



RAPPORT DE CONCEPTION

de 1^{ère} année du master d'informatique
parcours MIAGE, rédigés par

Sabrina Cheurf

Jérémy Paulino

Mathieu Sarlangue

Projet Cours Particuliers à Distance

Table des matières

| | |
|---|-----------|
| Introduction | 3 |
| 1 Cahier des charges | 4 |
| 1.1 Présentation du projet | 4 |
| 1.1.1 Sujet du projet | 4 |
| 1.1.2 Description du projet | 4 |
| 1.1.3 Identification des acteurs | 4 |
| 1.2 Descriptif des fonctionnalités | 5 |
| 1.2.1 Fonctionnalités essentielles | 5 |
| 1.2.2 Fonctionnalités optionnelles | 6 |
| 1.3 Gestion de projet | 7 |
| 1.3.1 Méthode de gestion de projet | 7 |
| 1.3.2 Planification | 8 |
| 1.3.3 Gestion des risques | 8 |
| 2 Diagramme de cas d'utilisation | 9 |
| 2.1 Diagramme uml | 9 |
| 2.2 Description des cas d'utilisation | 10 |
| 3 Diagramme de classes | 20 |
| 3.1 Diagramme uml | 20 |
| 3.2 Description détaillée des classes | 21 |
| 4 Conception objet du système | 23 |
| 4.1 Diagramme de séquences | 23 |
| 4.2 Diagramme d'états | 30 |
| 5 Eléments ergonomiques | 32 |
| 5.1 Diagramme de navigation | 32 |
| 5.2 Maquettages d'écrans | 33 |
| 6 Eléments techniques | 40 |
| 6.1 Choix des outils et technologies | 40 |
| 6.2 Pattern utilisés | 41 |
| Conclusion | 42 |
| Annexes | 43 |
| Annexe I - Diagramme de Gantt | 43 |
| Annexe II - Diagramme de Pert | 44 |
| Annexe III - Affectation des tâches par ressource | 45 |

Introduction



Dans le cadre de la premier année de maters informatique parcours MIAGE, nous devons effectuer, durant toute l'année scolaire, un projet. Nous avons choisi le projet n°2, à savoir "Cours particulier à distance".

Notre groupe a été formé par affinité et non pas par défaut. Nous nous connaissons et avons déjà travaillé ensemble, dans le passé (L3), sur un projet.

Nous détaillerons ce projet de "plate-forme interactive" tout au long de ce rapport de conception.

Dans une première partie, nous présenterons le cahier des charges, précédemment rédigé, afin de s'en servir d'appui, de justification, pour les parties de conception qui suivent. Par la suite, nous présenterons le diagramme de cas d'utilisation de manière globale, pour ensuite décrire, plus en détails, un par un, les cas d'utilisations les plus récurrent. Nous donnerons, ensuite, le diagramme de classe en le détaillant classe par classe.

Lors d'une quatrième partie, nous étudierons la conception objet du système en donnant plusieurs diagrammes de séquence UML ainsi que quelques diagrammes d'états. Nous détaillerons aussi quelques éléments ergonomiques (diagramme de navigation, maquettages d'écrans, etc...). Enfin nous présenterons les différents outils de développement que nous comptons utilisés durant la phase de développement.

1 Cahier des charges

1.1 Présentation du projet

1.1.1 Sujet du projet

"La société Aidadom permet à des internautes de bénéficier de cours particuliers à distance. Pour se faire, la société dispose d'un site internet correspondant à une plate-forme de partage entre professeurs et élèves. Le paragraphe qui suit présente le fonctionnement général de cette plate-forme.

Après inscription, un internaute achète un certain nombre de cours et sélectionne le ou les cours qui l'intéressent, ainsi que les professeurs disponibles, et les créneaux horaires. Professeurs et élèves se retrouvent aux créneaux horaires, par l'intermédiaire d'un système de vidéo par internet qui ne fait pas directement partie du site de la société. Après chaque cours l'enseignant saisit un compte rendu sommaire du cours : exercices réalisés, progrès, difficultés de l'élève, ...

Chaque élève dispose d'un espace personnel dans lequel les enseignants peuvent déposer des exercices, des fiches pratiques, des évaluations... Des profils d'élèves sont calculés par le système pour permettre de cibler les élèves sur de nouvelles offres. Ces profils sont calculés à partir de l'historique d'activité sur la plate-forme."

1.1.2 Description du projet

L'objectif du projet est donc de développer une plate-forme interactive, accessible sur internet, mettant en relation des élèves et des professeurs. Outre son aspect fonctionnel, la plate-forme en ligne devra être intuitive et simple d'utilisation afin de cibler un large panel d'élève (du simple collégien jusqu'à l'étudiant). Le site proposera une inscription (élève ou professeur) qui entraînera la création d'un "espace personnel" propre à l'utilisateur.

Le projet a pour but de gérer la mise en relation élève/professeur mais ne gérera pas le système de vidéo par internet qui fait office de cours. En d'autres termes, la plate-forme planifiera l' "iRencontre" élève/professeur (heure du cours, ...) mais ne **traitera** pas du "vidéo-cours" en lui-même. Le site mettra en relation les attentes des élèves ainsi que les disponibilités des professeurs.

Très grossièrement, le projet sera constitué d'une base de **donnée** et d'une interface graphique (site internet) interagissant ensemble. La base de données stockera toutes les données d'inscription, de créneaux horaires de cours disponibles, d'exercices,...

1.1.3 Identification des acteurs

Nous avons **identifiés** les acteurs suivants :

- **L'administrateur** : Comme dans la plupart des sites internets dynamiques, il est indispensable d'avoir un administrateur. L'administrateur aura pour mission de gérer le contenu , les permissions des élèves ainsi que des professeurs , il sera également en charge de valider les candidatures des professeurs, ou de renvoyer les professeurs qui ne sont pas performants. Tout cela sera possible dans une interface spécialement conçue pour l'administrateur.
- **L'élève** : Un élève est un internaute qui connecté, il s'est inscrit dans le but de suivre des cours. Il a renseigné un email et un mot de passe lors de son inscription, il est identifié par l'email
- **Le professeur** : Un Professeur est un internaute connecté, il s'est inscrit pour donner des cours et a été validé par l'administrateur en tant que professeur du site, il est identifié par son email.
- **Le visiteur** : Un visiteur est un internaute qui n'est pas inscrit sur le site, c'est l'utilisateur qui aura accès au moins de contenu.

1.2 Descriptif des fonctionnalités

Cette partie sera divisé en deux sous partie : les fonctionnalités essentielles (indispensable pour coller au sujet) et celles optionnelles qui seront ajoutées ou non au projet, suivant le temps de développement restant.

1.2.1 Fonctionnalités essentielles

D'après le sujet et nos envies personnelles, nous avons relevé les fonctionnalités suivantes :

Fonctionnalités pour l'administrateur :

- Se connecter : permet à l'utilisateur de se connecter, il sera reconnu par le site en tant que administrateur, élève ou professeur et aura accès au contenu qui lui est permis.
- Se déconnecter : permet à l'utilisateur de se déconnecter afin de sécuriser son compte
- Valider ou refuser un professeur
- Consulter les notes des professeurs et renvoyer un professeur
- Supprimer un compte utilisateur

Fonctionnalités pour un professeur :

- Se connecter (identique admin)
- Se déconnecter (identique admin)
- Envoyer un message à un élève
- Après chaque cours l'enseignant saisit un compte rendu sommaire du cours, les exercices réalisés, les progrès , les difficultés de l'élève.
- Le professeur peut déposer des documents sur l'espace personnel des élèves
- Envoyer un message à un élève et lire ses propres messages
- Consulter ses crédits obtenus
- Demander à échanger ses crédits contre un chèque
- Renseigner ses horaires de disponibilités
- Renseigner son prix horaire
- Consulter l'historique des cours donnés
- Consulter sa note moyenne (quand nombre d'élèves > 5)

Fonctionnalités pour un élève :

- Se connecter (identique admin)
- Se déconnecter (identique admin)
- Rechercher un cours par niveaux scolaire puis rechercher un professeur et les créneaux horaires
- Acheter des credits et consulter ses credits
- Modifier son profil (classe)
- Noter les professeurs
- Consulter l'historique des cours
- Envoyer un message à un professeur
- Accéder a son espace personnel

Fonctionnalités pour un visiteur :

- S'inscrire (en tant qu'élève ou professeur, si professeur envoyer CV)
- Consulter les offres
- Voir les cours et les professeurs disponibles

Fonctionnalités du système :

- Calcul profil de cours

1.2.2 Fonctionnalités optionnelles

Fonctionnalités pour l'administrateur :

- Ajouter des matières de manière dynamique
- Ajouter des niveaux (6e, 5e, ...) de manière dynamique

Fonctionnalités pour un professeur :

- Ecriture de plan du cours (résumé de cours)
- Signaler un élève à l'administrateur

Fonctionnalités pour un élève :

- Consulter son historique d'activité

Fonctionnalités pour un visiteur :

- Contacter l'administrateur

Fonctionnalités du système :

- Emailing promotion
- Emailing crédits presque épuisés (>15euros)
- Emailing pour confirmer certaines opérations (crédits ajoutés, crédits échangés contre chèque...)

1.3 Gestion de projet

Nous étudierons, dans cette partie, la gestion générale du projet en détaillant la méthode de gestion choisie, la planification et la gestion des risques.

1.3.1 Méthode de gestion de projet

Nous avons décidé de nous baser sur la méthode de projet agile "SCRUM" en effet il est important de définir une méthode de projet claire afin de ne pas se faire piéger par un manque de temps. Cette méthode nous semble adaptée pour un travail en petite équipe puisque lors des mêlée nous auront la possibilité d'être réactif si un membre du groupe a du mal effectuer sa tâche, il pourra alors se faire aider ou guider par un autre membre. Les mêlées ne pourront pas être quotidienne dû à notre charge de travail en dehors de ce projet mais nous essayerons d'en faire régulièrement afin de contrôler l'avancement du projet. Un des autres avantage de cette méthode est qu'elle "découpe" le projet en plusieurs "sprint" les premiers sprints seront donc réservés aux fonctionnalités les plus importantes, et les derniers pourront nous permettre éventuellement d'implémenter des fonctions supplémentaires non indispensable.

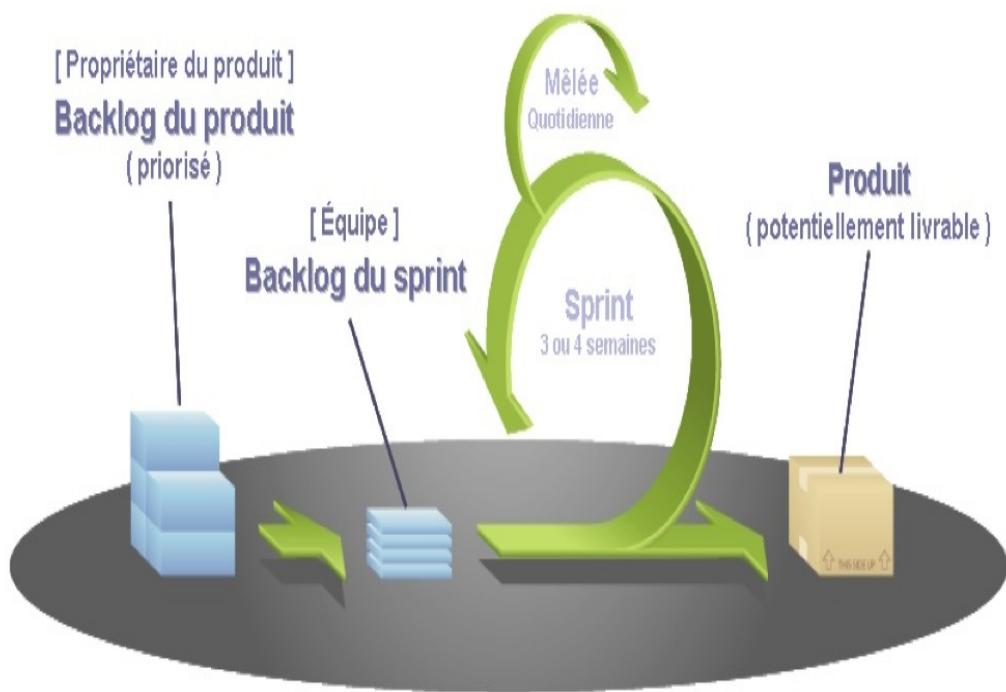


FIGURE 1 – Schéma récapitulatif de la méthode SCRUM

1.3.2 Planification

Nous détaillerons ici les différents diagrammes liés à la gestion de projet.

Diagramme de Gantt

Nous avons identifié deux grands processus dans ce projet : la phase de conception et celle de développement. La plupart des tâches du projet sont regroupées dans ces processus (sauf les phases de test). (Cf 1 -Annexe I Diagramme de Gantt)

Diagramme de Pert

Vous trouverez joint en annexe le diagramme de Pert résultant de celui de Gantt. (Cf 2 -Annexe II Diagramme de Pert)

1.3.3 Gestion des risques

Le risque majeur est d'avoir du retard dans le développement des fonctionnalités, la conséquence serait que certaines fonctionnalités, non indispensables au bon fonctionnement du site, seraient délaissées pour se concentrer sur les fonctionnalités indispensables.

Des risques techniques exigent aussi : L'apprentissage en J2EE, technologie imposée pour le projet, a été estimé à quelques semaines mais ce n'est qu'une estimation qui risque de retarder le projet.

Enfin, il existe de véritables risques de ressource. Nous sommes trois étudiants travaillant sur ce projet, chaque étudiant est rattaché à des tâches spécifique (Cf 3 -Annexe III Affectation des tâches par ressource), donc si un étudiant du groupe n'est pas assidu ou tombe malade par exemple, le planning serait modifié et certaines fonctionnalités secondaires, pourraient être, tout simplement, supprimées.

2 Diagramme de cas d'utilisation

2.1 Diagramme uml

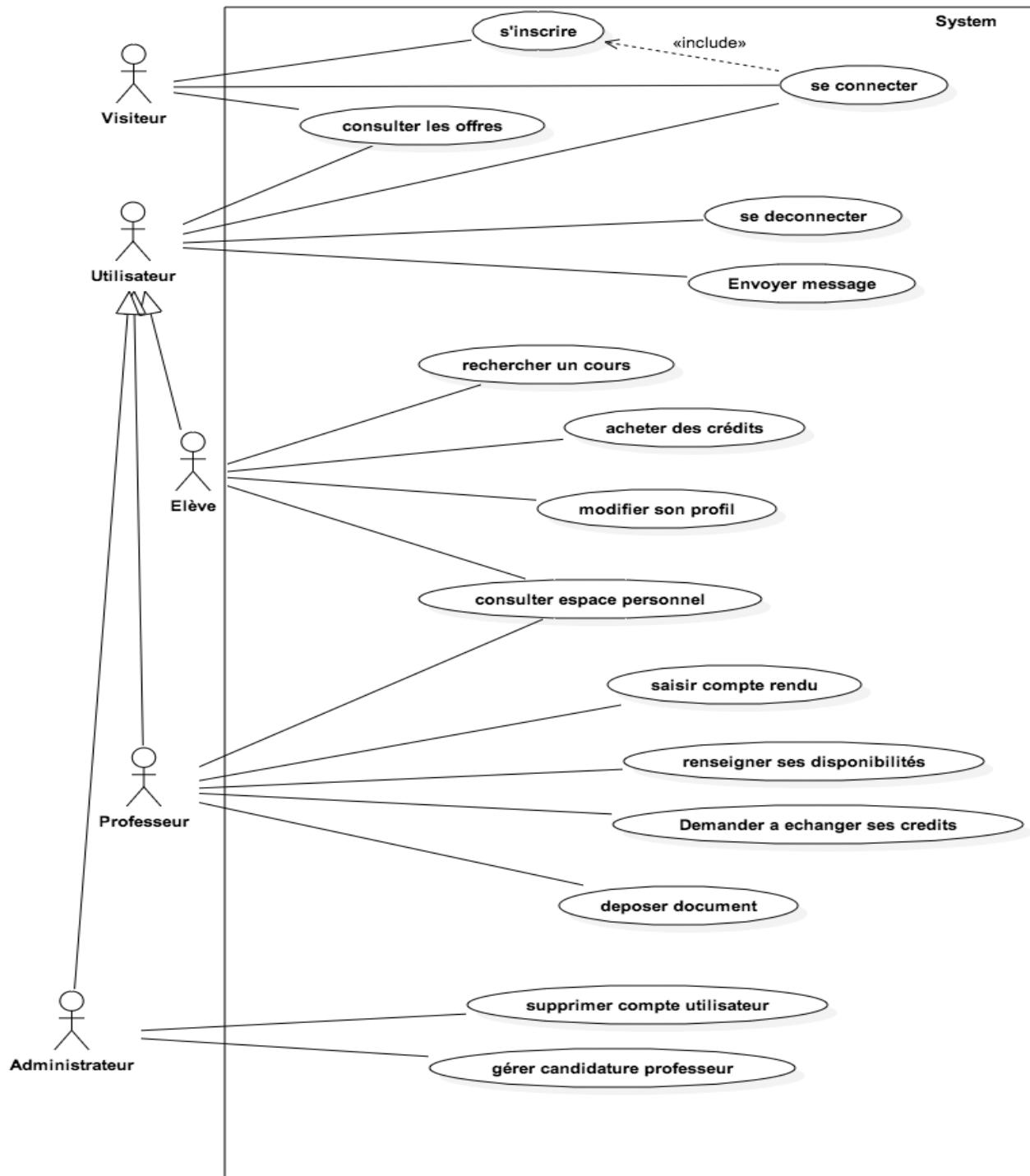


FIGURE 2 – Diagramme de cas d'utilisation

Dans le cahier des charges (chapitre 1), nous avions identifié quatre acteurs : l'administrateur, l'élève, le professeur, et le visiteur. Nous avons choisi de créer un acteur intermédiaire, à savoir l'utilisateur pour regrouper les actions communes entre professeur, élève et administrateur. Nous avons donc utilisé trois généralisations. Nous aurions pu, pour les mêmes raisons faire une généralisation

entre utilisateur et visiteur, mais il est impossible pour les élèves, professeurs et administrateurs de s'inscrire de nouveau. La généralisation n'est donc pas possible ici. Nous avons simplifié au maximum ce diagramme mais nous détaillerons plusieurs de ces actions, de façon plus complète, dans la partie suivante.

