

TABLE DES MATIERES

TABLE DES MATIERES	1
Table des figures.	3
Introduction.	4
I. Contexte et objectifs du projet	5
II. Analyse.....	5
II.1. Exigences fonctionnelles et non fonctionnelles.	5
II.1.1. Exigences fonctionnelles	5
II.1.2. Exigences non-fonctionnelles.....	6
II.2. Architecture logique de l'application.	6
II.3. Diagramme de contexte.	7
II.4. Diagramme de package.....	7
II.5. Diagramme de classes métier	7
II.6. Diagramme de cas d'utilisation.....	8
II.7. Diagrammes de séquence système.....	20
II.7.1. Diagramme de séquence système "Se connecter".....	20
II.7.2. Diagramme de séquence système "Noter un devoir"	21
II.7.3. Diagramme de séquence système "Modifier un cours"	21
II.7.4. Diagramme de séquence système "Consulter un cours"	22
II.7.5. Diagramme de séquence système "Soumettre un devoir"	22
II.7.6. Diagramme de séquence système "Se synchroniser"	23
II.7.7. Diagramme de séquence système "Ajout de fichiers privés"	23
II.7.8. Diagramme de séquence système "Inscrire un utilisateur à un cours"	24
II.7.9. Diagramme de séquence système "Désinscrire un utilisateur à un cours"	24
III. Conception.	25
III.1. Diagrammes de séquence technique.	25
III.1.1. Diagramme de séquence technique "Synchroniser"	25
III.1.2. Diagramme de séquence technique "Consulter un cours"	26
III.1.3. Diagramme de séquence technique "Modifier un cours"	26
III.1.4. Diagramme de séquence technique "Soumettre un devoir"	27

MOODLE CLIENT ANALYSE ET CONCEPTION

III.1.5	Diagramme de séquence technique "Ajout de fichiers privés"	27
III.1.6	Diagramme de séquence technique "Noter un devoir"	28
III.2.	Diagramme de classes technique de l'étudiant.	29
III.3.	Diagramme d'état-transition de la classe « devoir ».....	29
III.4.	Diagramme d'activités de navigation d'un étudiant	30
III.5.	Diagramme d'activité « Modifier un Cours ».....	31
IV.	Implémentation.	31
IV.1.	Technologies utilisées.....	31
IV.2.	Architecture de l'application (Diagramme de déploiement)	32
Conclusion.	33

Table des figures.

Figure 1 Architecture logique.....	6
Figure 2 Diagramme de contexte	7
Figure 3 Diagramme de package	7
Figure 4 Diagramme de classes métier	8
Figure 5 Diagramme de cas d'utilisation de l'étudiant	9
Figure 6 Diagramme de cas d'utilisation de l'enseignant	10
Figure 7 Diagramme de séquence système « Se connecter ».....	20
Figure 8 Diagramme de séquence système « Noter un devoir »	21
Figure 9 Diagramme de séquence système « Modifier un cours »	21
Figure 10 Diagramme de séquence système « Consulter un cours »	22
Figure 11 Diagramme de séquence système « Soumettre un devoir »	22
Figure 12 Diagramme de séquence système « Se synchroniser »	23
Figure 13 Diagramme de séquence système « Ajouter fichiers privés »	23
Figure 14 Diagramme de séquence système « Inscrire un utilisateur à un cours »	24
Figure 15 Diagramme de séquence système « Désinscrire un utilisateur à un cours »	24
Figure 16 Diagramme de séquence technique « Synchroniser »	25
Figure 17 Diagramme de séquence technique « Consulter un cours »	26
Figure 18 Diagramme de séquence technique « Modifier un cours »	26
Figure 19 Diagramme de séquence technique « Soumettre un devoir »	27
Figure 20 Diagramme de séquence technique « Ajouter fichiers privés »	27
Figure 21 Diagramme de séquence technique « Noter un devoir »	28
Figure 22 Diagramme de classes technique.....	29
Figure 23 Diagramme d'état transition d'un devoir	29
Figure 24 Diagramme d'activités de navigation d'un étudiant	30
Figure 25 Diagramme d'activité « Modifier un Cours »	31
Figure 26 Diagramme de déploiement.....	32

Introduction.

L'E-Learning est une problématique actuelle, avoir accès à des formations de qualité juste à partir de sa connexion Internet et apprendre sur un domaine. Cependant, vu notre contexte, une connexion stable est un concept homérique et ne saurait être une base. C'est pourquoi il a fallu penser à des solutions pour pallier ce problème, notamment Moodle. En effet, on peut avoir une application cliente qui se synchronise avec le serveur lorsqu'il y a de la connexion Internet, comme ça pas besoin d'être constamment connecté. Toute cette optique s'inscrit dans le cadre du cours d'IHM où il nous a été confié la tâche de continuer le travail mené par nos aînés, via l'ajout de fonctionnalités dans une logique open source. Dans ce but, nous allons d'abord présenter le contexte du projet et les objectifs recherchés

Puis, nous allons faire son analyse à travers la présentation des exigences, de l'architecture et des différents diagrammes ; notamment de contexte, de package, de cas d'utilisation et de séquences système.

Ensuite la conception, à travers les diagrammes de séquences techniques, de classes techniques, d'état transition et d'activité.

Enfin, l'implémentation via les technologies utilisées et l'architecture de l'application représentée par son diagramme de déploiement.

I. Contexte et objectifs du projet

La plateforme d'E-Learning Moodle permet à ses utilisateurs de suivre des cours à distance, à condition d'être connecté au serveur, donc à Internet dans certains cas. Cette contrainte de connectivité peut drastiquement influencer et freiner l'évolution de l'apprentissage des utilisateurs. Ceci s'explique par l'instabilité, voire l'absence de connexion Internet dans certaines zones, ou alors le manque de moyens financiers de l'utilisateur pour se procurer une connexion Internet stable. Il se pose donc le problème d'accessibilité aux ressources fournies par la plateforme. Il serait donc idéal de pouvoir rendre ces ressources accessibles offline, ce qui permettrait aux utilisateurs de pouvoir continuer à suivre les cours sans connexion Internet. C'est ce que nous nous proposons de réaliser par le développement de la présente application. Par la suite nous désignerons par compte distant le compte de l'utilisateur sur le serveur Moodle, et par compte local le compte de l'utilisateur sur le client Moodle.

Les objectifs /fonctionnalités de l'application sont les suivants :

- Permettre à l'utilisateur de relier son compte sur un serveur Moodle à son application Moodle client.
- Permettre à l'utilisateur de synchroniser son compte local à son compte distant en récupérant des cours, des devoirs et soumettant des fichiers privés.
- Télécharger le contenu des cours suivis par l'utilisateur depuis le serveur Moodle et le rendre accessible sur le client Moodle.
- Permettre à l'utilisateur de réaliser des soumissions à des devoirs sur le client Moodle, qui seront par la suite uploadés sur le serveur Moodle lorsqu'il sera connecté à celui-ci.
- Permettre à l'utilisateur d'ajouter une photo de profil.

Ces fonctionnalités devront aussi être implémentées pour les profils enseignants de la plateforme, ce qui leur permettra de télécharger les cours qu'ils dispensent et de pouvoir les modifier en local sans besoin d'être connecté au serveur. Modifications qui seront par la suite uploadées (téléversées) au niveau du serveur.

II. Analyse

II.1. Exigences fonctionnelles et non fonctionnelles.

II.1.1. Exigences fonctionnelles

Le système doit permettre de :

- Voir l'état d'avancement de la synchronisation
- Signaler lorsque la synchronisation est terminée
- Créer/Modifier/Supprimer un devoir
- Inscrire et désinscrire des utilisateurs à un cours
- Lancer une synchronisation
- Créer un forum de discussion entre les étudiants et les enseignants
- Ajouter une photo de profil
- L'implémentation en ligne de la fonctionnalité créer et supprimer un cours en ligne
- Voir la liste des étudiants qui ont été ajouté à un cours

II.1.2. Exigences non-fonctionnelles

- L'application doit être desktop
- Chaque utilisateur ne doit avoir accès qu'à sa vue

II.2. Architecture logique de l'application.

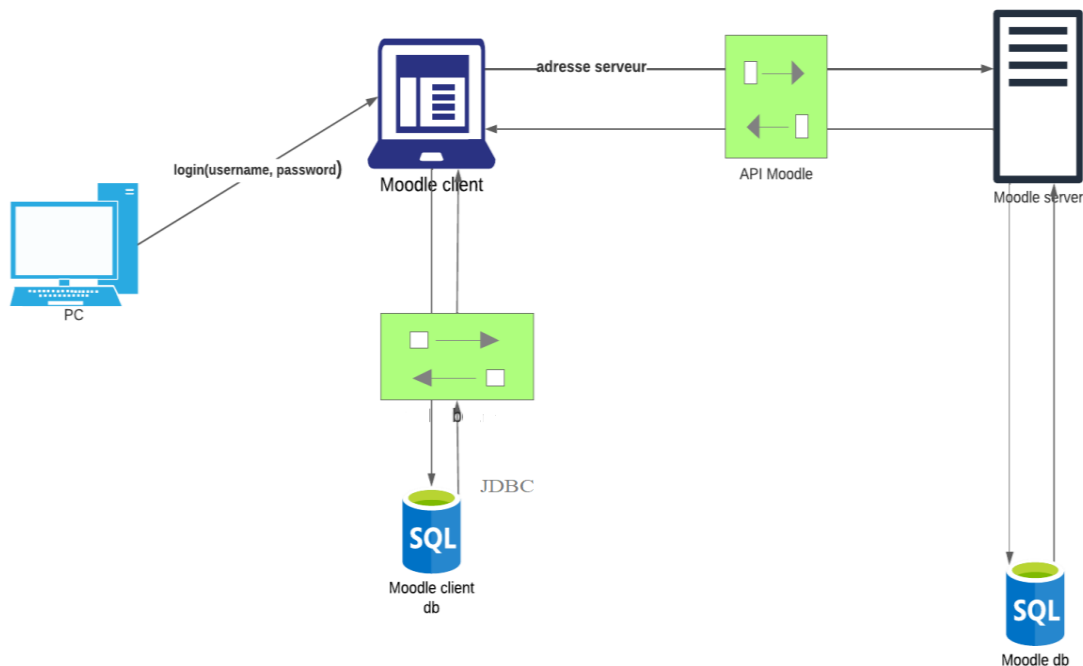


Figure 1 Architecture logique

II.3. Diagramme de contexte.

