

RAPPORT MINI-PROJET JAVA

DÉVELOPPEMENT D'UNE
APPLICATION DESKTOP POUR LA
GESTION D'UNE SALLE DU SPORT

Encadré par :
Mr. WALID

Présenté par:

- OURICH Outman
- AIT ABIDALLA Ayoub
- ACHIBANE Akram

Introduction générale

Dans les salles de sport, avoir une application de gestion dédiée est essentielle. Elle permet de centraliser les informations, d'automatiser les tâches administratives et de simplifier les procédures. Elle facilite la prise de décision et réduit les erreurs humaines.

L'importance de cette application réside dans sa capacité à simplifier et automatiser les tâches de gestion souvent fastidieuses et chronophages. En offrant une interface conviviale et intuitive, elle permettra aux administrateurs de gérer tous les aspects clés de la salle de sport de manière plus efficace, optimisant ainsi le temps et les ressources.

La gestion des clients sera grandement simplifiée grâce à cette application. Elle permettra de stocker toutes les informations relatives aux membres, telles que les coordonnées personnelles, les informations médicales, les horaires d'entraînement et les historiques de paiement.

De plus, cette application offrira une gestion efficace des équipements de la salle de sport. Elle permettra de tenir un inventaire précis des équipements disponibles, de planifier leur maintenance régulière, de gérer les réservations et de suivre leur état général. Grâce à ces fonctionnalités, les administrateurs pourront optimiser l'utilisation des équipements, éviter les surcharges ou les pénuries, et ainsi offrir une meilleure expérience aux clients.

Enfin, la gestion des abonnements sera automatisée et simplifiée grâce à cette application. Elle permettra de gérer les différentes formules d'abonnement, de suivre les paiements et les échéances, d'envoyer des rappels aux clients et de générer des rapports détaillés sur les revenus générés par les abonnements. Cette fonctionnalité facilitera la gestion financière de la salle de sport, assurant ainsi une meilleure rentabilité.

En résumé, une application de gestion pour les salles de sport offre des avantages tangibles en termes de personnalisation du service, d'efficacité opérationnelle et de pérennité commerciale.

Chapitre I :

PRÉSENTATION DU PROJET

1. Introduction :

Dans ce chapitre, On détermine le cahier de charge et les objectifs à atteindre de ce projet. Ainsi, on intéresse à présenter les outils utilisés dans la réalisation de ce projet.

2. Cahier des charges et objectifs :

a- Problématique :

Une **salle de sport** souhaite informatiser la gestion de ses activités dans le but d'améliorer le rendement du personnel. Cette informatisation vise à améliorer le rendement du personnel en simplifiant les procédures d'abonnement et d'accès pour les membres. Cette application vise à réduire l'utilisation du papier en favorisant l'utilisation de données numériques. Cela permettra également aux administrateurs d'effectuer des analyses statistiques en exploitant ces données. L'objectif final est d'optimiser les opérations de la salle de sport et maximiser les profits et l'efficacité globale de l'établissement.

b- Fonctionnalités :

Dans ce but, on se propose dans ce mini-projet d'**analyser**, faire la **conception** et **réaliser** une application desktop permettant de gérer les activités de cette salle de sport. Notre Application Java devra proposer à **l'administrateur** un menu permettant de naviguer dans les différentes fonctionnalités offertes par l'application :

➤ Gestion des clients :

- Ajouter un nouveau client avec toutes ses informations qui vous semblent utiles (code_client, nom complet, Adresse, N° tél,...etc.);
- Préciser le type d'abonnement qu'il a choisi (mensuel, trimestriel, annuel).
- Modifier, ou supprimer un client existant spécifié par un

nom.

- Afficher la liste des clients ordonnée par la date d'inscription.
- Rechercher un client par son Id et afficher l'enregistrement correspondant.
- Retour au menu principal.

