

Marvin ALVES DE JESUS – Fabien LEHOUEZEC – Yann MORTIER

Master 1 MIAGE par Apprentissage – Projet de Conception

2015/2016

UEVE - UPSaclay

MOOKY

Rapport de conception du projet



Ce document comporte le cahier des charges du projet MOOKY. Il inscrit les objectifs et les besoins attendus de l'application ainsi qu'une présentation des fonctionnalités, les spécifications et le détail des développements de celui-ci.

Informations du projet :

Nom du projet :	MOOKY
Groupe :	Marvin ALVES DE JESUS – Fabien LEHOUËZEC – Yann MORTIER
Version du document :	1.8
Statut :	Projet clôturé

Information Groupe :

PERSONNE	RÔLE
Marvin ALVES DE JESUS	Développement des classes et de la base de données distante. Mise en place du Team Foundation Server.
Fabien LEHOUËZEC :	Développement général. Rédaction des documents du projet.
Yann MORTIER :	Développement de l'IHM. Intégration : <ul style="list-style-type: none">- des classes et fonction BDD- de la librairie pour le streaming vidéo.- de la synthèse/reconnaissance vocale.

Révisions du document :

VERSION	NOM	DESCRIPTION
1.0	(all)	Création du document.
1.1	(all)	Ajout des contraintes / Outils et technologies utilisés / Description de l'existant / Maquette.
1.2	(all)	Ajout des diagrammes de cas d'utilisations.
1.3	(all)	Ajout du diagramme de classes et définition de la synthèse et reconnaissance vocale dans l'expression des besoins.
1.4	(all)	Ajout des diagrammes de séquences.
1.5	(all)	Ajout des diagrammes d'états. Ajout de la table des figures. Ajout du diagramme du package
1.6	(all)	Correction sur diagramme de classes Correction sur diagrammes de séquence
1.7	(all)	Ajout d'informations supplémentaires dans la partie Outils et technologies utilisés
1.8	(all)	Chapitre de conception du logiciel Ajout des fonctionnalités du logiciel

Tables des Matières

1	Présentation du projet	6
1.1	Contexte	6
1.2	Description de l'existant.....	6
1.3	Objectifs.....	6
2	Expression des besoins.....	7
2.1	Besoins Fonctionnels	7
2.2	Besoins Non Fonctionnels	9
3	Contraintes	9
3.1	Compatibilité	9
4	Outils et technologies utilisés.....	9
4.1	Environnement de développement et langages	9
4.2	Modèle utilisé	11
5	Spécifications Techniques et Fonctionnelles.....	12
5.1	Diagramme de cas d'utilisations	12
5.2	Présentation du périmètre fonctionnel de l'application.....	12
5.3	Détails des cas d'utilisations généraux.....	13
5.4	Détails des cas d'utilisations pour l'administrateur	14
5.5	Détails des cas d'utilisations pour l'étudiant	16
5.6	Détails des cas d'utilisations pour le professeur	19
6	Diagramme de classes	23
7	Diagrammes de séquence	27
8	Diagrammes d'états	32
9	Diagrammes de packages.....	35
10	Maquettes de départ.....	36
11	Gestion de la Base de données	38
11.1	Le cloud avec Microsoft Azure	38
11.2	EntityFramework et le modèle « Business Layer »	39
12	Fonctionnalités finales de MOOKY	39
12.1	Connexion.....	40
12.2	Espace Personnel.....	40

12.3	Gestion du profil.....	42
12.4	Accès à la session (étudiant)	43
12.5	Accès à la session (professeur).....	43
12.6	Mooky Replay.....	44
12.7	Mes Résultats	44
12.8	Gestion des sessions (professeur).....	45
12.9	Gestion des comptes MookyID (professeur administrateur).....	46
12.10	Paramètres	47
13	Table des figures.....	49

Description du besoin

1 Présentation du projet

1.1 Contexte

Dans le cadre de la formation Master 1 MIAGE Apprentissage, nous avons pour projet de réaliser une application. Le sujet choisi est un sujet libre validé par Mme BENZAKKI. Notre application Mooky s'inscrit dans le cadre de l'e-learning. Ce projet se décompose en trois phases.

Cette présentation du sujet décrit le contexte, les objectifs et les besoins de l'application (à rendre pour le 24 novembre 2015). Un rapport intermédiaire détaillant la modélisation et la spécification du système (à rendre pour le 10 mars 2015) puis une présentation finale de 50 minutes afin d'exposer le travail réalisé (le 23 juin 2016)

Ce document comporte la présentation du projet MOOKY. Il inscrit les objectifs et les besoins attendus de l'application ainsi qu'une présentation des fonctionnalités de celui-ci.

1.2 Description de l'existant

Aujourd'hui, il existe une plateforme en ligne pour suivre le cursus universitaire de la MIAGE nommé e-Miage. Nous retrouvons les cours écrits des modules enseignés ainsi que des exercices.

Les enseignants possèdent des statistiques mais propre à chaque université. Ils ont quelques informations sur le plan national (nombre d'inscrits, nombre d'étudiants inscrits à l'examen, nombre d'étudiants ayant réussi l'examen, ...). Sur le plan local, chaque université peut suivre les personnes ayant suivi les cours ou encore consulter les notes d'examens.

Ce système manque d'interactivité notamment dans l'échange entre les professeurs et les étudiants. Cela ne permet pas à l'étudiant d'interroger ce dernier pour obtenir plus d'informations sur un point du cours.

1.3 Objectifs

Mooky est une solution logicielle développée en technologie C# qui permettra d'accéder à distance à des cours sous forme numérique. Il s'inscrit dans la vision du MOOC (Formation en Ligne Ouverte à Tous).

Il s'agit :

- De cours diffusés en ligne
- De niveau universitaire
- Gratuit et libre d'accès

La structure d'un MOOC est caractérisée par 3 composantes :

- La vidéo
- Les exercices
- L'interaction entre le professeur et ses étudiants

La durée d'un MOOC sera ici d'environ 3h15. Nous reproduirons le même temps de travail qu'à l'Université.

Cette application a pour but de supprimer les frontières physiques d'un cours traditionnel. Pour y parvenir le logiciel devra recréer l'ambiance et les conditions d'un cours à l'Université dont le point clé sera l'échange entre l'enseignant et les élèves.

Le logiciel intégrera une technologie de streaming vidéo pour que l'enseignant diffuse à partir d'une webcam le cours depuis n'importe quel lieu où il se trouve. Les élèves pourront quant à eux communiquer avec l'enseignant à l'aide d'un salon de discussion.

2 Expression des besoins

2.1 Besoins Fonctionnels

Nous allons aborder les besoins qui caractérisent le projet. Nous les avons élaborés en fonction des deux parties présentes lors d'un cours : l'étudiant et l'enseignant.

Pour pouvoir définir ces besoins nous nous sommes penchés, dans un premier temps, sur nos propres attentes d'un cours en tant qu'étudiant, et dans un second temps nous avons cherché à satisfaire les espérances attendues vis-à-vis des enseignants.

Nous définirons les besoins dans le cas d'un étudiant, et dans celui de l'enseignant.

➤ Cas d'un étudiant :

Il devra être possible d'interagir pendant le cours en ligne, comme pour un cours traditionnel. C'est-à-dire avoir la possibilité de questionner et de communiquer en temps réel avec l'enseignant, mais aussi avec les autres étudiants qui participent au cours.

L'étudiant s'exercera à l'aide de plusieurs questions qui lui seront posées afin qu'il évalue sa compréhension du cours.

Enfin, il devra avoir un espace personnel pour pouvoir accéder :

- À l'ensemble des cours auxquels il a assisté.
- Aux résultats obtenus.
- A ces informations personnelles

➤ **Cas d'un enseignant :**

Le principal besoin caractérisé par le projet est de proposer des cours en ligne dans le but de donner un accès à des personnes étant situées dans des endroits différents.

Pour cela, il sera possible de créer un cours caractérisé par :

- Une durée
- Une matière
- Les notions du cours à valider
- Les points abordés
- La date et l'heure de la session

Il devra être possible d'interagir avec les étudiants pendant le cours afin de recevoir leurs questionnements et leurs remarques dans le cas d'incompréhensions sur des notions abordées.

L'enseignant pourra à tout moment durant le cours, et s'il le souhaite, évaluer à l'aide d'un questionnaire le niveau de compréhension du cours par les étudiants.

Enfin, il devra avoir accès à un espace personnel avec un suivi des cours donnés, de la moyenne obtenue des élèves pour un cours, des notions qui ont été évaluées. Il pourra aussi depuis cet espace, créer les sessions de cours, les modifier ou les supprimer.

➤ **Autres comptes :**

Le compte Administrateur permettra de gérer les enregistrements de la base de données à l'aide d'une interface qui lui sera dédié. Il pourra ainsi créer, supprimer ou modifier des enregistrements.

Il sera notamment chargé de créer les Mooky ID (identifiant) pour les utilisateurs.

2.2 Besoins Non Fonctionnels

L'application Mooky sera développée pour le système d'exploitation Microsoft Windows. Elle se présente sous forme d'un exécutable pour Windows (.EXE).

La principale contrainte sera la connexion à internet. En effet, toutes les données seront stockées sur un serveur distant (NAS).

3 Contraintes

3.1 Compatibilité

Mooky sera compatible avec Microsoft Windows, pour les versions suivantes :

- Windows 10
- Windows 8/8.1
- Windows 7

4 Outils et technologies utilisés

4.1 Environnement de développement et langages

Notre projet utilise le langage de programmation C# pour le développement du logiciel. L'écosystème créé autour du C# par Microsoft implique une évolution garantie grâce aux nombreuses mises à jour du langage. Celui-ci peut nous permettre de développer des applications multiplateformes (grâce à Mono sur Mac et Linux).

Le C# utilise le framework .NET (dotnet) qui permet de faciliter le développement grâce à des fonctionnalités prête à l'usage comme les expressions lambda ou les méthodes asynchrones.

Le langage SQL est utilisé pour exploiter des bases de données relationnelles. Il permet d'ajouter, modifier, rechercher ou supprimer des données dans une base de données.

La suite de logiciels de développement utilisée est Microsoft Visual Studio 2015. Microsoft fournit une version gratuite complète nommée Community. C'est l'environnement de développement principal pour le développement C#.

Dans cette suite nous utilisons Team Foundation Server (TFS) et principalement sa fonction de gestion de builds. Ce logiciel permet de sauvegarder le code source de Mooky et la chronologie de chaque fichier.

Pour mettre en collaboration notre travail, nous nous sommes appuyé sur l'outil Microsoft Team Foundation Server disponible grâce à l'environnement C#. Cela nous permet de mettre en relation notre travail ainsi que de maintenir plus facilement l'application grâce à un système de suivi des modifications.

Afin d'améliorer notre capacité de développement, nous nous sommes appuyé sur le plugin ReSharper Ultimate, conçu pour permettre l'augmentation de la productivité des développeurs de logiciels. Cet outil nous permet notamment une analyse statique du code de manière à repérer des erreurs formelles de programmation ou de conception, mais aussi pour déterminer la facilité ou la difficulté à maintenir le code. De plus, il aide à améliorer la lisibilité du code et ainsi qu'à sa maintenance grâce à un système de correcteur syntaxique de code et de complétion automatique.

Le système de mapping objet-relationnel Entity framework nous permet de disposer des données relationnelles sous la forme de propriétés et d'objets spécifiques aux domaines. Ainsi, nous pouvons réaliser toutes les opérations de CRUD (création, lecture, mise à jour, suppression) tout en éliminant la duplication de code pour ce type d'opération.

Le système de gestion de bases de données relationnelles (SGBDR) MySQL est utilisé pour la mise en place d'une base de données SQL. Cette base contiendra toutes les informations créées ou demandées par l'application Mooky.

MySQL est un des SGBDR les plus utilisés et est conçu pour les systèmes ayant besoin de performance. De plus, il présente les avantages d'être gratuit et open-source.

Cette base de données est hébergée sur un serveur que nous possédons pour en assurer la sécurité, la pérennité et permettre d'utiliser l'application Mooky en toute circonstance.

La plate-forme de cloud computing Microsoft Azure est utilisée pour le stockage des données et la mise à disposition de la base de données.

Le client FTP Filezilla Server, utilisé pour le transfert de donnée d'un ordinateur distant vers un ordinateur local et inversement, nous donne la possibilité de chiffrer les échanges et de compresser les données stockées. Dans notre application, il sert notamment de stockage et de mise à disposition des fichiers vidéos, audio et de toutes autres documents relatifs au cours proposé.

De plus, pour pouvoir compresser nos médias numériques, aussi bien auditifs que visuels, nous nous appuyons sur le programme Microsoft Expression Encoder.

4.2 Modèle utilité

Mooky est développé avec le modèle MVC (Modèle-Vue-Contrôleur). Ce patron de conception permet une programmation plus rapide et efficace grâce à une séparation du code en trois parties :

- Le modèle : Il contient les informations
- La vue : Il correspond à l'interface utilisateur et toute sa gestion
- Le contrôleur : Il fait le lien entre les informations et l'interface

5 Spécifications Techniques et Fonctionnelles

5.1 Diagramme de cas d'utilisations

L'élaboration des différents diagrammes de cas d'utilisation nous ont permis de construire la base de notre projet. Grâce à cela nous avons établi un aperçu synthétique de notre projet. Dans cette partie vous trouverez l'ensemble de nos cas d'utilisation.