2.2 Description des cas d'utilisation

Cas d'utilisation "se connecter" :

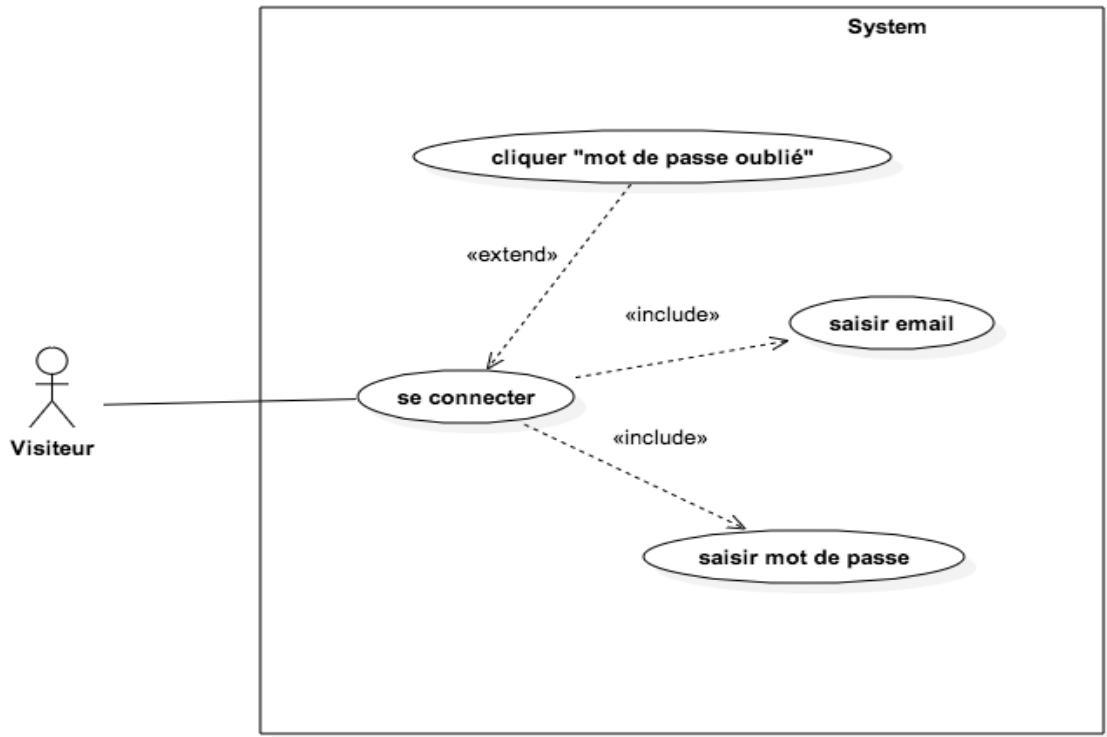


FIGURE 1 – Diagramme de cas d'utilisation "se connecter"

Système : Site iSchool

Acteur primaire : Le visiteur

Objectif : Se connecter sur son compte

Pré-condition : utilisateur déjà inscrit

Scénario nominal :

1. Le système demande au visiteur de renseigner son identifiant et son mot de passe.
2. Le visiteur saisit son identifiant et son mot de passe.
3. Le système valide et l'utilisateur se connecte sur sa session.

Scénario d'exceptions :

- 2a L'identifiant et/ou le mot de passe saisi par le visiteur ne correspond à aucune donnée dans notre base de données.
Le système informe le visiteur de son erreur et redémarre à l'étape 1.

Cas d'utilisation "s'inscrire" :

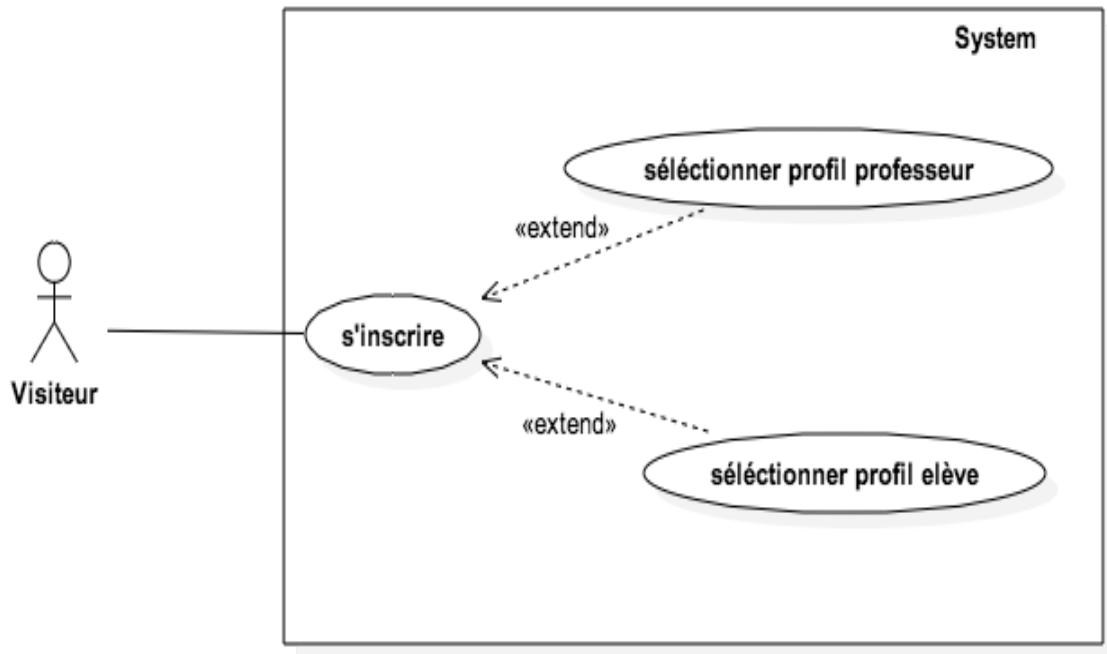


FIGURE 2 – Diagramme de cas d'utilisation "s'inscrire"

Système : Site iSchool

Acteur primaire : Le visiteur

Objectif : S'inscrire sur le site

Pré-condition : Le visiteur n'est pas déjà inscrit

Scénario nominal :

1. Le visiteur accède au site et clique sur s'inscrire.
2. Le système propose au visiteur de choisir un profil d'inscription (professeur ou élève).

Scénario d'exceptions :

- Aucun

Puisque les formulaires d'inscriptions professeur et élève sont différents (données supplémentaires nécessaires pour l'inscription des professeurs), on a deux scénario possible lors de l'inscription d'un membre :

- Une inscription pour les visiteurs qui ont le profil professeur
- Une inscription pour les visiteurs qui ont le profil élève

Cas d'utilisation "s'inscrire (professeur)" :

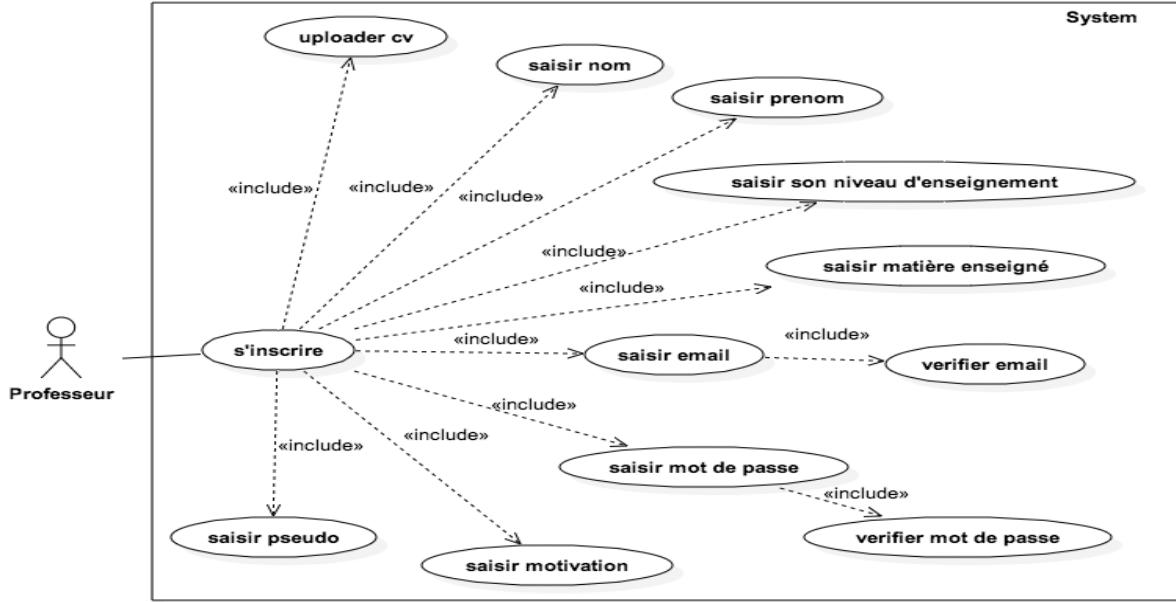


FIGURE 3 – Diagramme de cas d'utilisation "s'inscrire (professeur)"

Système : Site iSchool

Acteur primaire : Visiteur (professeur)

Objectif : S'inscrire sur le site

Pré-condition : Le professeur n'est pas déjà inscrit

Scénario nominal :

1. Le professeur accède au site et clique sur s'inscrire.
2. Le système propose au professeur de choisir un profil d'inscription (professeur ou élève).
3. Le professeur clique sur le profil d'inscription "professeur"
4. Le système affiche le formulaire d'inscription approprié.
5. Le professeur saisit les informations requises (mail, vérification mail, mot de passe, vérification mot de passe, niveau d'enseignement, adresse, pseudo).
6. Le professeur saisit ses motivations.
7. Le professeur upload son cv.
8. Le professeur valide l'inscription.
9. Le système confirme l'envoi de la demande d'inscription.

Scénario d'exceptions :

- 5a Le professeur n'utilise pas deux fois le même mot de passe pour les champs "mot de passe" et "vérification de mot de passe".
- 5b Le professeur n'utilise pas deux fois la même adresse email pour les champs "email" et "vérification d'email".
- 5 Le système avertit le professeur de son erreur par un message mais continu l'étape en cours.
- 9a Un ou plusieurs des champs requis n'ont pas été remplis.
- 9b L'adresse mail n'est pas au bon format.
- 9c Le pseudo est déjà utilisé par un autre inscrit.
- 9d Il y a déjà un compte d'enregistré avec cette adresse mail.
- 9e Le format du fichier cv n'est pas le bon.
- 9 Le système informe le professeur de son erreur et redémarre à l'étape 4.

Cas d'utilisation "s'inscrire (élève)" :

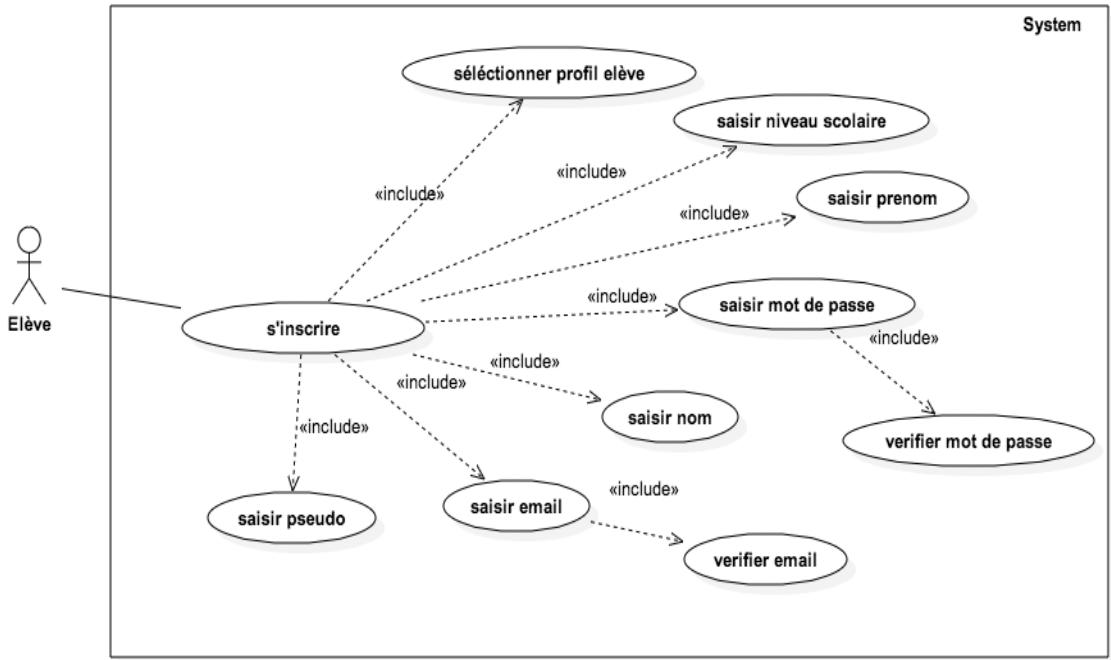


FIGURE 4 – Diagramme de cas d'utilisation "s'inscrire (élève)"

Système : Site iSchool

Acteur primaire : Visiteur (élève)

Objectif : S'inscrire sur le site

Pré-condition : L'élève n'est pas déjà inscrit

Scénario nominal :

1. L'élève accède au site et clique sur s'inscrire.
2. Le système propose au à l'élève de choisir un profil d'inscription (professeur ou élève).
3. L'élève clique sur le profil d'inscription "élève"
4. Le système affiche le formulaire d'inscription approprié.
5. L'élève saisit les informations requises (mail, vérification mail, mot de passe, vérification mot de passe, niveau scolaire, pseudo).
6. L'élève valide l'inscription.
7. Le système confirme l'envoi de la demande d'inscription.

Scénario d'exceptions :

- 5a L'élève n'utilise pas deux fois le même mot de passe pour les champs "mot de passe" et "vérification de mot de passe".
- 5b L'élève n'utilise pas deux fois la même adresse email pour les champs "email" et "vérification d'email".
- 5 Le système avertit l'élève de son erreur par un message mais continu l'étape en cours.
- 9a Un ou plusieurs des champs requis n'ont pas été remplis.
- 9b L'adresse mail n'est pas au bon format.
- 9c Le pseudo est déjà utilisé par un autre inscrit.
- 9d Il y a déjà un compte d'enregistré avec cette adresse mail.
- 9 Le système informe le professeur de son erreur et redémarre à l'étape 4.

