# Rapport de Conception : Système de Gestion des Masters Universitaires

## Table des matières

1. [Introduction](#introduction)
2. [Planification du Projet](#planification-du-projet)
   * [Vue d’ensemble des Sprints](#vue-densemble-des-sprints)
   * [Sprint 1 : Mise en place des bases](#sprint-1--mise-en-place-des-bases)
   * [Sprint 2 : Cœur fonctionnel pour les établissements](#X4474e901ef9f078b45c544b4c11ad2644b1328c)
   * [Sprint 3 : Fonctionnalités pour les étudiants](#Xbaf49b05dcbc2f40d3f416339957f5f8ccae84d)
3. [Diagrammes de Cas d’Utilisation](#diagrammes-de-cas-dutilisation)
   * [Cas d’Utilisation - Étudiant](#cas-dutilisation---étudiant)
   * [Cas d’Utilisation - Établissement](#cas-dutilisation---établissement)
4. [Diagrammes d’Activité](#diagrammes-dactivité)
   * [Soumission de candidature par l’étudiant](#soumission-de-candidature-par-létudiant)
   * [Examen des candidatures par l’établissement](#Xcd27e637430c55c0401803598d6f71749fb25a7)
   * [Définition des masters par l’établissement](#Xdf75e128857f729b2615aef32df681c67d1a823)
5. [Diagrammes de Séquence](#diagrammes-de-séquence)
   * [Soumission de candidature par l’étudiant](#Xe615cf4ec923dcf1d9e20fb4d651ba9982e7879)
   * [Examen des candidatures par l’établissement](#Xe1bf51814a020a6af95cb8a5aab086817088481)
   * [Définition des masters par l’établissement](#X847375a58760491d3469cdfd0f0860de8dd06a7)
6. [Diagramme de Classes](#diagramme-de-classes)
   * [Entités principales](#entités-principales)
   * [Relations principales](#relations-principales)
   * [Aspects techniques à considérer](#aspects-techniques-à-considérer-classes)
7. [Conclusion](#conclusion)

## Introduction

Ce rapport de conception présente une analyse détaillée du système de gestion des masters universitaires. Il se concentre sur les diagrammes UML essentiels et propose une explication complète de chaque aspect de la conception.

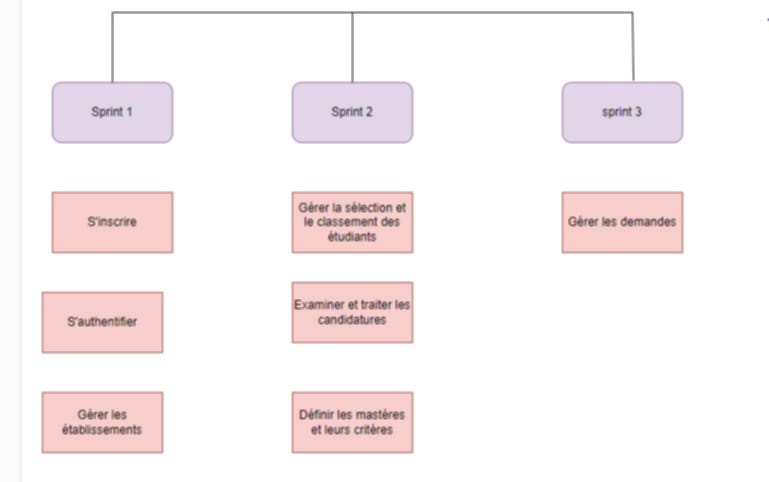
Le système vise à faciliter la gestion des programmes de master universitaires, en permettant aux établissements de créer et gérer leurs offres de formation, et aux étudiants de rechercher et postuler à ces programmes. Il comprend également des fonctionnalités d’administration pour les universités.

Ce document est destiné à servir de référence pour le développement du système, en fournissant une vision claire de sa structure et de son comportement.

