

Internships Management System

Système de gestion des stages au sein d'une faculté



Project: Méthodes et techniques de développement des logiciels

ÉTUDIANTS: MARUS CRISTINA & SOLOT SARA

GROUPE: 1230F

1) Acteurs:

Les acteurs sont les représentants d'entreprises, les étudiants qui accèdent à l'application à la recherche du stage, les mentors – les professeurs de la faculté et le doyen de la faculté, qui est également l'administrateur de la page.

2) Opérations:

- Le student peut:
 - connexion au compte;
 - créer un profil;
 - attachement CV;
 - recherche de stage par critères (domaine, rémunéré, non rémunéré, etc....)
 - fonction d'application pour le stage;
 - télécharger des documents supplémentaires (exemple: signature d'un accord de pratique);
- Le représentant de l'entreprise peut:
 - connexion au compte;
 - mise à jour du profil;
 - publier des annonces de stage;
 - liste des stagiaires;
 - bouton d'acceptation ou de rejet;
 - signer la convention-cadre de pratique
- Mentor – professeur FILS :
 - connexion au compte;
 - signer la convention-cadre de pratique
- Doyen FILS:
 - connexion au compte;
 - création de compte (exemple: étudiant, enseignant mentor, représentant);
 - lien: application - email;
 - voir la liste des accords signés / la liste des enseignants / la liste des entreprises;
 - signer la convention-cadre de pratique



3) Règles métier:

[RM1] La plate-forme peut être utilisée seulement par les personnes enregistrées.

[RM2] L'administrateur est le seul à pouvoir ajouter ou supprimer des comptes d'entreprise ou d'enseignants.

[RM3] L'entreprise peut voir uniquement le profil des candidats qui s'applique à ce stage.

[RM4] Les étudiants ne peuvent pas voir autres profils d'étudiants.

[RM5] Seule l'entreprise peut fermer la candidature pour ses propres stages.

Matrice RASIC

Project tasks	Product Owners	Business Analyst	Design Director	Backend Developer	Head of CRM	Frontend Developer
Research						
IDE	C	C	S	A	I	R
Features	C	C	S	A	I	R
Implementation						
Product Concept	A	R	S	S	C	S
Prototype	R	A	C	S	I	S
Charts	C	C	S	A	I	R
Backend Coding	I	I	C	A	I	S
Design	C	C	A	S	S	R
User Testing	I	I	S	I	A	I

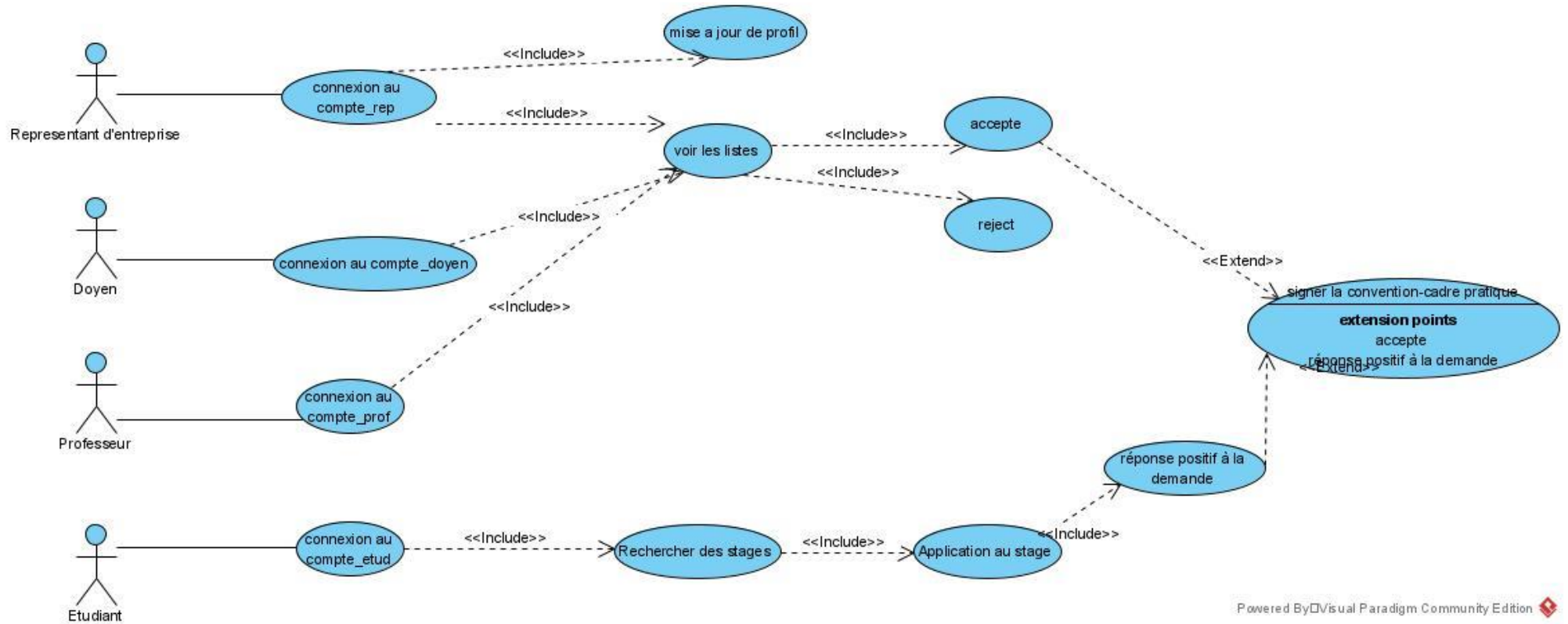
La légende:

Product Owners	L' équipe
Business Analyst	Iulia Stanica
Design Director	Marus Cristina
Backend Developer	Sara Solot
Head of CRM	Sara Solot
Frontend Developer	Marus Cristina