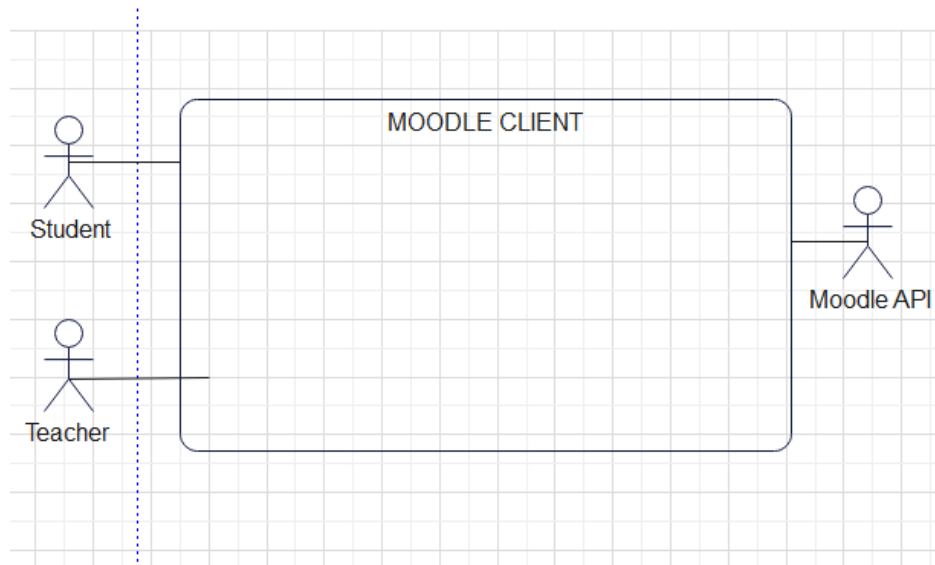


Figure 2 Diagramme de contexte

II.4. Diagramme de package.

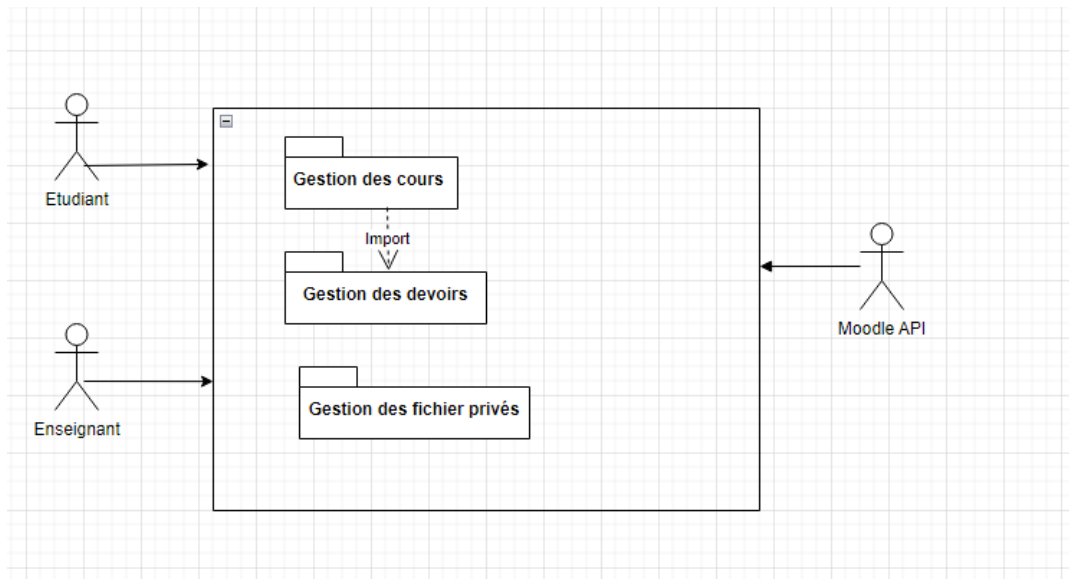


Figure 3 Diagramme de package

II.5. Diagramme de classes métier

Dans notre application, les entités que nous allons manipuler sont les suivantes : **user, course, section, submission, assignment, grade, file** et **module** comme le présente la figure 4.

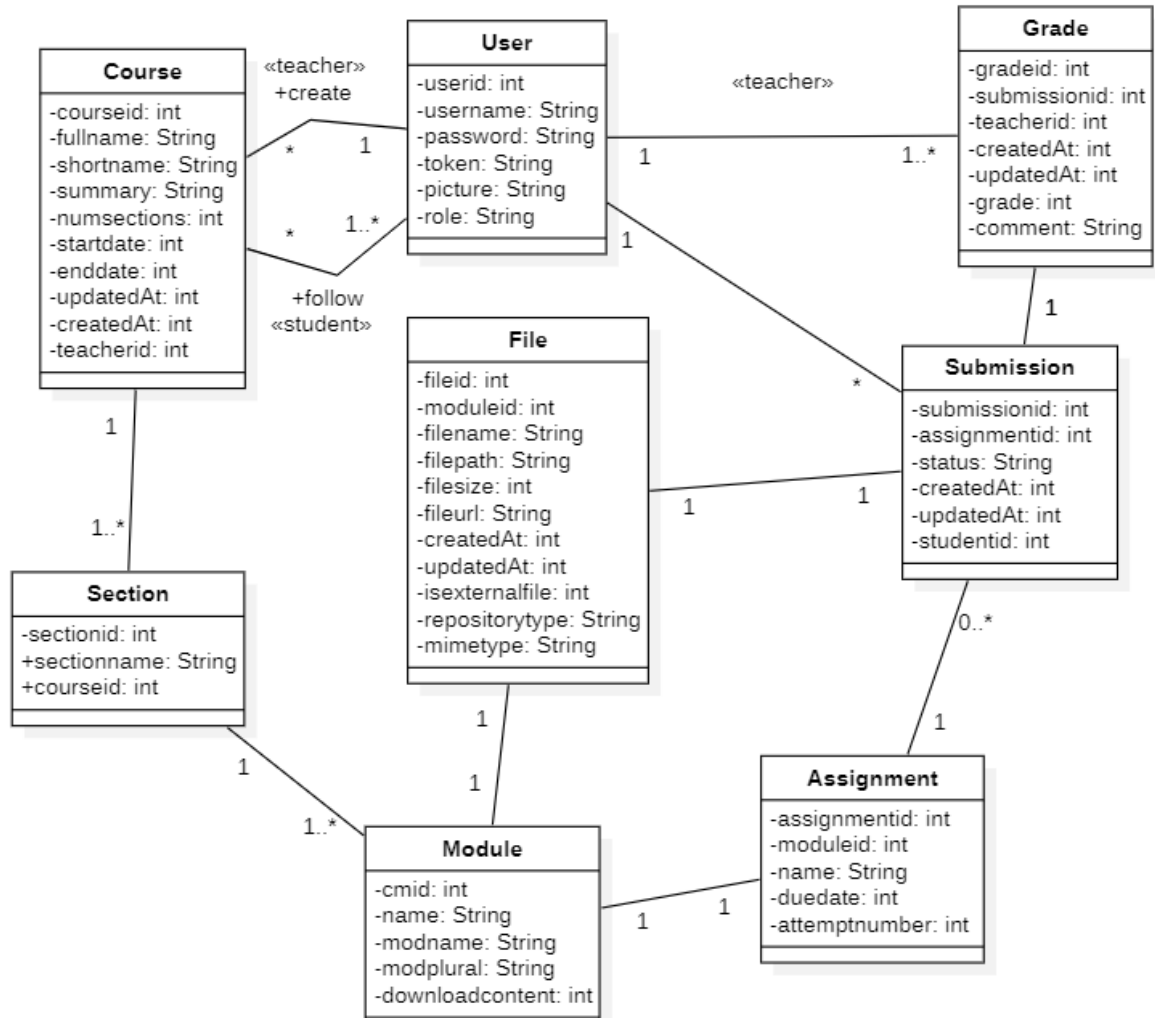


Figure 4 Diagramme de classes métier

II.6. Diagramme de cas d'utilisation.

Les cas d'utilisation communs à l'étudiant et à l'enseignant implémentés dans notre application sont :

- Se connecter
- Lier son compte distant à l'application
- Lancer la synchronisation
- Ajouter des fichiers privés

Les cas d'utilisation spécifiques à l'étudiant sont :

- Soumettre un devoir
- Consulter ses cours
- Ajouter une photo de profil

Les cas d'utilisation spécifiques à l'enseignant sont :

- Modifier un cours
- Noter un devoir
- Voir la liste des étudiants qu'il a ajouté à un cours
- Créer/supprimer un cours