## Planification du Projet

### Vue d’ensemble des Sprints

Le développement du système est divisé en trois sprints principaux, chacun se concentrant sur un ensemble spécifique de fonctionnalités:



Organisation des Sprints

### Sprint 1 : Mise en place des bases

#### Objectif

Établir les fondations du système en implémentant les fonctionnalités d’authentification et de gestion des établissements.

#### Fonctionnalités principales

1. **S’inscrire**
   * Création de compte pour les différents types d’utilisateurs (étudiants, établissements)
   * Formulaire d’inscription avec validation des données
   * Confirmation par email
2. **S’authentifier**
   * Système de connexion sécurisé
   * Gestion des sessions utilisateurs
   * Récupération de mot de passe
3. **Gérer les établissements**
   * Interface d’administration pour l’université
   * Ajout, modification et suppression d’établissements
   * Attribution des droits d’accès aux établissements

#### Livrables

* Base de données initiale avec les tables users, admin, university et institute
* Interface d’inscription et de connexion
* Panneau d’administration pour la gestion des établissements
* Documentation utilisateur pour ces fonctionnalités

#### Critères d’acceptation

* Les utilisateurs peuvent créer un compte et se connecter
* L’université peut gérer les établissements (CRUD)
* Les données sont correctement validées et stockées
* L’interface est responsive et accessible

### Sprint 2 : Cœur fonctionnel pour les établissements

#### Objectif

Développer les fonctionnalités essentielles permettant aux établissements de gérer leurs masters et les candidatures.

#### Fonctionnalités principales

1. **Définir les masters et leurs critères**
   * Création et configuration des programmes de master
   * Définition des critères d’admission
   * Paramétrage des dates de candidature
2. **Examiner et traiter les candidatures**
   * Interface de visualisation des candidatures reçues
   * Accès aux dossiers des étudiants
   * Système d’évaluation des candidatures
3. **Gérer la sélection et le classement des étudiants**
   * Acceptation ou rejet des candidatures
   * Classement des candidats selon les critères définis
   * Gestion des listes d’attente

#### Livrables

* Tables masters et applications dans la base de données
* Interface de gestion des masters pour les établissements
* Système de traitement des candidatures
* Notifications aux étudiants concernant leurs candidatures

#### Critères d’acceptation

* Les établissements peuvent créer et configurer des masters
* Les candidatures peuvent être examinées et traitées
* Le système permet de classer les candidats
* Les notifications sont envoyées correctement

### Sprint 3 : Fonctionnalités pour les étudiants

#### Objectif

Compléter le système avec les fonctionnalités destinées aux étudiants pour la recherche de masters et la gestion de leurs candidatures.

#### Fonctionnalités principales

1. **Gérer les demandes (candidatures)**
   * Interface de recherche et filtrage des masters
   * Formulaire de candidature
   * Téléversement des documents requis
   * Suivi de l’état des candidatures
2. **Fonctionnalités supplémentaires**
   * Système de feedback pour les étudiants
   * Notifications sur l’évolution des candidatures
   * Profil étudiant avec historique des candidatures

#### Livrables

* Tables students et feedback dans la base de données
* Interface de recherche et de candidature pour les étudiants
* Système de suivi des candidatures
* Fonctionnalité de feedback

#### Critères d’acceptation

* Les étudiants peuvent rechercher des masters et postuler
* Les documents peuvent être téléversés et associés aux candidatures
* Les étudiants peuvent suivre l’état de leurs candidatures
* Le système de feedback fonctionne correctement

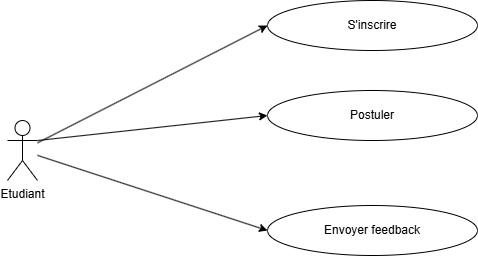
#### Interdépendances entre les Sprints

* Le Sprint 1 est un prérequis pour les Sprints 2 et 3, car il établit le système d’authentification et la structure des établissements.
* Le Sprint 2 doit être complété avant que les fonctionnalités du Sprint 3 puissent être pleinement utilisées, car les étudiants ont besoin de masters définis pour postuler.
* Les fonctionnalités développées dans chaque sprint s’intègrent pour former un système cohérent de gestion des masters universitaires.