➤ **Gestion des paiements :**

- Ajouter un nouveau paiement (en cas d'une nouvelle inscription d'un nouveau client, et en cas de renouvellement d'abonnement).
- Affichez les paiements en cours, c'est-à-dire les abonnements qui n'ont pas encore expiré.
- Affichez l'historique des abonnements, c'est-à-dire la liste des abonnements passés.
- Retour au menu principal.

➤ **Gestion des entraineurs :**

- Ajouter un nouveau entraineur avec toutes ses informations.
- Modifier, ou supprimer un entraineur existant spécifié par un nom.
- Spécifier, pour chaque entraineur un Planning associé.
- Afficher la liste des entraineurs ordonnée par la date de création.
- Rechercher un entraineur par son Id et afficher l'enregistrement correspondant.
- Retour au menu principal.

➤ **Gestion des équipements :**

- Ajouter un nouveau équipement avec toutes ses informations qui vous semblent utiles (code, description, status(bonne état, mauvaise état...)).
- Supprimer un équipement.
- Afficher la liste des équipement.
- Réclamer un équipement au propriétaire si son état est mauvais.

- Retour au menu principal.

Cette application doit offrir au propriétaire le même menu que celui de l'administrateur, ainsi que d'autres fonctionnalités supplémentaires.

➤ **Gérer les administrateurs :**

- Veuillez ajouter un nouvel administrateur en fournissant toutes les informations pertinentes (ID_admin, nom complet, adresse, numéro de téléphone, etc.), et n'oubliez pas d'inclure **un login** et un **mot de passe** pour accéder à la plateforme.
- Modifier, ou supprimer un administrateur existant spécifié par un nom.
- Afficher la liste des administrateurs ordonnée par la date d'inscription.
- Veuillez effectuer une recherche d'administrateur en utilisant son ID, puis afficher l'enregistrement correspondant.
- Retour au menu principal.

➤ **Consultation du bilan :**

- Permettra de consulter le nombre de membres et de coaches, de connaître les équipements endommagés, ainsi que d'obtenir une idée du chiffre d'affaires réalisé.
- Cette fonctionnalité permet de visualiser des graphiques qui affichent le pourcentage en fonction du genre des membres de la salle de sport, ainsi que selon le type d'abonnement. Ces graphiques offrent une représentation visuelle des données pour une meilleure compréhension des répartitions.

3- Outils et logiciels utilisés :

a- Logiciels :

Lors du développement de cette application, on a utilisé les logiciels suivants:

- Eclipse.
- Astah.
- XAMPP.
- PHPMyAdmin.

on va présenter ces différents logiciels dans la section suivante :

➤ **Eclipse :**

Eclipse est un environnement de développement intégré (IDE) largement utilisé dans le domaine de la programmation. Son rôle principal est de fournir un ensemble d'outils et de fonctionnalités pour faciliter la création, la modification et le débogage de logiciels. En tant qu'IDE, Eclipse offre un environnement convivial qui regroupe des fonctionnalités telles que l'éditeur de code, le débogueur, le compilateur et l'intégration avec des systèmes de contrôle de version. Grâce à sa flexibilité, Eclipse prend en charge différents langages de programmation tels que Java, C++, Python, PHP, et bien d'autres. Il permet également d'ajouter des plugins et des extensions pour étendre ses fonctionnalités de base. En résumé, Eclipse joue un rôle essentiel dans le développement logiciel en fournissant un environnement puissant et polyvalent pour les développeurs.