R	Responsible
A	Accountable
I	Informed
S	Support
C	Consulted

4. Cas d'utilisation

4.1. Diagramme UML



4.2. Description des cas

[UC1] Creation compte	
Acteurs	Etudiants, Professeurs, Représentant de l'entreprise, Doyen
Règles métier	RM1
Sommaire	Ce cas d'utilisation décrit la création du compte de tout utilisateur de la plateforme
Précondition	L'application doit contenir un nombre de comptes plus petit que la capacité maximale de l'application.
Postcondition	Un nouveau compte est ajouté dans la base de données
Pas	<ol style="list-style-type: none"> 1. Sélectionner l'option « Créez un compte » 2. Le système affiche la fenêtre correspondante à cette option 3. L'utilisateur introduit un nom d'utilisateur, une adresse de courriel, un mot de passe et la confirmation du mot de passe 4. L'utilisateur envoie le formulaire pour la vérification 5. Le système vérifie les données introduites 6. Le système affiche le message de succès et la page de connexion
Exceptions	<p>Au pas 5, si les données introduites ne sont pas valides, le système affiche un message d'erreur et on revient au pas 3.</p> <ol style="list-style-type: none"> 5.a. Il y a un champ vide 5.b. Le nom d'utilisateur est déjà associé à un autre compte 5.c. Le nom d'utilisateur a trop peu de lettres 5.d. L'adresse de courriel introduite n'a pas une forme valide 5.e. L'adresse de courriel est déjà associée à un autre compte 5.f. Le mot de passe introduit n'est pas valide – trop peu de caractères ou des caractères non-acceptés 5.g. Les deux mots de passe introduits sont différents

[UC2] Authentification	
Acteurs	Etudiants, Professeurs, Représentant de l'entreprise, Doyen
Règles métier	RM1
Sommaire	Ce cas d'utilisation décrit l'authentification de tout utilisateur de la plateforme
Précondition	L'utilisateur a un compte déjà créé
Postcondition	L'utilisateur a accès au contenu de la plateforme
Pas	<ol style="list-style-type: none"> 1. Sélectionner l'option « Connexion » 2. Le système affiche la fenêtre correspondante à cette option 3. L'utilisateur introduit son nom d'utilisateur et son mot de passe 4. L'utilisateur envoie le formulaire pour la vérification 5. Le système vérifie les données introduites 6. Le système affiche le message de succès et la page principale
Exceptions	<p>Au pas 5, si les données introduites ne sont pas valides, le système affiche un message d'erreur et on revient au pas 3.</p> <ol style="list-style-type: none"> 5.a. Il y a un champ vide 5.b. Le nom d'utilisateur n'est pas associé à aucun compte 5.c. Le mot de passe introduit est incorrect

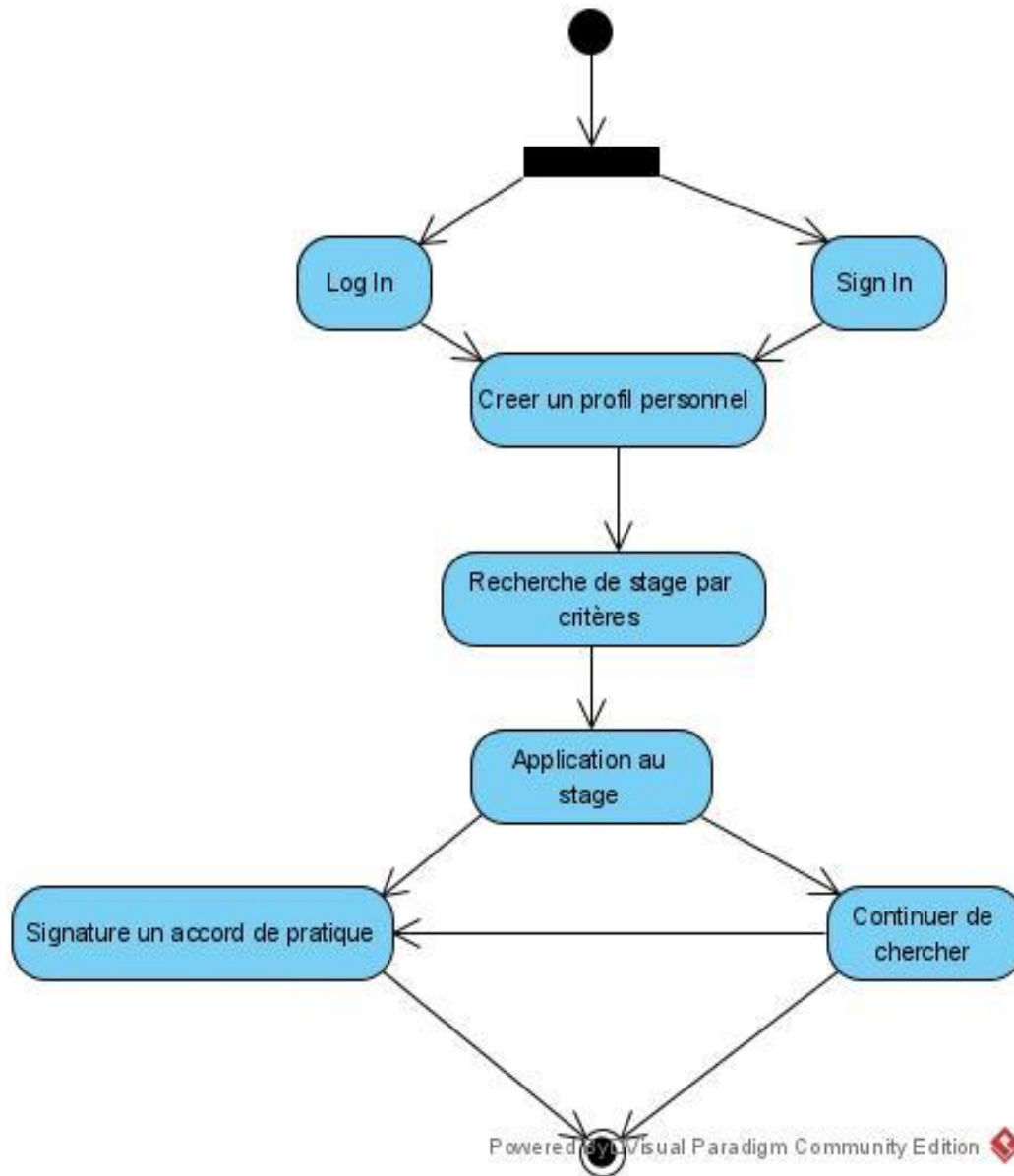
[UC3] Filtrage des stages	
Acteurs	Etudiants
Règles métier	RM1
Sommaire	Ce cas d'utilisation décrit l'option de filtrage des stages
Précondition	L'utilisateur est connecté
Postcondition	On affiche seulement les stages qui remplient les conditions du filtre
Pas	<ol style="list-style-type: none"> 1. Sélectionner l'option « Filtrer » 2. Le système affiche la fenêtre correspondante à cette option 3. L'utilisateur sélectionne les options qui lui conviennent 4. L'utilisateur envoie le formulaire pour le traitement 5. Le système recharge la fenêtre des stages 6. Le système affiche seulement les stages qui respectent les options choisies par l'utilisateur
Exceptions	<p>Au pas 5, si les données introduites ne sont pas valides, le système affiche un message d'erreur et on revient au pas 3.</p> <ol style="list-style-type: none"> 5.a. Il y a un champ vide 5.b. Le nom d'utilisateur n'est pas associé à aucun compte 5.c. Le mot de passe introduit est incorrect