## Diagrammes de Cas d’Utilisation

### Cas d’Utilisation - Étudiant

Le diagramme de cas d’utilisation pour l’étudiant représente les fonctionnalités principales accessibles à l’acteur “Étudiant” dans le système.



Cas d’Utilisation Étudiant

#### Acteur

* **Étudiant** : Représente un utilisateur qui souhaite s’inscrire et postuler à des programmes de master.

#### Cas d’utilisation principaux

1. **S’inscrire**
   * Description : Permet à un étudiant de créer un compte dans le système.
   * Préconditions : L’étudiant n’a pas encore de compte dans le système.
   * Flux principal :
     + L’étudiant accède à la page d’inscription.
     + Il remplit les informations requises (nom, email, mot de passe, etc.).
     + Le système vérifie la validité des informations.
     + Un nouveau compte étudiant est créé.
   * Postconditions : L’étudiant possède un compte et peut se connecter au système.
2. **Postuler**
   * Description : Permet à un étudiant de soumettre une candidature à un programme de master.
   * Préconditions : L’étudiant est inscrit et connecté au système.
   * Flux principal :
     + L’étudiant recherche et sélectionne un master.
     + Il remplit le formulaire de candidature.
     + Il téléverse les documents requis (relevés de notes, etc.).
     + Il soumet sa candidature.
   * Postconditions : La candidature est enregistrée dans le système avec le statut “en attente”.
3. **Envoyer feedback**
   * Description : Permet à un étudiant de fournir un retour ou une évaluation sur le système ou le processus de candidature.
   * Préconditions : L’étudiant est inscrit et connecté au système.
   * Flux principal :
     + L’étudiant accède à la fonctionnalité de feedback.
     + Il rédige son message et attribue une note.
     + Il soumet son feedback.
   * Postconditions : Le feedback est enregistré dans le système et peut être consulté par les administrateurs.

#### Importance dans le projet

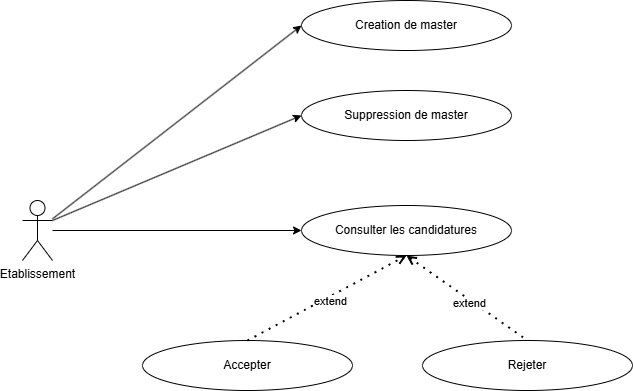
Ce diagramme définit les interactions principales que les étudiants auront avec le système. Il guide le développement des interfaces utilisateur et des fonctionnalités du Sprint 3.

#### Scénarios d’utilisation typiques

1. Un nouvel étudiant s’inscrit, recherche des masters qui l’intéressent, et postule à plusieurs d’entre eux.
2. Un étudiant déjà inscrit se connecte pour vérifier l’état de ses candidatures et envoyer un feedback sur son expérience.

### Cas d’Utilisation - Établissement

Le diagramme de cas d’utilisation pour l’établissement représente les fonctionnalités principales accessibles à l’acteur “Établissement” dans le système.



Cas d’Utilisation Établissement

#### Acteur

* **Établissement** : Représente une institution universitaire ou un institut qui propose des programmes de master.