Cas d'utilisation "rechercher cours" :

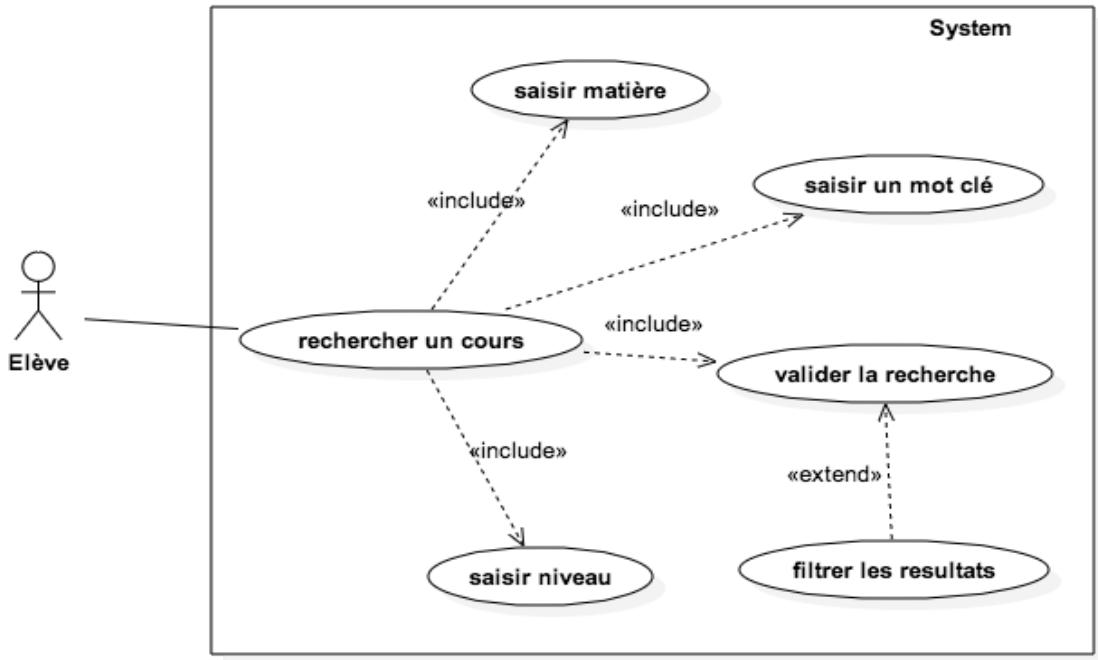


FIGURE 5 – Diagramme de cas d'utilisation "rechercher cours"

Système : Site iSchool

Acteur primaire : Elève

Objectif : Rechercher un cours disponible

Pré-condition : L'élève est authentifié

Scénario nominal :

1. L'élève clique sur "rechercher cours" dans son espace personnel.
2. Le système demande à l'élève de sélectionner une matière.
3. L'élève sélectionne la matière voulu.
4. Le système propose à l'élève de saisir un mot clé.
5. L'élève saisit un mot clé.
6. Le système pré-sélectionne le niveau scolaire de l'élève (niveau scolaire indiqué par l'élève lors de son inscription).
7. L'élève clique sur "lancer la recherche".
8. Le système fournit comme résultat une liste de cours disponible correspondant à la matière et au niveau scolaire indiqué par l'élève.

Scénario alternatif :

- 6a L'élève change la valeur du niveau scolaire, pré-sélectionné par le système, et saisit un autre niveau scolaire (pour des révisions par exemple).
- 7a L'élève choisit un filtre pour classer les résultats.
- 7 Le système prends en compte le filtre et classe ou supprime les cours en conséquence.

Scénario d'exceptions :

- 7a Il n'y a aucun cours disponible.
- 7 Le système redémarre à l'étape 2.

Cas d'utilisation "acheter crédit" :

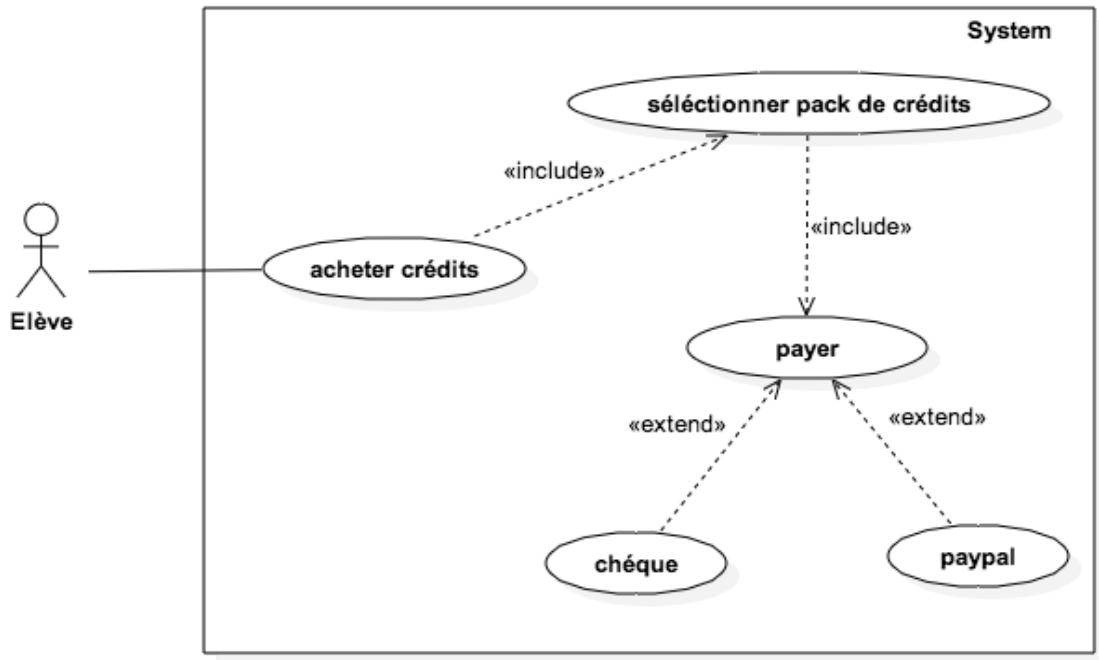


FIGURE 6 – Diagramme de cas d'utilisation "acheter crédit"

Système : Site iSchool/Plateforme PayPal

Acteur primaire : Elève

Objectif : Acheter des crédits

Pré-condition : L'élève est authentifié

Scénario nominal :

1. L'élève clique sur l'icône "panier" qui représente l'achat de crédit.
2. Le système demande à l'élève **d'indiquer** le nombre de crédits souhaité.
3. L'élève sélectionne le nombre de **crédit** souhaité.
4. L'élève valide son choix en cliquant sur le bouton de soumission.
5. Le système propose à l'élève deux moyens de paiement.
6. L'élève choisit PayPal comme moyen de paiement.
7. Le système redirige l'utilisateur vers la plateforme PayPal et attend la confirmation, par le système extérieur PayPal, du paiement.
8. Une fois le paiement reçu le système crédite le compte de l'élève des crédits achetés.

Scénario alternatif :

- 6a L'élève sélectionne le moyen de paiement "par chèque".
- 6 Le système indique à l'élève que son compte sera crédité une fois le chèque reçu et redémarre à l'étape 1.

Scénario d'exceptions :

- 7a Le système ne reçoit pas de confirmation de paiement par PayPal.
- 7 Le système indique à l'élève une erreur de transaction et redémarre à l'étape 2.

Cas d'utilisation "déposer document" :

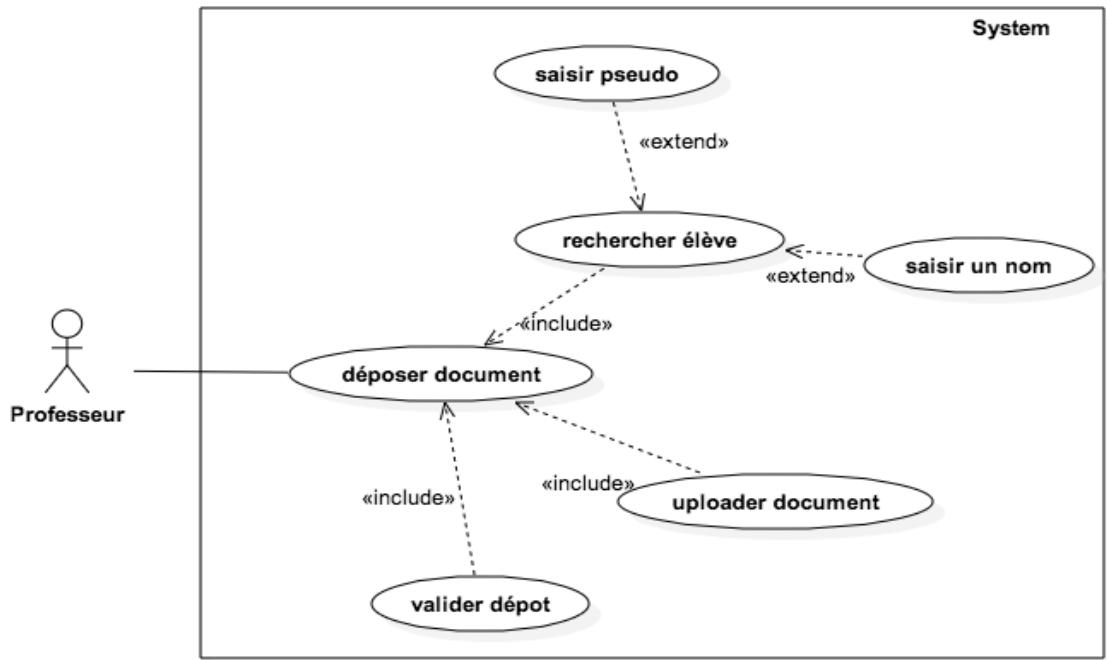


FIGURE 7 – Diagramme de cas d'utilisation "déposer document"

Système : Site iSchool

Acteur primaire : Professeur

Objectif : Déposer un document sur l'espace personnel d'un élève

Pré-condition : Le professeur est connecté et a déjà donné un cours à l'élève concerné

Scénario nominal :

1. Le professeur utilise la barre de recherche pour trouvé l'élève visé en saisissant son pseudo.
2. Le système renvoi au professeur la fiche de l'élève concerné.
3. Le professeur clique sur "soumettre un document".
4. Le système propose au professeur de uploader un document.
5. Le professeur upload le document.
6. Le professeur valide le dépôt en cliquant sur "valider".
7. Le système vérifie la taille du document.
8. Le système stock le document.
9. Le système indique un lien (lien de téléchargement du fichier en question) sur l'espace personnel de l'élève.
10. Le système indique au professeur que le document a bien été partagé.

Scénario d'exceptions :

- 1a Le professeur saisit le nom ou le prénom de l'élève.
- 1 Le système propose une liste d'élève résultant de la recherche et le professeur sélectionne l'élève en question.
- 7a Le fichier est trop volumineux pour être stocké.
- 7 Le système indique au professeur que le fichier est trop volumineux, lui indique aussi une borne de taille maximale et redémarre à l'étape 4.

Cas d'utilisation "envoyer message" :

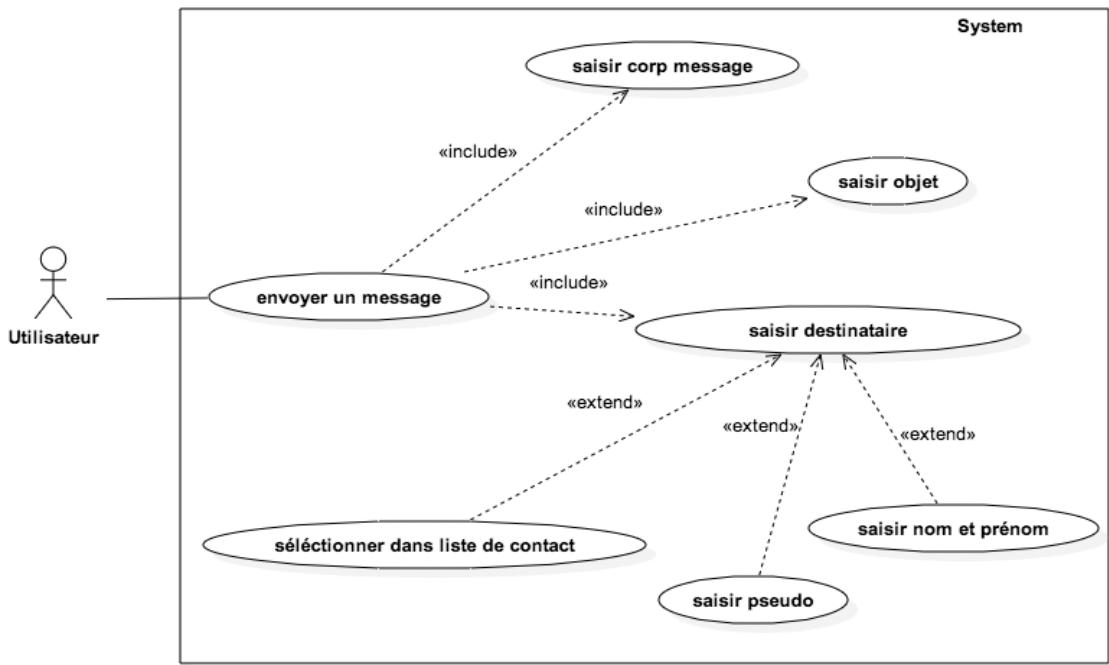


FIGURE 8 – Diagramme de cas d'utilisation "envoyer message"

Système : Site iSchool

Acteur primaire : Utilisateur

Objectif : Envoyer un message à un autre utilisateur.

Pré-condition : L'utilisateur est authentifié et est dans la rubrique contact

Scénario nominal :

1. L'utilisateur clique sur la rubrique contact.
2. Le système propose un formulaire contenant trois champs de saisie ("destinataire", "objet", "corp").
3. L'utilisateur saisit le corps du message ainsi que l'objet du message.
4. L'utilisateur clique sur le champ destinataire.
5. Le système affiche la liste des contacts de l'utilisateur (professeur qui a déjà donné un cours pour l'élève et élève à qui le professeur a déjà donné un cours pour le professeur).
6. L'utilisateur clique sur le destinataire voulu.
7. L'utilisateur clique sur "envoyer message".
8. Le système indique à l'utilisateur que son message a bien été envoyé.

Scénario alternatif :

- 5a L'utilisateur saisit directement le pseudo du destinataire.
- 5b L'utilisateur saisit directement le nom et prénom du destinataire
- 5 Le système n'affiche aucune liste et indique à l'utilisateur s'il a reconnu le destinataire en question.

Scénario d'exceptions :

- 5aa Le système ne trouve pas le pseudo dans la base de données.
- 5ab Le système ne trouve pas la correspondance nom/prénom dans la base de données.
- 5 Le système indique à l'utilisateur que le destinataire n'existe pas et redémarre à l'étape 5.

Cas d'utilisation "supprimer compte utilisateur" :

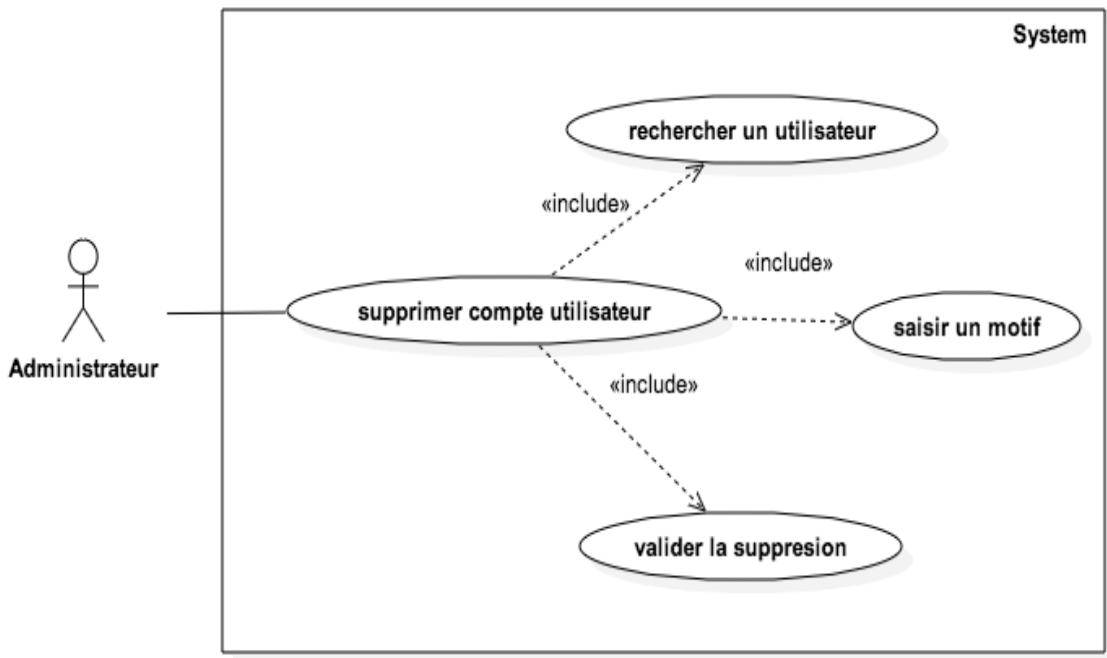


FIGURE 9 – Diagramme de cas d'utilisation "supprimer compte utilisateur"

Système : Site iSchool

Acteur primaire : Administrateur

Objectif : Supprimer un compte utilisateur.

Pré-condition : L'administrateur est authentifié en tant que tel.