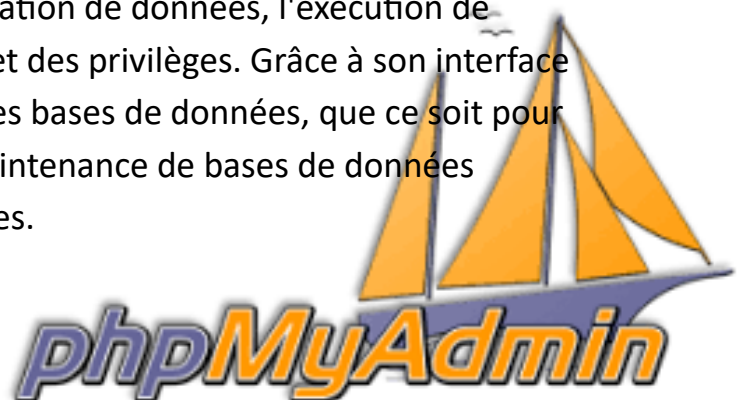
➤ **XAMPP :**

XAMPP est une solution logicielle populaire qui facilite la mise en place d'un environnement de développement web local. Il combine les principaux composants nécessaires à la création et à l'exécution d'applications web : Apache, MySQL, PHP et Perl. Cette introduction vise à souligner l'importance de XAMPP en tant qu'outil polyvalent pour les développeurs web, offrant un moyen pratique de tester et de déployer des applications localement. Grâce à sa facilité d'installation et à sa configuration simplifiée, XAMPP permet aux développeurs de se concentrer sur le développement et le test de leurs applications sans se soucier de la mise en place complexe de l'environnement de production.



➤ **PHPMyAdmin :**

PhpMyAdmin est une interface web conviviale et puissante pour gérer les bases de données MySQL. En offrant une solution accessible à travers un navigateur, phpMyAdmin simplifie la gestion et la manipulation des données dans les bases de données. Cette introduction met en évidence l'importance de phpMyAdmin en tant qu'outil essentiel pour les administrateurs de bases de données et les développeurs, offrant des fonctionnalités avancées telles que la création de tables, l'importation et l'exportation de données, l'exécution de requêtes SQL et la gestion des utilisateurs et des privilèges. Grâce à son interface intuitive, phpMyAdmin facilite la gestion des bases de données, que ce soit pour la création de nouvelles applications, la maintenance de bases de données existantes ou l'analyse des données stockées.



➤ **Astah :**

Astah Community est un outil de modélisation et de conception de logiciels gratuit et open-source. Il permet aux développeurs et aux concepteurs de créer des diagrammes UML (Unified Modeling Language) pour représenter visuellement la structure et les interactions d'un système logiciel. Cette introduction met en évidence l'importance d'Astah Community en tant qu'outil précieux pour la modélisation des logiciels, offrant une interface conviviale et des fonctionnalités avancées pour créer des diagrammes de classes, des diagrammes de cas d'utilisation, des diagrammes de séquence et bien plus encore. Avec Astah Community, les développeurs peuvent mieux comprendre les exigences et la logique d'un système, faciliter la communication entre les membres de l'équipe et documenter efficacement la conception d'un projet. Que ce soit pour des projets personnels ou des projets collaboratifs, Astah Community se révèle être un outil incontournable pour les professionnels du développement logiciel.



2. Langages :

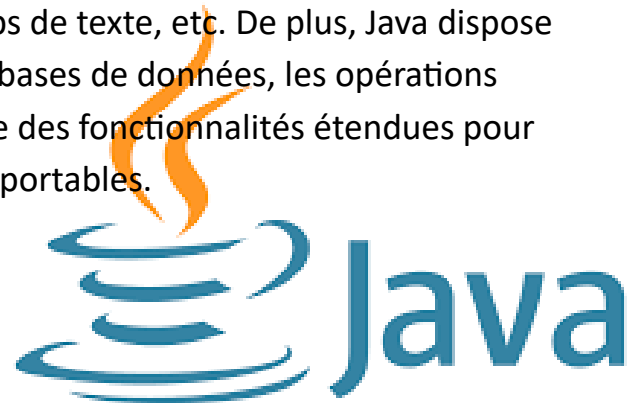
Lors du développement de cette application, les langages suivants ont été utilisés :

- ✓ JAVA.
- ✓ SQL.
- ✓ UML

On va les présenter dans la section suivante :

➤ **JAVA :**

Java est un langage de programmation polyvalent utilisé pour le développement d'applications de bureau. Sa portabilité permet aux applications Java de s'exécuter sur différents systèmes d'exploitation. Java offre des bibliothèques telles que Swing et JavaFX pour créer des interfaces graphiques interactives avec des composants tels que des boutons, des champs de texte, etc. De plus, Java dispose de nombreuses bibliothèques pour gérer les bases de données, les opérations réseau, le cryptage, etc. En résumé, Java offre des fonctionnalités étendues pour créer des applications de bureau robustes et portables.