[UC4]] Sauvegarde des stages	
Acteurs	Etudiants
Règles métier	RM1
Sommaire	Ce cas d'utilisation décrit le processus de sauvegarde des stages
Précondition	L'utilisateur est connecté
Postcondition	L'utilisateur a des stages sauvegardés dans son compte ; l'utilisateur reçoit des recommandations des stages
Pas	<ol style="list-style-type: none"> 1. Sélectionner l'option « Sauvegarder stages » 2. Le système traite la commande de sauvegarde 3. Le système affiche un message de succès
Exceptions	Au pas 2, s'il y a une erreur pendant la sauvegarde du stages, le système affiche un message d'erreur et on revient au pas 1.

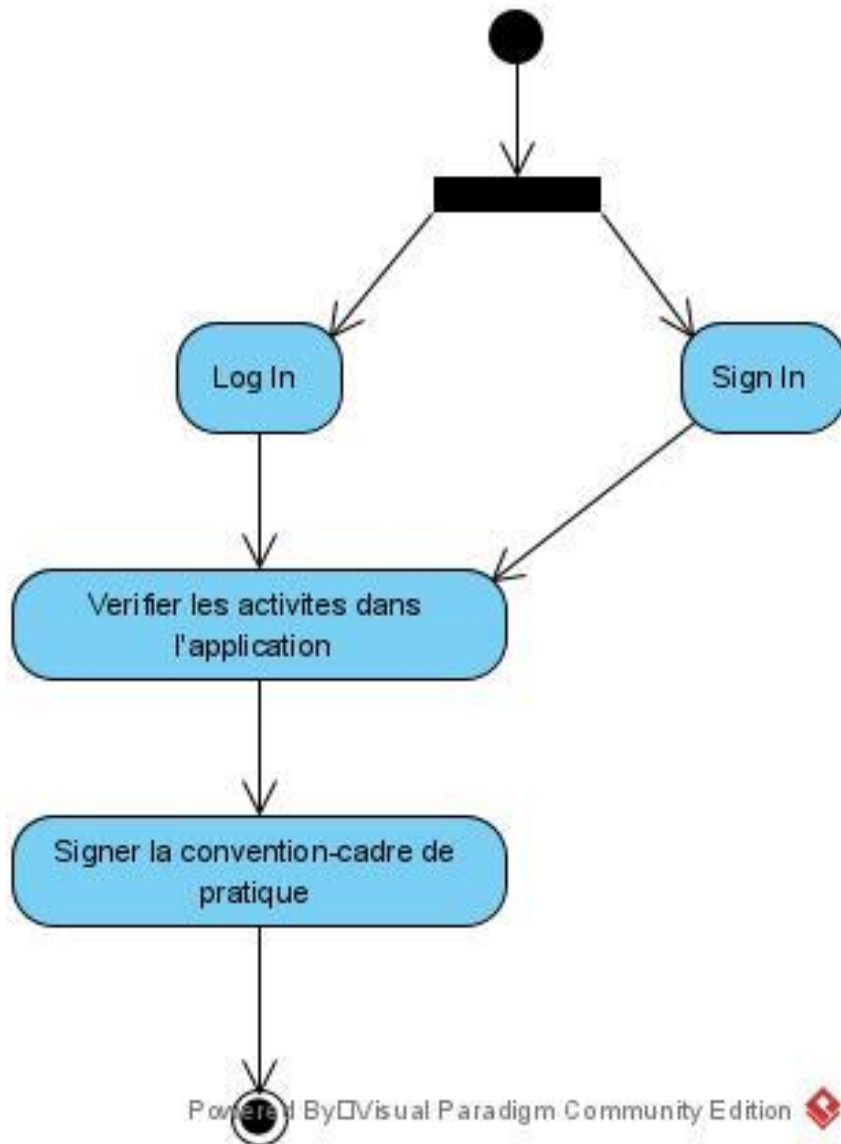
[UC5]] Effacement des comptes d'utilisateur	
Acteurs	Doyen
Règles métier	RM1, RM4
Sommaire	Ce cas d'utilisation décrit le processus de effacer des comptes
Précondition	L'utilisateur est connecté en tant qu'administrateur
Postcondition	Le compte d'utilisateurs est effacé de la base de données
Pas	<ol style="list-style-type: none"> 1. Sélectionner l'option « Eliminer compte » 2. Le système affiche une fenêtre de confirmation 3. Le système élimine le compte de la base de données 4. Le système affiche un message de succès 5. Le système envoie un courriel à l'utilisateur qui détenait le compte pour lui informer de son effacement
Exceptions	Au pas 3, s'il y a une erreur pendant l'élimination du compte, le système affiche un message d'erreur et on revient au pas 1.