Scénario nominal :

1. L'administrateur saisit le pseudo de l'utilisateur visé.
2. Le système redirige l'administrateur sur la fiche d'activité de l'utilisateur saisi.
3. L'administrateur clique sur supprimer ce compte.
4. Le système affiche une boîte de texte vide.
5. L'administrateur saisit les motifs qui entraînent la suppression du compte (ces motifs seront indiqué à l'utilisateur quand il essayera de se connecter à son compte).
6. L'administrateur clique sur "valider la suppression".
7. Le système supprime le compte de l'utilisateur dans la base de donnée et indique à l'administrateur que le membre a été supprimé.

Scénario d'exceptions :

- 2a Le système ne trouve pas le pseudo dans la base de donnée.
- 2 Il indique à l'administrateur qu'il y a une erreur de saisie et redémarre à l'étape 1.
- 6a Le champs de texte dédié aux motifs de suppression est vide.
- 6 Le système indique qu'il faut un motif de suppression et redémarre à l'étape 4.

Cas d'utilisation "renseigner ses disponibilités" :

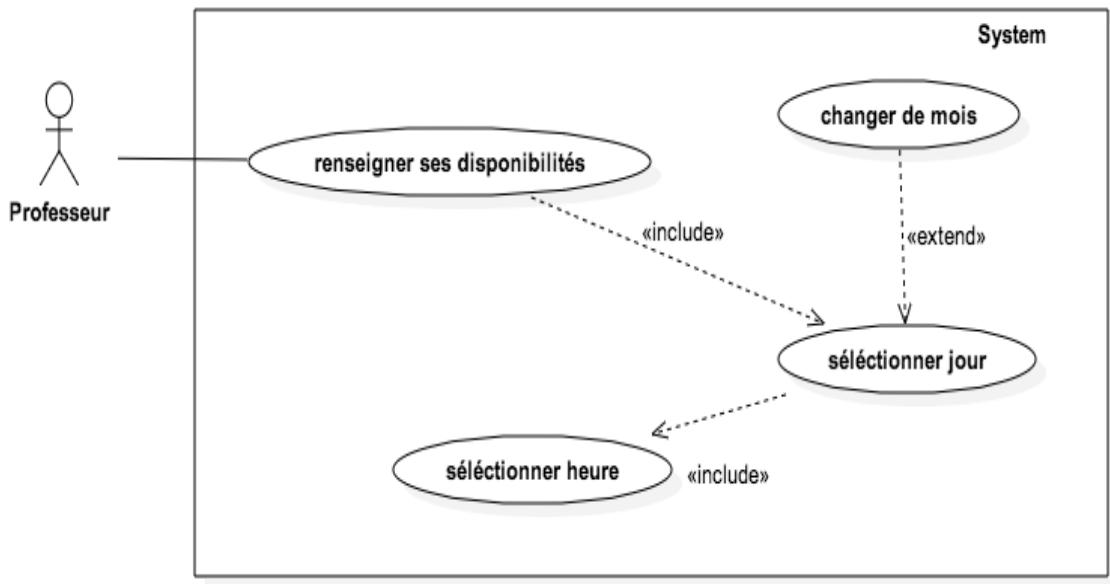


FIGURE 10 – Diagramme de cas d'utilisation "renseigner ses disponibilités"

Système : Site iSchool

Acteur primaire : Professeur

Objectif : Renseigner ses disponibilités pour donner un cours particulier.

Pré-condition : Le professeur est authentifié et est sur son espace personnel

Scénario nominal :

1. Le professeur clique sur "voir disponibilités".
2. Le système redirige le professeur sur la page de ses disponibilités (sous forme de calendrier).
3. Le professeur sélectionne le jour voulu dans le calendrier (il navigue dans le calendrier en faisant défiler les mois via deux bouton : précédent,suivant).
4. Le système ouvre une fenêtre détaillant les heures de la journée sélectionnée.
5. Le professeur coche les heures de disponibilité et ferme la fenêtre.
6. Le système ajoute les disponibilités du professeur dans la base de donnée.
7. Le système indique au professeur que ses disponibilité ont été mis à jour.

Scénario d'exceptions :

- aucune

3 Diagramme de classes

3.1 Diagramme uml

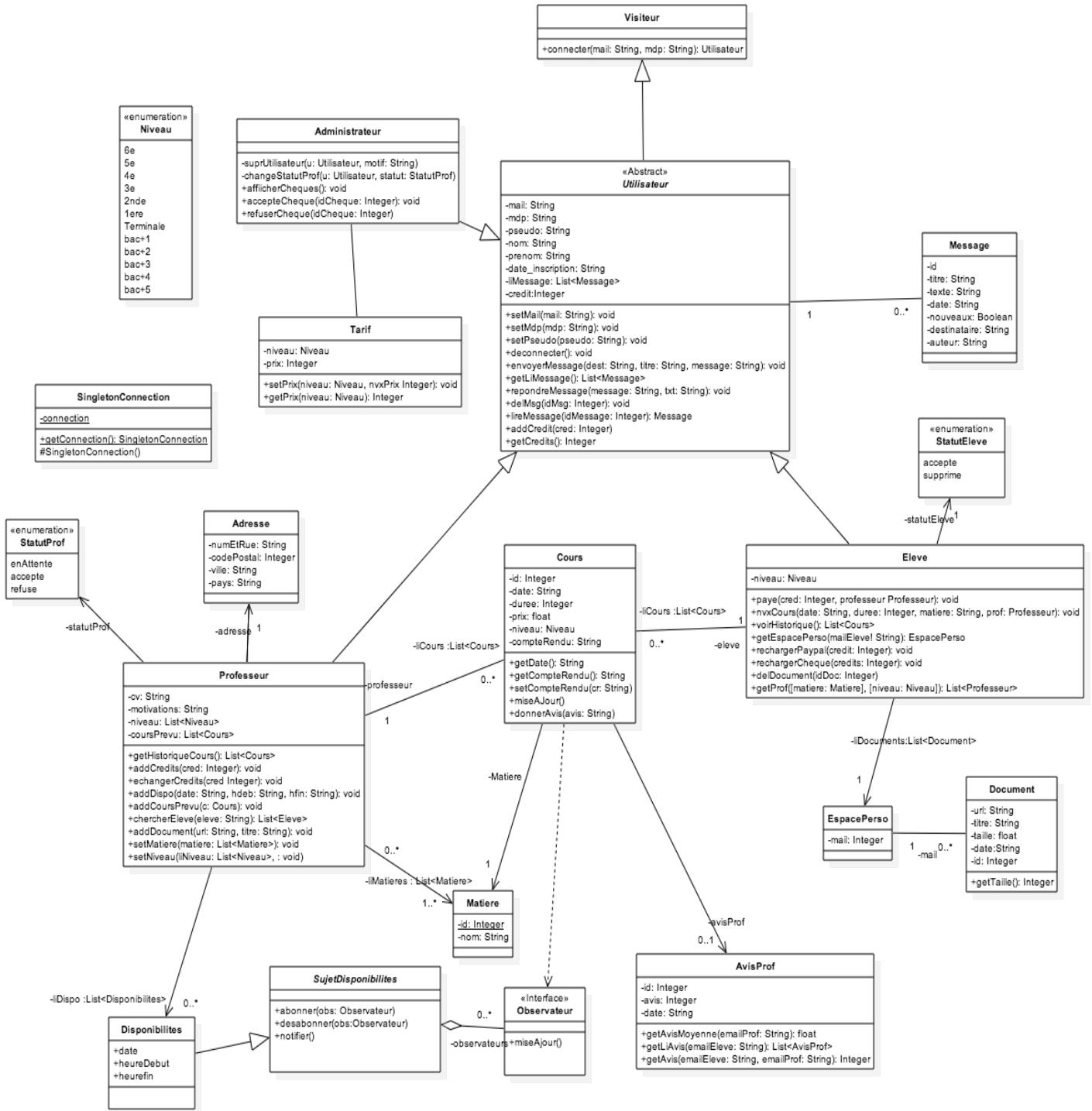


FIGURE 11 – Diagramme de classe

Nous voulions dans un premier temps déstructurer notre diagramme de classe pour le présenter sous trois **couche** : modèle, vue, contrôleur, donc en trois diagrammes de classe distinct. Mais le diagramme de classes représentant les vues se trouve être le diagramme de navigation (Cf partie 5 - Elements Ergonomiques), il n'y avait donc pas grand intérêt à le détailler dans cette partie. De même pour le diagramme de classes des contrôleurs, il n'est pas très utile en terme de conception de créer 6 ou 7 classes de servlet qui n'ont aucune interaction entre elles. Nous avons donc décidé de présenté le diagramme de classes dit "de modèle" qui nous semble être le plus intéressant.

3.2 Description détaillée des classes

Nous décrirons ici les différentes classes du diagramme une par une en détaillant notamment les méthodes et attributs de chacune d'entre elle.

Par soucis de clarté et afin d'aérer notre diagramme de classe nous n'y avons pas inclus les constructeurs de classes.

Visiteur : La classe visiteur représente un internaute quelconque qui n'a pas de statut défini (enseignant ou étudiant).

Utilisateur : La classe utilisateur représente un professeur un élève ou l'administrateur, ils peuvent avoir des messages qui sont regroupés dans une liste de message (liMessage) celle-ci est vide à la création de l'utilisateur, il possède également un nombre de crédit.

Professeur : Cette classe représente les professeurs, elle hérite de la classe Utilisateur, elle a en attribut statut qui est de type StatutProf et qui représente les différentes étapes de la candidature de l'enseignant(voir diagramme état), elle comporte également un cv qui est de type String représentant l'url de l'emplacement du CV une fois uploadé lors de son inscription, lors de cette dernière il fournit une adresse qui se fait via la classe Adresse. Le professeur peut assurer des cours à des niveaux différents (niveau : List<niveau>) mais également des matières différentes, ses deux informations sont renseigné lors de l'inscription et modifiable par la suite, il doit fournir ses disponibilités ce qui se fait avec la classe disponibilité. Il peut déposer des documents dans l'espace personnelle de l'élève avec la méthode addDocument(url :String, titre :String) : void de la classe EspacePerso. Le professeur accumule des crédits dès qu'un cours de ses cours est réservé par un élève, il pourra à tout moment les échanger contre un chèque ou un paiement paypal.

Administrateur : Il s'agit de la classe représentant les administrateurs du site, ils ont la possibilité de supprimer un utilisateur avec un motif suprUtilisateur(u :Utilisateur, motif :String), ce qui a pour effet de passer l'élève au statut supprime et le professeur au statut refuse(voir diagramme d'état) , c'est à l'administrateur aussi de gérer les chèques reçus grâce à afficherCheques, il pourra les accepter ou refuser grâce à accepterCheque ou refuserCheque.

Elève : La classe Eleve représente les élèves, elle hérite aussi de la classe Utilisateur, il peut rechercher des cours, s'inscrire à des cours, voir son historique de cours passés, il peut supprimer des documents de son espace personnelle. Elle a en attribut niveau, qui représente son niveau scolaire. Il peut ajouter des crédits à son compte en les achetant par chèque ou par Paypal , s'il n'a plus de crédits alors il ne peut s'inscrire à de nouveaux cours . Lorsqu'il réserve un cours, il utilise la méthode paye pour payer un professeur.

Statuts Elève : Il s'agit de l'énumération représentant le statut d'un élève qui peut être : accepté ou refusé. Par défaut un élève est accepté lors de son inscription, il devient refusé si l'administrateur le supprime.

Statuts Prof : StatutProf est une énumération représentant le statut d'un professeur qui peut être : en Attente, accepté, refusé.

Statuts Niveau : Niveau est une énumération représentant les niveaux scolaire : collège, lycée, supérieur... elle est utilisée pour définir le niveau de l'élève, ou la liste des niveaux enseignés par un professeur.

AvisProf : La classe AvisProf nous sera utile pour recueillir les avis sur les professeurs auprès des élèves ainsi le professeur pourra éventuellement être renvoyé s'il est mal évalué par plusieurs élèves. Elle a pour attribut un ID unique , un avis qui est un entier qui va de 0 à 5, il sera représenté sur le site par des étoiles Les élèves peuvent attribuer un avis à chacun de leurs cours passé au cours des deux derniers mois grâce à la méthode setAvis(in emailEleve :String, in emailProf :String, in avis :Integer) . Si le professeur a plus de 3 notes alors on peut lui calculer la moyenne de ses avis grâce à la méthode getAvisMoyenne(in emailProf :String) : float qui renverra un float. Chaque élève peut consulter la liste des notes qu'il a donné grâce à getLiAvis(in emailEleve :String) : List<AvisProf> et les modifier toujours grâce à setAvis (uniquement au cours des deux mois qui suivent le cours)

Tarif : La classe Tarif représente le prix horaire des cours de chaque Niveau. L'administrateur peut les modifier grâce à la méthode setPrix qui prend en paramètre le Niveau et son nouveau prix correspondant.

Matière : Chaque professeur enseigne au moins une matière, les matières qu'il enseigne sont stockées dans une liste de matière (liMatieres : List<Matiere>) les cours ont également une matière en attribut.

Espace Personnel : Chaque élève dispose d'un espace perso qui est construit lors de l'inscription il peut le consulter à sa guise, dans cette espace perso , les professeurs avec lesquels il a eu des cours peuvent uploader des documents (pdf, word, image ...) , chaque espace perso d'un élève est donc constitué d'une liste de documents (liDocuments) , chaque document possède un id unique, un titre, une taille(en Mo) , une URL qui correspond à l'emplacement du document, et un type.

Disponibilités : La classe Disponibilites représente les dates et heures de disponibilité du professeur.

Cours : Chaque cours concerne un professeur et un élève , il est défini par un ID unique, la matière enseignée, une durée d'enseignement de type entier , un prix(calculé en fonction du prix horaire et de la durée du cours et également un attribut compteRendu de type String, il s'agit du compte rendu que le professeur fera après avoir effectué le cours.

Document : Un document contient une url qui est l'emplacement du document sur le serveur, un titre et une taille.

Message : Il s'agit de la classe contenant les messages des utilisateurs, elle a comme attribut nouveau de type boolean qui détermine si le message a été lu ou pas encore une zone de texte pour l'écriture du message de type String ainsi que le nom de l'auteur et du destinataire qui sont de type string .

SingletonConnection : C'est l'objet pour accéder à la base de données.

SujetDisponibilité : C'est une classe abstraite qui permet de créer les disponibilités puis l'interface observateur rend compte des changements de disponibilités au cours qui se met à jour.

4 Conception objet du système

Dans cette partie, nous détaillerons dans un premier temps plusieurs diagramme séquence puis nous donnerons deux diagrammes d'état.

4.1 Diagramme de séquences

La description des différents diagrammes de séquence s'approche significativement au cas d'utilisation correspondant à la partie 2 (Cf Cas d'utilisation)

Interaction connexion :

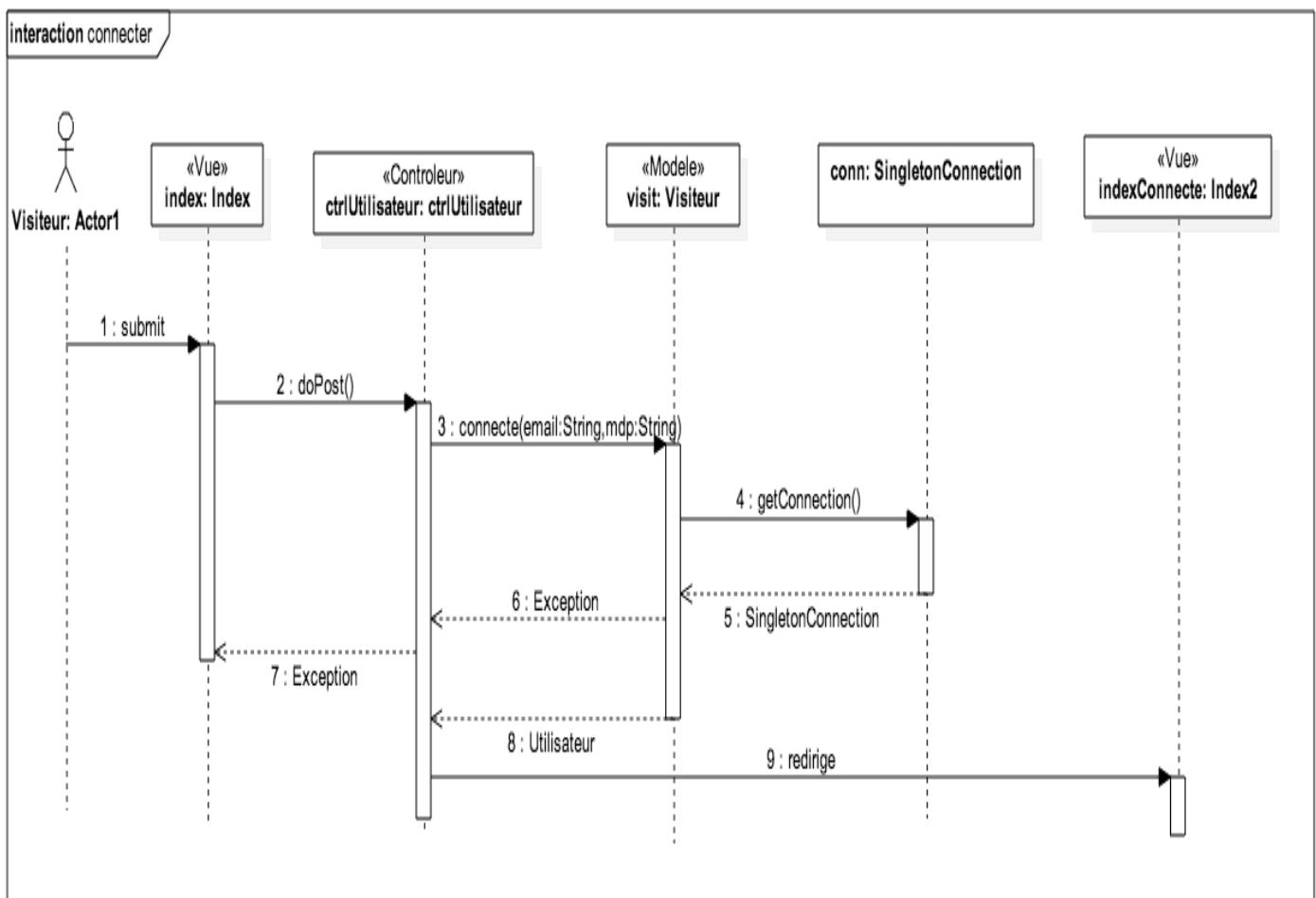


FIGURE 12 – Diagramme de séquence interaction connexion

Un visiteur souhaite se connecter, il remplit le formulaire de connexion , quand le visiteur valide le formulaire, les données sont transmis au contrôleur avec la méthode doPost qui va envoyer l'email et le mot de passe à la méthode connecter de la classe visiteur, qui renvoi l'utilisateur et le redirige vers la page index2.