#### Cas d’utilisation principaux

1. **Création de master**
   * Description : Permet à un établissement de créer un nouveau programme de master dans le système.
   * Préconditions : L’établissement est enregistré et connecté au système.
   * Flux principal :
     + L’établissement accède à l’interface de gestion des masters.
     + Il remplit les informations du master (nom, description, nombre maximum d’étudiants, dates de candidature, etc.).
     + Le système valide les informations.
     + Le nouveau master est créé et disponible pour les candidatures.
   * Postconditions : Le master est enregistré dans le système et peut recevoir des candidatures.
2. **Suppression de master**
   * Description : Permet à un établissement de supprimer un programme de master existant.
   * Préconditions : L’établissement est connecté et le master à supprimer existe dans le système.
   * Flux principal :
     + L’établissement accède à la liste de ses masters.
     + Il sélectionne le master à supprimer.
     + Il confirme la suppression.
     + Le système supprime le master.
   * Postconditions : Le master est retiré du système et n’est plus disponible pour les candidatures.
3. **Consulter les candidatures**
   * Description : Permet à un établissement de visualiser les candidatures reçues pour ses masters.
   * Préconditions : L’établissement est connecté et des candidatures ont été soumises.
   * Flux principal :
     + L’établissement accède à l’interface de gestion des candidatures.
     + Il peut filtrer les candidatures par master, statut, etc.
     + Il visualise les détails des candidatures (informations de l’étudiant, documents, etc.).
   * Extensions :
     + **Accepter** : L’établissement peut accepter une candidature après examen.
     + **Rejeter** : L’établissement peut rejeter une candidature après examen.
   * Postconditions : Les statuts des candidatures peuvent être mis à jour (acceptées ou rejetées).

#### Extensions

1. **Accepter**
   * Description : Extension du cas d’utilisation “Consulter les candidatures” permettant d’accepter une candidature.
   * Flux principal :
     + L’établissement examine les détails d’une candidature.
     + Il décide d’accepter la candidature.
     + Le système met à jour le statut de la candidature à “Acceptée”.
     + Une notification est envoyée à l’étudiant concerné.
2. **Rejeter**
   * Description : Extension du cas d’utilisation “Consulter les candidatures” permettant de rejeter une candidature.
   * Flux principal :
     + L’établissement examine les détails d’une candidature.
     + Il décide de rejeter la candidature.
     + Le système met à jour le statut de la candidature à “Rejetée”.
     + Une notification est envoyée à l’étudiant concerné.

#### Importance dans le projet

Ce diagramme définit les interactions principales que les établissements auront avec le système. Il guide le développement des interfaces administratives et des fonctionnalités du Sprint 2.

#### Aspects techniques à considérer

* **Création de master** : Cette fonctionnalité est fondamentale car elle alimente le catalogue de masters disponibles pour les étudiants.
* **Suppression de master** : Cette action doit être gérée avec précaution, notamment si des candidatures sont déjà associées au master.
* **Consulter les candidatures** avec les extensions **Accepter** et **Rejeter** : Ces fonctionnalités sont essentielles pour le processus de sélection et nécessitent une interface intuitive pour faciliter la prise de décision.

## Diagrammes d’Activité

Les diagrammes d’activité représentent les flux de travail et les processus métier du système de gestion des masters universitaires. Ils montrent la séquence d’actions et les décisions prises par les différents acteurs.

### Soumission de candidature par l’étudiant Diagramme d’activité - Soumission Candidature

#### Objectif

Ce diagramme modélise le flux de travail séquentiel et les décisions prises par un étudiant lorsqu’il postule à un master.

#### Description du processus

1. L’étudiant s’authentifie sur le système
2. Il recherche les masters disponibles
3. Il consulte les détails des masters qui l’intéressent
4. Il sélectionne un master pour postuler
5. Il remplit le formulaire de candidature
6. Il télécharge les documents requis (relevés de notes, CV, etc.)
7. Le système valide les informations saisies
8. Si la validation échoue, l’étudiant corrige les erreurs
9. Si la validation réussit, la candidature est soumise
10. Le système confirme la soumission de la candidature

#### Points clés

* Le diagramme inclut des points de décision pour la validation des données
* Il montre clairement la séquence chronologique des actions
* Il illustre l’interaction entre l’étudiant et le système

### Examen des candidatures par l’établissement Diagramme d’activité - Examen Candidatures

#### Objectif

Ce diagramme décrit le processus suivi par un établissement pour évaluer les candidatures reçues.