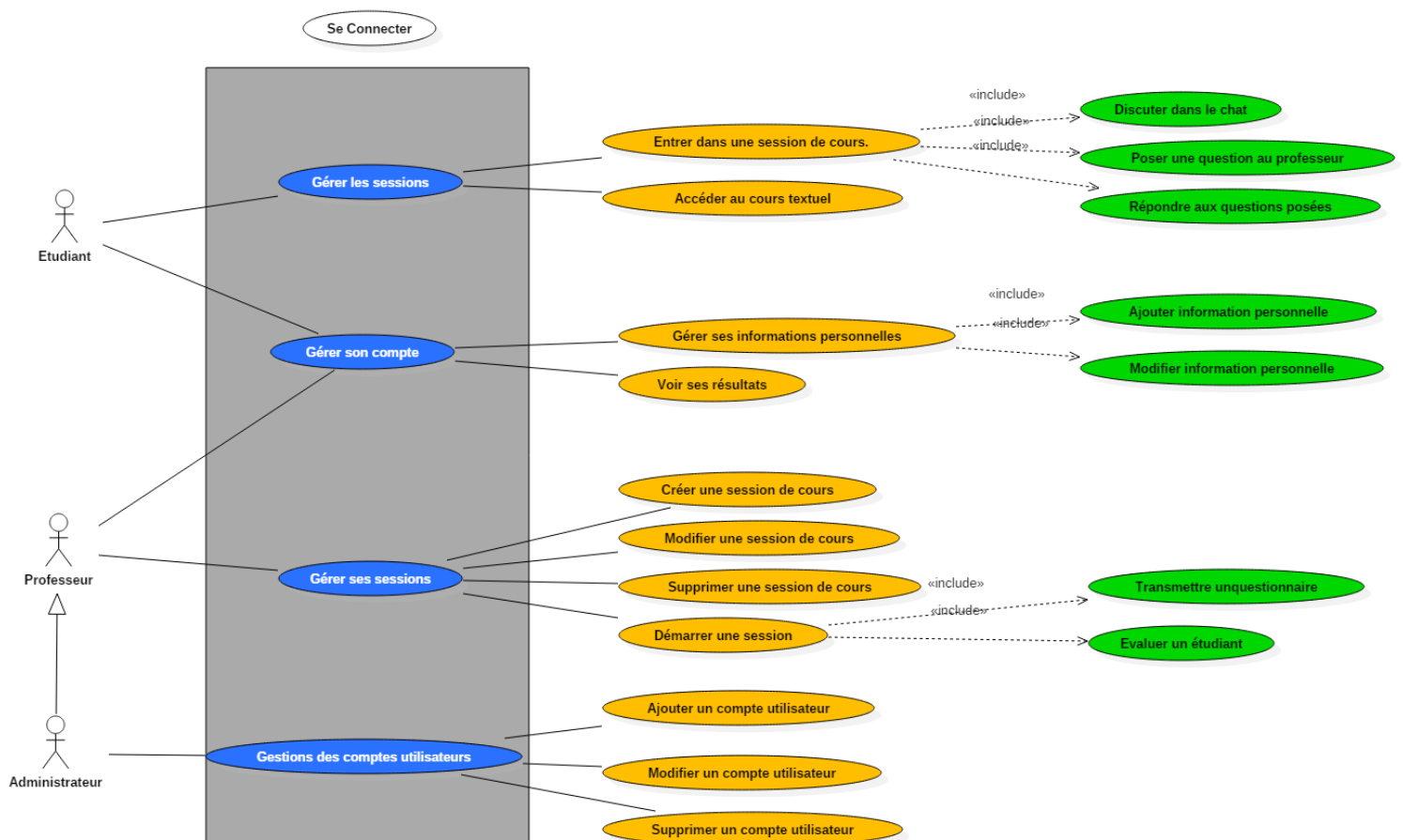


Figure 1: Diagramme de cas d'utilisations

5.2 Présentation du périmètre fonctionnel de l'application

Il y aura 3 cas d'utilisations au sein de l'application : l'étudiant, le professeur et l'administrateur :

Cas étudiant :

- Gestion de son compte : l'étudiant consultera ses informations personnelles. L'étudiant peut accéder aux résultats obtenus lors des sessions de cours.
- Gestion de ses sessions : L'étudiant peut entrer dans une session de cours où il est inscrit. De plus, le cours lui sera accessible de façon textuel s'il le souhaite pour revoir des notions clés.

Cas professeur :

- Gestion de son compte : le professeur pourra gérer son compte au même titre ainsi qu'avec les mêmes fonctionnalités.
- Gestion de ses sessions Le professeur aura la possibilité de créer une session de cours, de la démarrer, de la modifier ou encore de l'annuler.

Cas administrateur :

- Gestion des comptes utilisateurs : l'administrateur sera chargé d'ajouter des comptes utilisateurs, de les modifier ou encore d'en supprimer.

5.3 Détails des cas d'utilisations généraux

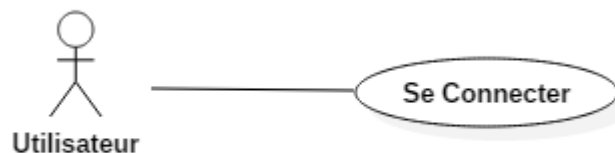


Figure 3 : Connexion à un compte

Intitulé :

Se connecter à son compte

Acteur :

Utilisateur (Etudiant, Professeur ou Administrateur)

Données d'entrée :

- L'utilisateur possède son adresse mail et son mot de passe

Scénario :

1. Le système ouvre sur la fenêtre de connexion
2. L'utilisateur saisit son mail ainsi que le mot de passe associé
3. L'utilisateur clique sur « Connexion »

5.4 Détails des cas d'utilisations pour l'administrateur



Figure 4 : Ajout d'un compte utilisateur

Intitulé :

Ajouter un compte utilisateur

Acteur :

Administrateur

Données d'entrée :

- L'administrateur est connecté dans l'application.
- L'administrateur possède le nom, prénom, adresse mail, mot de passe, classe (si étudiant), type de l'utilisateur.

Scénario :

1. Le système affiche la fenêtre de création d'un utilisateur
2. L'administrateur inscrit le nom, le prénom, l'adresse mail, le mot de passe, le type de l'utilisateur et la classe (si étudiant)
3. L'administrateur clique sur « Ajouter utilisateur »



Figure 5 : Modification d'un compte utilisateur

Intitulé :

Modification d'un compteur utilisateur

Acteur :

Administrateur

Donnée d'entrée :

- L'administrateur est connecté dans l'application

Scénario :

1. Le système affiche la liste des utilisateurs
2. L'administrateur choisit l'utilisateur qu'il souhaite modifier
3. L'administrateur clique sur « Modifier utilisateur »
4. Le système affiche la fenêtre de modification d'utilisateur avec les informations de l'utilisateur sélectionné préinscrites
5. L'administrateur modifie les informations erronées
6. L'administrateur clique sur le bouton « Modifier utilisateur »
7. Le système demande confirmation à l'administrateur
8. L'administrateur clique sur le bouton « Valider »



Figure 5 : Suppression d'un compte utilisateur

Nom :

Supprimer un compte utilisateur

Acteur :

Administrateur

Données d'entrée :

- L'administrateur est connecté dans l'application.
- L'administrateur choisit un utilisateur parmi une liste d'utilisateur.

Scénario :

1. Le système affiche la liste des utilisateurs
2. L'administrateur choisit l'utilisateur qu'il souhaite supprimer
3. L'administrateur clique sur « Supprimer utilisateur »
4. Le système demande la confirmation à l'administrateur
5. L'administrateur clique sur le bouton « Valider »

5.5 Détails des cas d'utilisations pour l'étudiant

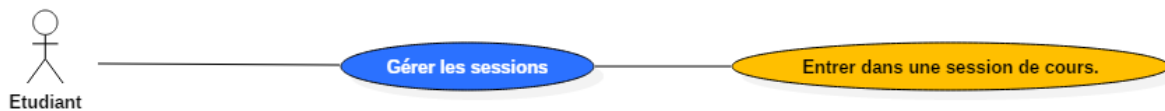


Figure 6 : Entrer dans une session de cours

Nom :

Entrer dans une session de cours

Acteur :

Etudiant

Données d'entrée :

- L'étudiant est connecté à l'application.

Scénario :

1. L'étudiant clique sur « Gérer les sessions »
2. Il clique ensuite sur « Entrer dans une session de cours »
3. Il peut choisir parmi la liste des sessions disponibles
4. L'étudiant est maintenant connecté à la session

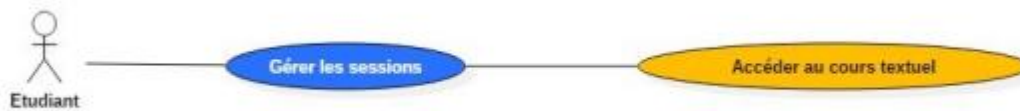


Figure 7 : Accéder aux cours textuels

Intitulé :

Accéder au cours textuel

Acteur :

Etudiant

Données d'entrée :

- L'étudiant est connecté dans l'application

Scénario :

1. L'étudiant clique sur « Accéder aux cours textuels »
2. L'étudiant sélectionne un cours textuel et clique sur « Ouvrir »
3. Le système ouvre le fichier correspondant au cours textuel sélectionné



Figure 8 : Voir ses résultats

Intitulé :

Voir ses résultats

Acteur :

Etudiant

Données d'entrée :

- L'étudiant est connecté dans l'application

Scénario :

1. L'étudiant clique sur « Voir les sessions passées »
2. Le système ouvre la fenêtre des sessions passées
3. L'étudiant voit le résultat de chaque session, il clique sur « Détails » de la session sélectionnée
4. Le système ouvre la fenêtre de la session sélectionnée, le résultat de chaque question s'affiche

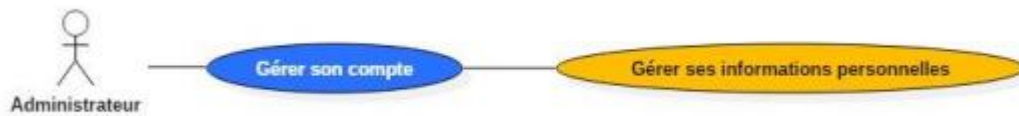


Figure 9 : Gérer ses informations personnelles

Intitulé :

Gérer ses informations personnelles

Acteur :

Utilisateur

Données d'entrée :

- L'utilisateur est connecté dans l'application

Scénario :

1. Le système affiche la fenêtre de gestion des informations personnelles
2. L'utilisateur clique sur le bouton « Modifier » de l'information qu'il veut modifier
3. Le système demande la confirmation
4. L'utilisateur clique sur le bouton « Valider »

5.6 Détails des cas d'utilisations pour le professeur

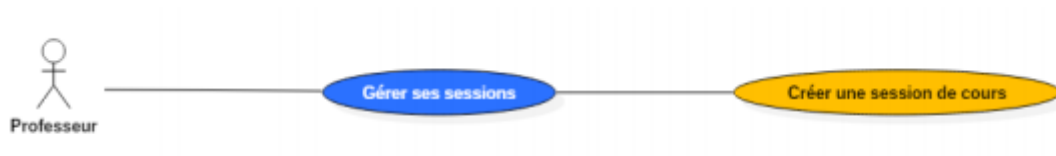


Figure 10 : Créer une session de cours

Intitulé :

Créer une session de cours

Acteur :

Professeur

Données d'entrée :

- Le professeur est connecté à l'application
- Le professeur est dans la partie « gestion de session »

Scénario principal :

1. Le professeur clique sur « Créer une session de cours »
2. Le système demande au professeur de remplir un formulaire de création de session de cours
3. Le professeur clique sur « Valider »
4. Le système lui affiche un récapitulatif des informations renseignées par le professeur
5. Le professeur clique sur « Accepter »
6. Le système informe le professeur que la création de la session de cours a bien été effectuée.

Scénario d'erreur :

- 3a. Un (plusieurs) champs est (sont) mal renseigné(s), le système affiche un message d'erreur et le (les) champs concerné(s) est (sont) marqué(s) en rouge.

Scénario alternatif :

3b. Le professeur clique sur « Retour »

5a. Le professeur clique sur « Retour »



Figure 11 : Modifier une session de cours

Intitulé :

Modifier une session de cours

Acteur :

Professeur

Données d'entrée :

- Le professeur est connecté à l'application
- Le professeur est dans la partie « gestion de session »

Scénario :

1. Le professeur clique sur « Modifier une session de cours »
2. Le système affiche la liste des sessions de cours
3. Le professeur clique sur la session de cours qu'il souhaite modifier
4. Le système affiche le formulaire de la session de cours choisi avec les informations correspondantes
5. Le professeur modifie les informations présentes dans le formulaire
6. Le professeur clique sur « Valider »
7. Le système lui affiche un récapitulatif des informations renseignées par le professeur
8. Le professeur clique sur « Accepter »
9. Le système informe le professeur que la modification de la session de cours a bien été effectuée.

Scénario d'erreur :

- 6a. Un (plusieurs) champs est (sont) mal renseigné(s), le système affiche un message d'erreur et le (les) champs concerné(s) est (sont) marqué(s) en rouge. Retour à l'étape 6.

Scénario alternatif :

3a. Le professeur clique sur « Retour ».

5a. Le professeur clique sur « Retour ». Retour à l'étape 2.

8a. Le professeur clique sur « Retour ». Retour à l'étape 4.



Figure 12 : Supprimer une session en cours

Intitulé :

Supprimer une session de cours

Acteur :

Professeur

Données d'entrée :

- Le professeur est connecté à l'application
- Le professeur est dans la partie « gestion de session »

Scénario :

1. Le professeur clique sur « Supprimer une session de cours »
2. Le système affiche la liste des sessions en cours
3. Le professeur clique sur la session de cours qu'il souhaite supprimer
4. Le système lui affiche un message de confirmation
5. Le professeur clique sur « Accepter »
6. Le système informe le professeur que la suppression de la session de cours a bien été effectuée.

