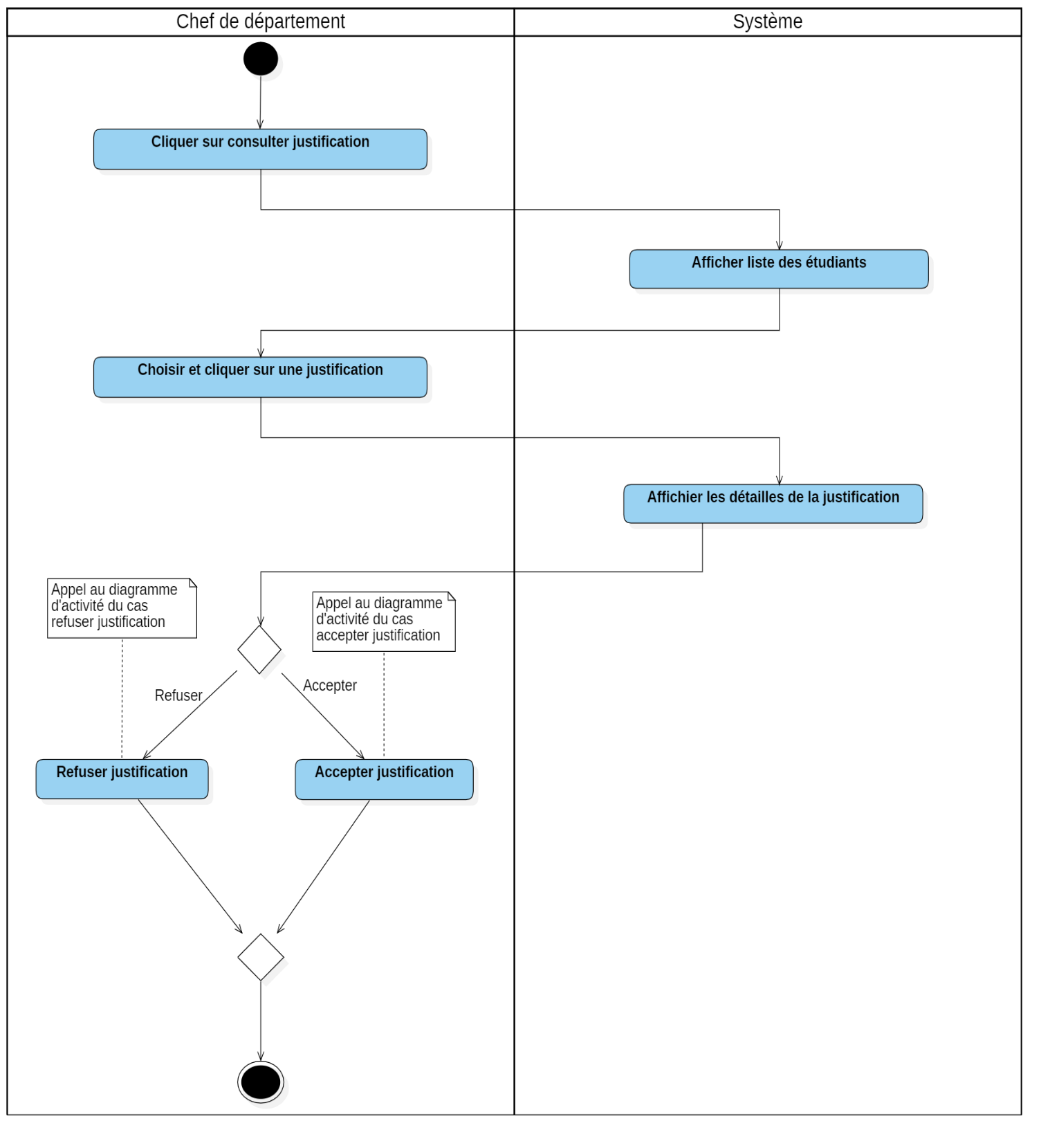


Figure : Diagramme d'activité du cas « Consulter justification ».