★ Diagramme de cas d'utilisation de l'étudiant

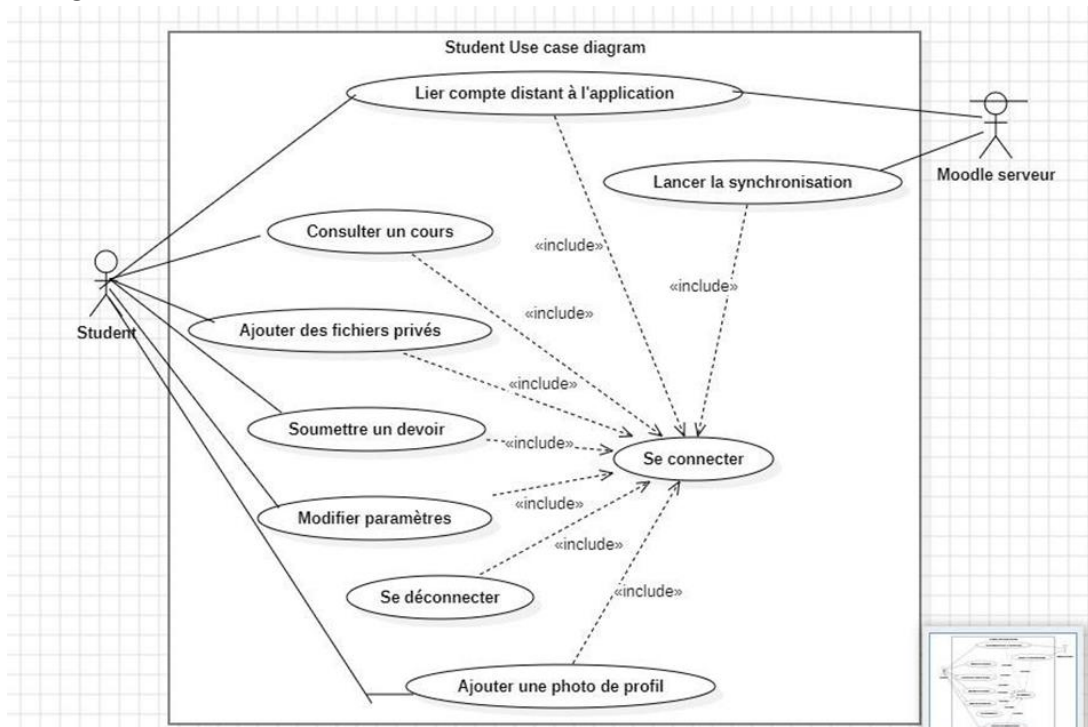


Figure 5 Diagramme de cas d'utilisation de l'étudiant

★ Diagramme de cas d'utilisation de l'enseignant

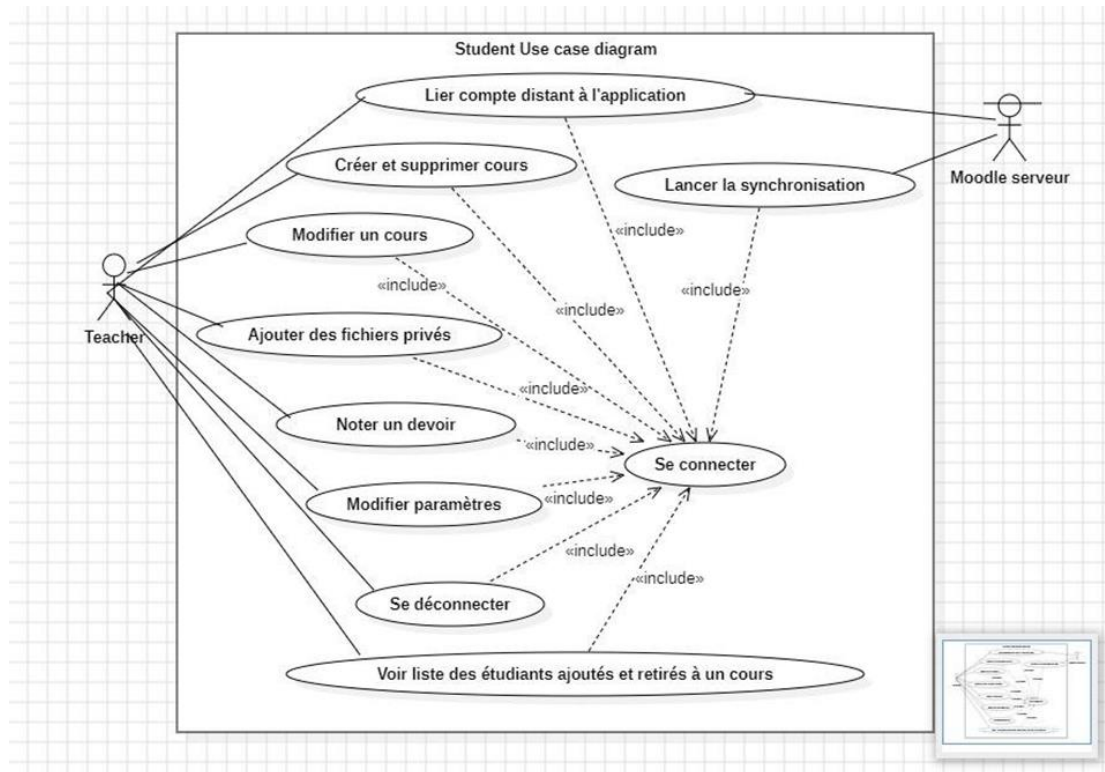


Figure 6 Diagramme de cas d'utilisation de l'enseignant

★ Description textuelle des cas d'utilisation

Nom du cas d'utilisation	Modifier paramètres
Objectif	<ul style="list-style-type: none"> - Changer le mode de synchronisation - Changer l'adresse du serveur Moodle
Acteurs concernés	<ul style="list-style-type: none"> - L'enseignant - L'étudiant

MOODLE CLIENT ANALYSE ET CONCEPTION

Précondition	Être authentifié dans l'application
Scénario nominal	<ol style="list-style-type: none">1. Accès au menu supérieur droit de l'application2. Choisir le paramètre à modifier3. Le système sauvegarde les modifications
Scénario alternatif	Aucun

MOODLE CLIENT ANALYSE ET CONCEPTION

Nom du cas d'utilisation	Se connecter
Objectif	<ul style="list-style-type: none">- Permettre à l'utilisateur de lier son application Moodle Client à son compte sur Moodle Serveur
Acteurs concernés	<ul style="list-style-type: none">- L'utilisateur- Moodle server API
Précondition	<ul style="list-style-type: none">- Être connecté à Internet- L'adresse du serveur Moodle doit être indiquée
Scénario nominal	<ol style="list-style-type: none">1. Le système affiche la page de login2. L'utilisateur remplit son username et mot de passe3. Le système vérifie la conformité des informations fournies4. Le système connecte l'utilisateur et lie son compte distant à l'application

MOODLE CLIENT ANALYSE ET CONCEPTION

Scénario alternatif	Informations incomplètes ou incorrectes : <ul style="list-style-type: none"> - Le système réaffiche la page de login - L'utilisateur corrige ses informations - Le cas d'utilisation reprend à l'action 3 du scénario nominal
---------------------	--

Nom du cas d'utilisation	Lancer la synchronisation
Objectif	Permettre que les modifications faites de part et d'autre puissent être téléchargées dans l'application ou dans Moodle serveur
Acteurs concernés	<ul style="list-style-type: none"> - L'utilisateur - Moodle server API
Precondition	<ul style="list-style-type: none"> - Être connecté à Internet - L'adresse du serveur Moodle doit être indiquée
Scénario nominal	1. L'utilisateur appuie sur le bouton de synchronisation 2. Téléchargement des ressources et des données sur Moodle Client 3. Upload des ressources et des données vers Moodle serveur
Scénario alternatif	Pas de connexion Internet ou erreur de serveur :

MOODLE CLIENT ANALYSE ET CONCEPTION

	<ul style="list-style-type: none">- L'utilisateur vérifie la connexion Internet- L'utilisateur corrige l'adresse du serveur dans les paramètres de l'application- Le cas d'utilisation reprend à l'action 1 du scénario nominal
--	---

Nom du cas d'utilisation	Consulter un cours
Objectif	Permettre à l'utilisateur d'avoir accès à son cours et les ressources correspondantes
Acteurs concernés	<ul style="list-style-type: none">- L'étudiant- L'enseignant
Précondition	Etre authentifié sur l'application
Scénario nominal	<ol style="list-style-type: none">1. Accès au dashboard de l'application2. Le système affiche la liste des cours auxquels l'étudiant est inscrit3. L'utilisateur choisi sur l'un d'eux4. Le système affiche les ressources correspondantes
Scénario alternatif	Aucun

Nom du cas d'utilisation	Soumettre un devoir
Objectif	Permet à l'étudiant de rendre un devoir dans l'application étant hors connexion
Acteur concerné	L'étudiant
Précondition	Etre authentifié sur l'application
Scénario nominal	1.L'utilisateur choisit le devoir 2. Le système affiche le formulaire de Soumission 3. L'utilisateur téléverse ses fichiers 4. Le système les sauvegarde en attendant la synchronisation
Scénario alternatif	Aucun

Nom du cas d'utilisation	Ajouter une photo de profil
Objectif	Permettre à l'utilisateur d'ajouter une photo de son choix qui pourrait permettre à l'étudiant de l'identifier ou non

MOODLE CLIENT ANALYSE ET CONCEPTION

Acteur concerné	L'étudiant
Précondition	Avoir un compte
Scénario nominal	L'utilisateur se connecte puis ajoute sa photo de profil
Scénario alternatif	Aucun

Nom du cas d'utilisation	Voir la liste des étudiants connectés à un cours
Objectif	Permettre à l'enseignant de voir les étudiants qu'il a ajouté à son cours
Acteur concerné	L'enseignant
Précondition	Etre authentifié sur l'application
Scénario nominal	1. Accès au dashboard de l'application 2. Le système affiche le cours de l'enseignant 3.L'enseignant peut à présent voir la liste des étudiants qu'il a ajouté à son cours
Scénario alternatif	Aucun

Nom du cas d'utilisation	Ajouter des fichiers privés
Objectif	Permet à l'utilisateur d'ajouter des fichiers qu'il pourra consulter ou utiliser à sa guise dans l'application
Acteurs concernés	<ul style="list-style-type: none">- L'enseignant- L'étudiant
Précondition	Etre authentifié sur l'application
Scénario nominal	<ol style="list-style-type: none">1. Accès à la barre de menu gauche de L'application2. Cliquer sur « Private Files »3. L'application affiche la page de soumission des fichiers4. L'utilisateur soumet les fichiers et clique sur « Save changes »5. Le système sauvegarde les fichiers en attendant la synchronisation
Scénario alternatif	Aucun

MOODLE CLIENT ANALYSE ET CONCEPTION

Nom du cas d'utilisation	Noter un devoir
Objectif	Permettre à un enseignant de corriger les devoirs soumis relatifs à son cours hors connexion
Acteur concerné	L'enseignant
Precondition	Etre authentifié sur l'application
Scénario nominal	<ol style="list-style-type: none">1. L'utilisateur choisi parmi la liste des devoirs2. Le système affiche le formulaire de correction3. L'utilisateur note le devoir4. Cliquer sur « Sauvegarder »5. Sauvegarde par le système en attendant la synchronisation
Scénario alternatif	Aucun

Nom du cas d'utilisation	Modifier un cours
Objectif	Permettre à l'enseignant d'ajouter, de supprimer des ressources concernant un cours

MOODLE CLIENT ANALYSE ET CONCEPTION

Acteur concerné	L'enseignant
Précondition	Etre authentifié sur l'application
Scenario nominale	<ol style="list-style-type: none">1. L'utilisateur choisi parmi la liste des cours2. Le système affiche les ressources dédiées3. L'utilisateur supprime ou ajoute et modifie des fichiers4. Cliquer sur « Sauvegarder »5. Sauvegarde par le système en attendant la synchronisation
Scénario alternatif	Aucun

Nom du cas d'utilisation	Se déconnecter
Objectif	Permet à l'utilisateur de délier son application à son compte distant sur Moodle serveur
Acteurs concernés	<ul style="list-style-type: none">- L'étudiant- L'enseignant
Précondition	Etre authentifié sur l'application
Scénario nominal	<ol style="list-style-type: none">1. Accès au menu supérieur droit de

	<p>L'application</p> <p>2. Cliquer sur « Log out »</p> <p>3. L'application renvoie automatiquement la page de login</p>
Scénario alternatif	Aucun

II.7. Diagrammes de séquence système.