Scénario alternatif :

5a. Le professeur clique sur « Refuser ». Retour à l'étape 2.

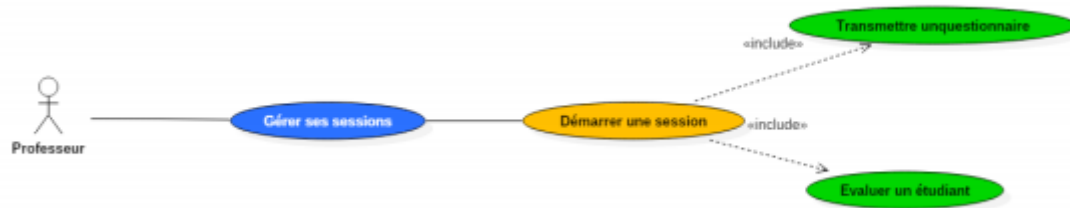


Figure 13 : Démarrer une session de cours

Intitulé :

Démarrer une session de cours

Acteur :

Professeur

Données d'entrée :

- Le professeur est connecté à l'application
- Le professeur est dans la partie « gestion de session »

Scénario :

1. Le professeur clique sur « Démarrer une session de cours »
2. Le système affiche les sessions de cours
3. Le professeur clique sur la session de cours qu'il souhaite démarrer
4. Le système lui affiche la fenêtre de cours
5. Le professeur fait son cours
6. Le professeur clique sur « Transmettre un questionnaire »
7. Les étudiants travaillent sur le questionnaire sur un temps limité
8. Le système stoppe la session de travail
9. Le professeur continue son cours
10. Le professeur transmet oralement les bonnes réponses au questionnaire.
11. Le professeur termine son cours
12. Le professeur clos la session de cours

13. Le système lui retourne les questionnaires remplis par les étudiants
14. Le professeur évalue les étudiants

6 Diagramme de classes

La solution Mooky contient 4 projets dont voici les diagrammes de classes :