➤ **SQL :**

SQL est un langage de requête utilisé pour interagir avec les bases de données relationnelles. Il permet de créer, de modifier et de manipuler les données stockées dans une base de données. SQL permet d'effectuer des opérations telles que la création de tables, l'insertion, la mise à jour et la suppression de données, ainsi que l'exécution de requêtes complexes pour récupérer des informations spécifiques. SQL est essentiel pour la gestion et l'interrogation efficaces des données dans les applications web et les systèmes d'information.



➤ **UML :**

UML (Unified Modeling Language) est un langage de modélisation visuel largement utilisé dans le domaine du génie logiciel. Il offre un ensemble de notations graphiques standardisées pour représenter visuellement les différentes perspectives d'un système logiciel. UML propose plusieurs types de diagrammes qui permettent de capturer et de communiquer efficacement les différentes facettes d'un système.

Parmi les principaux diagrammes UML, on trouve le diagramme de cas d'utilisation, qui décrit les fonctionnalités du système du point de vue des utilisateurs. Le diagramme de classe représente les classes du système, leurs attributs, leurs relations et leurs méthodes. Le diagramme d'objets est utilisé pour montrer les instances spécifiques des classes et leurs relations à un moment donné. Le diagramme de séquence illustre les interactions entre les objets au fil du temps, tandis que le diagramme d'état représente les différents états qu'un objet peut prendre et les transitions entre ces états.



3. Outil de conception d'interface graphique :

Pour concevoir une interface graphique en Java, nous avons utilisé ces outils :

- JavaFX.
- Scene Builder.

➤ **JavaFX :**

JavaFX est une bibliothèque de développement d'interfaces graphiques moderne et riche en fonctionnalités, intégrée à la plateforme Java. Elle offre une approche plus avancée et flexible que Swing pour la création d'interfaces graphiques dans les applications de bureau Java. JavaFX prend en charge des fonctionnalités telles que les effets visuels avancés, les animations, la mise en page flexible, les contrôles personnalisés et la gestion des médias. Cette bibliothèque permet également une séparation plus claire entre la logique métier et la présentation visuelle grâce à l'utilisation du langage de balisage FXML pour définir les interfaces utilisateur.



➤ **Scene Builder :**

SceneBuilder est un outil de conception graphique qui facilite la création d'interfaces utilisateur avec JavaFX. Il permet aux développeurs de concevoir des interfaces graphiques de manière visuelle en utilisant un système de glisser-déposer pour ajouter et positionner les composants graphiques. SceneBuilder offre une interface conviviale et intuitive, permettant aux développeurs de visualiser et de manipuler facilement les éléments de l'interface utilisateur, tels que les boutons, les champs de texte, les menus, les dispositions, etc. Il permet également de personnaliser les propriétés des composants, de définir des styles CSS et d'associer des événements aux actions utilisateur.