II.7.1. Diagramme de séquence système "Se connecter"

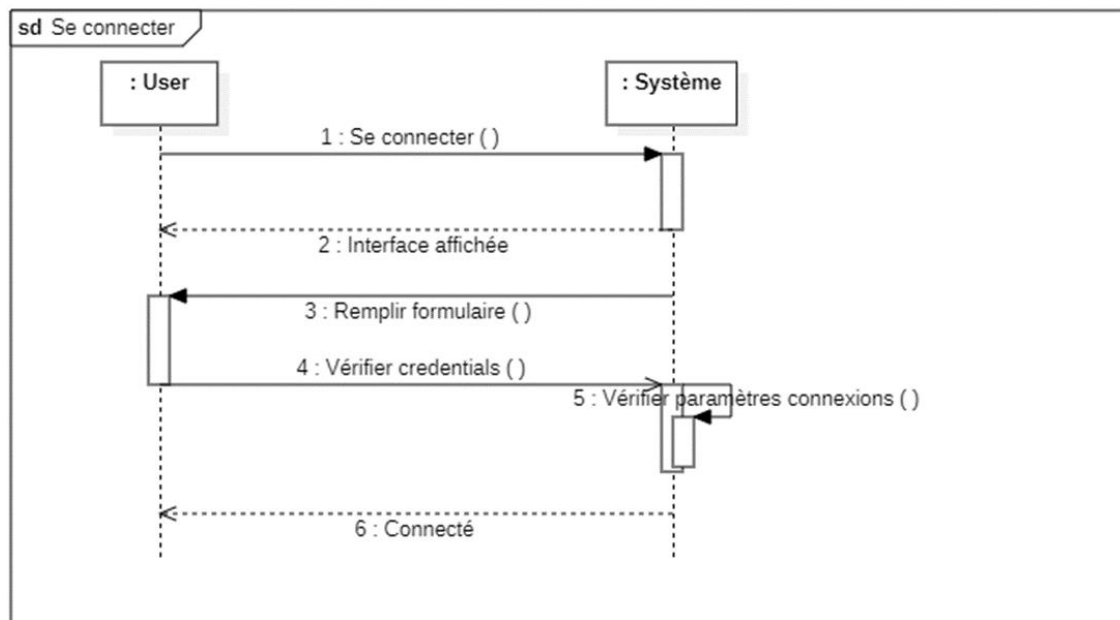


Figure 7 Diagramme de séquence système « Se connecter »

II.7.2. Diagramme de séquence système "Noter un devoir"

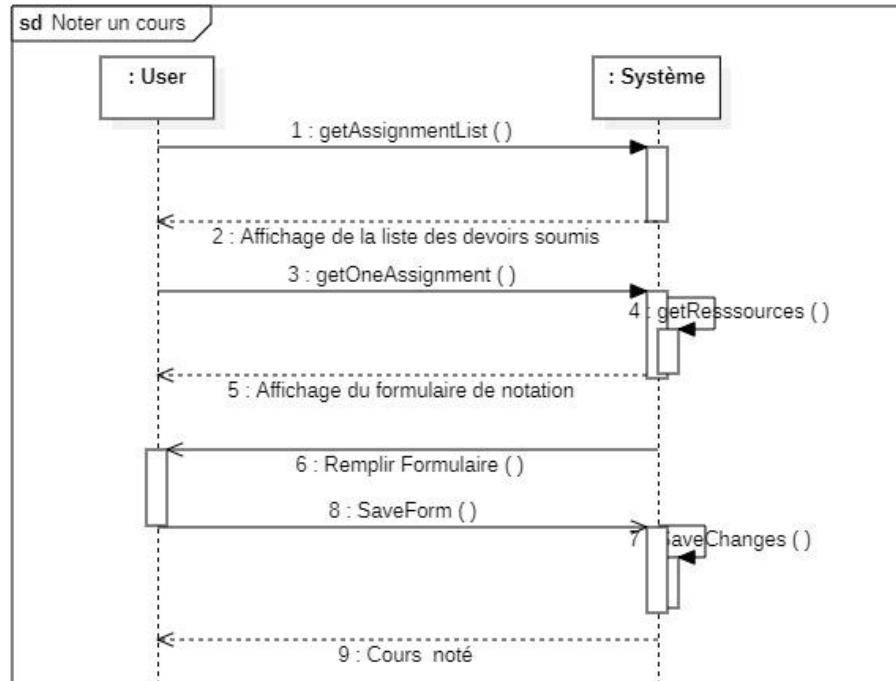


Figure 8 Diagramme de séquence système « Noter un devoir »

II.7.3. Diagramme de séquence système "Modifier un cours"

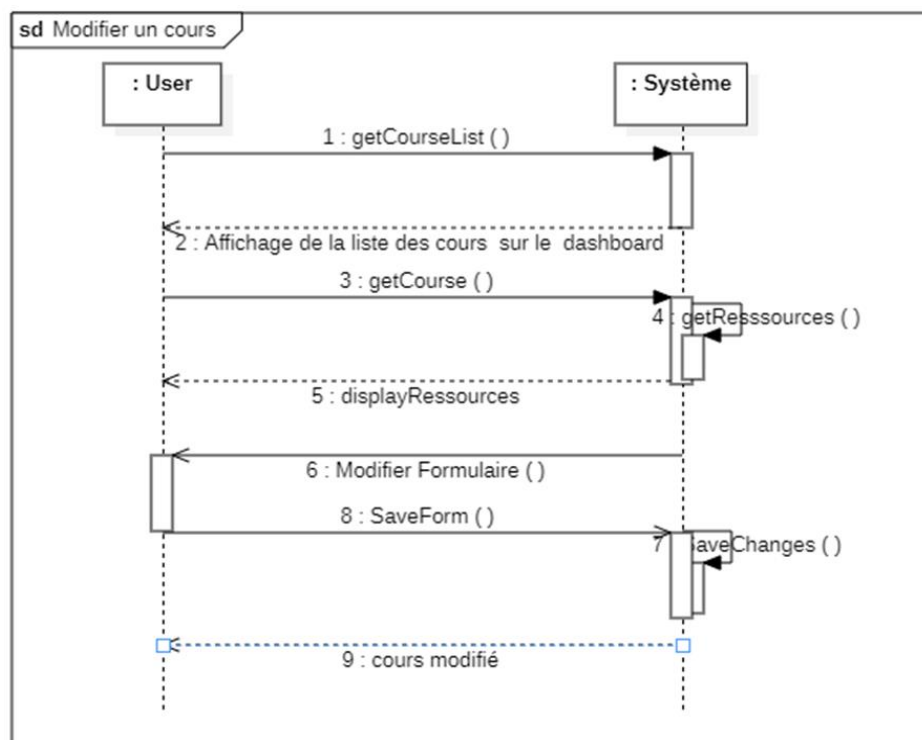


Figure 9 Diagramme de séquence système « Modifier un cours »

II.7.4. Diagramme de séquence système "Consulter un cours"

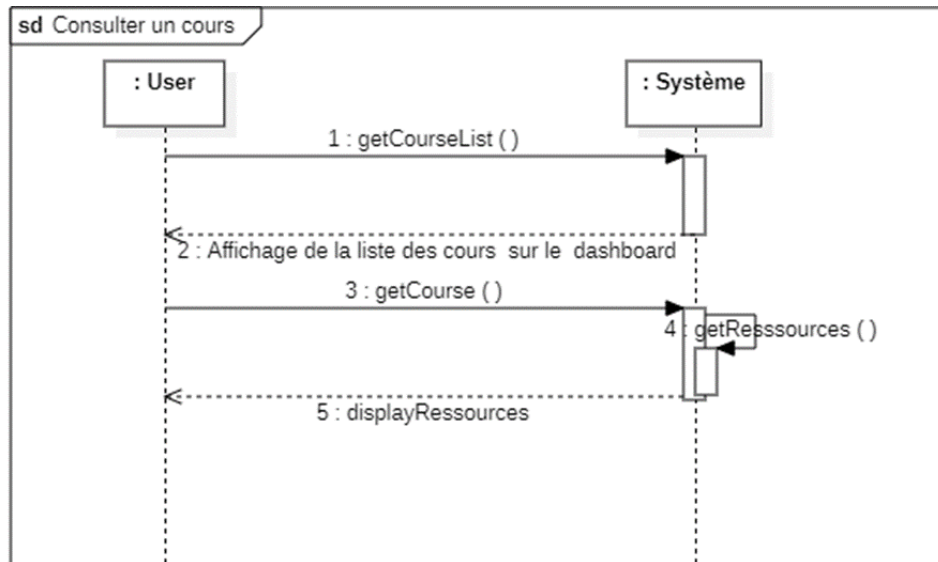


Figure 10 Diagramme de séquence système « Consulter un cours »

II.7.5 Diagramme de séquence système "Soumettre un devoir"