Interaction inscription :

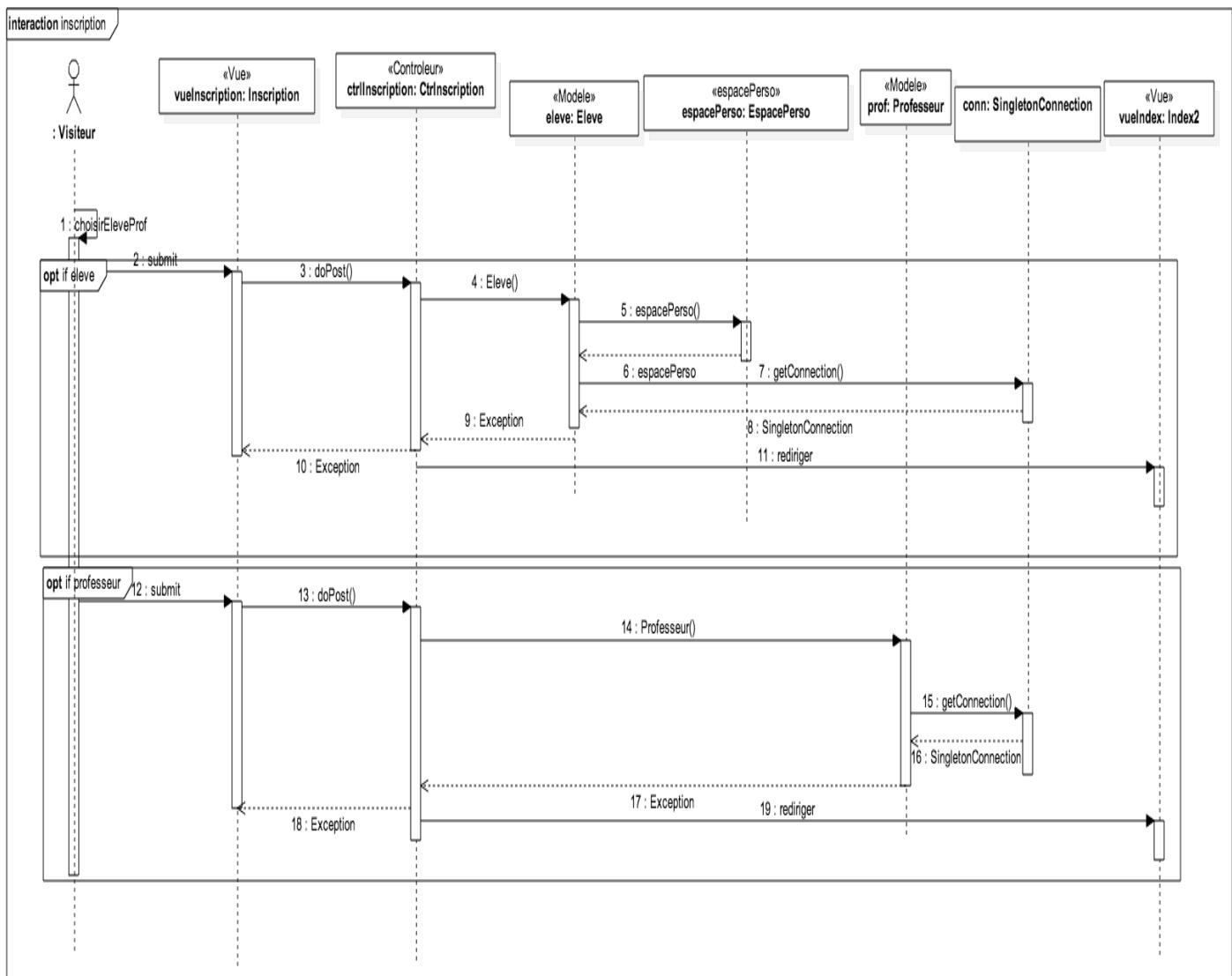


FIGURE 13 – Diagramme de séquence interaction inscription

Un Visiteur souhaite s’inscrire, pour cela il se rend sur la vue inscription, là il a le choix de s’inscrire en tant qu’élève ou en tant que professeur, le formulaire adéquat est affiché en fonction de son choix grâce au Javascript.

Le visiteur rempli le formulaire, les champs du formulaire sont transmis au contrôleur avec la méthode doPost qui va créer un objet de type Eleve ou Professeur en fonction du choix du visiteur. Les champs demandés sont passé en paramètre dans le constructeur des classes Eleve et Professeur.

Une inscription d’un élève entraîne la création de son espace personnel (EspacePerso).

Avant tout accès à notre base de données nous devons appeler la méthode getConnection du SingletonConnection, cela est vrai pour tous les diagrammes de séquences. Si une exception est levée le visiteur reste sur la vue Inscription et le message d’erreur sera afficher. Par soucis de clarté nous n’avons pas encadré les levées d’exceptions, il en est de même pour les diagrammes suivants. Si aucune exception n’est levée le visiteur est redirigé vers la page index2.

Interaction acheter crédit :

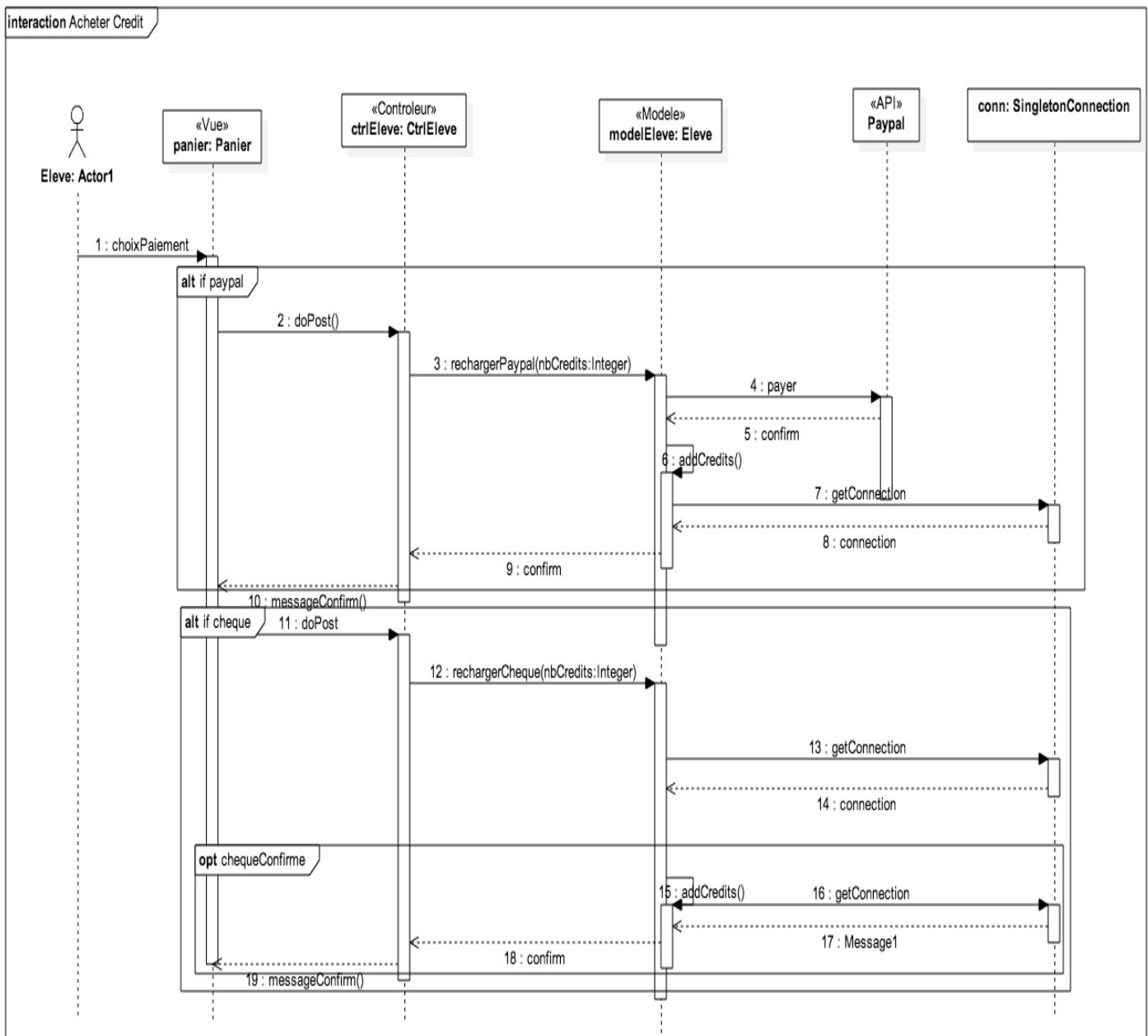


FIGURE 14 – Diagramme de séquence interaction acheter crédit

Un élève se rend sur la page Panier, il choisit le type de paiement (paypal ou chèque), la méthode doPost permet au contrôleur de récupérer le champ représentant le nombre de crédits voulus, s'il s'agit d'un paiement via paypal alors on utilisera la méthode rechargePaypal(nbC credits) qui va communiquer avec l'API PayPal, on supposera que l'API nous renvoie une confirmation, on peut alors ajouter des crédits à l'élève grâce à la méthode addCredits.

S'il s'agit d'un chèque alors on utilisera la méthode rechargeCheque afin d'enregistrer le chèque en base de donnée.

Une fois le chèque confirmé par l'administrateur (grâce à la méthode accepteCheque) les crédits sont ajoutés à l'élève via la méthode addCredits. Dans les deux cas quand le crédit est ajouté on envoie une confirmation à l'élève

Interaction déposer document :

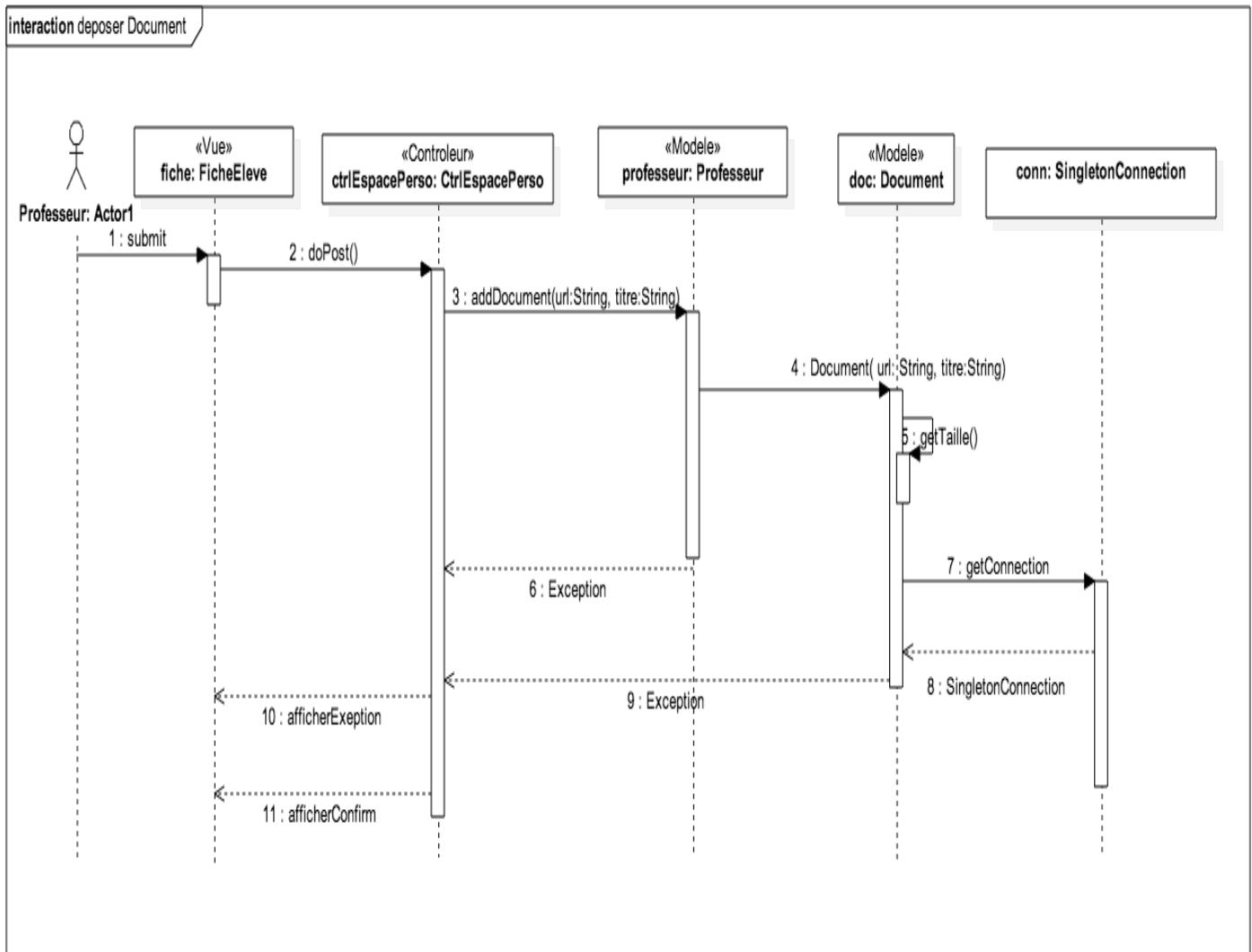


FIGURE 15 – Diagramme de séquence interaction déposer document

Pour déposer un document un professeur doit se rendre sur la fiche d'un élève , il clique sur le bouton déposer un document, il choisit le document à uploader et valide, l'url de l'emplacement du document sur le serveur est transmise au contrôleur via doPost, on utilisera la méthode addDocument de la classe Professeur, qui construira notre document via la classe Document avec les paramètres url et titre , sa taille sera calculer par getTaille() à la construction, une fois le document déposé l'élève reçoit un message de confirmation.