5. Diagrammes d'activité

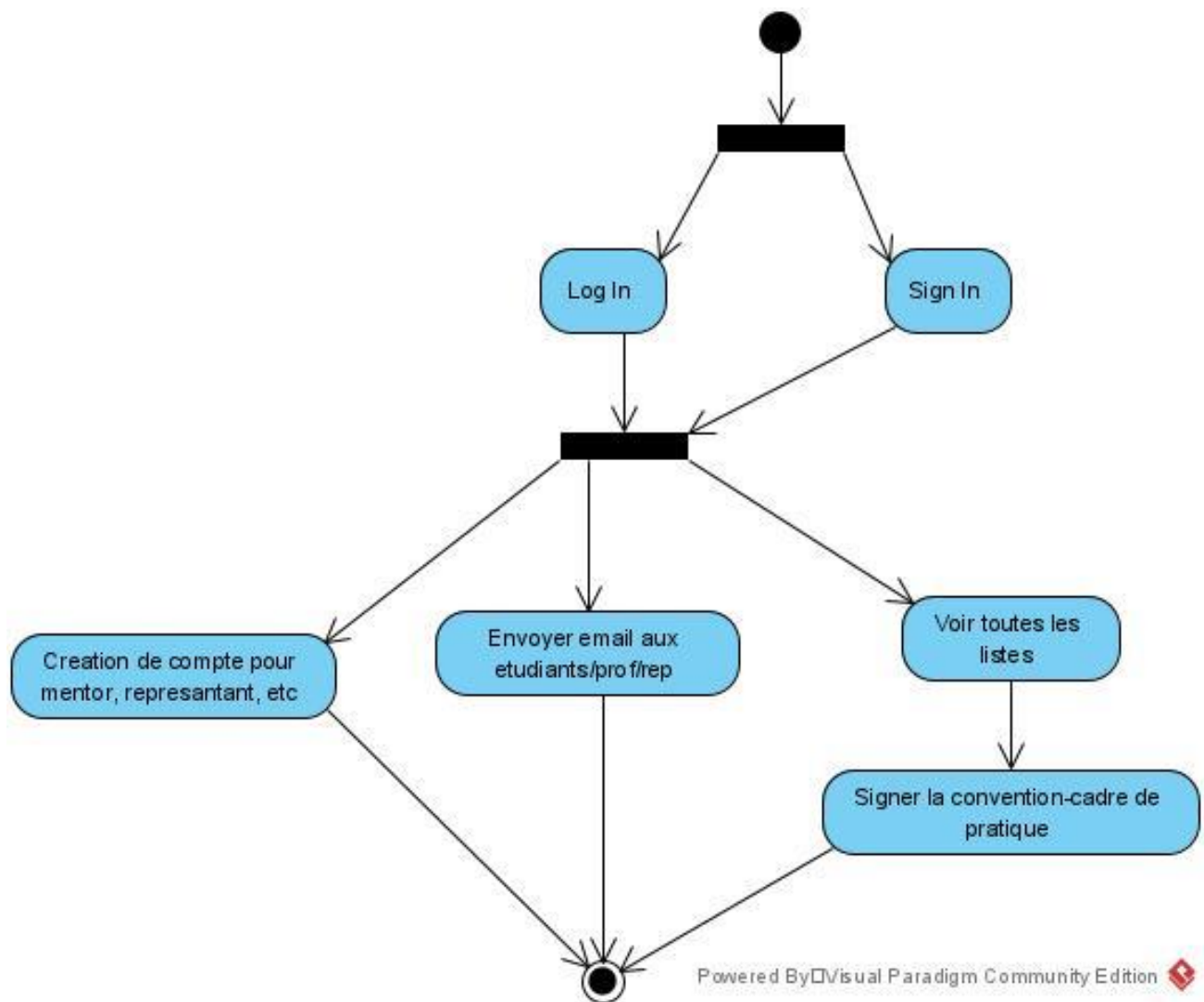
- pour étudiant:



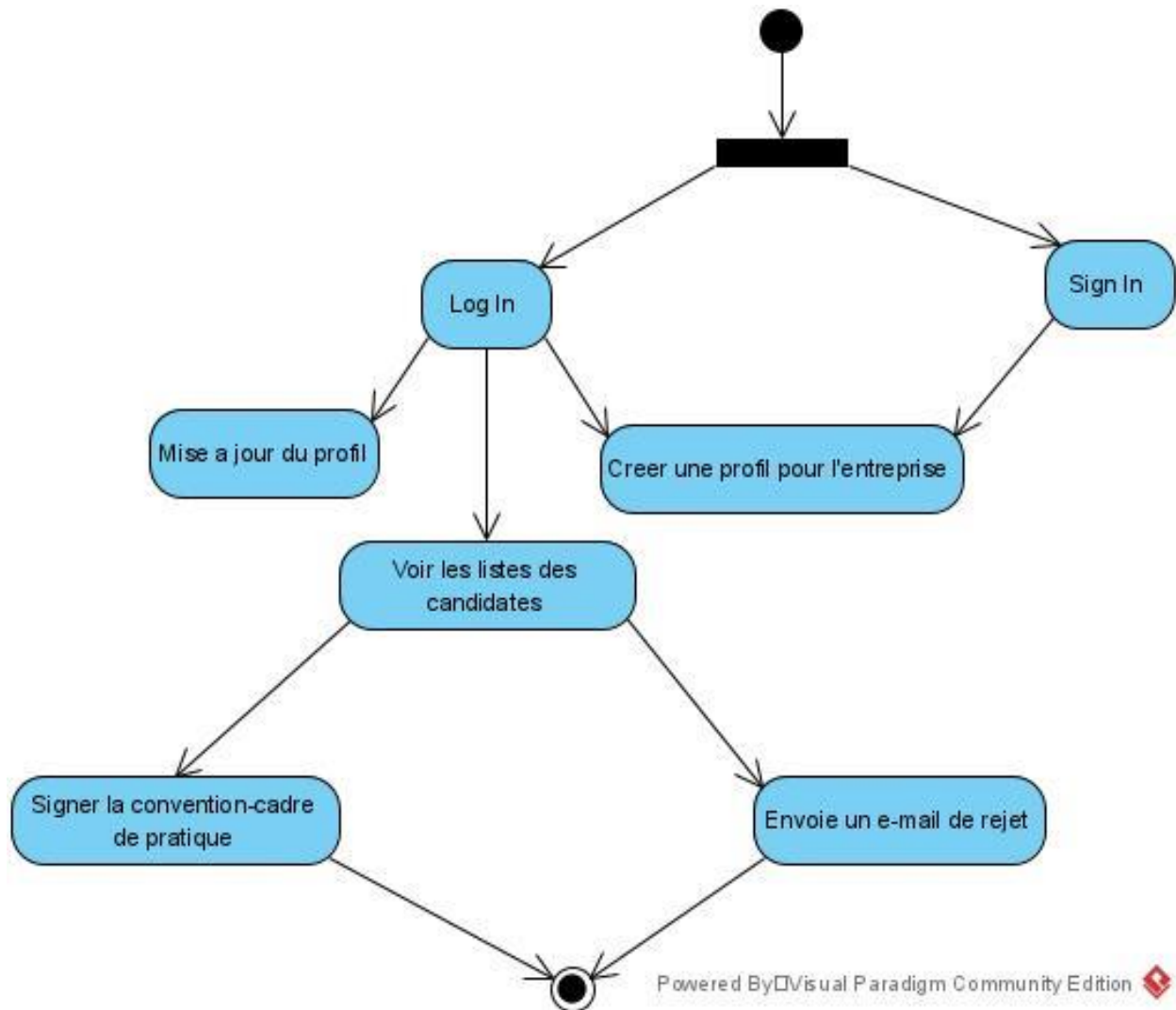
- pour professeur :



- pour Doyen

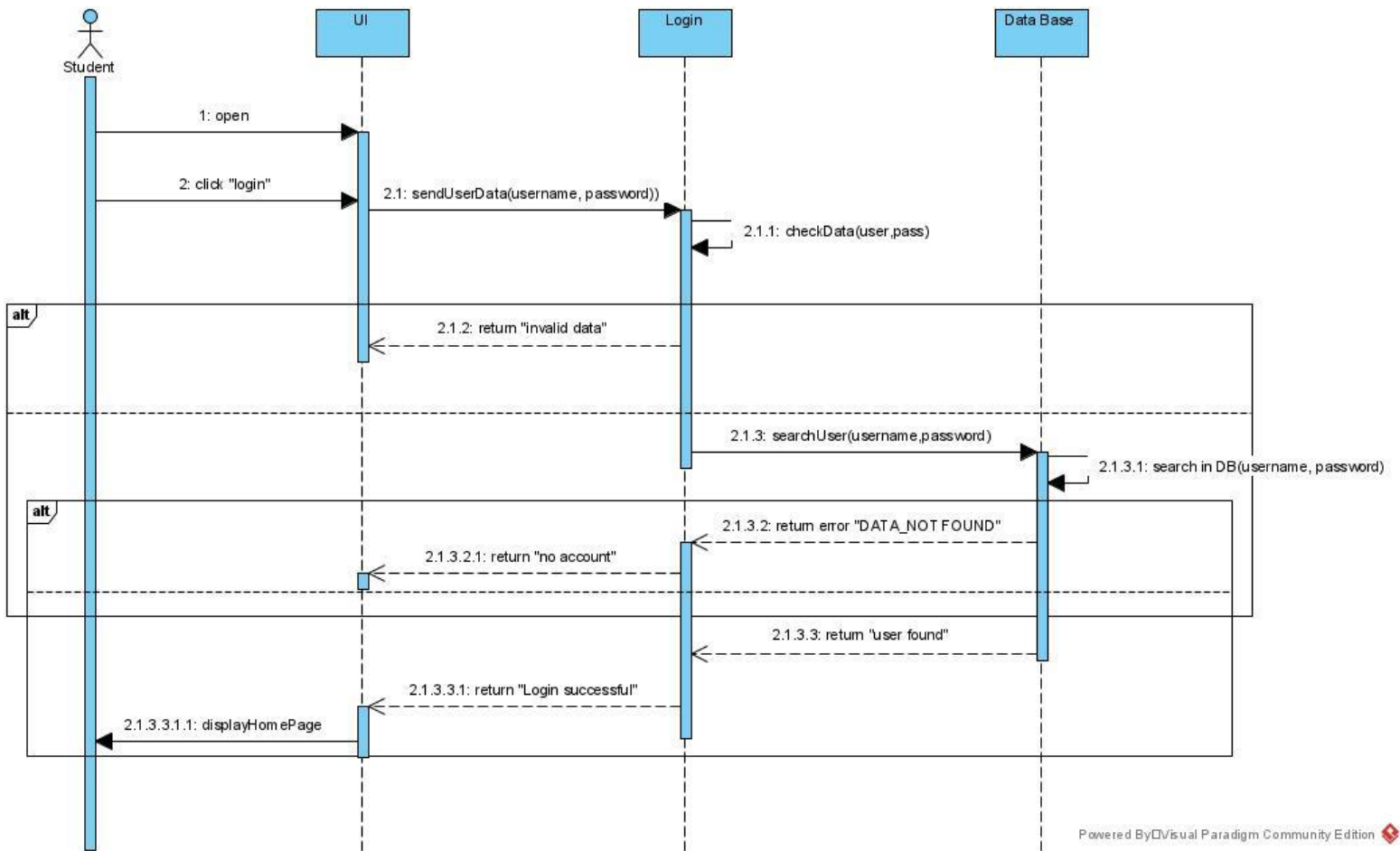


- pour la representant de l'entreprise :