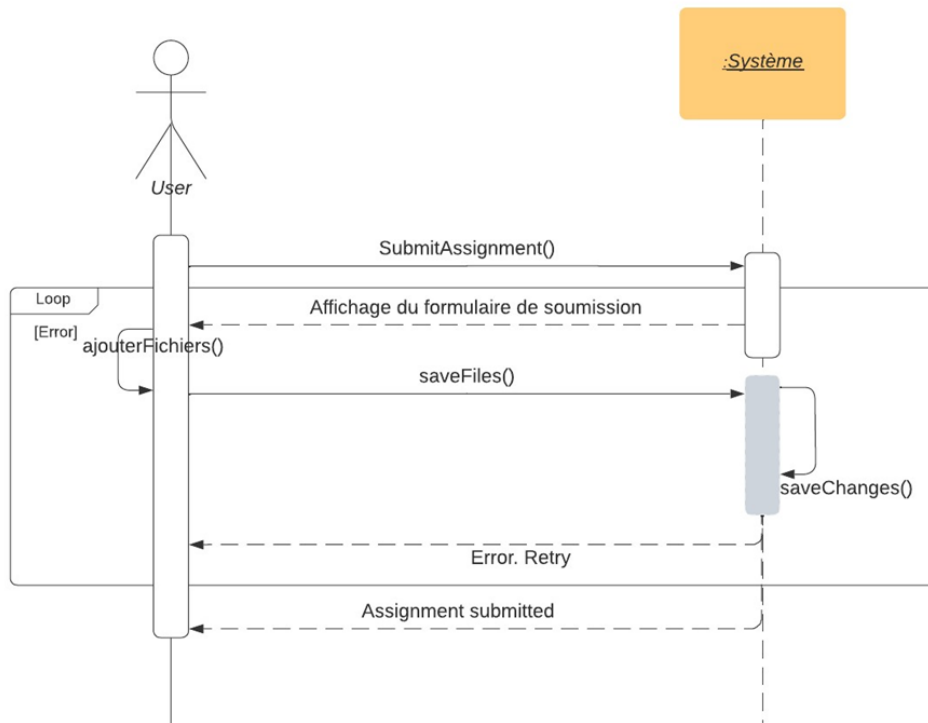


Figure 11 Diagramme de séquence système « Soumettre un devoir »

II.7.6. Diagramme de séquence système "Se synchroniser"

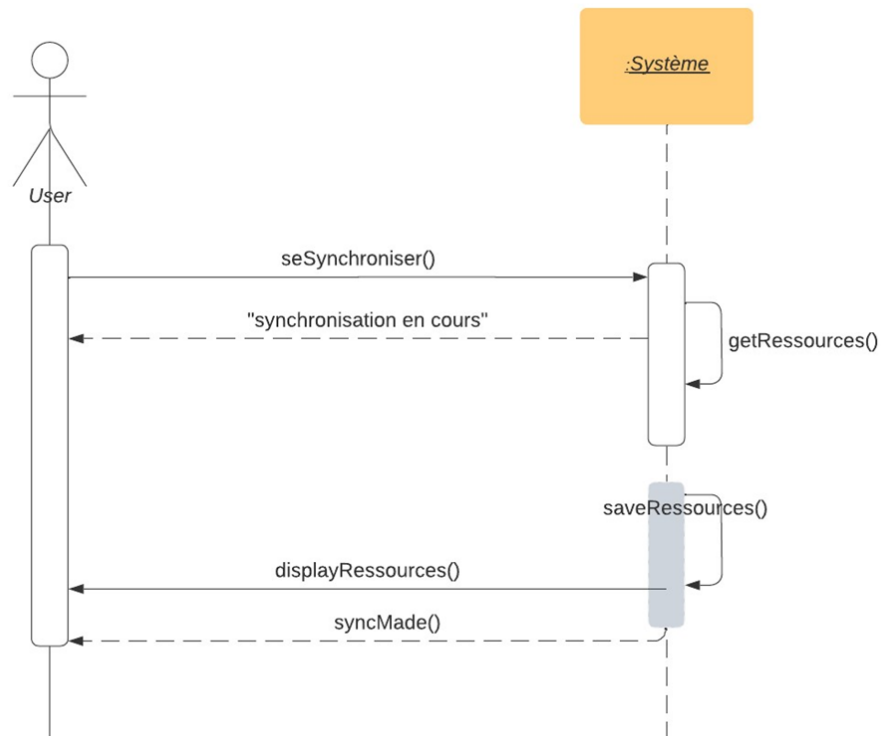


Figure 12 Diagramme de séquence système « Se synchroniser »

II.7.7. Diagramme de séquence système "Ajout de fichiers privés"

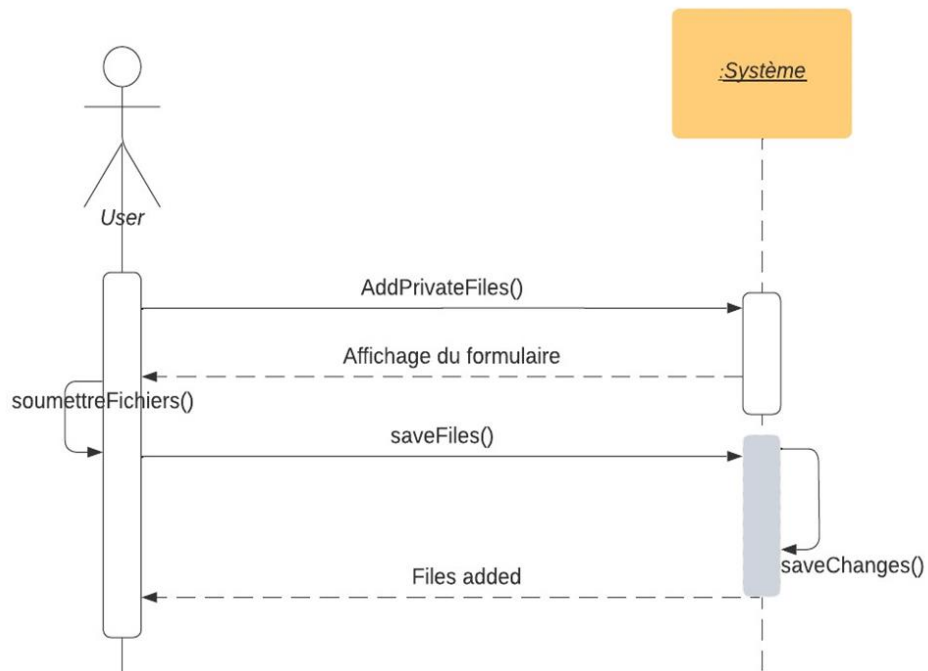


Figure 13 Diagramme de séquence système « Ajouter fichiers privés »

II.7.8. Diagramme de séquence système "Inscrire un utilisateur à un cours"

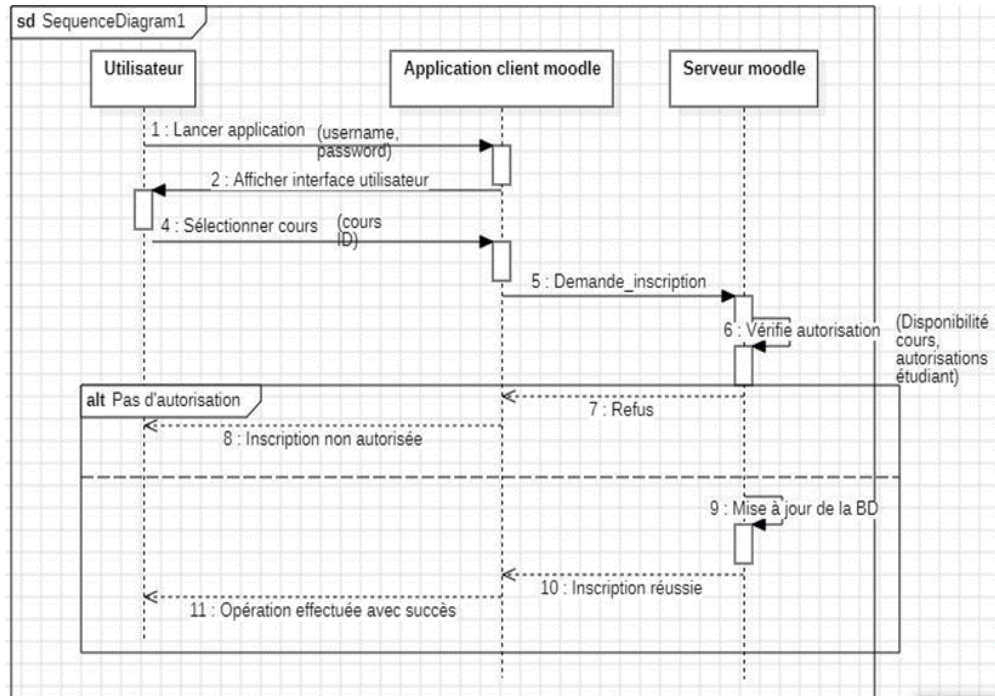


Figure 14 Diagramme de séquence système « Inscrire un utilisateur à un cours »

II.7.9. Diagramme de séquence système "Désinscrire un utilisateur à un cours"

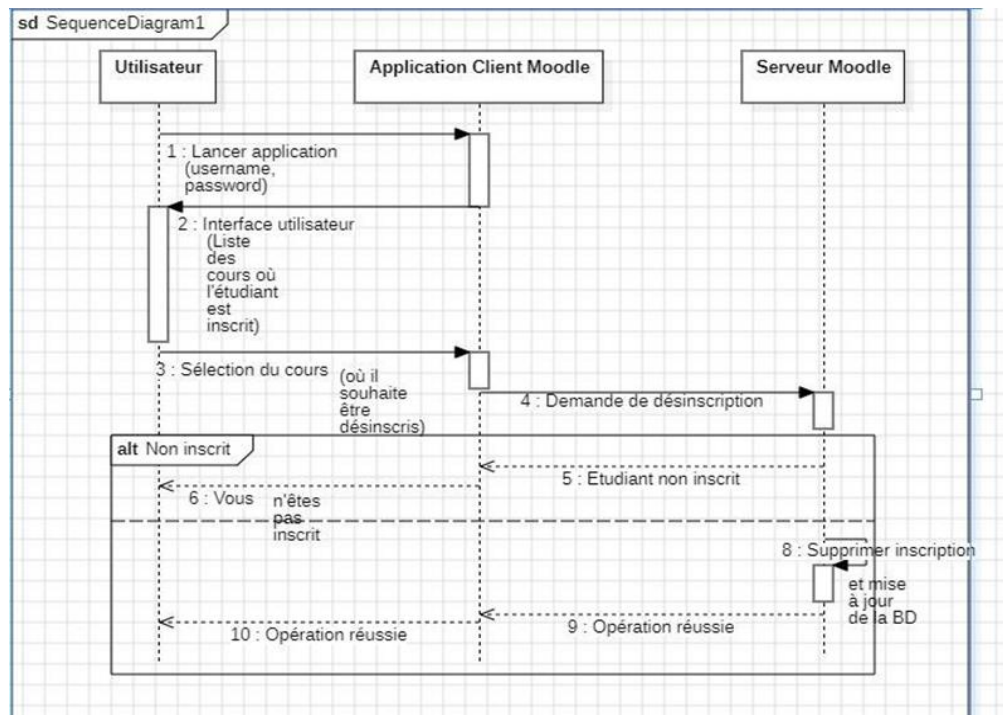


Figure 15 Diagramme de séquence système « Désinscrire un utilisateur à un cours »

III. Conception.

III.1. Diagrammes de séquence technique.