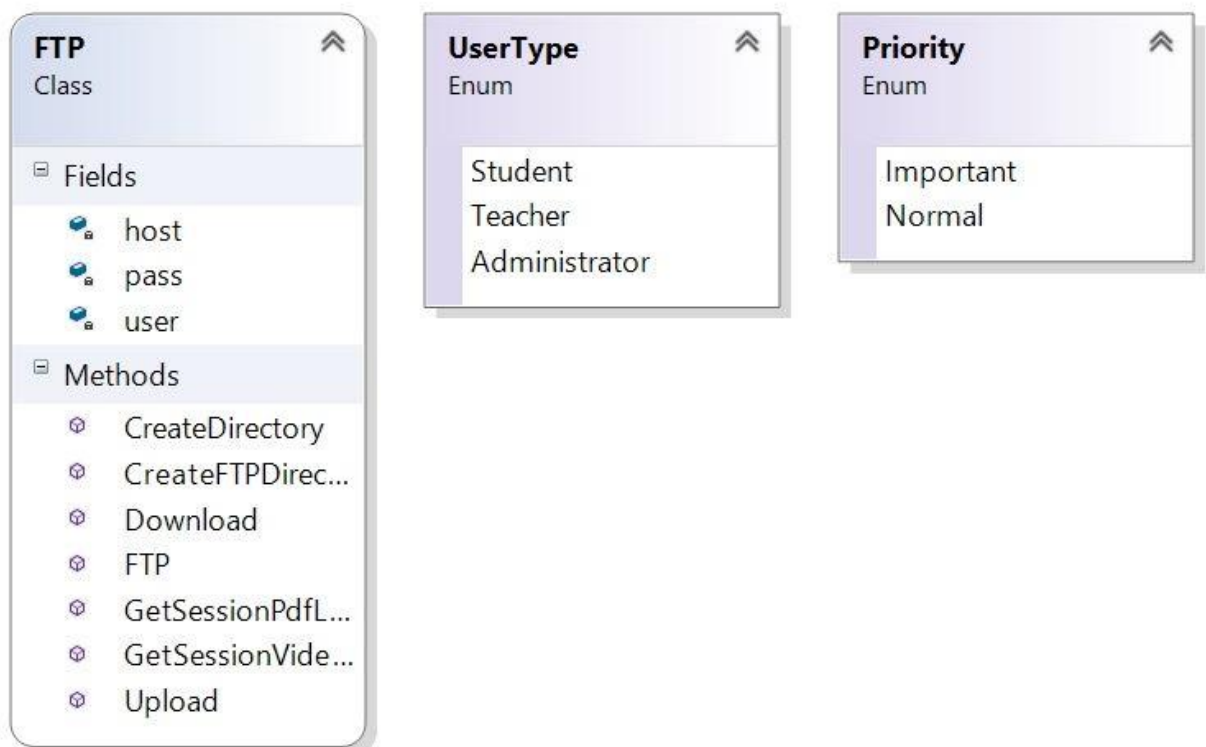


Figure 14 : Diagramme de classe de MookyCommon

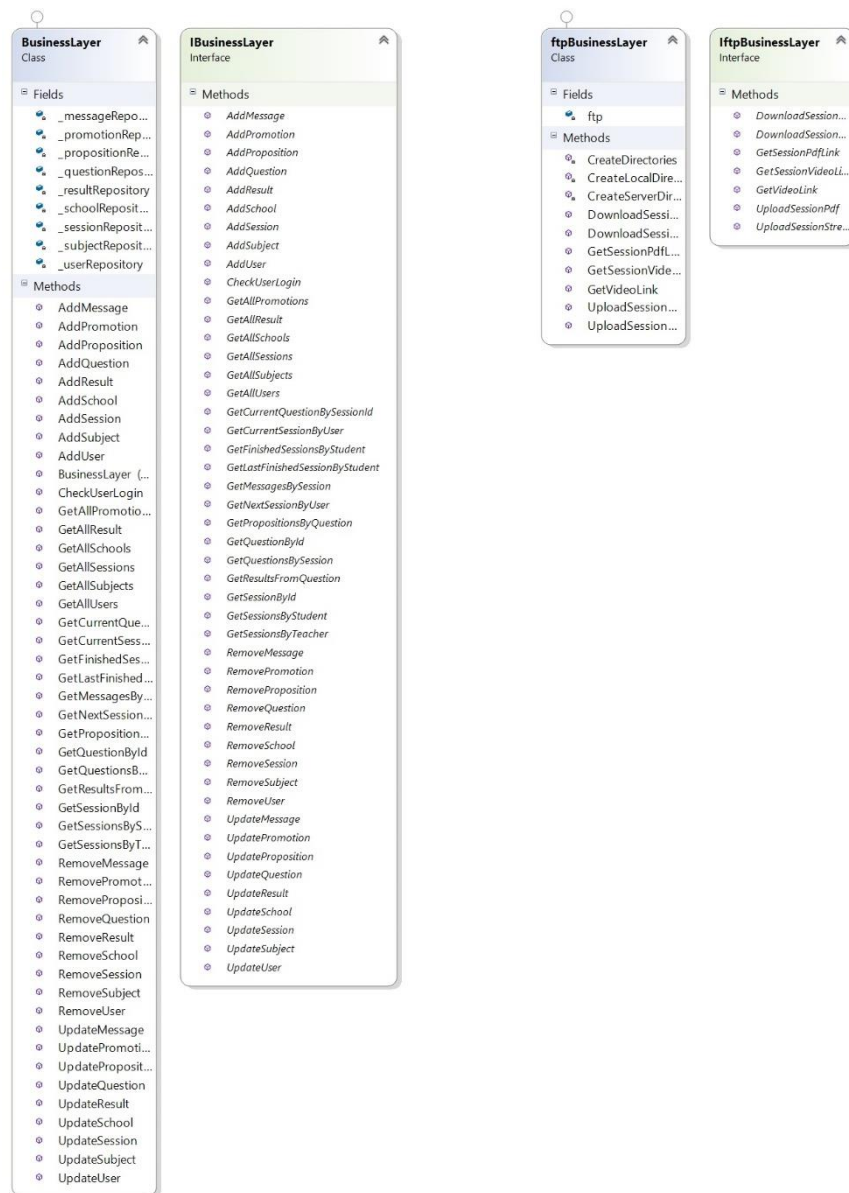


Figure 15 : Diagramme de classes de MookyBusiness

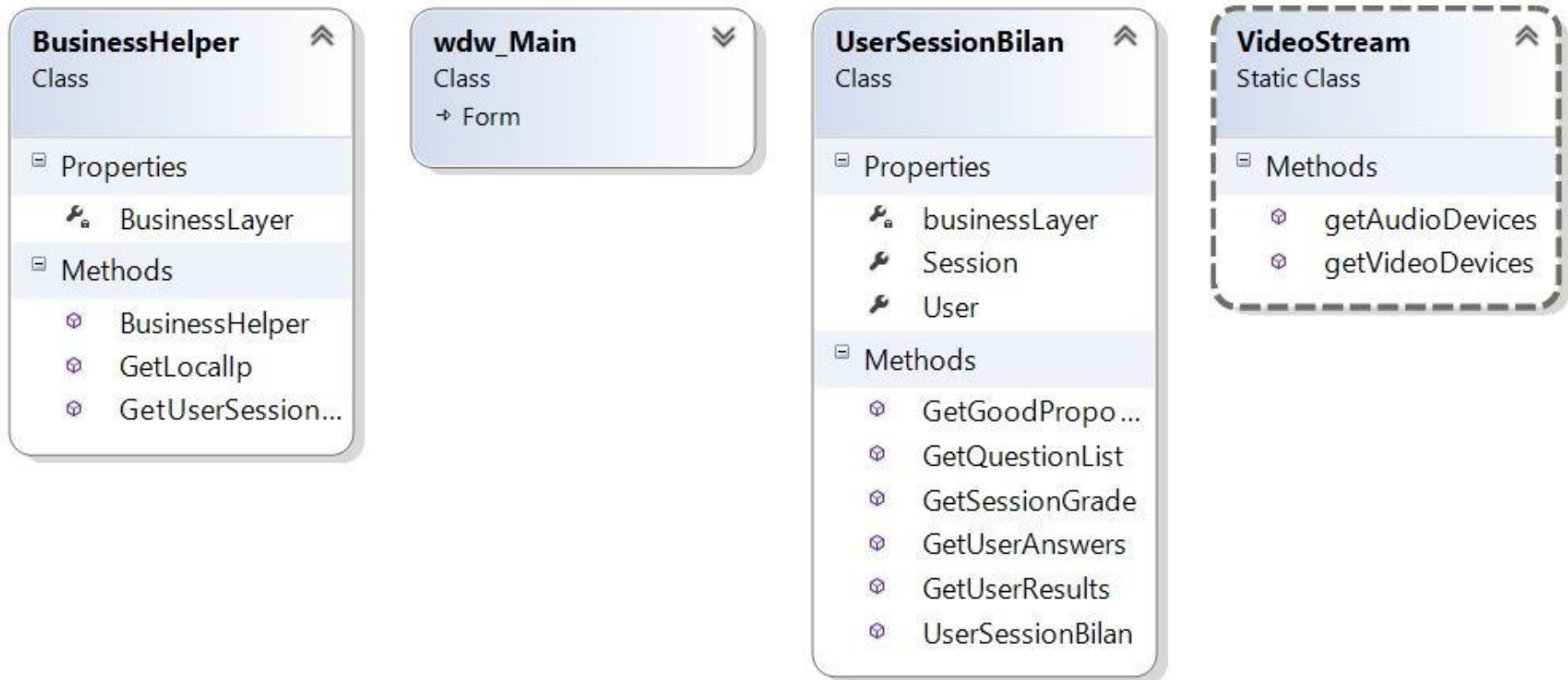


Figure 16 : Digramme de classes de Mookky

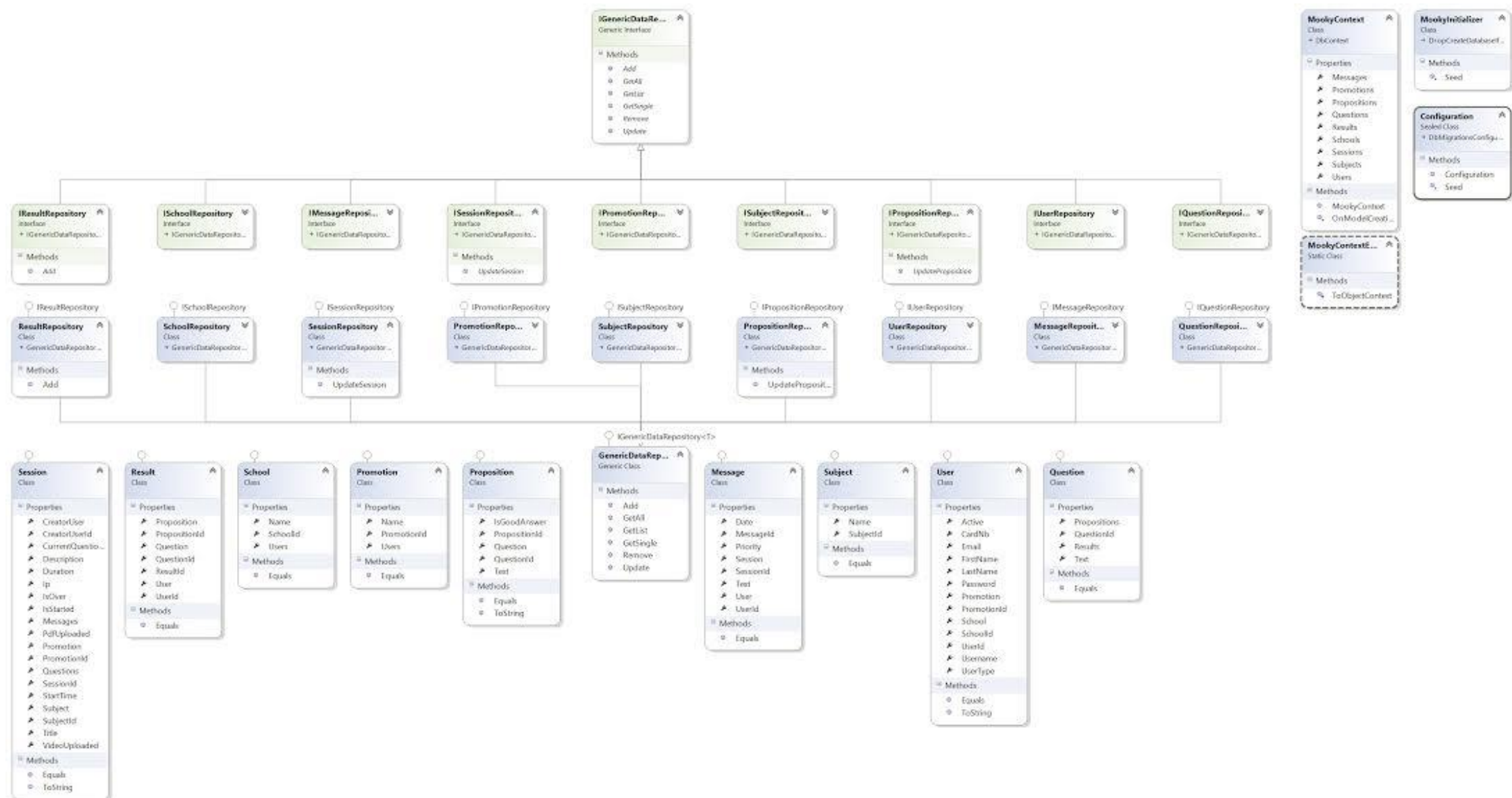


Figure 17 : Diagramme de classes de MookyDatabase

7 Diagrammes de séquence

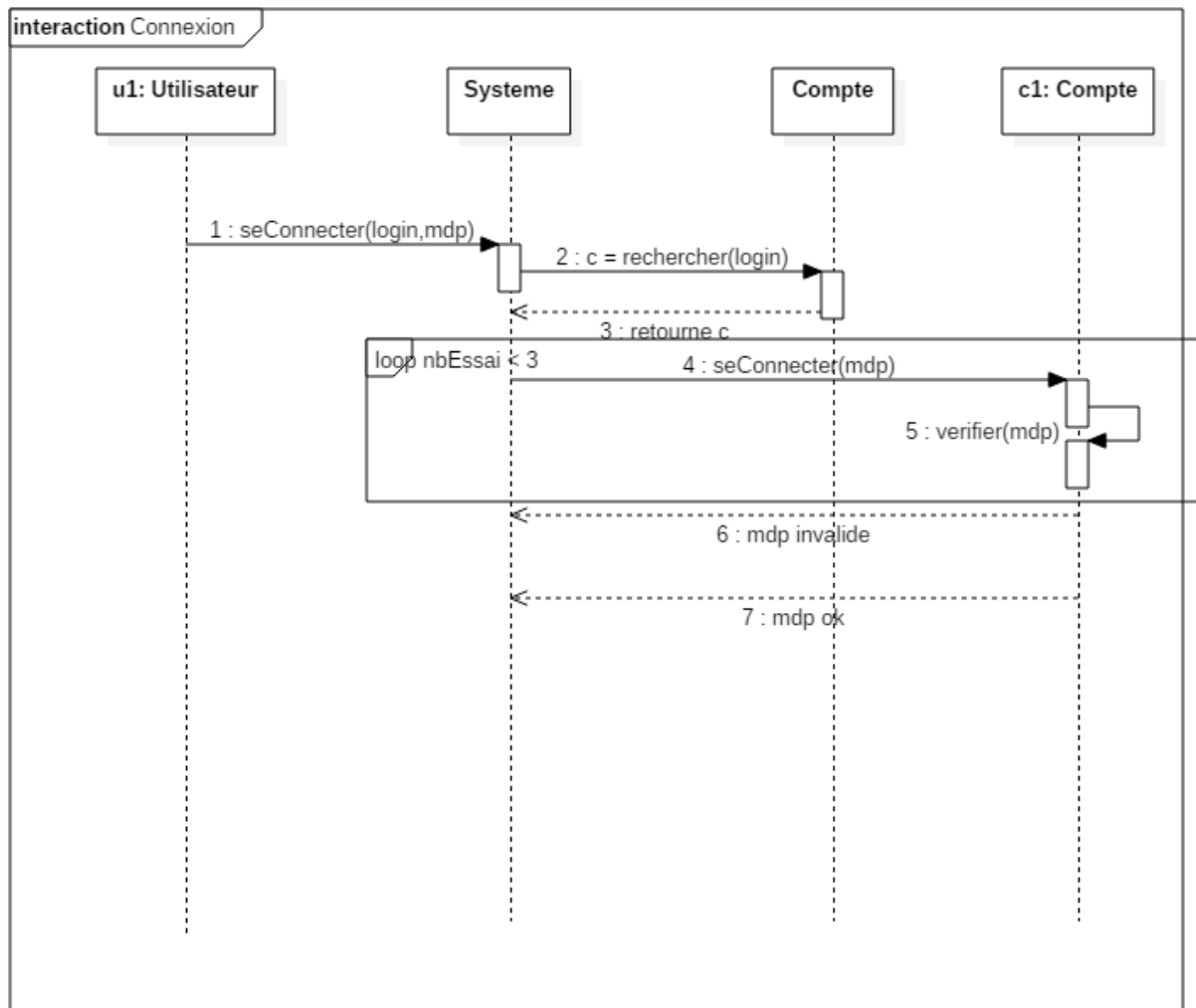


Figure 18 : Connexion au service

La figure 14 représente le diagramme de séquence pour l'action de se connecter sur son compte.

Ici, l'utilisateur saisie un nom d'utilisateur et un mot de passe. Le système recherche un compte associé au nom d'utilisateur.

Si oui, il retourne le compte. Une fois le compte retourné, le système vérifie le mot de passe. Le système effectue 3 tentatives de connexion. Dans le cas où le mot de passe est exact, l'utilisateur accède à son compte sinon il doit saisir de nouveau ses informations d'authentification.

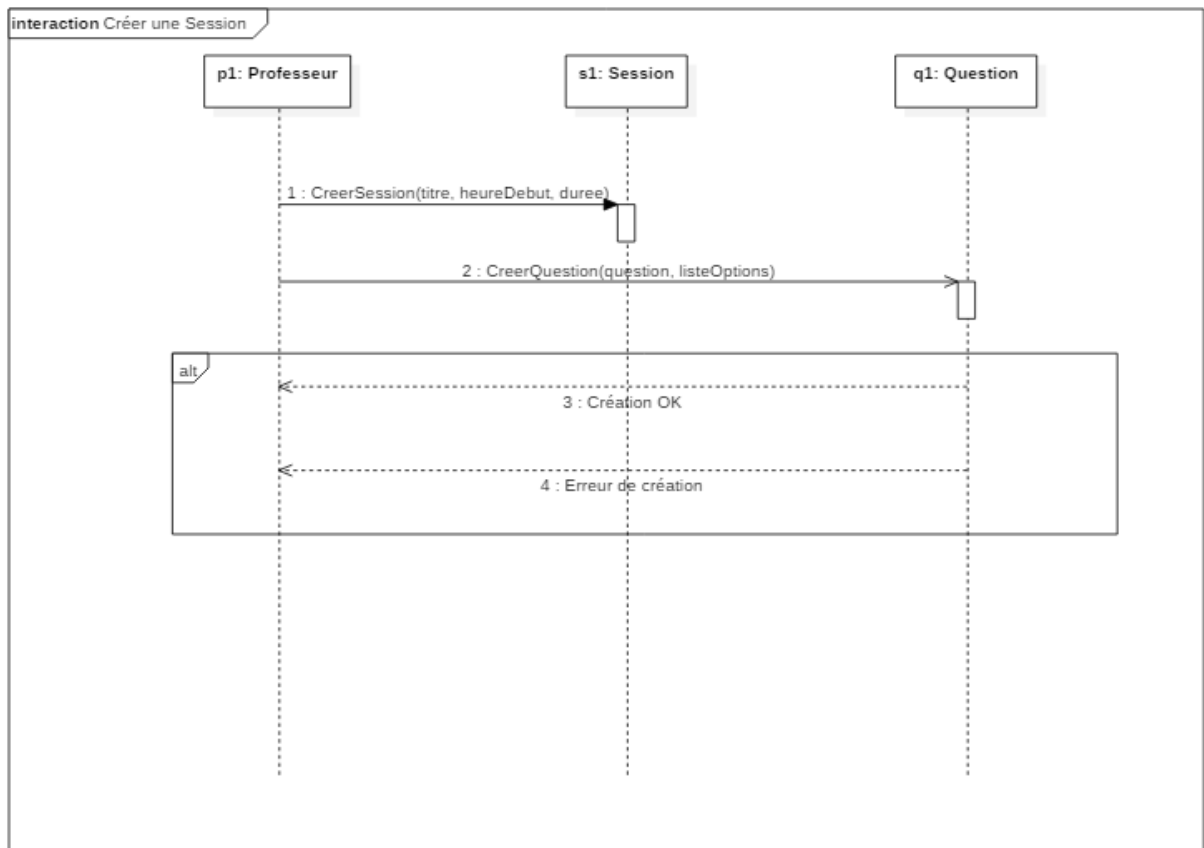


Figure 19 : Créer une session

La figure 15 représente le diagramme de séquence pour l'action créer une session.

Ici, le professeur qui se situe dans le menu de gestion des sessions, crée une nouvelle session. Pour cela il crée une session en remplissant un formulaire avec le titre, l'heure de début ainsi que la durée.

Deuxième étape, il crée les questions associées à ce cours. Ce questionnaire sera diffusé pendant le cours. Si la saisie est conforme, la nouvelle session sera programmée sinon le professeur devra vérifier les champs du formulaire mal saisi.

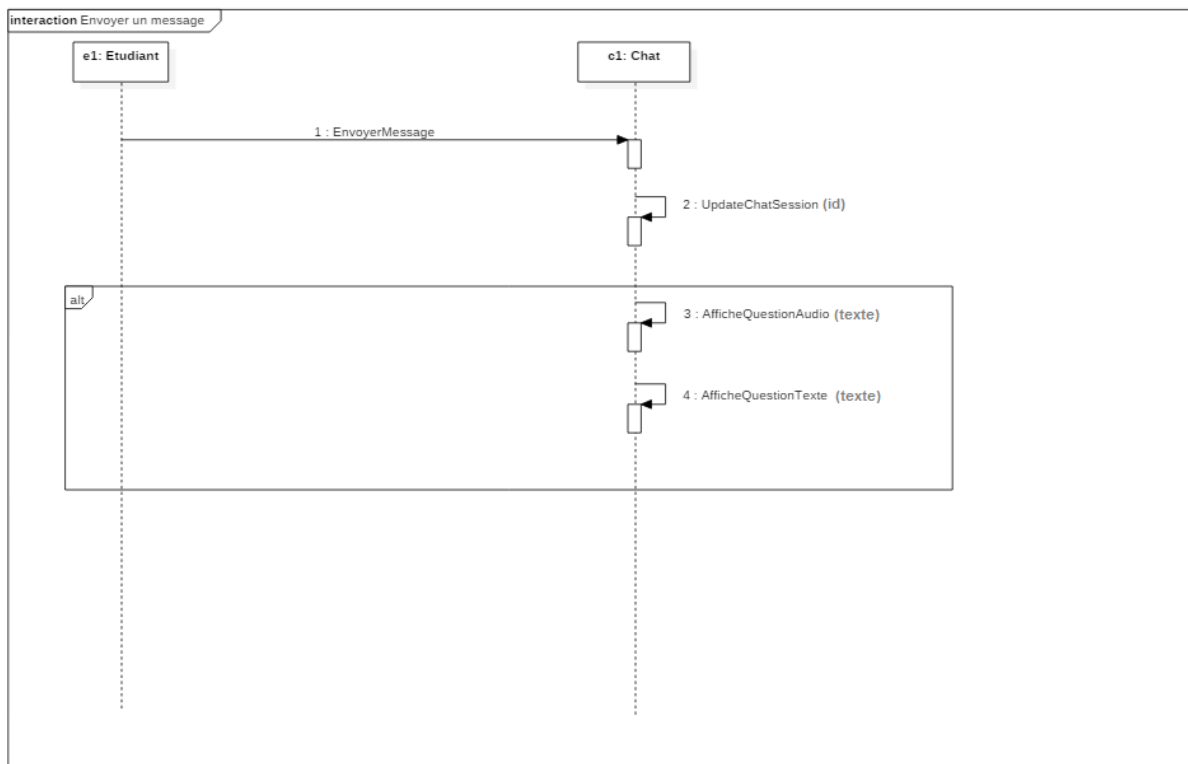


Figure 20 : Envoyer un message

La figure 16 représente le diagramme de séquence pour l'action d'envoyer un message.