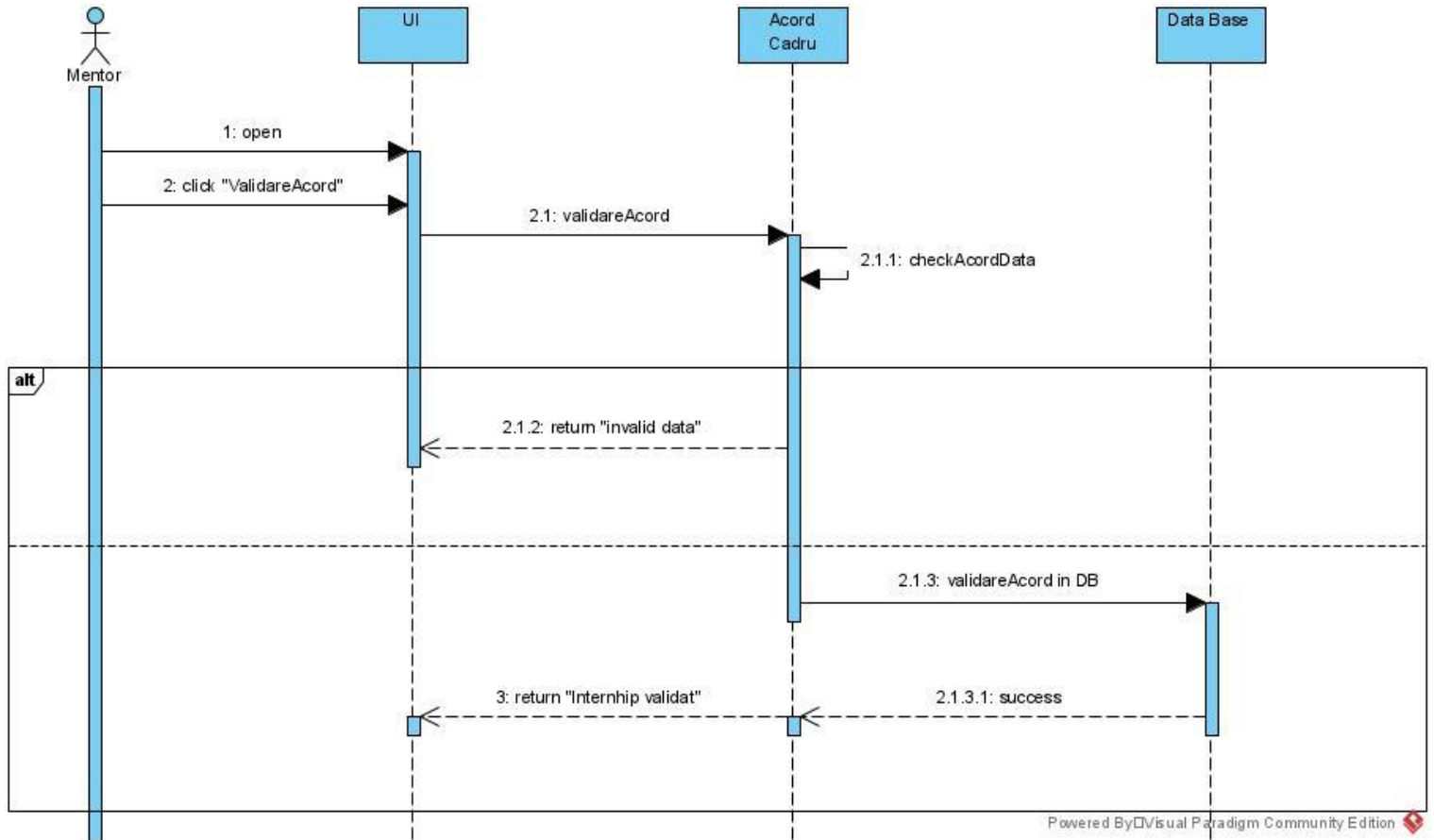


6. Diagrammes de séquence

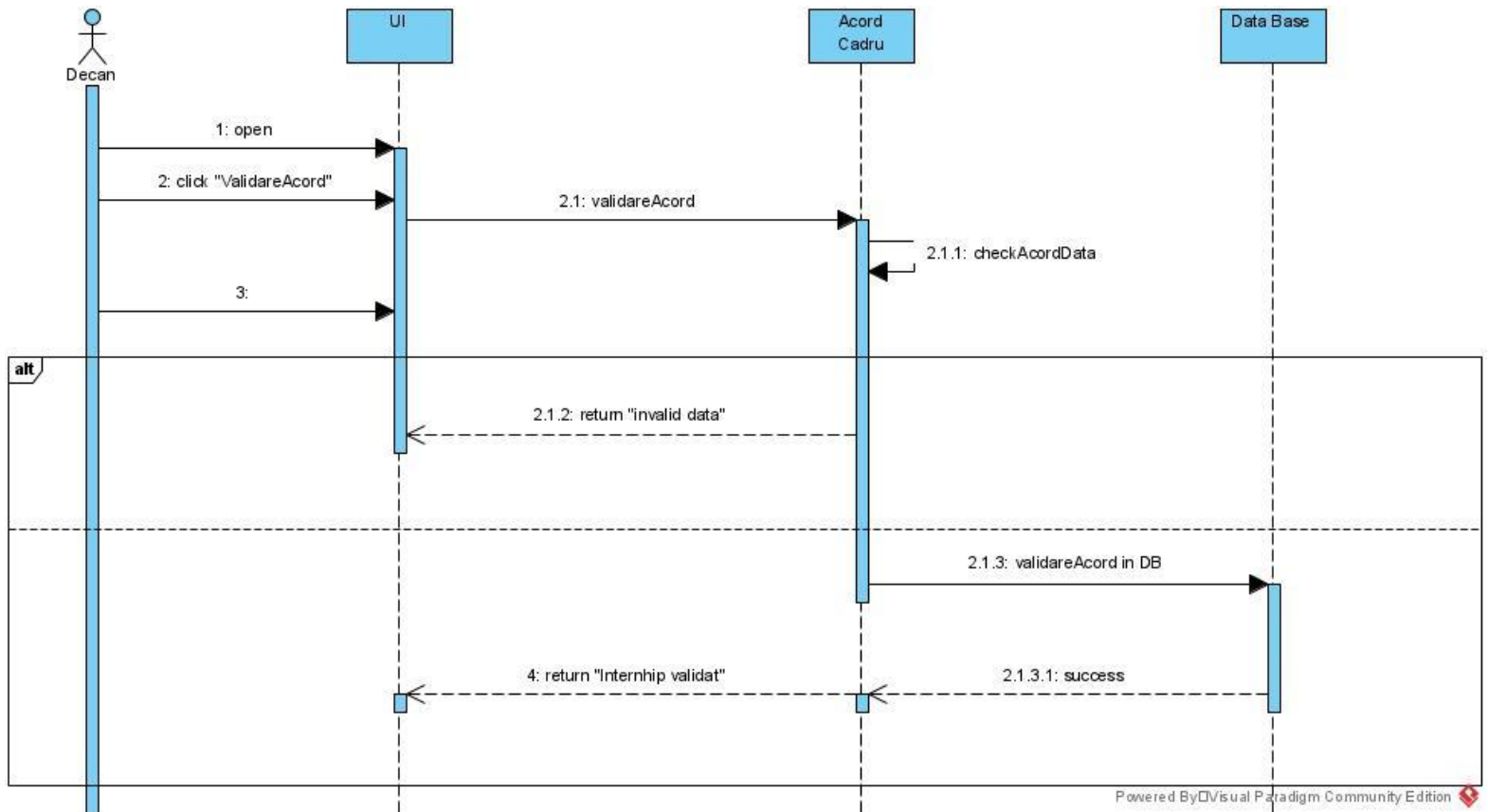
- Log in



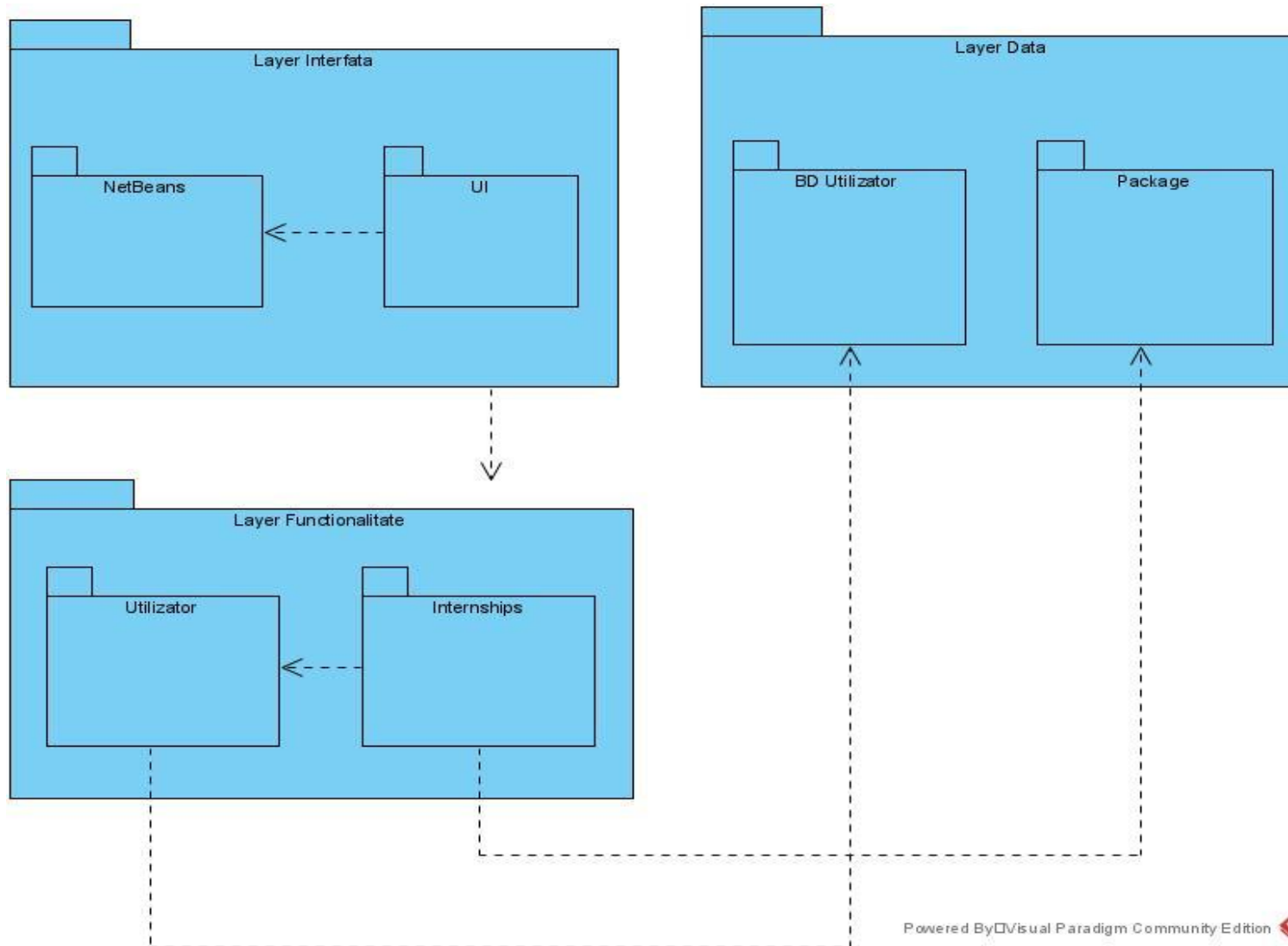
- Accord-cadre du mentor



- Accord-cadre du doyen



7. Diagrammes de paquetage



ETAPE II

1. Modèle architecturale

Pour notre application, nous avons choisi le modèle architectural 3-tier. Ce modèle est approprié parce qu'il reflète la séparation des couches existante dans notre projet. De plus, il a l'avantage qu'un peut modifier un des couches sans interférer avec les autres.