III.1.1 Diagramme de séquence technique "Synchroniser"

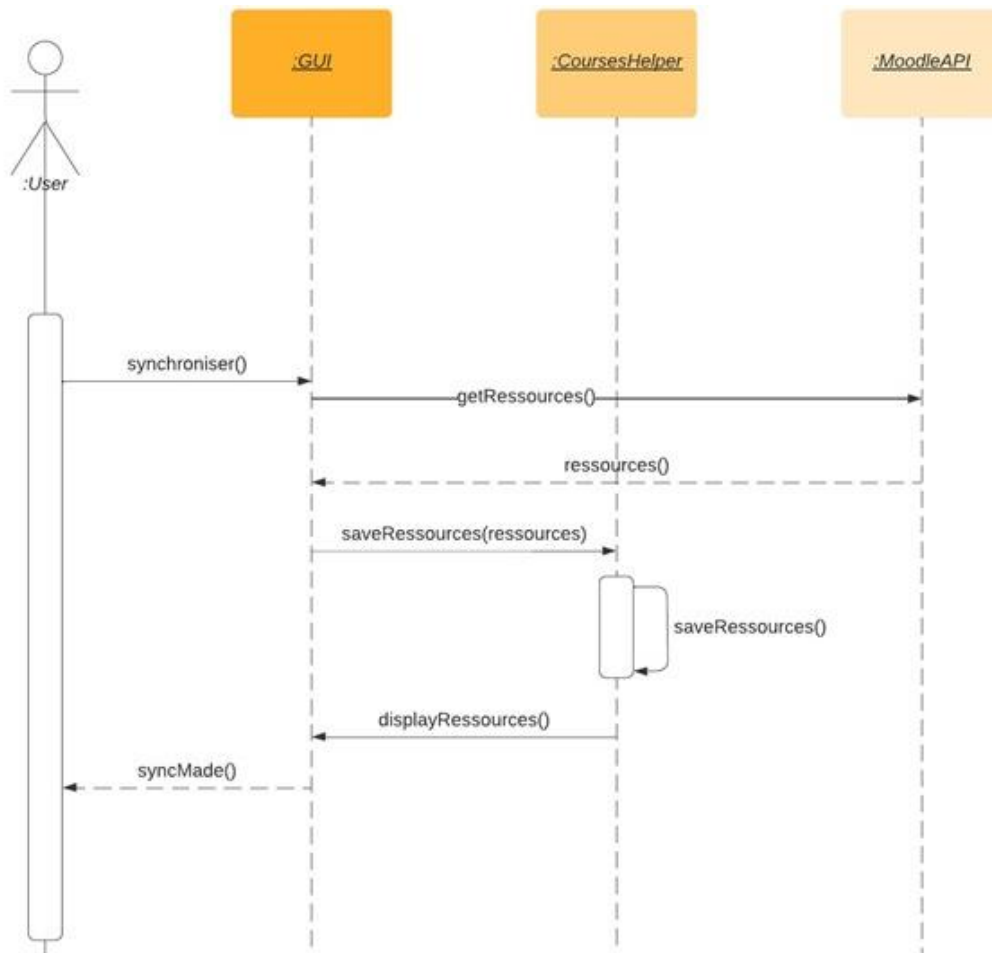


Figure 16 Diagramme de séquence technique « Synchroniser »

III.1.2. Diagramme de séquence technique "Consulter un cours"

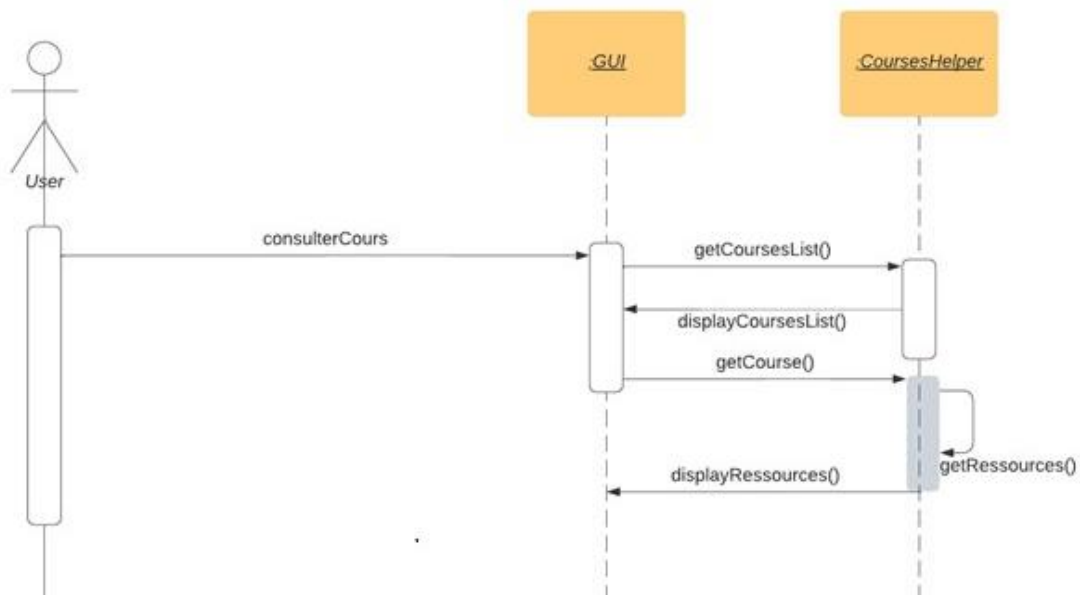


Figure 17 Diagramme de séquence technique « Consulter un cours »

III.1.3. Diagramme de séquence technique "Modifier un cours"

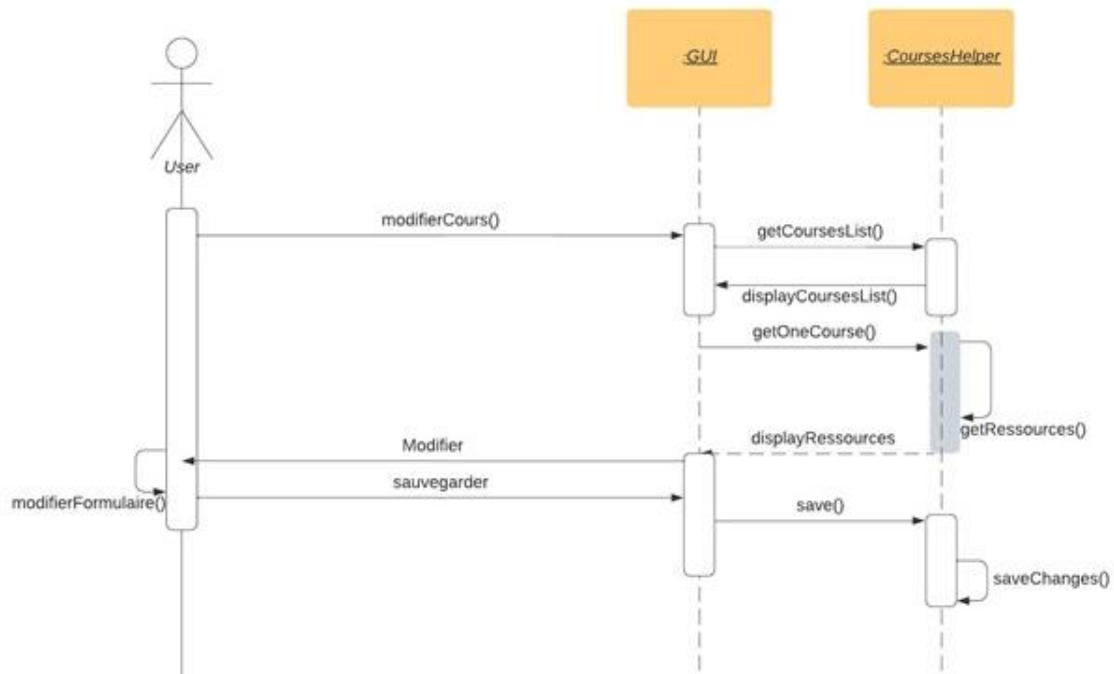


Figure 18 Diagramme de séquence technique « Modifier un cours »

III.1.4 Diagramme de séquence technique "Soumettre un devoir"

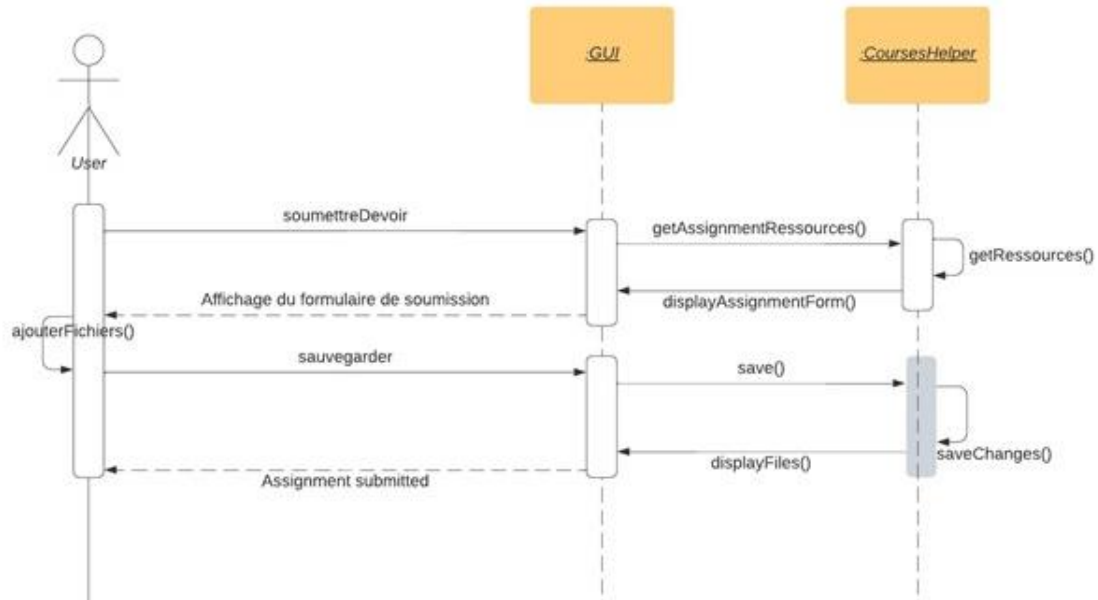


Figure 19 Diagramme de séquence technique « Soumettre un devoir »