Interaction envoyer message :

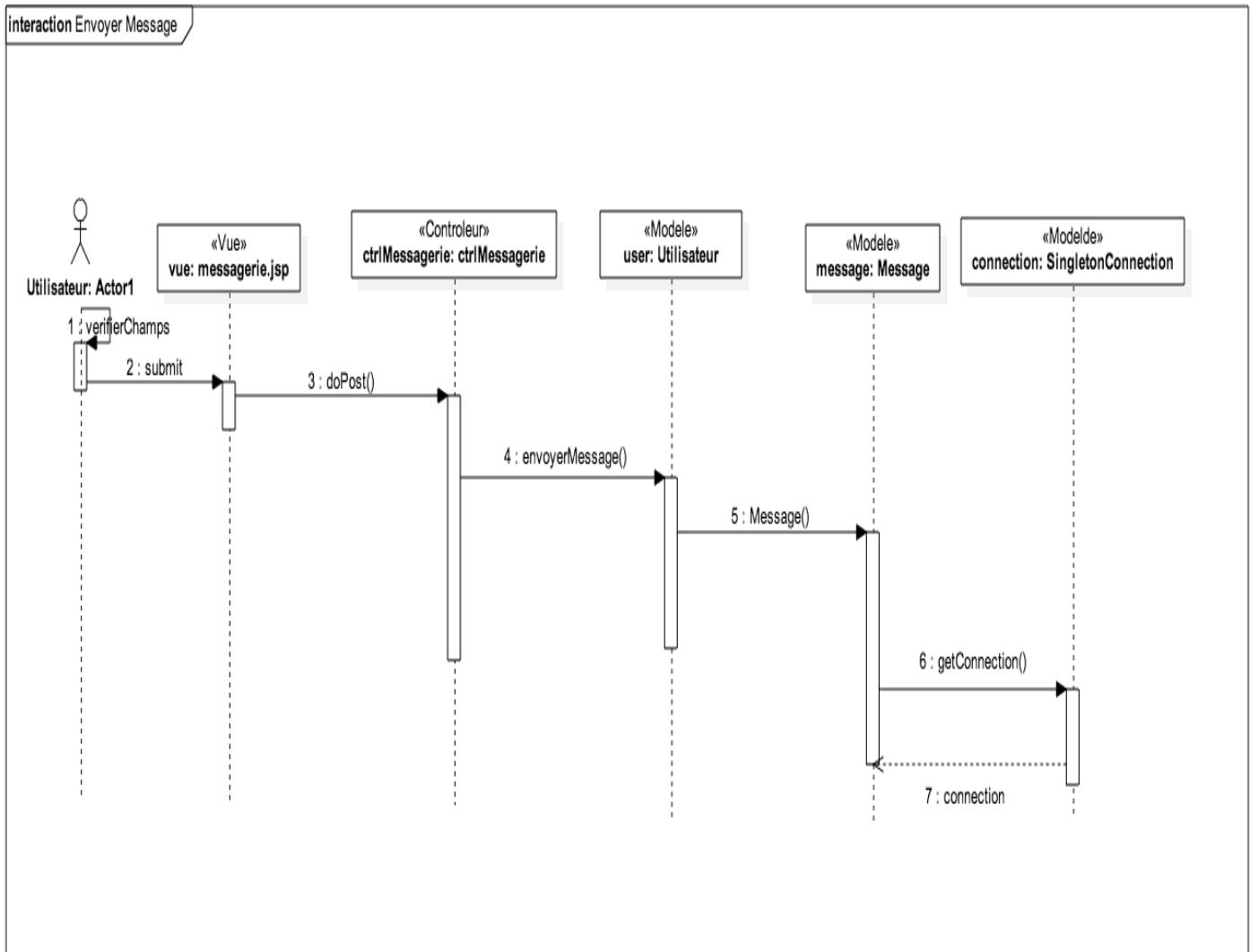


FIGURE 16 – Diagramme de séquence interaction envoyer message

Pour envoyer un message l'utilisateur se rend sur la fiche de l'utilisateur avec lequel il veut communiquer, il clique sur le bouton envoyer un message, on transmet ces variables au contrôleur via la méthode doPost, le contrôleur l'envoi au modèle Utilisateur qui via la méthode envoyerMessage qui prend en paramètre l'email du destinataire, le titre et le message et créer un objet de type Message grâce à son constructeur, par défaut ce constructeur fixe le booleen nouveaux à true, il passera à false une fois le message lu, la date elle sera définie à la construction avec la méthode java adéquat.

Interaction recherche cours :

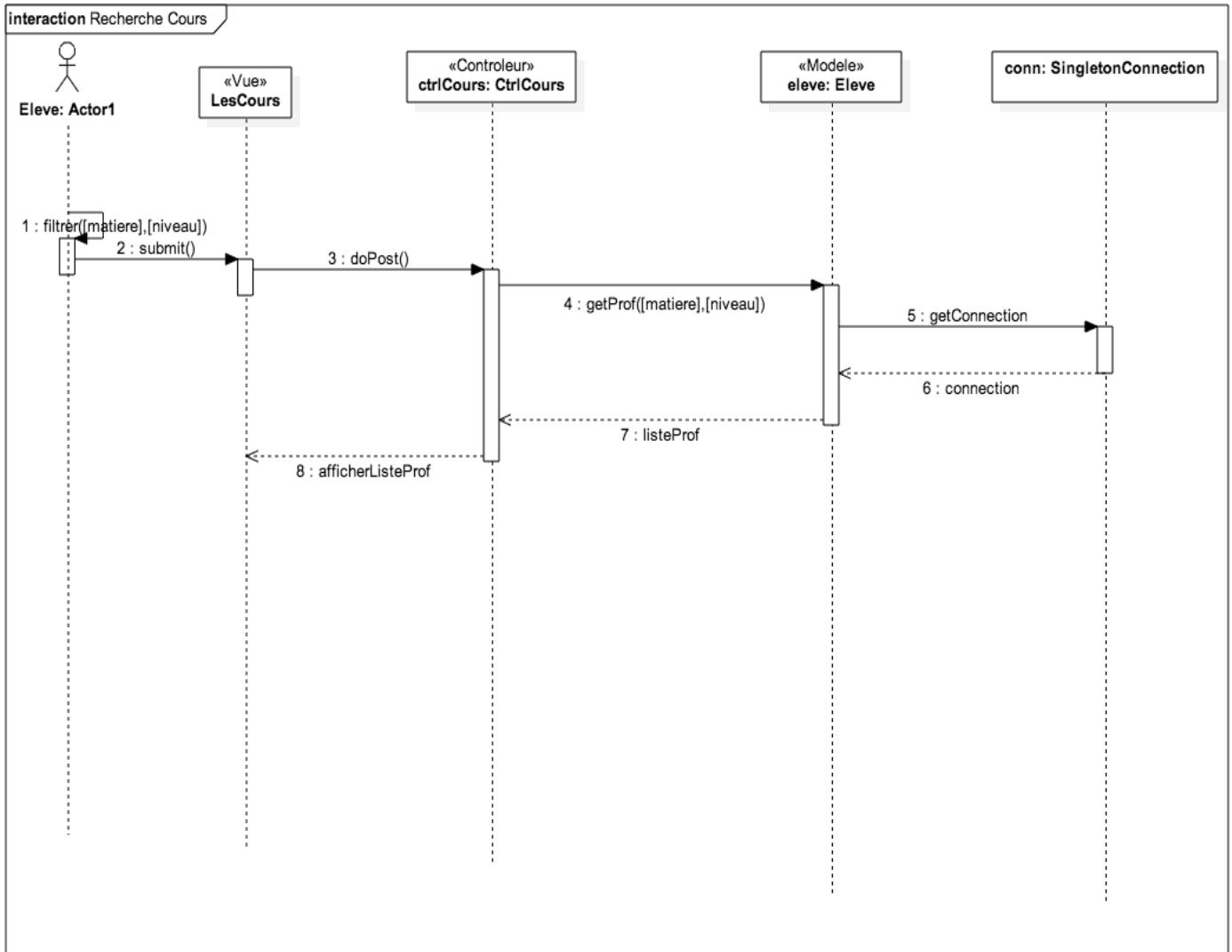


FIGURE 17 – Diagramme de séquence interaction rechercher cours

Pour chercher un cours un élève se rend sur la page LesCours, il peut choisir différents filtre. Nous avons choisis de développer d'abord les filtres indispensable et donc nous n'avons que deux filtres prévus pour le moment, un filtre pour définir la date du cours recherché sera surement mis en place. Les filtres sont transmis au contrôleur via `doPost`, puis au modèle Eleve via la méthode `getProf`, celle si prend deux paramètres optionnelles : matière et niveau.

Il n'est pas possible de définir des paramètres optionnelles en java cependant nous pouvons aisément le simuler, c'est ce que nous ferons. Le modèle nous renverra alors une liste de professeur qui sera affiché sur notre page LesCours. Dans un premier temps nous développerons ça de façon simple puis si le temps nous le permet nous utiliserons AJAX afin de ne pas recharger la page à chaque nouvelle recherche.

Interaction rechercher élève :

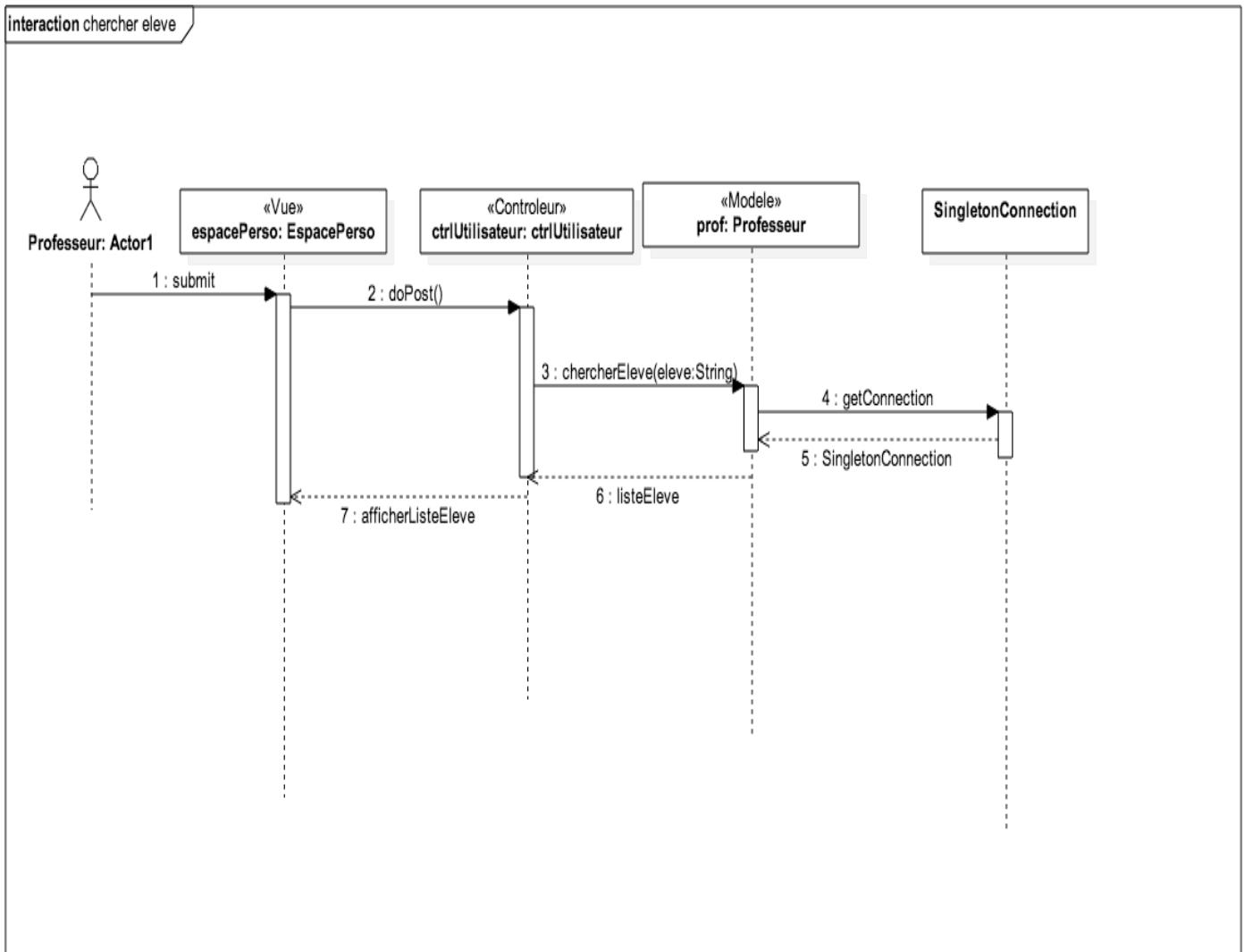


FIGURE 18 – Diagramme de séquence interaction rechercher élève

Pour chercher un élève, un professeur entre sur sa page espacePerso, de là il pourra effectuer une recherche grâce à la barre de recherche , le contrôleur récupère la string de la barre de recherche et la transmet au modèle Professeur qui cherchera en base de donnée grâce à la méthode chercherEleve.

Dans un premier temps nous développerons ceci de façon simple et si nous avons du temps supplémentaires nous effectuerons la recherche en ajax afin que dès que l'on tape une lettre dans la barre de recherche, celle-ci nous propose d'auto-compléter notre recherche

4.2 Diagramme d'états

Les diagrammes d'états montrent les différents états d'un objet ainsi que les transitions entre ces états. Nous avons choisi de vous présenter deux diagrammes d'états, à savoir le diagramme d'états représentant les différents états d'une session utilisateur et celui représentant l'évolution de l'état d'un utilisateur.

Etats d'une session d'utilisateur :

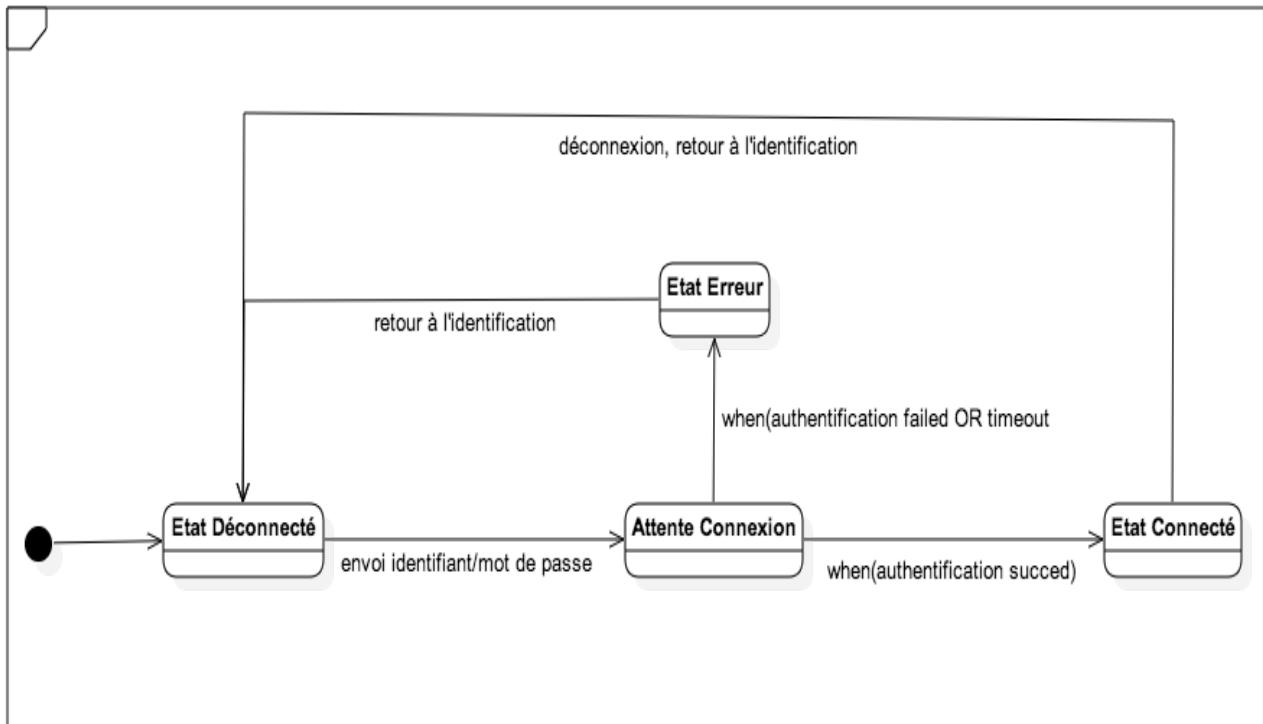


FIGURE 19 – Diagramme d'états session utilisateur

Etats :

- **Déconnecté** : C'est l'état par défaut de la session utilisateur. En réalité, on ne parle pas encore, ici, d'utilisateur mais bien de visiteur étant donné que ce dernier n'est toujours pas identifié.
- **Attente connexion** : Il s'agit de l'état de la session lorsque la combinaison identifiant/mot de passe a été soumise par le visiteur. A cet état, il y a attente d'une réponse du système (temps de recherche et vérification du compte en question dans la base de données).
- **Erreur** : C'est l'état de la session lorsque le système, à l'état précédent, n'a pas trouvé le compte en question ou que la combinaison identifiant/mot de passe fourni par le visiteur n'est pas correcte, ou encore que le temps de traitement d'ouverture de session a été dépassé. (l'état attente connexion dure trop longtemps).
- **Connecté** : Il s'agit de l'état de la session utilisateur (professeur, administrateur ou élève) lorsque le traitement dédié à l'identification s'est déroulé avec succès.