#### Description du processus

1. L’établissement s’authentifie sur le système
2. Il accède à la liste des candidatures en attente
3. Il sélectionne une candidature à examiner
4. Il consulte les détails de l’étudiant et les documents fournis
5. Il évalue la candidature selon les critères du master
6. Il prend une décision (Accepter/Rejeter)
7. Le système met à jour le statut de la candidature
8. Une notification est envoyée à l’étudiant
9. L’établissement passe à la candidature suivante ou termine le processus

#### Points clés

* Le diagramme illustre la boucle d’examen des candidatures
* Il montre le processus de prise de décision
* Il inclut la mise à jour du statut et la notification

### Définition des masters par l’établissement Diagramme d’activité - Définition Master

#### Objectif

Ce diagramme montre les étapes de création ou de modification d’un programme de master par un établissement.

#### Description du processus

1. L’établissement s’authentifie sur le système
2. Il choisit entre créer un nouveau master ou modifier un existant
3. S’il crée un nouveau master :
   * Il saisit les informations de base (nom, description, etc.)
   * Il définit les critères d’admission
   * Il ajoute les spécialités associées
   * Il configure les dates de candidature
4. S’il modifie un master existant :
   * Il sélectionne le master à modifier
   * Il met à jour les informations nécessaires
5. Le système valide les informations
6. Si la validation échoue, l’établissement corrige les erreurs
7. Si la validation réussit, les modifications sont enregistrées

#### Points clés

* Le diagramme montre les deux chemins possibles (création/modification)
* Il détaille les étapes de configuration d’un master
* Il inclut la validation des données par le système

#### Valeur ajoutée pour les développeurs

* Ils fournissent une vue séquentielle des actions à implémenter
* Ils identifient les points de décision et les branches conditionnelles
* Ils clarifient les interactions entre l’utilisateur et le système

#### Lien avec les sprints

* Le diagramme “Soumission de candidature” correspond aux fonctionnalités du Sprint 3
* Le diagramme “Examen des candidatures” correspond aux fonctionnalités du Sprint 2
* Le diagramme “Définition des masters” correspond également aux fonctionnalités du Sprint 2

## Diagrammes de Séquence

Les diagrammes de séquence illustrent les interactions chronologiques entre les différents objets et acteurs du système de gestion des masters universitaires. Ils montrent l’ordre des messages échangés et les opérations effectuées.

### Soumission de candidature par l’étudiant (Séquence) Diagramme de séquence - Soumission Candidature

#### Objectif

Ce diagramme illustre les interactions chronologiques entre l’Étudiant, le Système et la Base de Données lors de la soumission d’une candidature.

#### Description des interactions

1. L’étudiant s’authentifie auprès du système
2. Le système vérifie les identifiants auprès de la base de données
3. L’étudiant demande la liste des masters disponibles
4. Le système récupère les informations des masters depuis la base de données
5. L’étudiant sélectionne un master et demande ses détails
6. Le système récupère et affiche les détails du master
7. L’étudiant remplit le formulaire de candidature
8. L’étudiant téléverse les documents requis
9. Le système valide les informations saisies
10. Le système enregistre la candidature dans la base de données
11. Le système confirme la soumission à l’étudiant

#### Points clés

* Le diagramme montre clairement la séquence temporelle des interactions
* Il met en évidence les échanges entre l’étudiant, le système et la base de données
* Il illustre les opérations de lecture et d’écriture dans la base de données

### Examen des candidatures par l’établissement (Séquence) Diagramme de séquence - Examen Candidatures

#### Objectif

Ce diagramme détaille les interactions lors de l’évaluation d’une candidature par un établissement.