III.1.5 Diagramme de séquence technique "Ajout de fichiers privés"

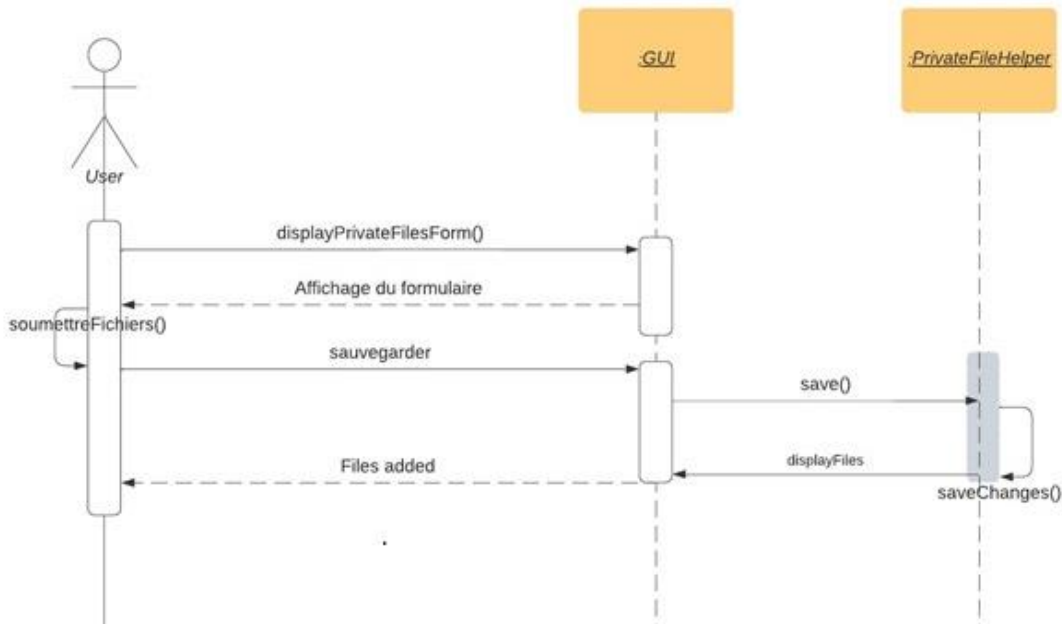


Figure 20 Diagramme de séquence technique « Ajouter fichiers privés »

III.1.6 Diagramme de séquence technique "Noter un devoir"

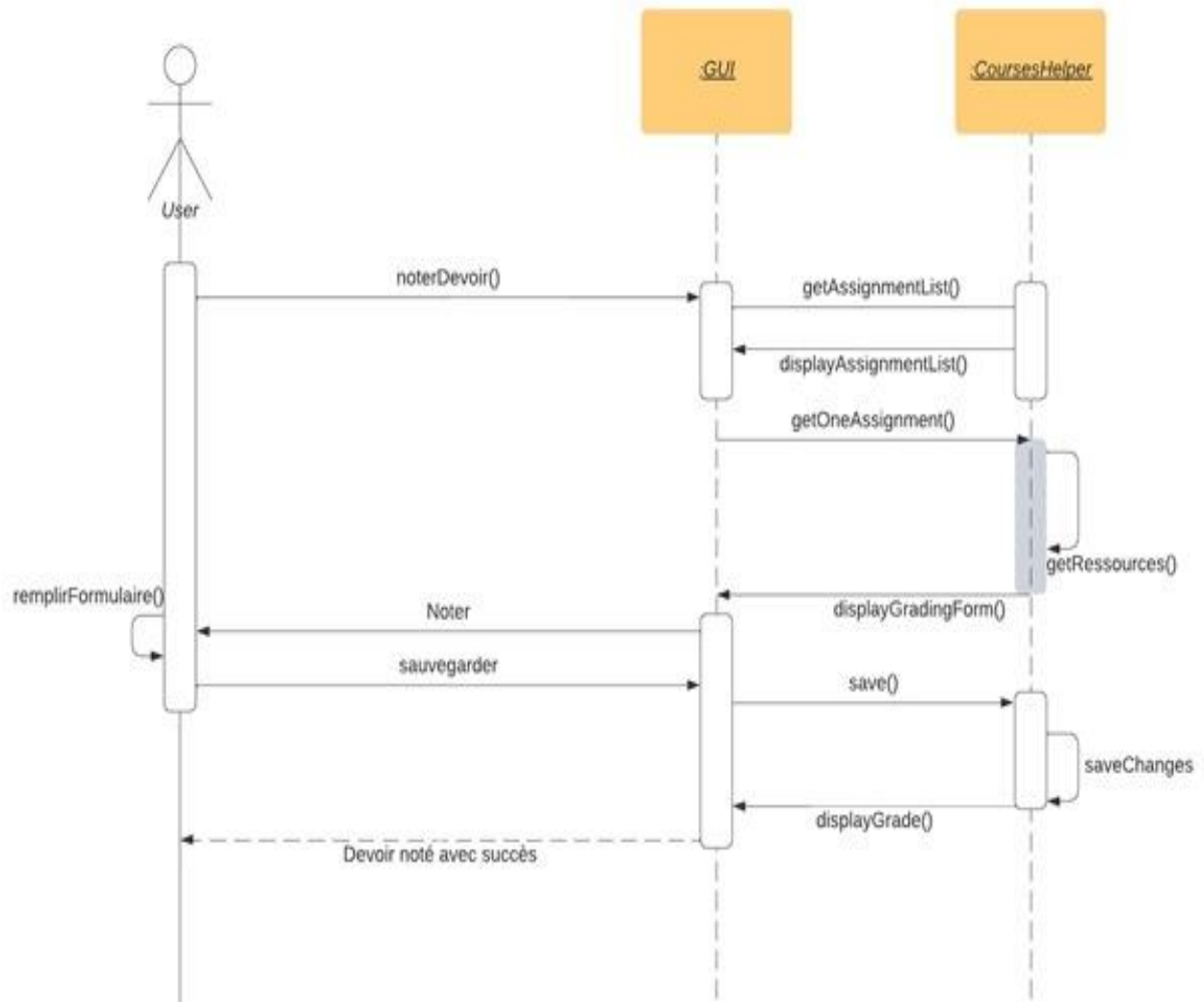


Figure 21 Diagramme de séquence technique « Noter un devoir »

III.2. Diagramme de classes technique de l'étudiant.

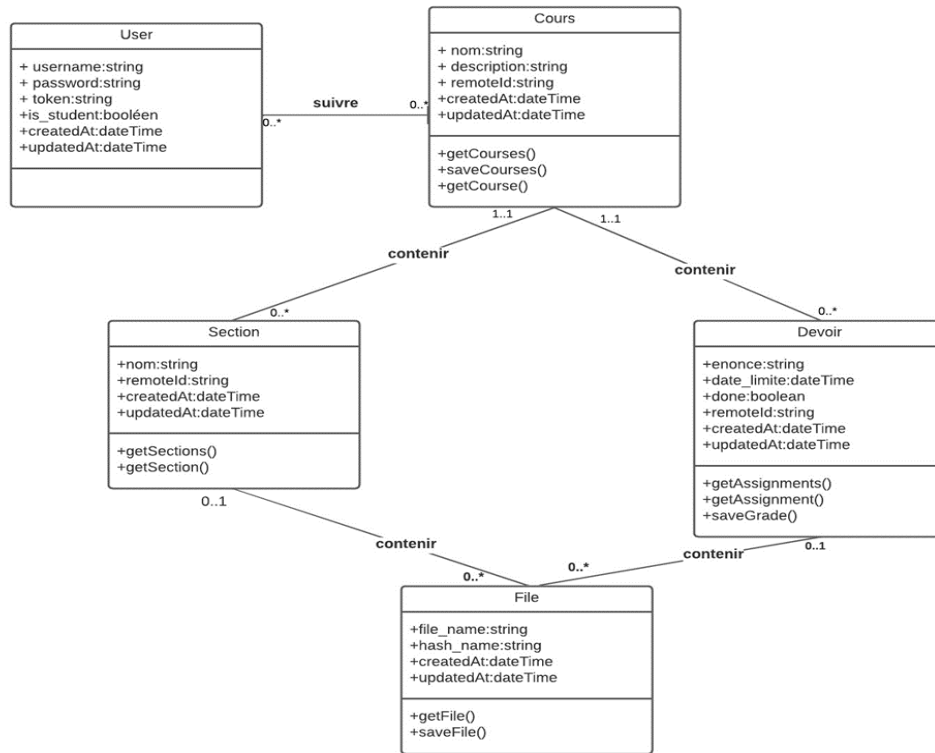


Figure 22 Diagramme de classes technique

III.3. Diagramme d'état-transition de la classe « devoir ».