Ici, l'étudiant qui est connecté à une session en cours peut envoyer des messages. Pour cela, l'étudiant saisit le message dans la textbox de la fenêtre, il appuie sur « envoyer ». Le chat de la session se met à jour. Selon ce que l'utilisateur a paramétré, le message envoyé est lu à l'oral et/ou est seulement à l'écrit.

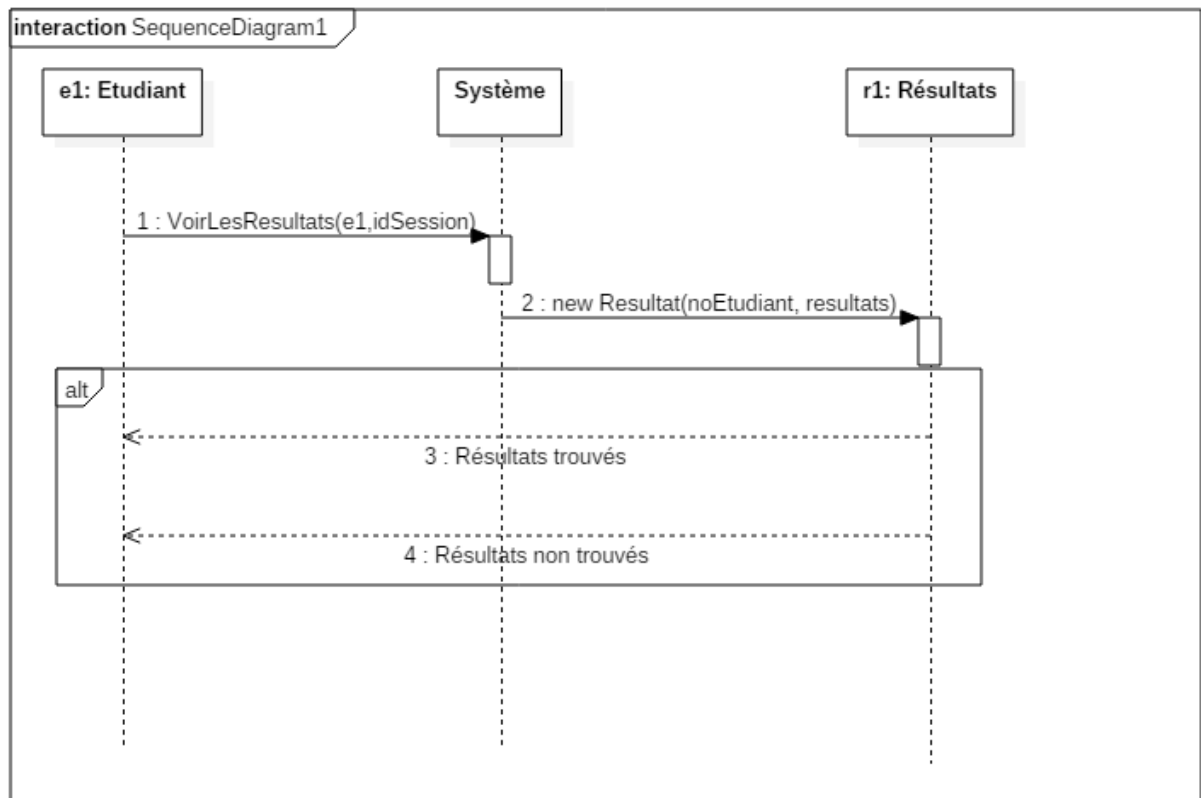


Figure 21 : Voir ses résultats

La figure 17 représente le diagramme de séquence pour l'action de voir les résultats d'une session.

Ici, l'étudiant qui est connecté à son compte, clique sur « Voir mes résultats ». Il choisit une session. Le système va récupérer les résultats trouvés ou non trouvés. Dans le cas où il trouve, il affichera les résultats à l'étudiant sinon un message d'erreur apparaîtra.

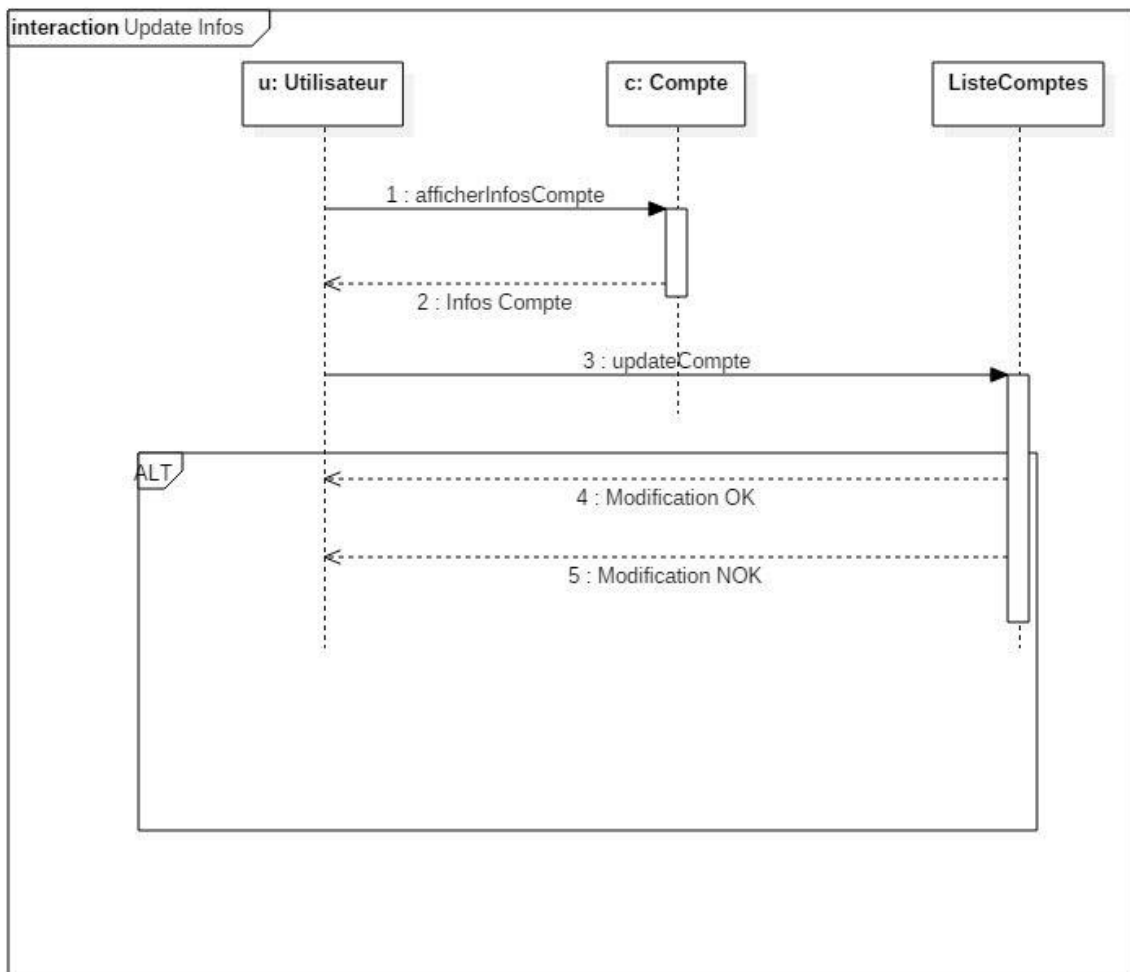


Figure 22 : Mise à jour des informations personnelles

La figure 18 représente le diagramme de séquence pour l'action de mettre à jour les informations personnelles de l'utilisateur.

Ici, l'utilisateur qui est connecté à son compte, clique sur « Mettre à jour mes informations personnelles ». Il met à jour ses informations à l'aide des champs textes associés. Une fois terminé, il clique sur le bouton « Mettre à jour »

8 Diagrammes d'états

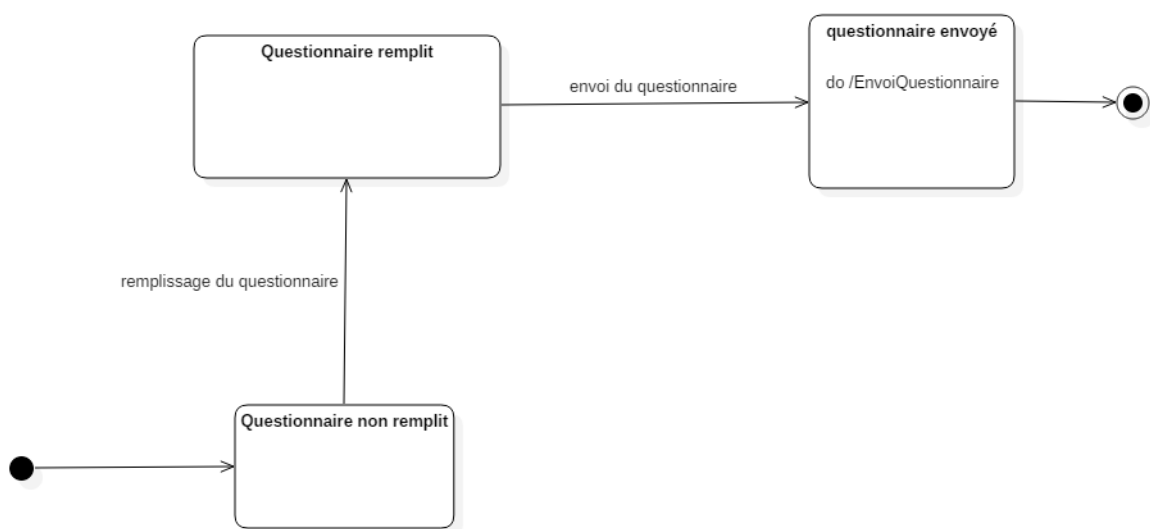


Figure 23 : Remplissage du questionnaire

La figure 19 représente le diagramme d'état pour l'action de répondre à un questionnaire.

Ici, l'utilisateur qui est connecté à une session de cours, reçoit un questionnaire vierge de réponses choisies. Il remplit alors le questionnaire. Une fois celui-ci rempli, le questionnaire est envoyé lorsque l'utilisateur clique sur « Valider ». Les réponses sont enregistrées et seront utilisées pour obtenir les résultats.

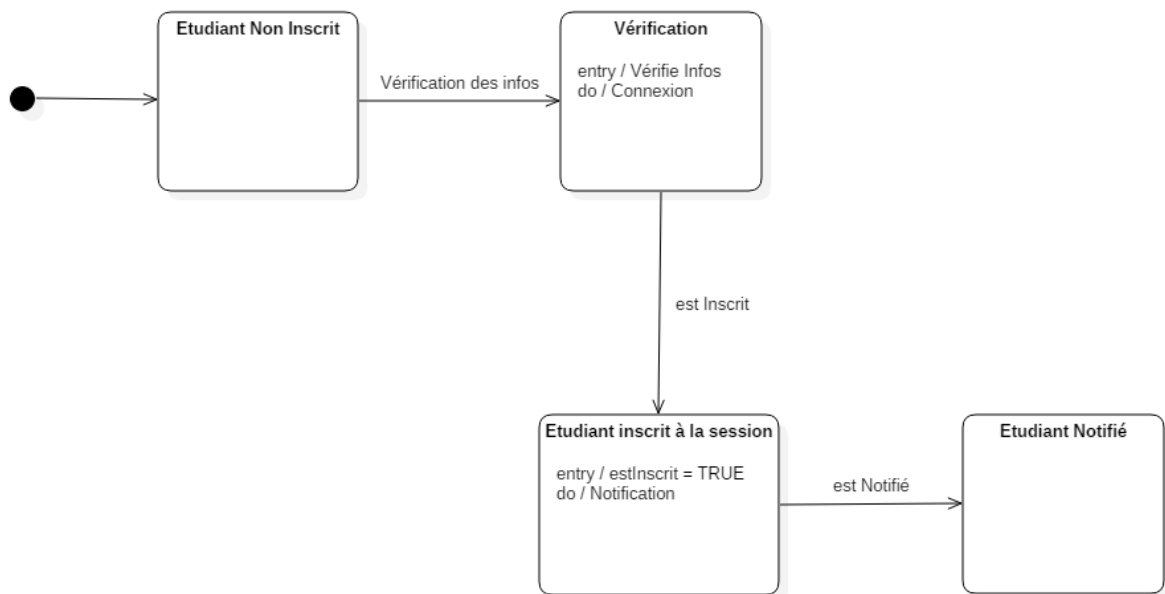


Figure 24 : Inscription à une session de cours

La figure 20 représente le diagramme d'état pour l'action d'inscrire un étudiant à une session de cours

Ici, le professeur va inscrire les étudiants à inscrire pour une session de cours. Le programme vérifie les étudiants du professeur puis donne la liste d'étudiants/promotion dont le professeur peut ajouter à la session. Une fois enregistré, les étudiants reçoivent une notification les informant du nouvel évènement.