#### Diagramme d’activité du cas « Établir la liste des étudiants exclus » :

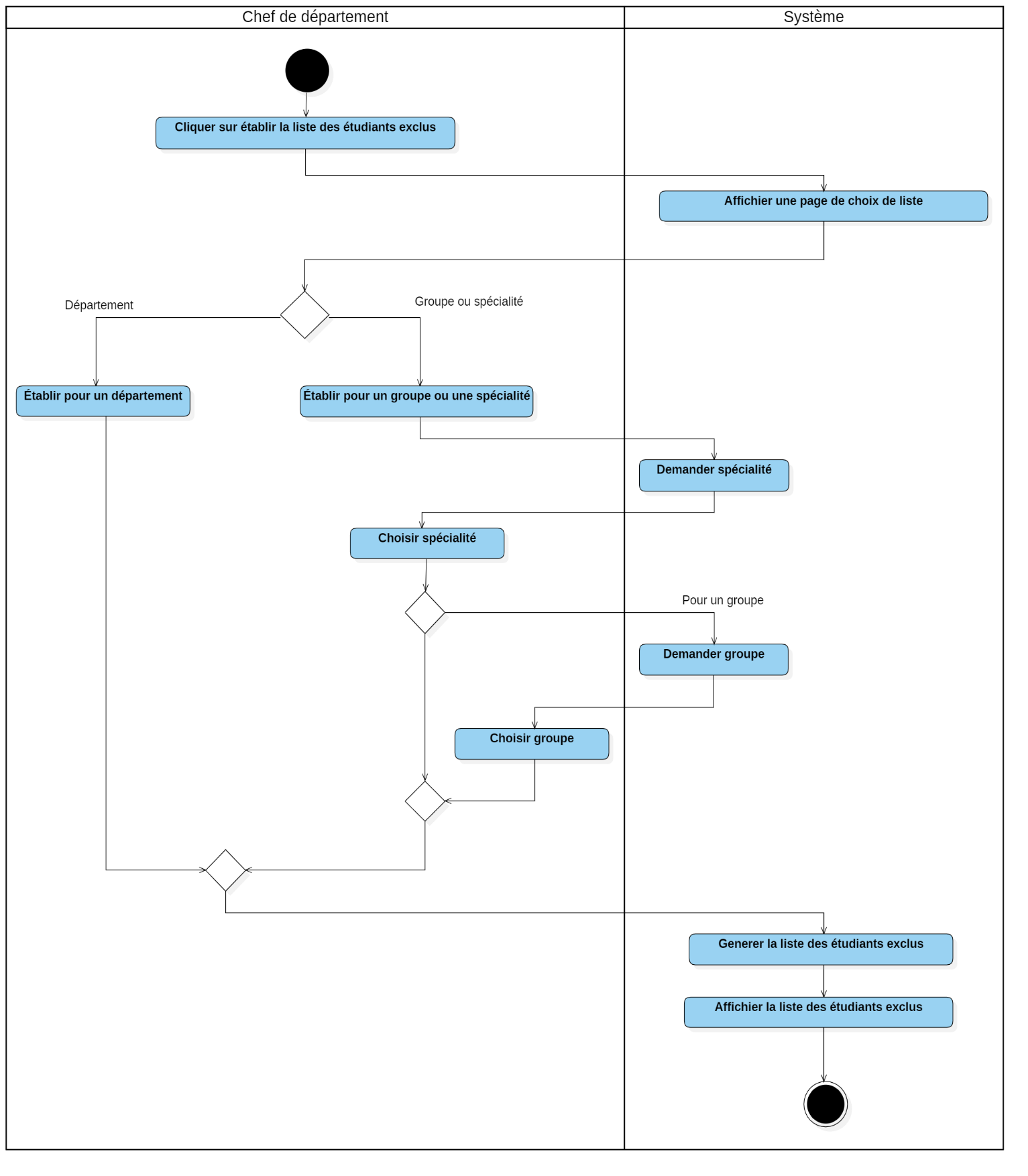


Figure : Diagramme d'activité du cas « Établir la liste des étudiants exclus ».

## Conclusion :

À la fin de ce chapitre, nous avons obtenu un modèle de conception détaillé qui décrit comment sont réalisées les fonctionnalités de l’application que l’on a recensées dans le chapitre précédent. Ce modèle contient le diagramme de classes conceptuel représentant l’aspect statique du système, et des diagrammes d’activité qui représentent son aspect dynamique. Nous avons également réalisé le passage du diagramme classes vers le modèle relationnel et ainsi disposer d’un schéma relationnel de la base de données. Dans le chapitre suivant, nous allons aborder les différentes technologies utilisées dans la réalisation de l’application.