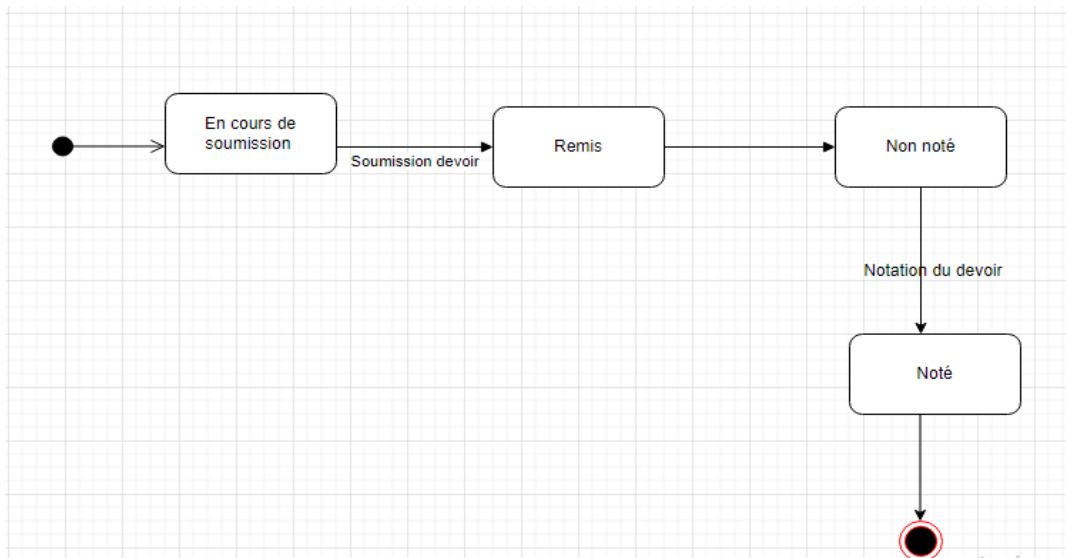


Figure 23 Diagramme d'état transition d'un devoir

III.4. Diagramme d'activités de navigation d'un étudiant

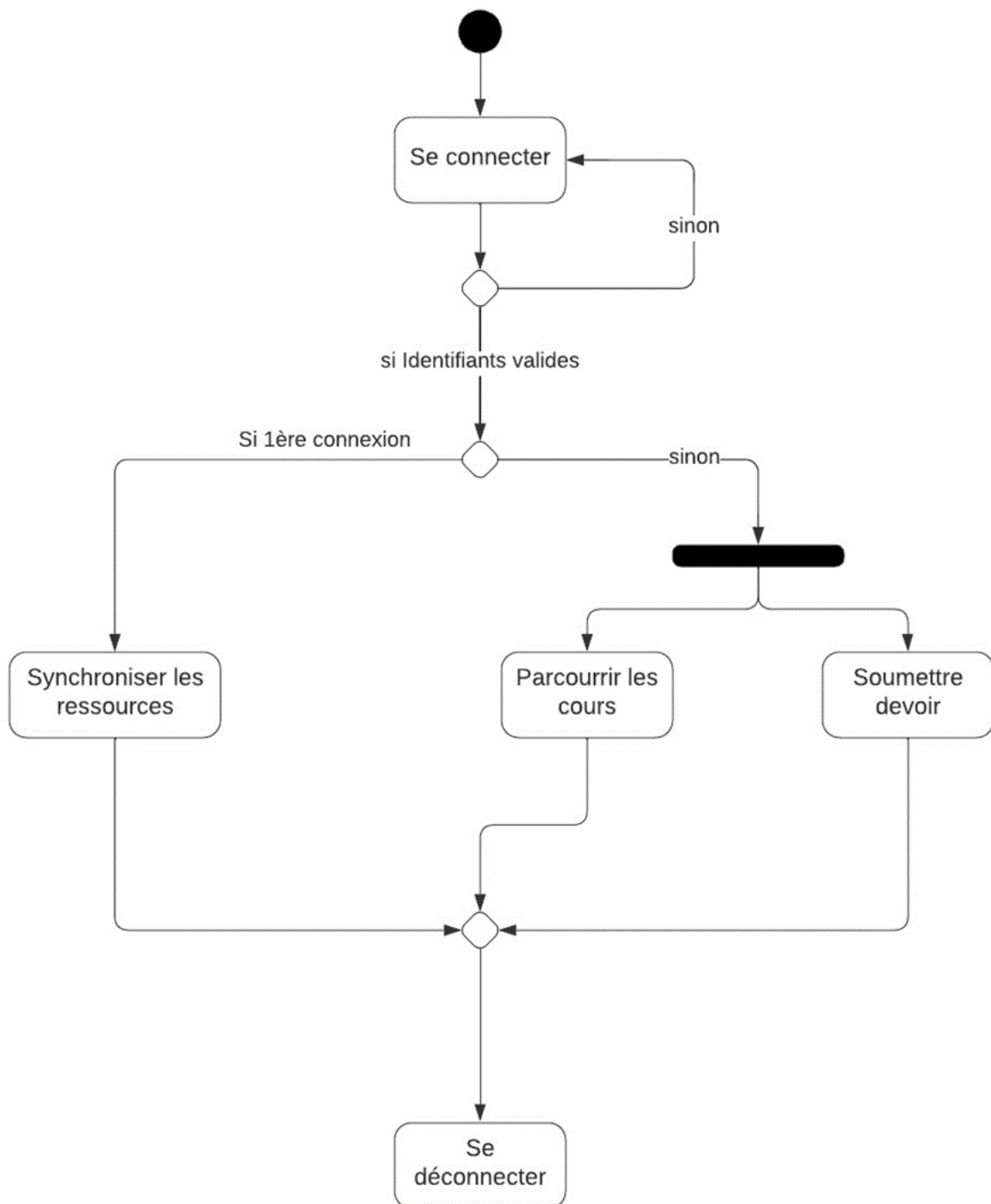


Figure 24 Diagramme d'activités de navigation d'un étudiant

III.5. Diagramme d'activité « Modifier un Cours »

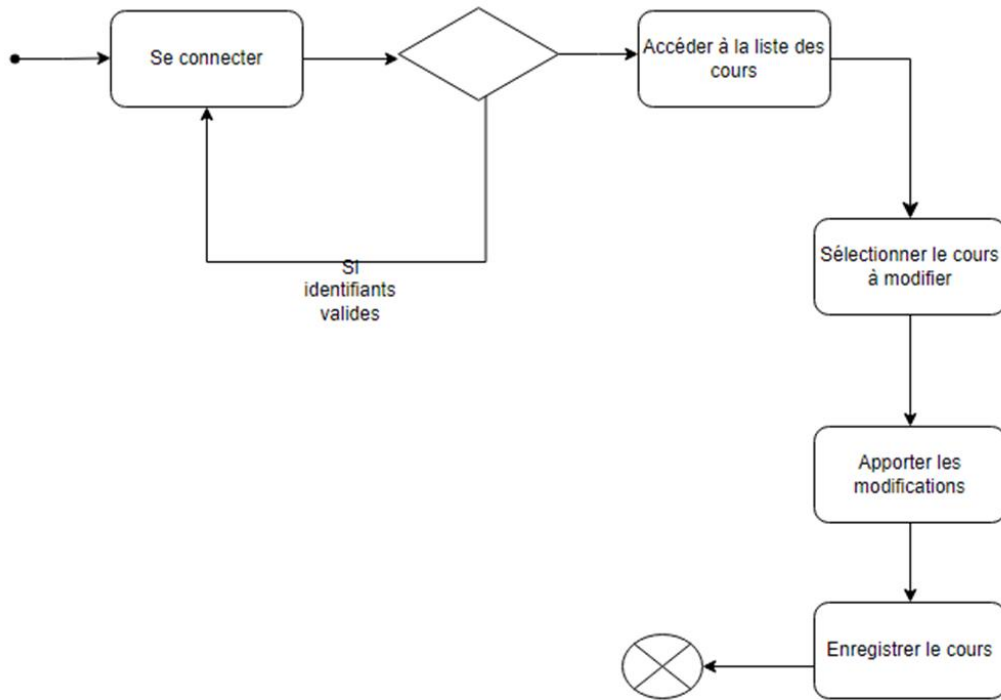


Figure 25 Diagramme d'activité « Modifier un Cours »

IV. Implémentation.

IV.1. Technologies utilisées.

Pour la réalisation de ce projet nous avons utilisé

- Base de données MySQL au niveau du serveur Moodle
- JavaFX pour les interfaces du client Moodle
- Base de données MySQL pour le client Moodle
- JDBC pour la communication entre l'application et la base de données Moodle client

IV.2. Architecture de l'application (Diagramme de déploiement)

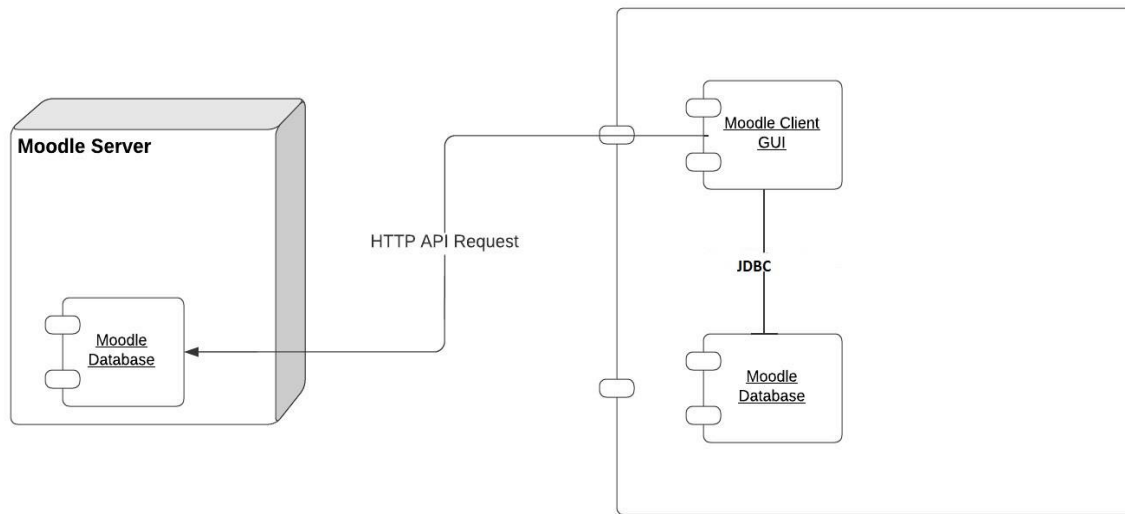


Figure 26 Diagramme de déploiement

L'application Moodle Client a pour composants majeurs les interfaces utilisateurs qui communiquent avec sa base de données locale via JDBC. Cette base de données stocke tous les fichiers et données relatives à l'utilisateur. Ces données proviennent de la base de données de Moodle Serveur ; données accessibles à partir des requêtes faites sur l'API de Moodle par l'application lors de la synchronisation.

Conclusion.

En conclusion, il était question de présenter l'analyse, la conception et l'implémentation du projet en plus de fonctionnalités ajoutées car il n'y avait pas besoin de s'étendre sur les fonctionnalités préalablement implémentées.

Ainsi, nous avons ajouté les fonctionnalités d'ajout d'avatar, d'inscription à un cours, de désinscription à un cours et du statut de synchronisation ; mais cela n'a pas été de tout repos.

En effet, le premier challenge rencontré fut la compilation qui posa problème chez la plupart d'entre nous, malgré le fait que nous disposions de tous les prérequis. On a dû se borner à l'environnement prédéfini dans un but de continuité et avoir une compréhension profonde du code pour savoir où effectuer des modifications. Après ajout, il était aussi question de vérifier le fonctionnement global pour toujours avoir un projet qui marche.