Etats de l'objet utilisateur :

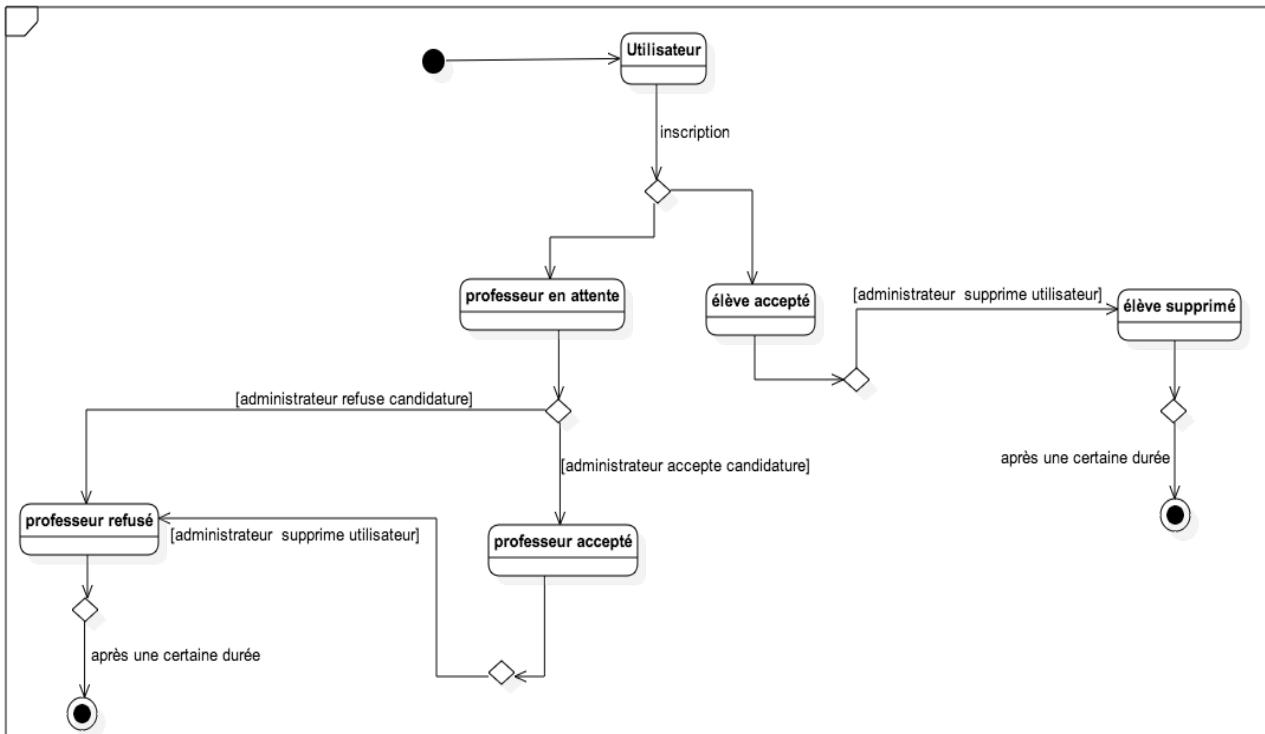


FIGURE 20 – Diagramme d'états de l'objet utilisateur

Etats :

- **Utilisateur** : C'est l'état initial de l'objet utilisateur. La spécification entre élève et professeur, faite lors de l'inscription, n'est pas encore réalisé.
- **Professeur en attente** : Il s'agit de l'état de l'objet utilisateur spécifié professeur lors de l'inscription. A cet état, il y a attente de la décision de l'administrateur (accepter candidature ou non).
- **Professeur accepté** : C'est l'état de l'objet utilisateur après acceptation de candidature par l'administrateur. Cet état de l'objet peut perdurer très longtemps.
- **Professeur refusé** : Il s'agit de l'état de l'objet utilisateur spécifié professeur après refus de la candidature par l'administrateur, ou après suppression par ce dernier du compte utilisateur. Nous avons décidé de garder ces objets après "suppression" par l'administrateur pour pouvoir afficher les motifs de suppression de compte à l'utilisateur concerné (après sa connexion).
- **Elève** : Il s'agit de l'état de l'objet utilisateur spécifié élève à l'inscription. Cet état de l'objet peut perdurer très longtemps.
- **Elève supprimé** : Il s'agit de l'état de l'objet utilisateur spécifié élève après suppression par l'administrateur du compte utilisateur. Au même titre que l'objet professeur supprimé nous avons décidé de garder ces objets après "suppression" par l'administrateur pour pouvoir afficher les motifs de suppression de compte à l'utilisateur concerné (après sa connexion).

5 Eléments ergonomiques

5.1 Diagramme de navigation

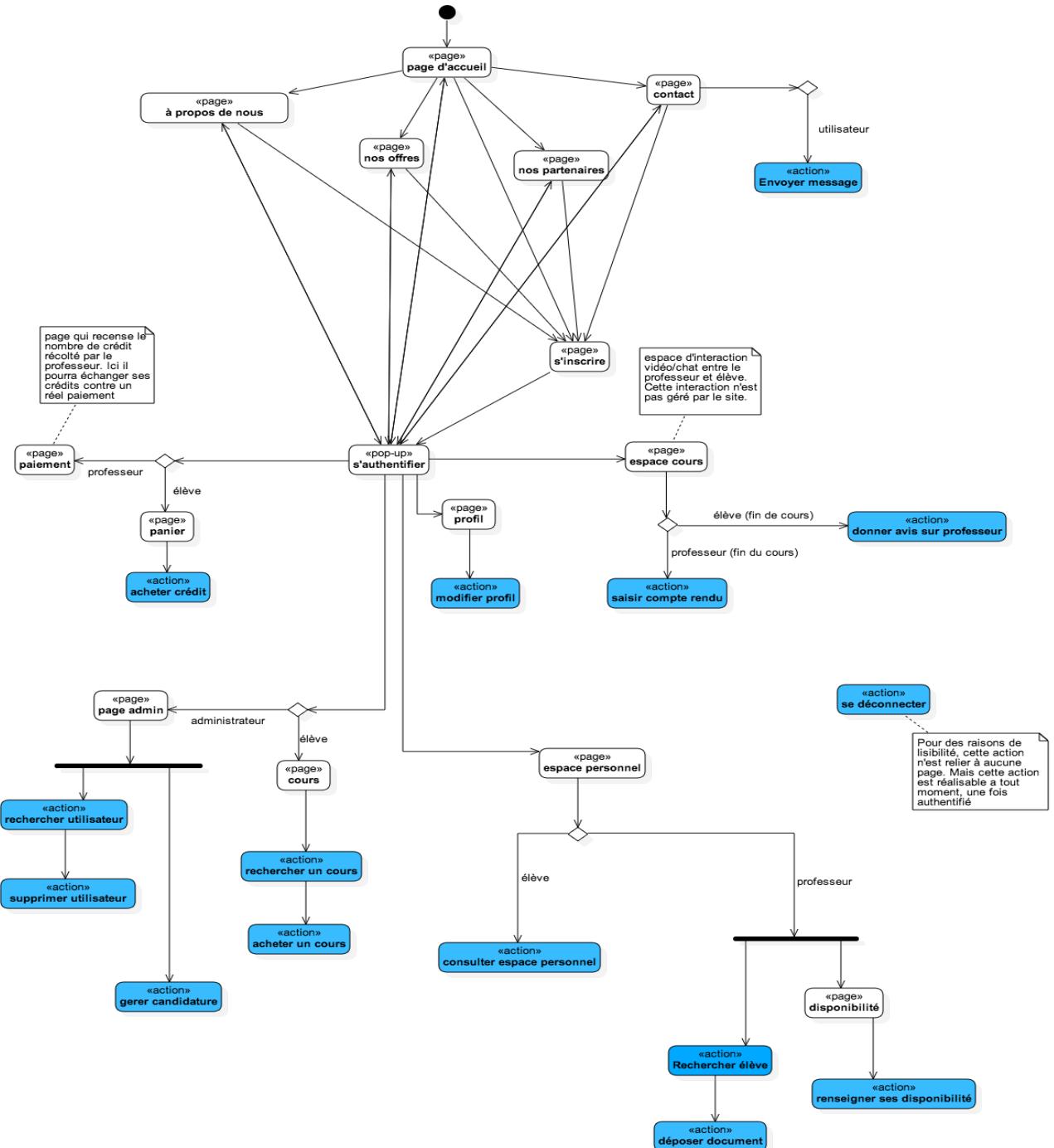


FIGURE 21 – Diagramme de navigation

Dans les standards du langage de modélisation UML, le diagramme de navigation n'existe pas. Nous avons donc utilisé les outils d'un diagramme d'activité UML pour modéliser celui-ci. Ce diagramme présente donc toutes les pages du futur site ainsi que quelques actions (distinguées en bleu) spécifiques à ces pages. Nous avons modélisé les différentes pages en les regroupant par acteur (professeur, élève ou administrateur) en utilisant le symbole de condition. Comme indiqué par des flèches à double sens, les pages accessibles via la barre de menu ("à propos de nous", "nos offres", "contact", "nos partenaires") sont toujours accessibles après identification de l'utilisateur.

5.2 Maquettages d'écrans

Dans cette section, nous détaillerons nos maquettes du projet. Durant cette phase d'analyse, nous avons développé en parallèle une partie de la vue du projet (vue du modèle MVC). La conception de ces vues nous a permis de mieux cibler les actions (des cas d'utilisation) nécessaires ainsi que les classes essentielles au projet. Nous avons choisi cette façon de faire pour ne négliger aucune fonctionnalité. De ce fait, nous pourrons générer, la base de donnée nécessaire, à partir des logiciels avec lesquels nous avons créé les diagrammes UML, sans craindre des modifications de celle-ci par la suite.

Page d'accueil visiteur

The screenshot shows the homepage of the iSchool website. At the top, there is a navigation bar with links for 'S'IDENTIFIER' and 'S'INSCRIRE'. Below the navigation bar is the iSchool logo and the tagline 'LECOLE VIRTUELLE INTELLIGENTE'. The main content area features a large image of a smiling woman wearing a headset, with the text 'Des Professeurs Qualifiés À Votre Disposition' below it. There are three small dots at the bottom left of the image, suggesting a slide show. Below this, the heading 'Nous Vous Ouvrons La Voie Du Succès!' is displayed. A paragraph of text follows, stating: 'iSchool est une plateforme interactive qui met à la disposition d'élève, un large panel de cours virtuels. Que vous soyez collégien ou étudiant en école d'ingénieur, iSchool est fait pour vous! En effet des professeurs de collège, de lycée et universitaire, sélectionnés par nos soins, vous attendent.' Underneath this text are three sections: 'Les Meilleurs Pour Vous' (with an image of three professionals), 'deuxieme zone' (with an image of hands holding a small plant), and 'troisieme zone' (with an image of two people shaking hands). Each section has a 'Plus' button at the bottom. At the very bottom of the page, there is a footer with copyright information: '© 2014 iSchool' and 'Website project by Evry University'.

FIGURE 22 – Vue écran d'accueil

Page d'inscription visiteur (profil professeur)

The screenshot shows the iSchool website's registration form for visitors. At the top, there is a header with the iSchool logo, navigation links for 'S'IDENTIFIER' and 'S'INSCRIRE', and social media sharing options. Below the header, a question asks 'Quelle type d'inscription souhaitez-vous?' with two buttons: 'Professeur' (selected) and 'Elève'. The main form area is titled 'Nom et prénom *' and contains fields for first name and last name. It includes sections for 'Adresse' (street address and secondary line), 'Ville' and 'État/Province', 'Code Postal' and 'pays', and 'Email' with a confirmation field. There are dropdown menus for 'Niveau d'enseignement' (College, Lycée, Université) and 'Spécialité'. A large text area labeled 'Vos Motivations' is present. Below this, there is a file upload field for 'Charger ici votre CV' (with a note that no file has been selected) and a message input field with a placeholder 'Entrer le message suivant *'. At the bottom, there is another dropdown for 'Niveau scolaire' and a 'Valider' button.

FIGURE 23 – Vue écran d'inscription (profil professeur)

Dans cet exemple, le profil professeur est sélectionné. Nous avons dû différencier les deux formulaires d'inscription étant donné que le professeur n'a pas les mêmes attributs que l'élève (voir diagramme de classe - Chapitre 3). Les deux différents processus d'inscription sont détaillés en scénario d'utilisation (Cas d'utilisation - Chapitre 2). Cette page d'inscription sera accessible par le visiteur via le bouton "s'inscrire" qui est en section "HEADER" de la page d'accueil. A tout moment de la navigation du visiteur, ce bouton restera accessible.

Page de connexion

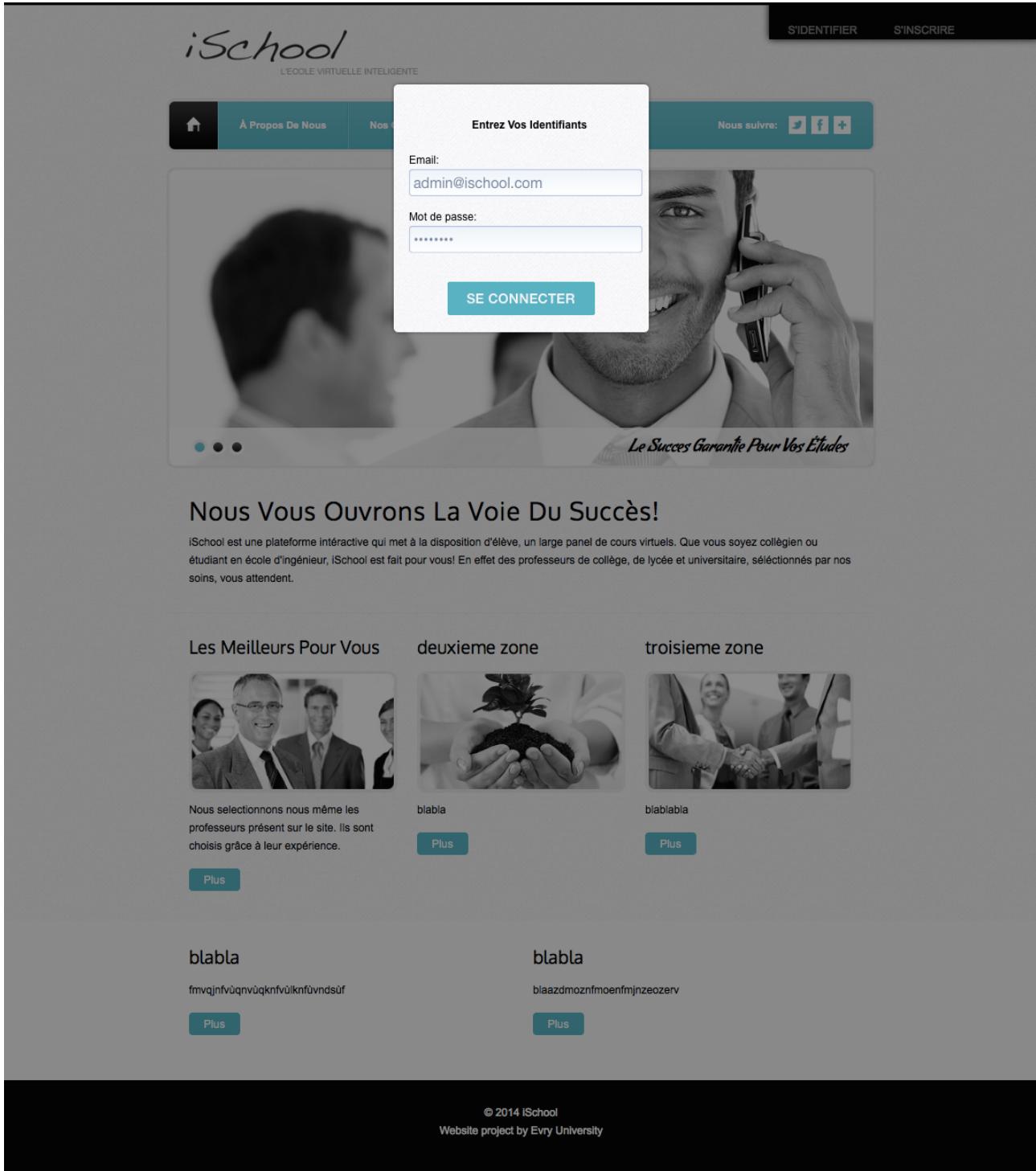


FIGURE 24 – Vue écran connexion

Une fois inscrit, le visiteur pourra se logger avec ses identifiants (adresse mail/mot de passe). Le bouton "se connecter" est, comme le bouton "s'inscrire", accessible en haut à droite de page, à tout moment pour le visiteur.