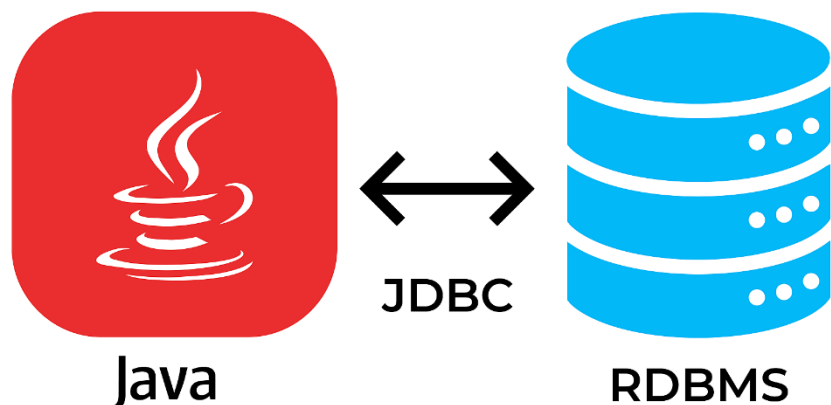


e- Outil de persistance de données :

Pour la persistance des objets en Java nous avons utiliser **JDBC (Java DataBase Connectivity)** :

C'est une API (Interface de Programmation Applicative) Java qui fournit un moyen standard pour établir une connexion et interagir avec une base de données. JDBC permet aux développeurs d'accéder et de manipuler des données stockées dans une variété de bases de données relationnelles, telles que MySQL, Oracle, PostgreSQL, etc.

En utilisant JDBC, les développeurs peuvent établir une connexion à une base de données, exécuter des requêtes SQL pour récupérer, insérer, mettre à jour ou supprimer des données, et gérer les transactions. JDBC fournit également des fonctionnalités telles que la préparation des requêtes, la gestion des résultats et la récupération des métadonnées de la base de données.



CHAPITRE II :

CONCEPTION DE L'APPLICATION

1- Introduction :

On expose, dans ce chapitre, la solution conceptuelle que on a proposée et cette conception du système à réaliser qui a pour but de rendre flexible la tâche de la gestion.

En d'autre terme, ce chapitre devrait répondre à la question : *comment faire ?*

La structure de ce chapitre dépend de la nature de ce projet. On a conçu la phase de conception d'un système d'information qui nécessite des méthodes permettant de mettre en place un modèle.