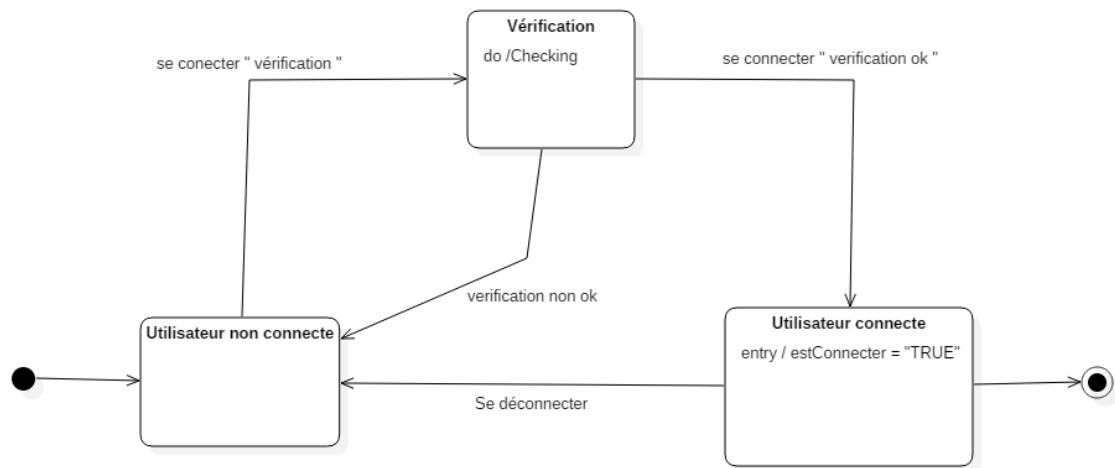


Figure 25 : Connexion à un compte utilisateur

La figure 21 représente le diagramme de séquence pour l'action de se connecter à l'application Mooky.

Ici, au lancement de l'application, le programme affiche un formulaire de connexion. L'utilisateur saisit ses identifiants de connexion. Ensuite, le système vérifie que les informations sont correctes. Si oui, l'utilisateur est connecté et accède au menu principal de Mooky. Dans le cas contraire, l'utilisateur se voit renvoyer au formulaire de connexion avec le message de l'erreur correspondant.

9 Diagrammes de packages

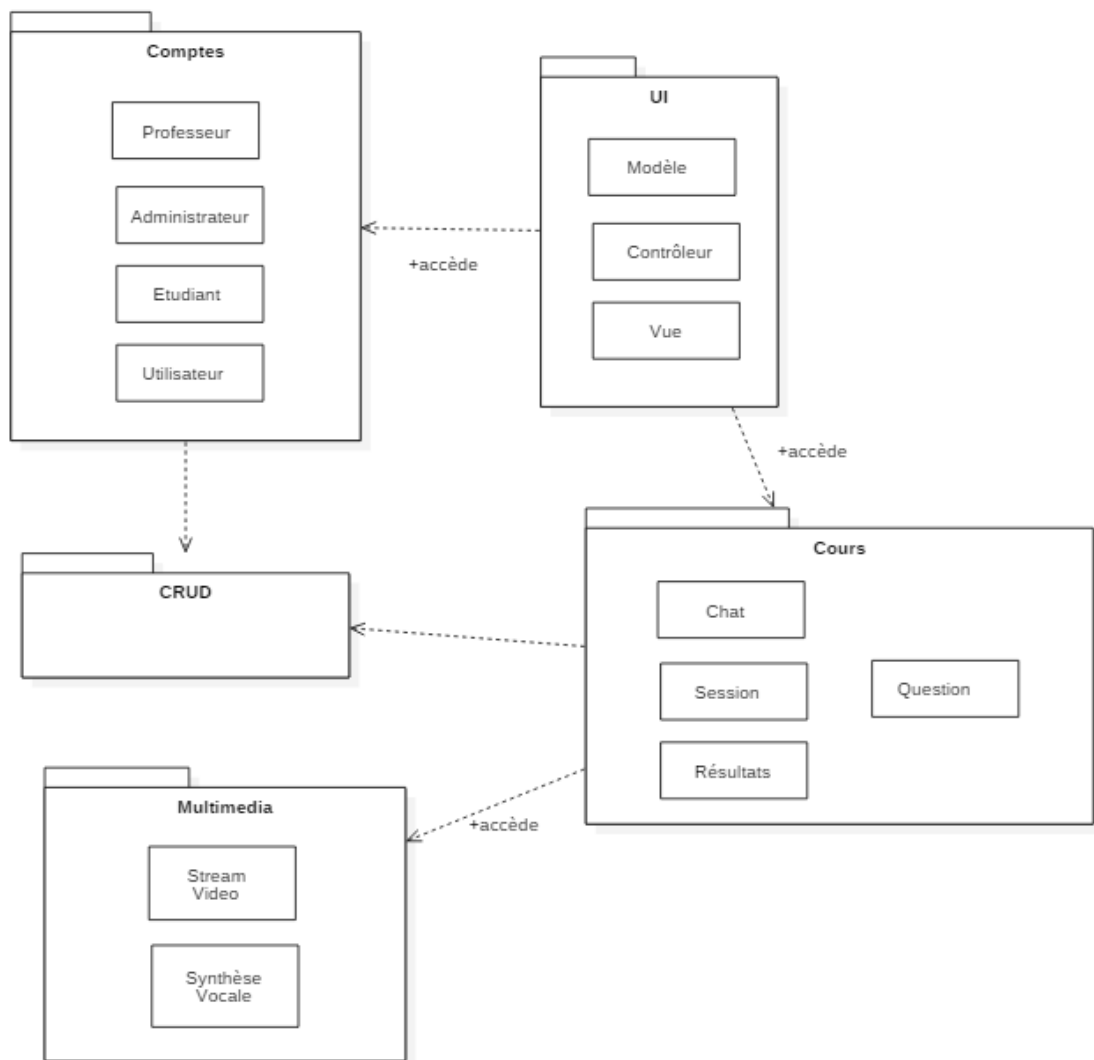


Figure 26 : Diagramme de packages

- **Comptes** : Paquet regroupant les différentes classes de comptes.
- **UI** : Paquet regroupant le modèle MVC (modèle/contrôleur/vue). Il permet de gérer l'interface utilisateur du programme de façon propre en séparant les trois parties. Cela permet de bénéficier d'une clarté dans l'architecture. Le MVC facilite par exemple la maintenance de l'application.
- **CRUD** : Paquet regroupant toutes les classes qui accèdent à la base de données. Il permet de séparer les accès à la base de données du reste de l'application afin de ne pas impacter les différents composants du programme.

- **Cours** : Paquet regroupant les classes liées à la gestion des cours ainsi que les composants d'une session.
- **Multimédia** : Paquet regroupant les bibliothèques de streaming vidéo et synthèse vocale

10 Maquettes de départ

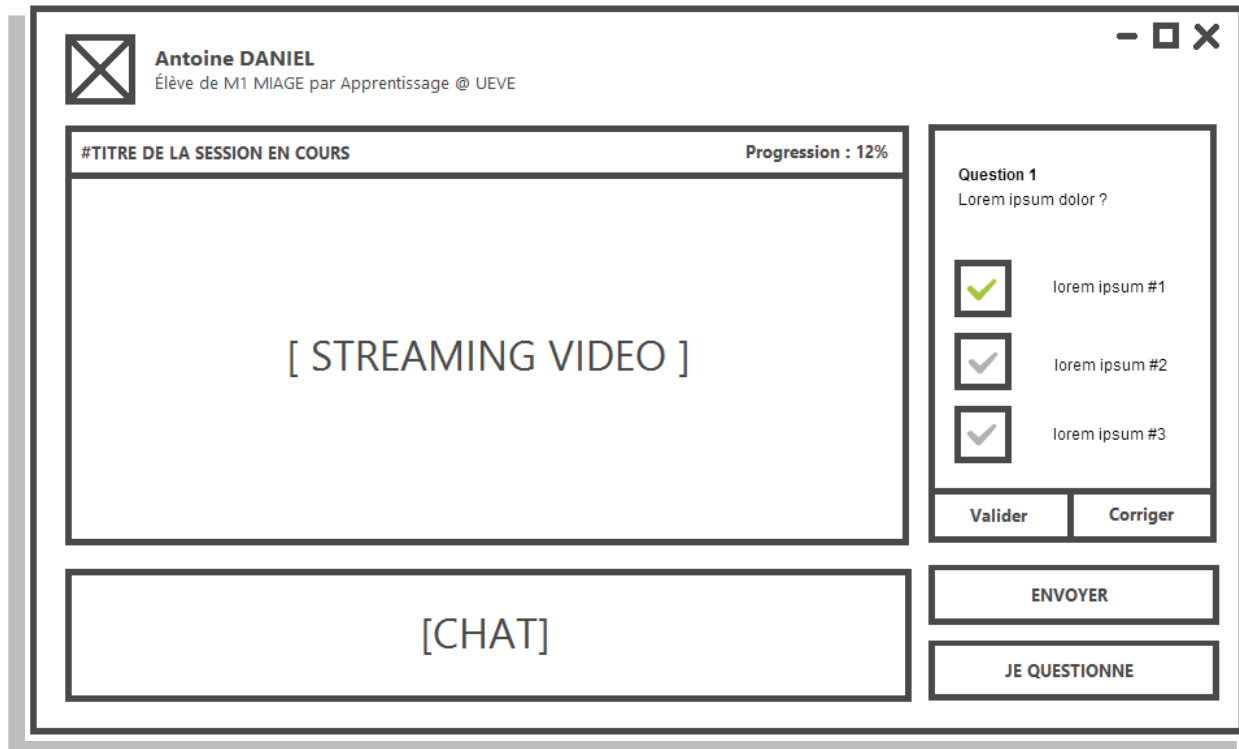


Figure 27 : IHM de session pour l'étudiant

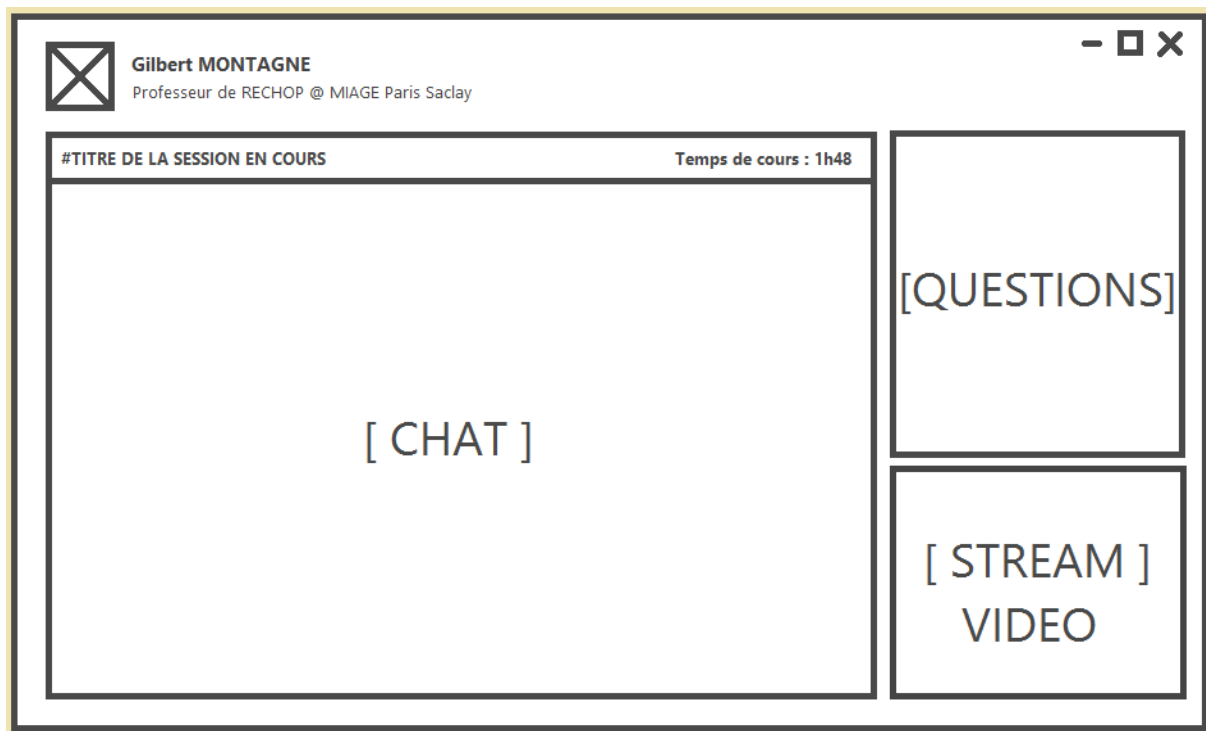


Figure 28 : IHM de professeur pour la session

Conception du logiciel

11 Gestion de la Base de données

11.1 Le cloud avec Microsoft Azure

La base de données est stockée sur la plateforme en nuage Microsoft Azure. Cette plateforme a plusieurs avantages :

- La base de données est accessible partout
- Le serveur s'adapte automatiquement à la charge
- Le risque de perte de donnée est extrêmement faible
- La disponibilité des serveurs est très élevée (supérieure à 99.9%)
- Un pare-feu permet de gérer les accès à la base

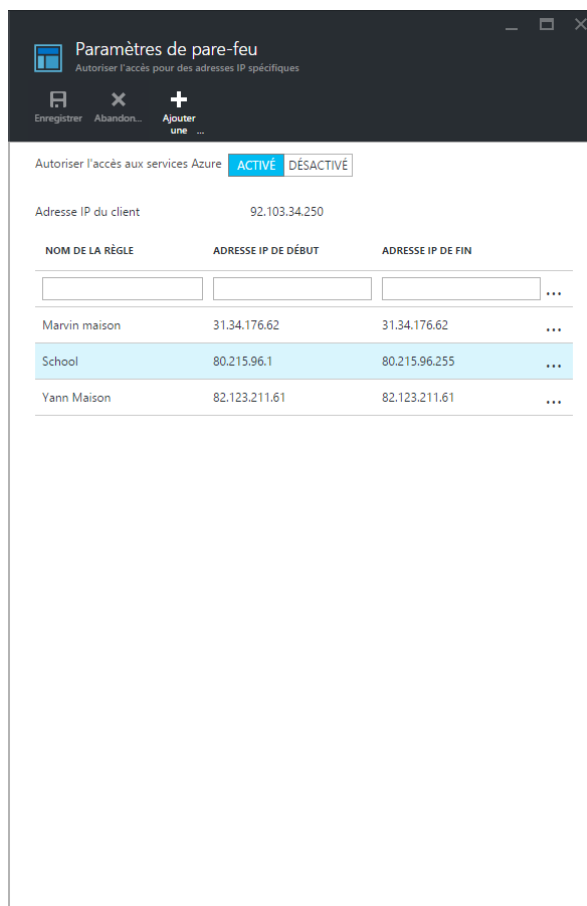


Figure 29 : Paramètres de pare-feu sur Microsoft Azure

11.2 EntityFramework et le modèle « Business Layer »

Pour la gestion de la base de données nous utilisons l'ORM Entity Framework. Nous appliquons le développement « model first » pour la base de données, ce qui signifie qu'Entity créera automatiquement une table par classe du model.