Page d'accueil utilisateur

The screenshot shows the user homepage of iSchool. At the top, there is a navigation bar with links for "ESPACE PERSONNEL", "SE DÉCONNECTER", and a shopping cart icon. Below the navigation bar is a header with the "iSchool" logo and the tagline "L'ÉCOLE VIRTUELLE INTELLIGENTE". A horizontal menu bar follows, featuring a home icon, "À Propos De Nous", "Nos Offres", "Nos Partenaires", "Contacts", and social media links for Twitter, Facebook, and Google+. The main content area features a large, smiling woman holding a microphone, with the text "Des Cours Personnalisés Selon Vos Besoins" below her. Below this is a section titled "Nous Vous Ouvrons La Voie Du Succès!" with a subtext explaining that iSchool is an interactive platform providing virtual courses for students and professionals. It highlights a selection of professors from various educational levels. The page includes several visual elements: three small images (professors, a plant, and people shaking hands), descriptive text, and "Plus" buttons. At the bottom, there is a black footer bar with copyright information: "© 2014 iSchool" and "Website project by Evry University".

FIGURE 25 – Vue écran d'accueil utilisateur

Une fois connecté, l'utilisateur n'a plus accès aux bouton "s'inscrire" et "se connecter". Ils ont été remplacé par "espace personnel" et "se déconnecter". Nous avons rajouté un panier, pour l'utilisateur élève, pour lui permettre d'acheter des crédits. Nous avons choisi de garder, dans toutes les vues, une barre de menu statique présentant les sections qui doivent être le plus mis en avant (contact, les offres, l'accueil, l'accès au partage sur réseau sociaux, un descriptif du projet, et un icône d'accueil).

Page de contact

The screenshot shows the 'Contact' page of the iSchool website. At the top, there's a navigation bar with links for 'S'IDENTIFIER' and 'S'INSCRIRE'. Below the header, there's a menu bar with 'Accueil', 'À Propos De Nous', 'Nos Offres', 'Nos Partenaires', 'Contacts', and a 'Nous suivre:' section with social media icons for Twitter, Facebook, and Google+. The main content area has a heading 'Nous Contacter'. It includes a map showing the location of iSchool at 'IBISC - IBGBI - 2ème étage, 23, Boulevard de France, 91034 - EVRY'. Below the map are address details: 'IBISC - IBGBI - 2ème étage', '23, Boulevard de France', '91034 - EVRY', 'Telephone: 01 64 98 78 87', 'Fax: 01 64 98 78 88', and 'E-mail: support@ischool.fr'. To the right of the contact information is a large graphic of an open blue envelope containing a white card with the text 'Bonjour iSchool,' and 'MESSAGE:'. At the bottom of the page, a black footer bar contains the copyright notice '© 2014 iSchool' and 'Website project by Evry University'.

FIGURE 26 – Vue écran contact

Voila une vue représentant un des scénario détaillés dans le chapitre 2 (cas d'utilisation - envoyer message). Nous sommes, ici, dans le cas de figure où l'utilisateur décide d'envoyer un message à un administrateur.

Page de contact 2

The screenshot shows the contact page of the iSchool website. At the top right are links for "S'IDENTIFIER" and "S'INSCRIRE". Below the header is a navigation bar with a home icon and links for "À Propos De Nous", "Nos Offres", "Nos Partenaires", "Contacts", and "Nous suivre:" followed by social media icons for Twitter, Facebook, and Google+.

The main content area is titled "Nous Contacter". It features a map of Evry with a red marker indicating the location of IBISC - IBGBI. Below the map is the address: "IBISC - IBGBI - 2ème étage, 23, Boulevard de France, 91034 - EVRY". Contact details include: Telephone: 01 64 98 78 87; Fax: 01 64 98 78 88; E-mail: support@ischool.fr. A "Signé," field and an "EMAIL:" field are also present.

To the right is a large graphic of an open blue envelope containing a white card with the text "Bonjour iSchool, VOTRE MESSAGE:". Below the card is the message "Signé, NOM:" and "EMAIL:". At the bottom of the envelope graphic is the text "Maintenant, j'envoi, merci!".

At the very bottom of the page is a black footer bar with the copyright notice: "© 2014 iSchool" and "Website project by Evry University".

FIGURE 27 – Vue écran contact 2

Comme explicité dans la page précédente, cette rubrique "contact" sera toujours présente pendant la navigation, sur le site, d'un utilisateur.

Maquette de "disponibilité du professeur"

La maquette de la page 'disponibilité du professeur' est présentée dans un cadre avec un header et un footer.

Header:

- Logo 'Ischool' à gauche.
- Boutons 'espace perso' et 'se deconnecter' à droite.
- Menu horizontal avec les options 'Accueil', 'a propos de nous', 'nos offres', 'nos partenaires', 'contacts' et icônes pour Twitter, Facebook et Google+.

Titre central:

Vos disponibilité

Calendrier:

| Calendrier | | | | | | | |
|------------|----|---------|----|----|----|----|----|
| | | Fevrier | | | | | |
| 30 | 31 | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 | 1 |

Horaire 09/02:

| horaire 09/02 |
|---|
| <input checked="" type="checkbox"/> 9h-10h |
| <input type="checkbox"/> 10h-11h |
| <input type="checkbox"/> 11h-12h |
| <input type="checkbox"/> 12h-13h |
| <input type="checkbox"/> 13h-14h |
| <input type="checkbox"/> 14h-15h |
| <input type="checkbox"/> 15h-16h |
| <input checked="" type="checkbox"/> 16h-17h |
| <input type="checkbox"/> 17h-18h |
| <input type="checkbox"/> 18h-19h |

Footer:

footer

FIGURE 28 – Maquette disponibilité professeur

Nous n'avons pas eu encore le temps de finir cette vue en .jsp. Nous vous présenterons donc que la maquette de la page "vos disponibilités" pour l'utilisateur professeur. (voir Chapitre 2 - Cas d'utilisation)

6 Eléments techniques

6.1 Choix des outils et technologies

Langages utilisés



L'intégralité du projet sera développé en Java Entreprise Edition. C'est une technologie de développement totalement inconnue pour nous, mais une documentation et plusieurs cours sur le sujet devraient nous suffire pour le développement du projet.

IDE



Nous avons choisis de programmer, tout les trois, avec le même IDE à savoir Eclipse, celui-ci étant l'IDE gratuit le plus utilisé en entreprise, il permet l'utilisation de nombreux plugins qui peuvent nous être utile.

Système de gestion de base de donnée



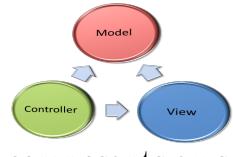
Nous avons choisi d'utiliser MySQL comme base de donnée. Malgré qu'un SGBD comme Oracle serait plus adapté pour interagir avec du J2EE, MySQL est très populaire, donc très documentée, et nous avons, tous les trois, déjà travaillé avec celle-ci.

Outil de gestion des versions



Le code source devant être déposé sur github et ayant des cours de Git à l'université, nous avons décidé de nous servir de ce logiciel libre dans le but de faciliter notre travail d'équipe et nos phases de débogage/test.

Modèle de conception



Le modèle de conception imposé, pour le projet, est le modèle MVC ("Model" "View" "Controller") est un patron de conception destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective. Notre projet de L3 suivait ce modèle précis. Nous n'avons donc pas à consacrer du temps de documentation concernant ce modèle.

Logiciel de conception



Nous avons utilisé StarUML pour tous nos diagrammes de conception (diagramme de séquences, de classes, de navigations, d'états, de cas d'utilisation). Ce logiciel permet de générer du code à partir du diagramme UML, ce qui nous permettra d'obtenir directement le code source de la base de donnée.

6.2 Pattern utilisés

A fin d'appliquer des solutions standards, et pour des raisons plus techniques (détailé plus bas) nous avons choisi d'utiliser trois patrons de conception : l'observateur, pour gérer les mises à jour automatiques entre la base de données et le calendrier personnel du professeur (indiquant ses disponibilités) ; le singleton, pour les connexions, qui répond à la question de comment garantir qu'une classe dispose d'une et une seule instance et fournir un accès unique et global à cette instance ; et le design pattern stratégie pour gérer les différents "pack" de crédit proposés à l'achat pour l'élève. Nous avons abandonné l'idée d'utiliser ce dernier pour simplifier le projet (les pack seront statique). Nous détaillerons leurs utilisations ci-dessous.

Le singleton

Dédié à la création d'instance de classe, le singleton est l'un des rares patrons de conception à ne pas faire usage de l'abstraction pour fournir une solution générique et réutilisable. Ce qui est en fait normal vu son objectif, à savoir assurer l'existence et l'unicité de l'instance d'une classe.

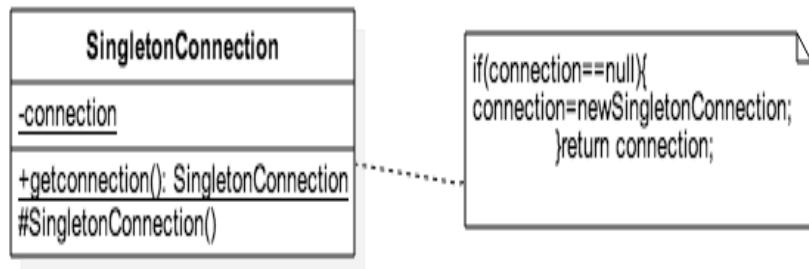


FIGURE 29 – Design pattern singleton

La solution architecturale proposé réside donc dans l'utilisation d'un attribut et d'une méthode de classe dans la classe dont on veut garantir l'existence et l'unicité de l'instanciation. En fait, l'existence sera garantie une fois le premier appel au constructeur effectué via la méthode `getConnection()` :`SingletonConnection`.

L'observateur

Le design pattern observateur permet que les instances cours soit toujours informés des nouvelles disponibilités des professeurs. En effet, nous avons une méthode public `notifier()` dans la classe abstraite "SujetDisponibilité" qui permet, comme son nom l'indique de notifier à l'observateur un changement d'état (ajout ou suppression de disponibilité d'un professeur). Par ce procédé, les cours proposés sur notre plateforme seront, en temps réel attitrés à un professeur disponible.

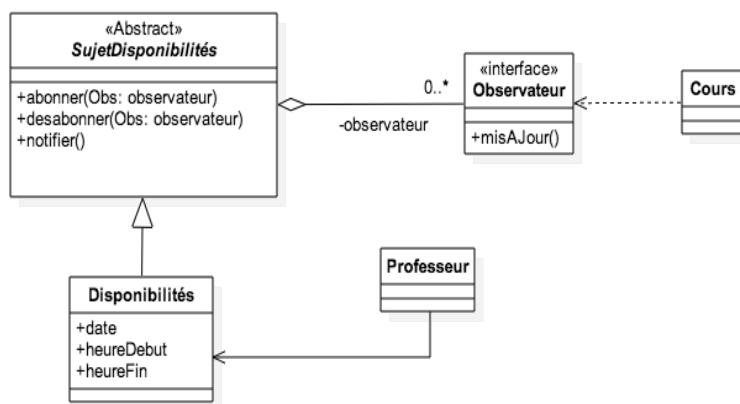


FIGURE 30 – Design pattern observateur

Conclusion

Notre rapport de conception répond bien aux attentes stipulées dans le cahier des charges. Nous avons respecté le diagramme de gantt (fourni en annexe Cf - Annexe I - Diagramme de Gantt) ainsi que la répartition des tâches fixées lors de cette phase de conception du projet (Cf - Annexe III - Affectation des tâches par ressource).

Malgré quelques désaccords, notamment sur le diagramme de classe, le déroulement de cette phase d'analyse s'est passé sans grandes difficultés. De plus, une bonne ambiance de travail réside dans notre groupe ce qui apporte une motivation supplémentaire. La communication était véritablement omniprésente durant cette phase de conception notamment par le biais de conférences, travaux en groupe à l'université ainsi que chez autrui. Cette communication nous a permis de nous conseiller, guider et à anticiper les problèmes de conception.

Cette phase d'analyse nous a permis de mettre en application un grand nombre de cours de notre cursus universitaire, notamment les cours de modélisation objet (pour tous les diagrammes UML) ou encore ceux de génie logiciel (pour les patrons de conception)

Nous avions pour habitude, lors du développement d'ancien projet web d'utilisé un ou plusieurs framework. Mais, ayant peu de connaissance en J2EE, nous avons décidé de nous en passer.

Cette approche théorique, nous a permis de bien déterminer l'aspect architectural du projet : la partie "Eléments Techniques" (partie 6) recense l'intégralité des outils que nous allons ou avons utilisé pour le projet. La partie concernant les cas d'utilisation (partie 2) nous a aidé à déterminer les besoins de chaque acteur. Les parties 3 et 4, elles, nous ont permis de structurer précisément l'architecture du projet alors que la partie "Elements Ergonomiques" détaille l'aspect visuel du projet.

C'est la complémentarité entre tous ces différents diagrammes qui nous permettent d'avoir un aspect global de la partie de développement qui suit.

Annexe

Annexe I - Diagramme de Gantt

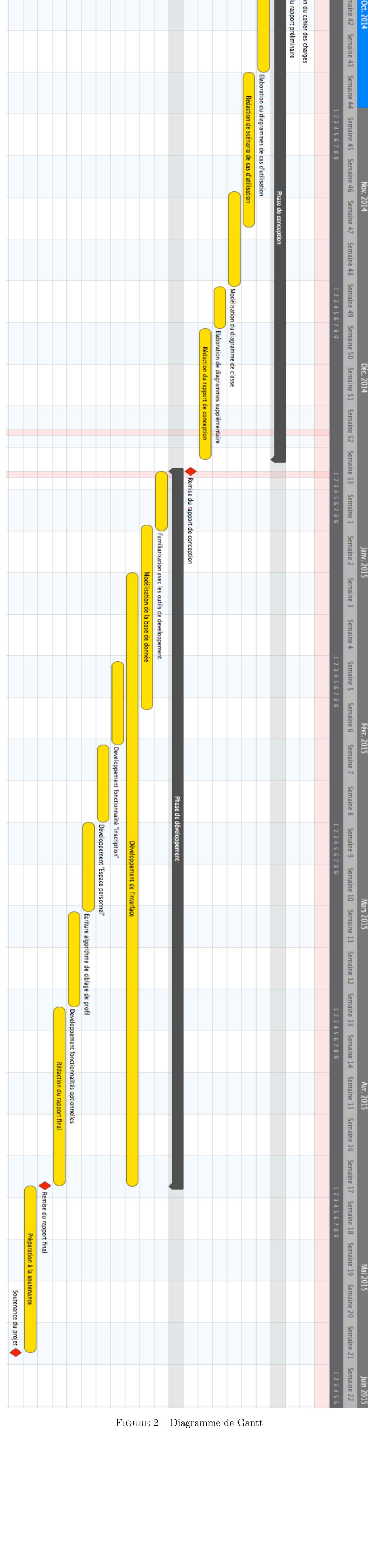
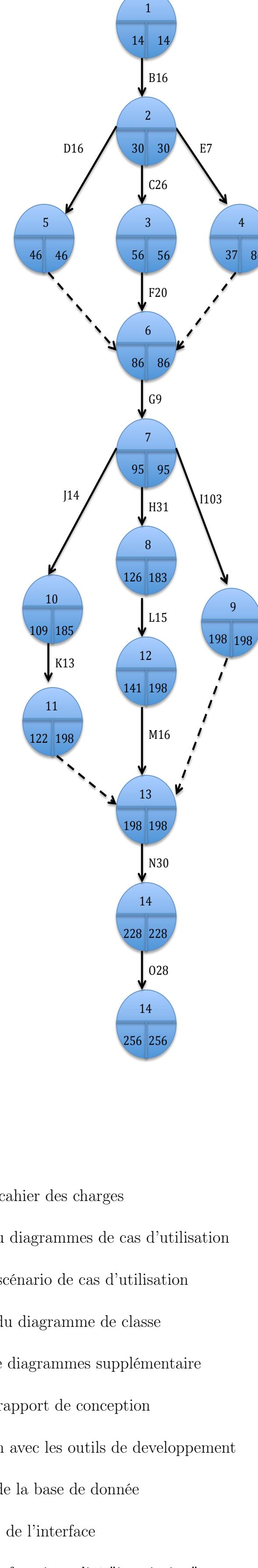


FIGURE 2 – Diagramme de Gantt

Annexe II - Diagramme de Pert



Légende :

- A) Rédaction du cahier des charges
- B) Elaboration du diagrammes de cas d'utilisation
- C) Rédaction de scénario de cas d'utilisation
- D) Modélisation du diagramme de classe
- E) Elaboration de diagrammes supplémentaire
- F) Rédaction du rapport de conception
- G) Familiarisation avec les outils de développement
- H) Modélisation de la base de donnée
- I) Développement de l'interface
- J) Développement fonctionnalité "inscription"
- K) Développement "Espace personnel"
- L) Ecriture algorithme de ciblage de profil
- M) Développement fonctionnalités optionnelles
- N) Rédaction du rapport final
- O) Préparation à la soutenance