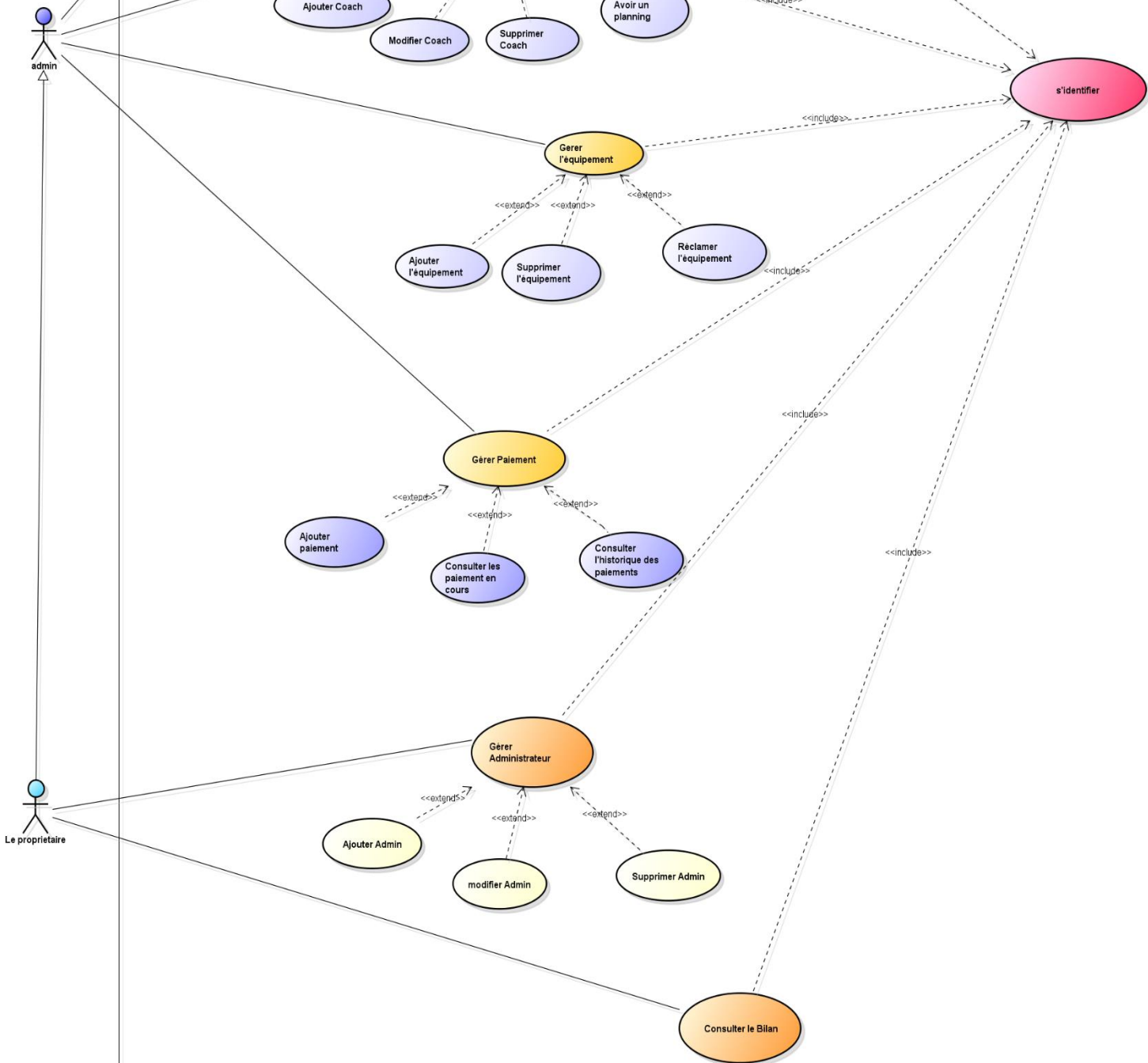
Il existe plusieurs méthodes d'analyse, la plus utilisée étant **la conception UML**.

2- Les différents diagrammes UML de notre application :

a. Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation est un outil de modélisation utilisé dans le domaine du développement logiciel pour représenter les interactions entre les acteurs et le système.

Voici notre diagramme :



b- Diagramme de classe :

Le diagramme de classes est un outil de modélisation utilisé dans le développement logiciel pour représenter la structure et les relations entre les classes d'un système.

Voici notre diagramme :

Equipment
- code int
- nom char
- status char
- nombre int
+ getEquipementCode int: void
+ getEquipementStatus char: void
+ getEquipementNombre int: void

Coach
- id int
- nom char
- prenom char
- adresse char
- email char
- salaire float
+ getCoachId int: nom char prenom char adresse char tel int salaire float: void
+ modifierCoachId int: nom char prenom char adresse char tel int salaire float: void
+ deleteCoachId int: void
+ checkCoachId int: void

Proprietaire
- id int
- nom char
- prenom char
- adresse char
- password char
- tel char
+ sauvegarderProprietaire char password char: void
+ modifierProprietaire char: void

Sale
- code int
- nom char
- prenom char
- adresse char
- employes Array<int>
- equipments Array<int>
+ getSaleCode int: nom char prenom char adresse char employes Array<int> equipments Array<int>: void
+ modifierSaleCode int: nom char prenom char adresse char employes Array<int> equipments Array<int>: void
+ deleteSaleCode int: void

Login
- id char
- password char
- loginType char
+ login char password char loginType char: void
+ modifierLogin char password char: void

Patient
- code int
- id client int
- typeAbonnement char
- dateDebut Date
- dateFin Date
- adresse char
+ getPatientCode int: id client int typeAbonnement char dateDebut Date dateFin Date adresse char: void
+ modifierPatientCode int: id client int typeAbonnement char dateDebut Date dateFin Date adresse char: void
+ deletePatientCode int: void
+ checkPatientCode int: void