La création de la base de données est gérée par les classes de contexte et « l'initializer ».

Tout d'abord le contexte est le point d'entrée pour toute action sur la base de données. Il contient la chaine de connexion qui permet de se connecter (lien du serveur, nom d'utilisateur, mot de passe, paramètres divers, nom de la base de données), la configuration du contexte, le lien vers l'initializer ainsi que les différentes classes qui seront matérialisées en table dans la base de données.

Ensuite l'initializer permet au démarrage de l'application de choisir entre :

- `CreateDatabaseIfNotExists` : Crée une base de données si elle n'existe pas (En production)
- `DropCreateDatabaseIfModelChanges` : Recrée la base si le model a changé (En production)
- `DropCreateDatabaseAlways` : Effacer et recrée la base à chaque lancement de l'application. (Pour les tests)

L'initializer permet aussi d'introduire des informations dans la base de données.

Pour les actions CRUD (Create, Read, Update, Delete), nous avons la classe générique `GenericDataRepository`. La généricité nous permet d'avoir une classe `Repository` pour chaque model qui hérite de `GenericDataRepository`, ceci évite le code inutile et permet de créer des méthodes uniques à chaque model.

Cependant nous n'utilisons pas directement les classes `Repository` qui donnent trop de libertés. Nous profitons d'une classe `BusinessLayer`, qui instancie tous les `Repository` de chaque model.

Cependant nous n'utilisons pas directement les classes `Repository` qui donnent trop de libertés. Nous profitons d'une classe `BusinessLayer`, qui instancie et utilise tous les `Repository` de chaque model, et écrivons chaque méthode qui nous seront nécessaires en rapport avec la base de données.

12 Fonctionnalités finales de MOOKY

12.1 Connexion

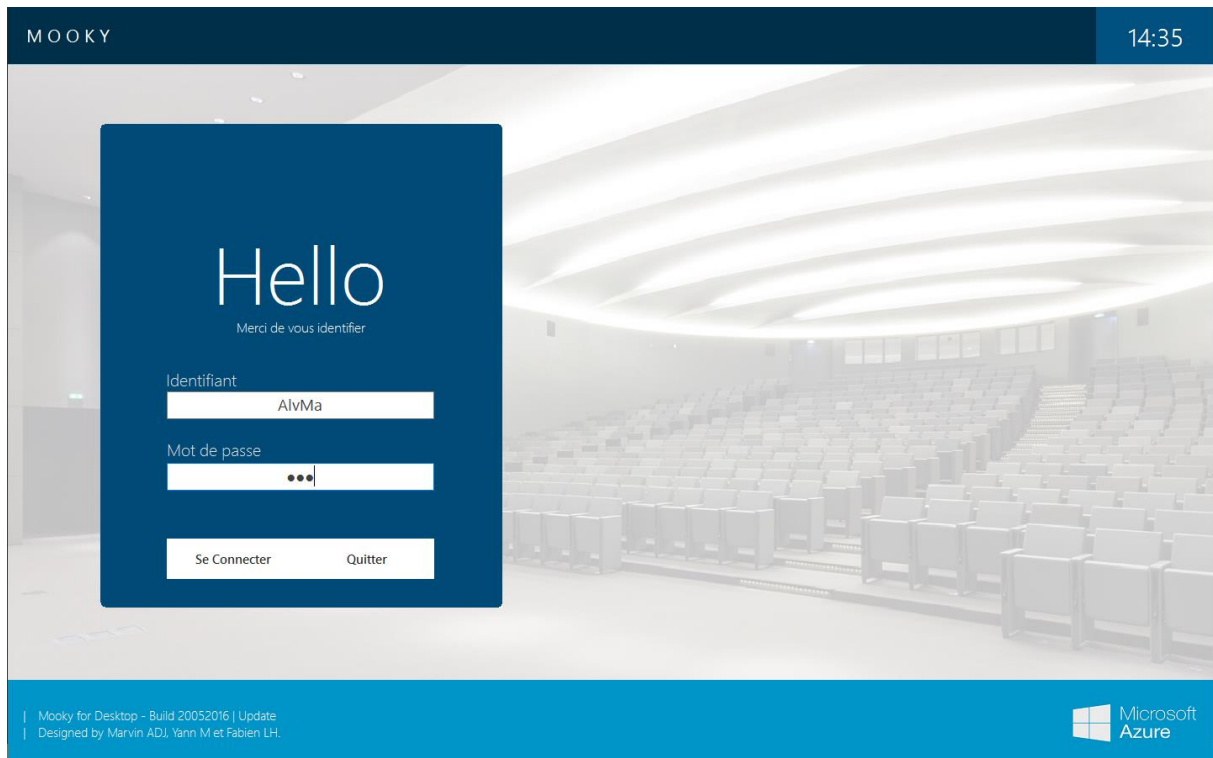


Figure 30 : Ecran de connexion

Pour la phase « identification », l'utilisateur entre son MookylD ainsi que son mot de passe associé. Il peut dans cet écran « Se connecter » ou « Quitter » Mooky. Il informe aussi l'utilisateur sur la version actuelle du programme dite « Numéro de Build ».

12.2 Espace Personnel

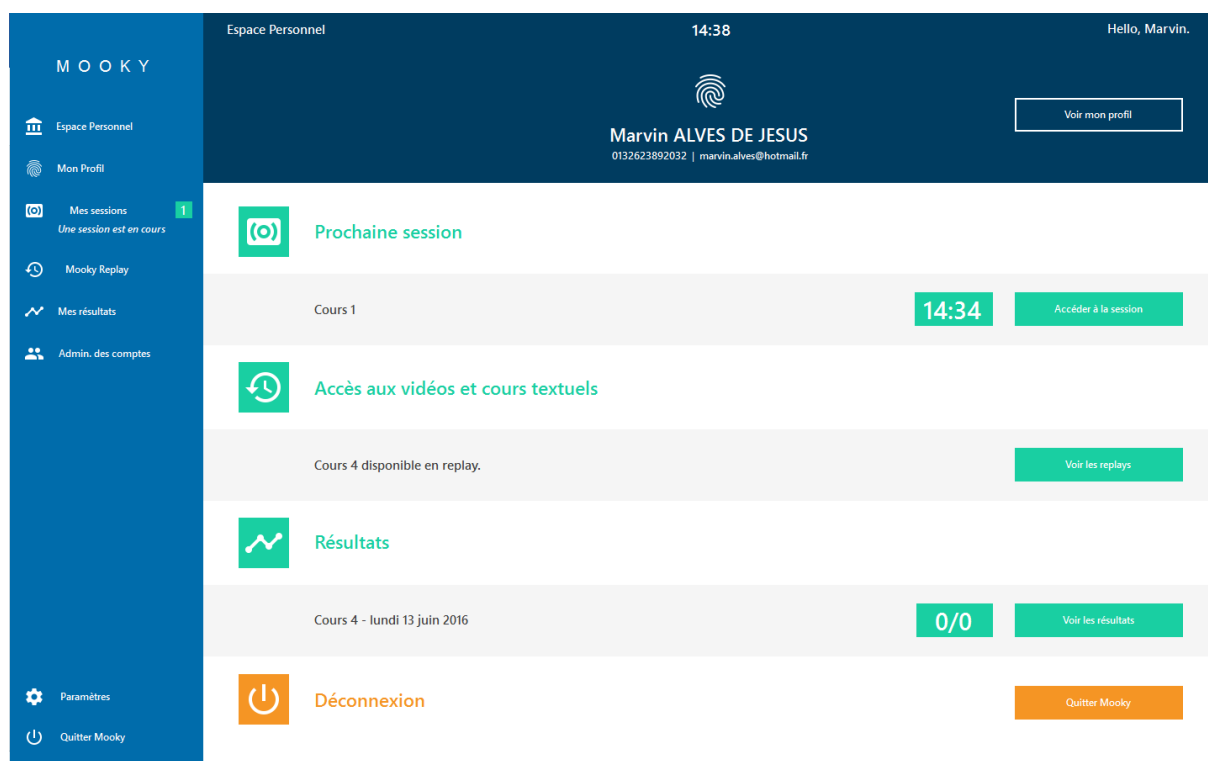


Figure 31: Espace Personnel

Une fois que l'utilisateur est connecté, ses informations de compte sont récupérées et MookY affiche « l'Espace Personnel ». Cet écran est un résumé des différentes notifications d'activités de l'utilisateur. C'est lors de l'identification que la barre latérale nommée « Access Bar » s'affiche. Elle permet de naviguer rapidement dans les menus de l'application. Cette dernière s'adapte au type de compte de l'utilisateur en cours.

L'écran contient une barre latérale d'informations de profil avec le nom, le mail ainsi que le n° de carte universitaire. Un bouton mène directement à la gestion du profil identifié.

En dessous, nous retrouvons pour un étudiant :

- **La session en cours** (ou la prochaine session programmée) : Ici est affiché le titre, la description de la session ainsi que l'heure programmée. Un bouton d'accès à la session est affiché.
- **Le résultat de la dernière session effectuée** : Nous retrouvons ici le titre de la session, le score ainsi qu'un bouton d'accès à la gestion des résultats.

- **Le Replay** : Est affiché, le titre de la dernière session mise en ligne sur Mooky Replay et un bouton d'accès rapide au Mooky Replay.
- **Le bouton de déconnexion** de Mooky.

Pour un professeur (*en plus des fonctionnalités Etudiant*) :

- **La gestion des sessions** : remplace la fonction Mooky Replay. Permet de gérer les sessions du professeur identifié (Création, modification, suppression)

Pour un professeur administrateur (*en plus des fonctions Etudiant et Professeur*) :

- **La gestion des comptes MookyID** : permet de gérer les comptes Mooky (Création, modification, suppression)

12.3 Gestion du profil

The screenshot displays the 'Mon profil' (My profile) page in the Mooky application. The sidebar on the left contains the following links: Espace Personnel, Mon Profil, Mes sessions (with a notification 'Une session est en cours'), Mooky Replay, Mes résultats, Admin. des comptes, Paramètres, and Quitter Mooky. The main content area is titled 'Mon profil' and shows the user's name 'Marvin Alves de Jesus' and student ID 'N°Etudiant : 0132623892032'. Below this, there are two sections: 'Informations de connexion' (Login information) and 'Informations complémentaires' (Additional information). The login section includes fields for 'Identifiant' (AlvMa) and 'Mot de passe' (mdp). The additional information section includes fields for 'Mail' (marvin.alves@hotmail.fr), 'Établissement' (Master 1 MIAAGE Apprentissage), and 'Promotion' (Evry Val d'Essonne). A green 'Valider' button is located at the bottom of the form.

Figure 32 : Gestion du profil

Dans cet écran, la gestion du profil permet d'éditer certaines informations du compte comme le mot de passe, l'établissement ainsi que la promotion dans le cas d'un étudiant.

Une fois les changements effectués et validés, une notification de confirmation apparaît afin d'informer l'utilisateur si l'opération a réussi ou non.

12.4 Accès à la session (étudiant)

L'accès à la session permet pour l'étudiant de se connecter à la session démarrée par un professeur. Dans cet écran, nous retrouvons :

- Le lecteur vidéo du flux en direct (Composant Windows Media Player)
- Le chat qui permet de communiquer entre étudiants et de poser une question au professeur.
- Le questionnaire en temps réel
- Le temps écoulé de la session.

C'est dans celui-ci que l'étudiant va participer à la session et ainsi enregistrer un résultat.

12.5 Accès à la session (professeur)

L'accès à la session permet pour le professeur de démarrer sa session. Dans cet écran, nous retrouvons :

- Le lecteur vidéo du flux en direct
- Le chat qui permet de visualiser les questions des étudiants.
- Le bouton de navigation pour la question suivante.

Une fois celle-ci terminée, le professeur pourra choisir de la publier sur la plateforme Mooky Replay pour que les étudiants ayant participé à ce cours puissent le revoir, visionner le PDF du cours ou encore le questionnaire avec les réponses correctes.