#### Description des interactions

1. L’établissement s’authentifie auprès du système
2. Le système vérifie les identifiants auprès de la base de données
3. L’établissement demande la liste des candidatures en attente
4. Le système récupère les candidatures depuis la base de données
5. L’établissement sélectionne une candidature spécifique
6. Le système récupère les détails de l’étudiant et de sa candidature
7. L’établissement examine les informations et prend une décision
8. L’établissement soumet sa décision (accepter/rejeter)
9. Le système met à jour le statut de la candidature dans la base de données
10. Le système envoie une notification à l’étudiant concerné

#### Points clés

* Le diagramme montre les interactions entre l’établissement, le système et la base de données
* Il illustre le processus de prise de décision et de mise à jour du statut
* Il inclut la notification envoyée à l’étudiant

### Définition des masters par l’établissement (Séquence) Diagramme de séquence - Définition Master

#### Objectif

Ce diagramme montre les interactions pour la création ou la modification d’un master.

#### Description des interactions

1. L’établissement s’authentifie auprès du système
2. Le système vérifie les identifiants auprès de la base de données
3. L’établissement accède à l’interface de gestion des masters
4. Si création d’un nouveau master :
   * L’établissement saisit les informations du master
   * Le système valide les données
   * Le système enregistre le nouveau master dans la base de données
5. Si modification d’un master existant :
   * L’établissement demande la liste de ses masters
   * Le système récupère les masters depuis la base de données
   * L’établissement sélectionne un master à modifier
   * Le système récupère les détails du master
   * L’établissement modifie les informations
   * Le système valide les modifications
   * Le système met à jour les informations dans la base de données

#### Points clés

* Le diagramme illustre les deux scénarios possibles (création/modification)
* Il montre les opérations de validation des données
* Il détaille les interactions avec la base de données pour l’insertion ou la mise à jour

#### Valeur ajoutée pour les développeurs

* Ils montrent précisément quand et comment les objets interagissent
* Ils identifient les appels à la base de données et les validations nécessaires
* Ils aident à concevoir les API et les interfaces entre composants

## Diagramme de Classes

Le diagramme de classes représente la structure statique du système de gestion des masters universitaires. Il montre les entités principales, leurs attributs et les relations entre elles.