Admin
- id int
- nom char
- prenom char
- login char
- password char
- tel char
+ getAdminId int: nom char prenom char adresse char tel int login char password char: void
+ modifierAdminId int: nom char prenom char adresse char tel int login char password char: void
+ deleteAdminId int: void
+ checkAdminId int: void

Client
- id int
- nom char
- prenom char
- adresse char
- tel char
- description Date
+ getClientId int: nom char prenom char adresse char tel int: void
+ modifierClientId int: nom char prenom char adresse char tel int: void
+ deleteClientId int: void
+ checkClientId int: void

Abonnement
- code int
- type char
- date int
+ getAbonnementCode int: type char date int: void
+ modifierAbonnementCode int: type char date int: void
+ deleteAbonnementCode int: void
+ checkAbonnementCode int: void

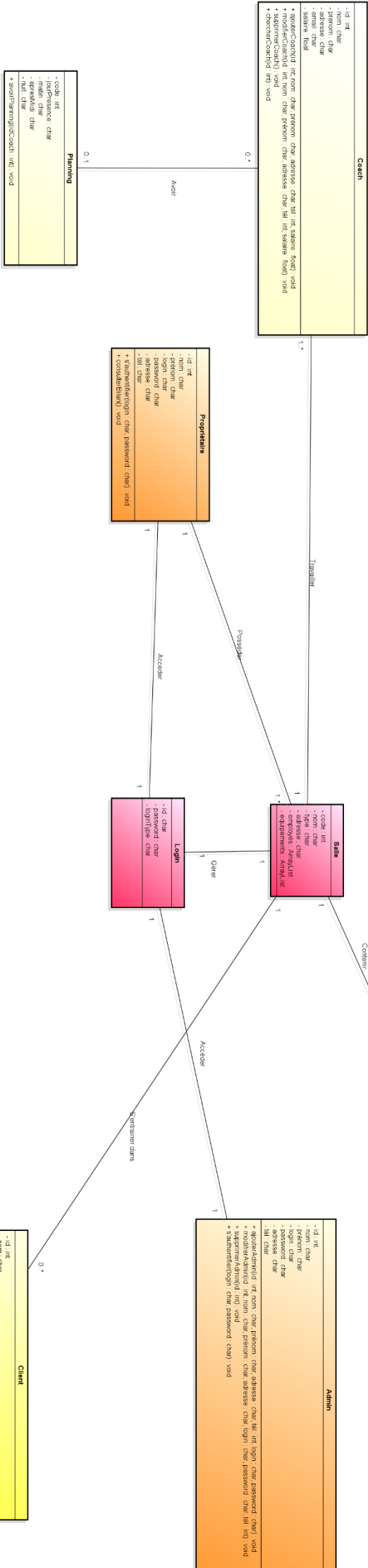
Abonnement
- code int
- type char
- date int
+ getAbonnementCode int: type char date int: void
+ modifierAbonnementCode int: type char date int: void
+ deleteAbonnementCode int: void
+ checkAbonnementCode int: void

Abonnement
- code int
- type char
- date int
+ getAbonnementCode int: type char date int: void
+ modifierAbonnementCode int: type char date int: void
+ deleteAbonnementCode int: void
+ checkAbonnementCode int: void

Abonnement
- code int
- type char
- date int
+ getAbonnementCode int: type char date int: void
+ modifierAbonnementCode int: type char date int: void
+ deleteAbonnementCode int: void
+ checkAbonnementCode int: void

Abonnement
- code int
- type char
- date int
+ getAbonnementCode int: type char date int: void
+ modifierAbonnementCode int: type char date int: void
+ deleteAbonnementCode int: void
+ checkAbonnementCode int: void