12.6 Mooky Replay

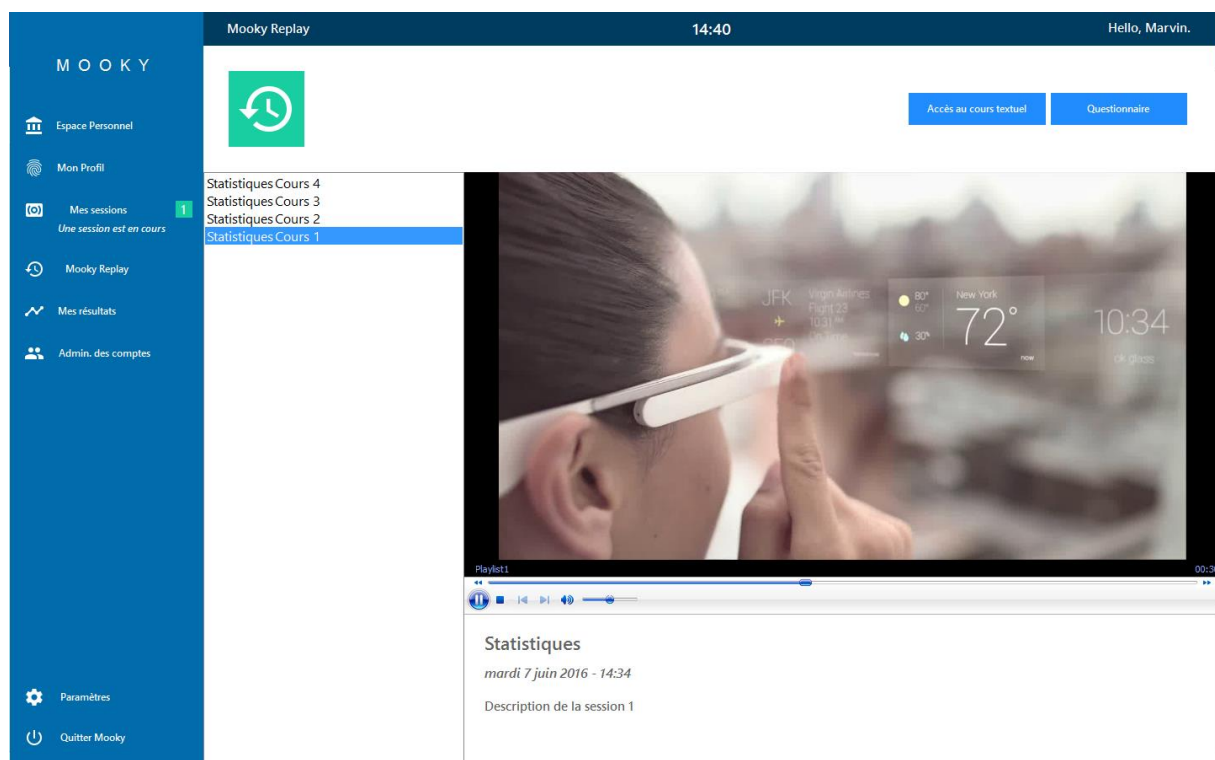


Figure 33: Mooky replay

Mooky Replay est la plateforme de Cours à la demande de Mooky. Celle-ci permet à un étudiant de retrouver les cours suivis. A l'aide d'un simple clic sur la zone de sélection, une fenêtre, contenant un lecteur vidéo et une description, s'affiche. Deux boutons sont disponibles :

- **Accès au cours textuel** : permet de visualiser le cours sous forme de fichier PDF qui a été associé à la session sélectionnée.
- **Questionnaire** : permet de visualiser les questions posées lors de la session.

12.7 Mes Résultats

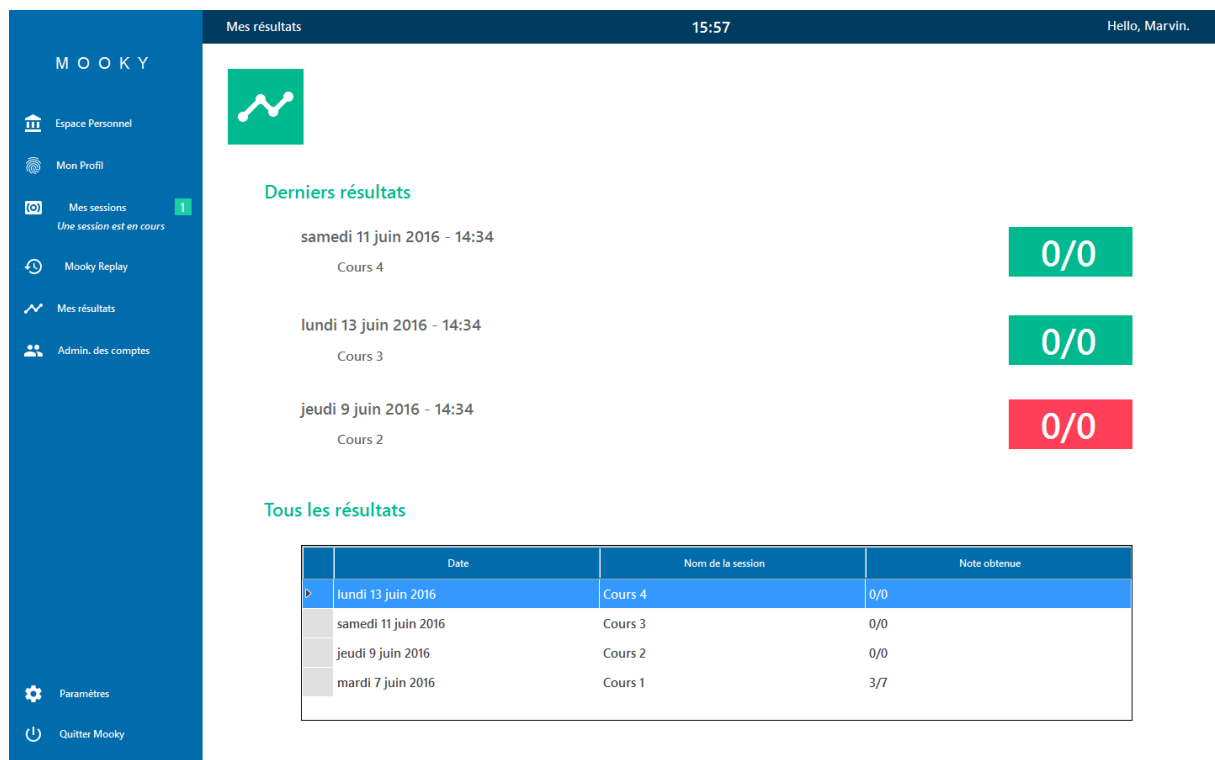


Figure 34 : Mes résultats

Mes Résultats est l'écran de visualisation des résultats obtenus. Il permet principalement de consulter les notes obtenues ainsi que la correction du questionnaire, le tout associé à une session.

12.8 Gestion des sessions (professeur)

MOOKY

Espace Personnel

Mon Profil

Démarrer la session

Une session est programmée

Gestion des sessions

Mes résultats

Paramètres

Quitter Mooky

Gestion des sessions

09:00

Hello, Jean.

Créer une session

Modifier une session

Supprimer une session

Titre :

Cours N°3 - Graphes Complexes

Date et heure :

13 : 00

jeudi 16 juin 2016

Durée :

2 : 10

Matière :

Statistiques

Promotion :

Master 1 MIAGE Apprentissage

Description :

Cours traitant le chapitre sur les graphes complexes.
Avec évaluation.

PDF :

C:\Users\Devolga\Pictures\CM4.pdf

...

Valider cette nouvelle session

Figure 35: Gestion des sessions

L'écran de gestion de sessions permet de manager les sessions du professeur identifié. A l'aide de trois onglets, il peut créer, modifier ou supprimer une session.

12.9 Gestion des comptes MookyID (professeur administrateur)

MOOKY

Espace Personnel

Mon Profil

Mes sessions

Une session est en cours

MookY Replay

Mes résultats

Admin. des comptes

Paramètres

Quitter MookY

Administration des comptes MookYID

08:56

Hello, Marvin.

Créer un utilisateur

Modifier un utilisateur

Supprimer un utilisateur

Alves de Jesus Marvin (0132623892032)

N° de carte :

0132623892032

Identifiant :

AlvMa

Nom :

Alves de Jesus

e-Mail :

marvin.alves@hotmail.fr

Mot de Passe :

mdp

Prénom :

Marvin

Etablissement :

Evry Val d'Essonne

Promotion :

Type Utilisateur :

Student

Modifier cet utilisateur

Figure 36 : Gestion des comptes MookYID

La fenêtre de gestion des comptes permet à un professeur administrateur de pouvoir manager les comptes MookYID. Il peut créer, modifier ou suspendre un compte.

12.10 Paramètres

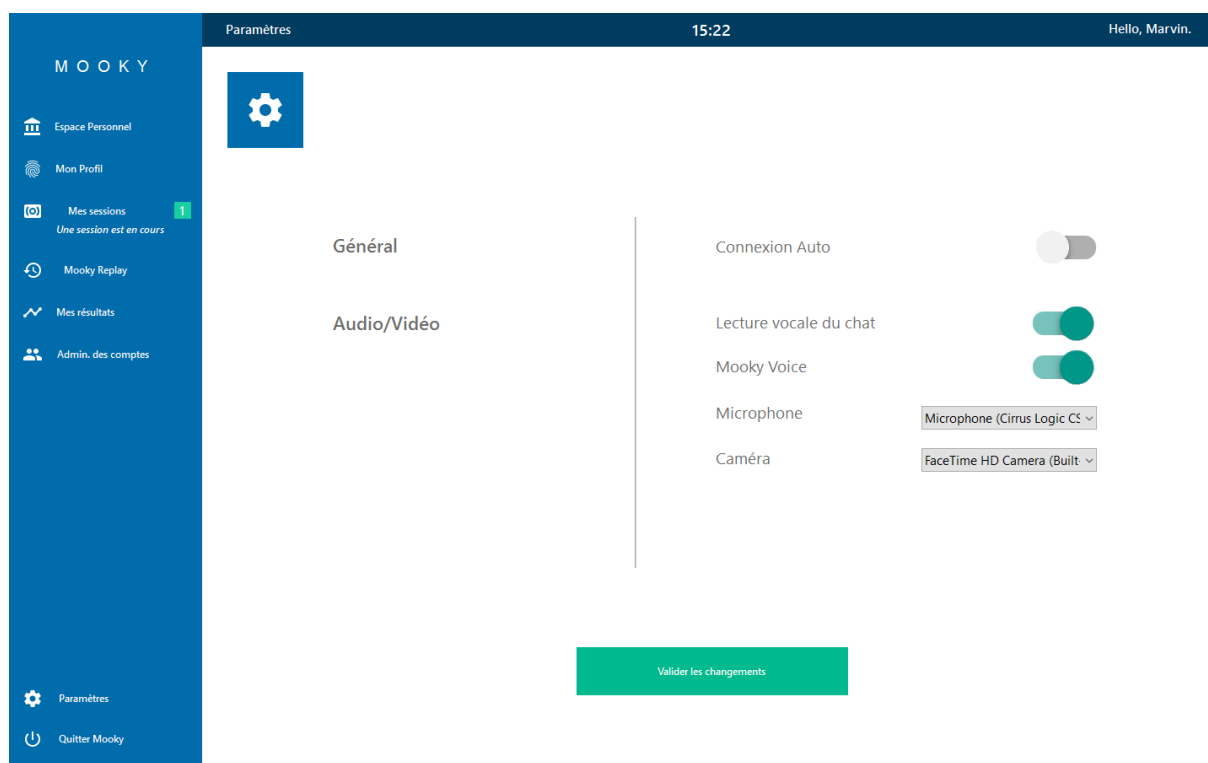


Figure 37 : Ecran de paramétrage

L'écran de paramétrage permet de personnaliser le client à l'aide de diverses options. Dans celui-ci il est possible de paramétrer les options suivantes :

- **Lecture vocale** : cette option active la lecture à l'aide d'un synthétiseur vocal, la question posée par un étudiant.
- **Mooky Voice** : option qui permet, lors d'une session, d'interagir avec le logiciel à l'aide de la reconnaissance vocale pour un professeur.
- **Microphone** : permet de sélectionner le périphérique microphone par défaut.
- **Caméra** : permet de sélectionner le périphérique webcam par défaut.

13 Table des figures

Figure 1: Diagramme de cas d'utilisations	12
Figure 2 : Cas d'utilisation général	12
Figure 3 : Connexion à un compte.....	13
Figure 4 : Ajout d'un compte utilisateur	14
Figure 5 : Modification d'un compte utilisateur.....	15
Figure 6 : Entrer dans une session de cours	16
Figure 7 : Accéder aux cours textuels.....	17
Figure 8 : Voir ses résultats	17
Figure 9 : Gérer ses informations personnelles	18
Figure 10 : Créer une session de cours.....	19
Figure 11 : Modifier une session de cours	20
Figure 12 : Supprimer une session en cours.....	21
Figure 13 : Démarrer une session de cours.....	22
Figure 14 : Diagramme de classe de MookyCommon.....	23
Figure 15 : Diagramme de classes de MookyBusiness	24
Figure 16 : Diagramme de classes de Mooky	25
Figure 17 : Diagramme de classes de MookyDatabase	26
Figure 18 : Connexion au service.....	27
Figure 19 : Créer une session	28
Figure 20 : Envoyer un message.....	29
Figure 21 : Voir ses résultats	30
Figure 22 : Mise à jour des informations personnelles	31
Figure 23 : Remplissage du questionnaire	32
Figure 24 : Inscription à une session de cours	33
Figure 25 : Connexion à un compte utilisateur	34
Figure 26 : Diagramme de packages	35
Figure 27 : IHM de session pour l'étudiant	36
Figure 28 : IHM de professeur pour la session.....	37
Figure 29 : Paramètres de pare-feu sur Microsoft Azure.....	38
Figure 30 : Ecran de connexion	40
Figure 31: Espace Personnel.....	41
Figure 32 : Gestion du profil	42
Figure 33: Mooky replay.....	44
Figure 34 : Mes résultats.....	45
Figure 35: Gestion des sessions.....	46
Figure 36 : Gestion des comptes MookyID	47
Figure 37 : Ecran de paramétrage	48