La représentation des couches est la suivante :

- couche de présentation : l'interface graphique de l'application
- couche logique : le traitement des données introduites, des actions des utilisateurs et de leurs recherches
- couche de stockage : la base de données avec les utilisateurs et les stages

2. Modèles de conception (Design patterns)

Pour notre projet, nous avons choisi les modèles de conception suivants :

Pour notre projet, nous avons choisi les modèles de conception suivants :

➤ **Model-View-Controller Pattern :**

- pourquoi : ce modèle permet la séparation facile entre les trois niveaux de l'application, en correspondance avec le modelé architectural ;
- dans notre projet : le « Model » est représenté par les classes Student, Internship, etc. ; le « View » est représenté par l'interface graphique, réalisé en JavaFX ; le « Controller » est représenté par les classes LoginController, RegisterController, etc.

➤ **Singleton Pattern :**

- pourquoi : ce modèle assure qu'il y a une seule instance d'une classe dans l'application ;
- dans notre projet : il est appliqué quand on fait la connexion avec la base de données : on fait une seule connexion, c'est-à-dire qu'on instancie un

seul objet de type ConnectionProvider

Composantes

Dans notre projet, les composantes principales sont les suivantes :

- l'interface : est réalisé avec NetBeans
- API: fait la liaison entre l'interface et l'application ; il prend les données introduites dans les champs de texte et les autres informations fournies par la fenêtre graphique et les envoie vers la couche logique de l'application pour être traitées
- le service : c'est la couche logique de l'application ; c'est le niveau où a lieu le traitement des données et aussi où il y a la seule liaison proprement dite avec la base de données
- la base de données : réalisé avec MySQL, elle contient les utilisateurs de la plateforme, les stages enregistrés, et des détails concernant les deux ; la connexion avec la base de données est faite au moment où on ouvre l'application, dans la fenêtre d'authentification

Diagramme de la base de données

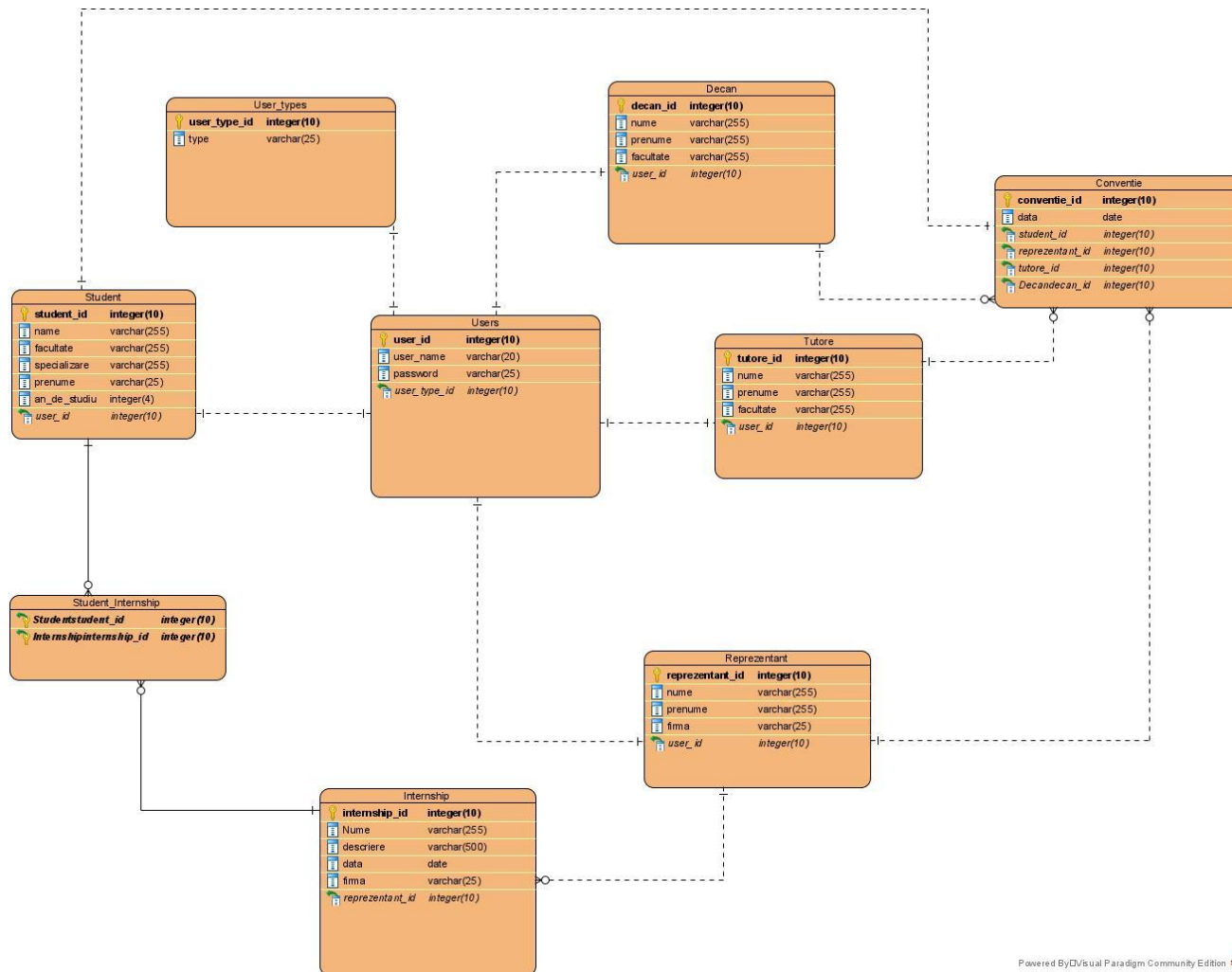


Diagramme de classes