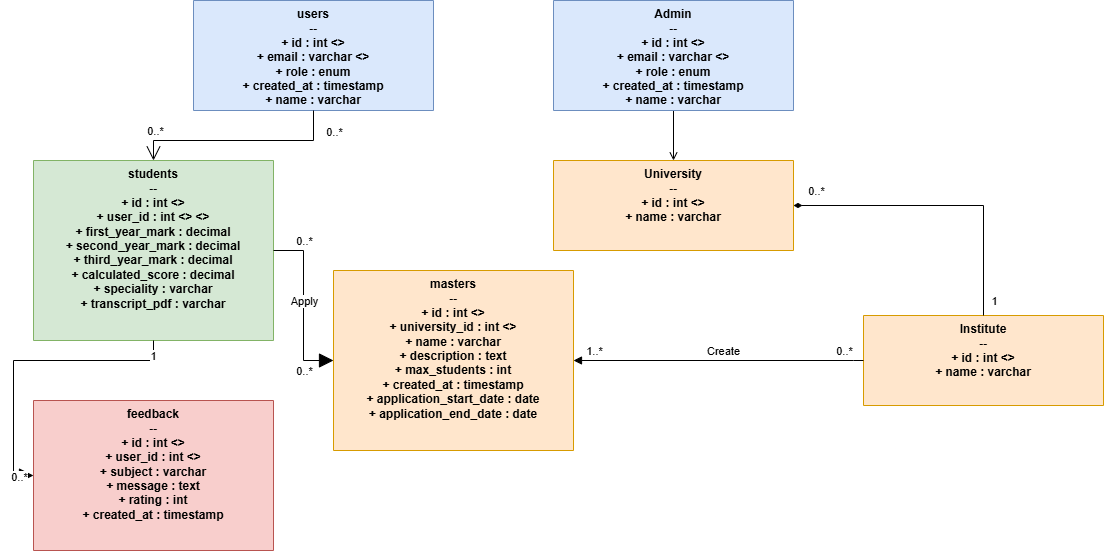


Diagramme de Classes

### Entités principales

1. **users** (Utilisateurs)
   * Classe de base pour tous les utilisateurs du système
   * Attributs: id, email, role, created\_at, name
2. **Admin** (Administrateur)
   * Hérite de la classe users
   * Représente les administrateurs du système
   * Attributs: id, email, role, created\_at, name
3. **students** (Étudiants)
   * Liée à la classe users (relation 0..\*)
   * Représente les étudiants qui peuvent postuler aux masters
   * Attributs: id, user\_id, first\_year\_mark, second\_year\_mark, third\_year\_mark, calculated\_score, speciality, transcript\_pdf
4. **University** (Université)
   * Liée à la classe Admin (relation 0..\*)
   * Représente les universités qui gèrent les masters
   * Attributs: id, name
5. **Institute** (Institut)
   * Liée à la classe University (relation 0..\*)
   * Représente les instituts qui peuvent être associés à une université
   * Attributs: id, name
6. **masters** (Masters)
   * Liée aux classes University (relation 1..*) et students (relation 0..*)
   * Représente les programmes de master proposés
   * Attributs: id, university\_id, name, description, max\_students, created\_at, application\_start\_date, application\_end\_date
7. **feedback** (Retours)
   * Liée à la classe students (relation 0..\*)
   * Représente les retours fournis par les étudiants
   * Attributs: id, user\_id, subject, message, rating, created\_at

### Relations principales

1. **University - masters**: Relation “Create” (Créer) avec cardinalité 1..\* du côté University et 0..\* du côté masters, indiquant qu’une université peut créer plusieurs masters.
2. **students - masters**: Relation “Apply” (Postuler) avec cardinalité 0..\* des deux côtés, indiquant qu’un étudiant peut postuler à plusieurs masters et qu’un master peut recevoir des candidatures de plusieurs étudiants.
3. **University - Institute**: Relation avec cardinalité 1 du côté University et 0..\* du côté Institute, indiquant qu’un institut appartient à une seule université, mais qu’une université peut avoir plusieurs instituts.
4. **users - students**: Relation avec cardinalité 0..\* des deux côtés, indiquant qu’un utilisateur peut être associé à plusieurs profils étudiants et vice versa.
5. **students - feedback**: Relation avec cardinalité 1 du côté students et 0..\* du côté feedback, indiquant qu’un étudiant peut fournir plusieurs retours.

### Aspects techniques à considérer (Classes)

* **Héritage** : La relation entre users et Admin utilise l’héritage, ce qui peut être implémenté de différentes façons en base de données (table unique, table par classe, etc.).
* **Relations many-to-many** : La relation entre students et masters (Apply) suggère une table de jointure qui n’est pas explicitement représentée dans le diagramme mais qui sera nécessaire dans l’implémentation.
* **Attributs calculés** : L’attribut calculated\_score dans la classe students est probablement dérivé des notes des trois années (first\_year\_mark, second\_year\_mark, third\_year\_mark), ce qui nécessitera une logique de calcul dans l’application.

## Conclusion

Ce rapport de conception fournit une documentation complète et cohérente du système de gestion des masters universitaires. Il couvre les aspects essentiels de la conception (structure de données, comportement, cas d’utilisation) et fournit une base solide pour le développement du projet selon les sprints définis.

Les diagrammes UML présentés offrent différentes perspectives sur le système, permettant aux différentes parties prenantes (développeurs, architectes, testeurs, etc.) de comprendre le système selon leur point de vue spécifique.

La planification en trois sprints permet un développement incrémental du système, avec des livrables fonctionnels à la fin de chaque sprint, tout en maintenant une vision cohérente du produit final.

Les fonctionnalités sont clairement définies et organisées pour répondre aux besoins des différents acteurs (étudiants, établissements, universités), assurant ainsi que le système final sera complet et adapté aux exigences du projet.