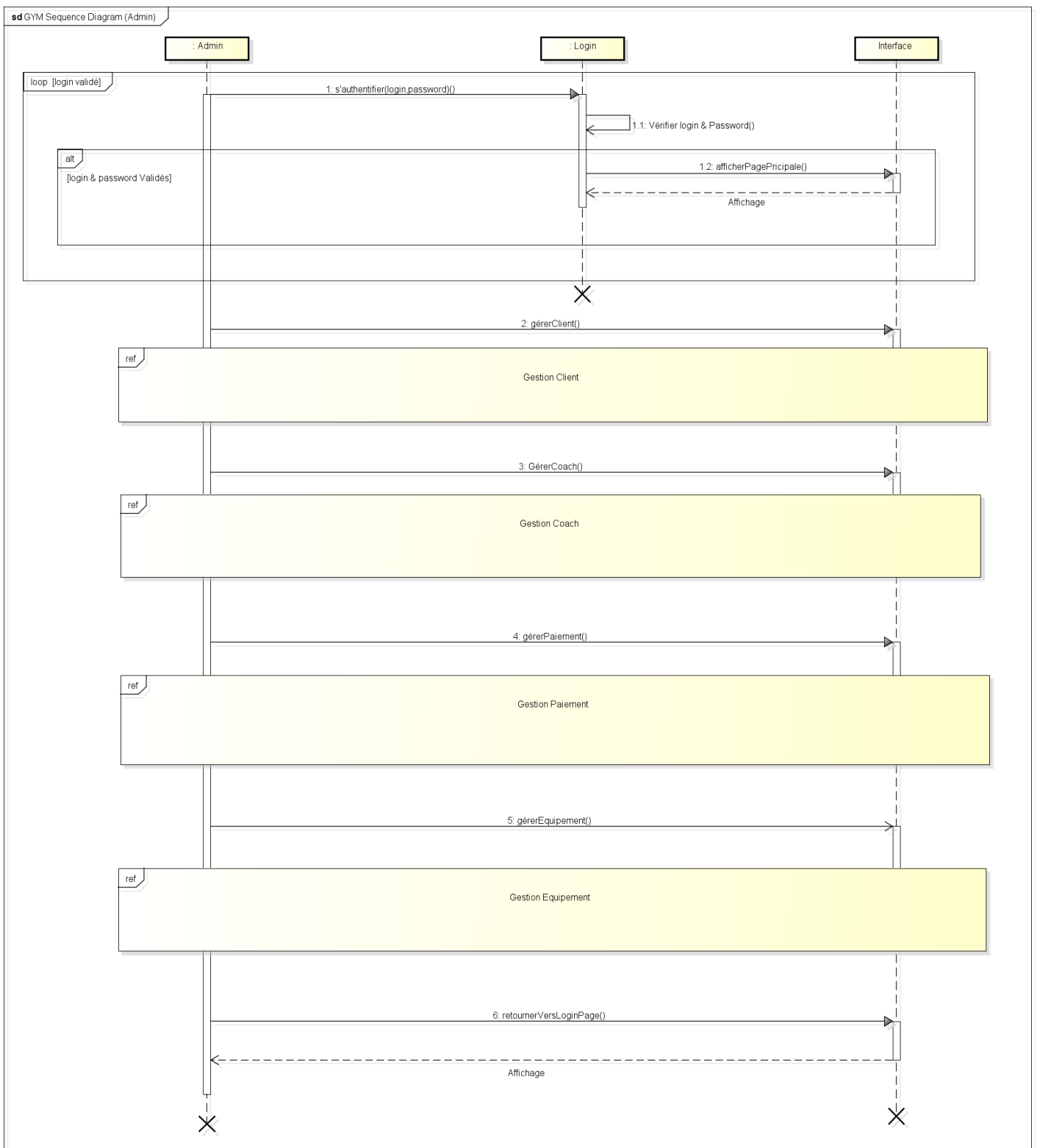
Planning
- code int
- jourAbonnement char
- heureAbonnement char
- repete char
- nat char
+ voirPlanningCoach int: void



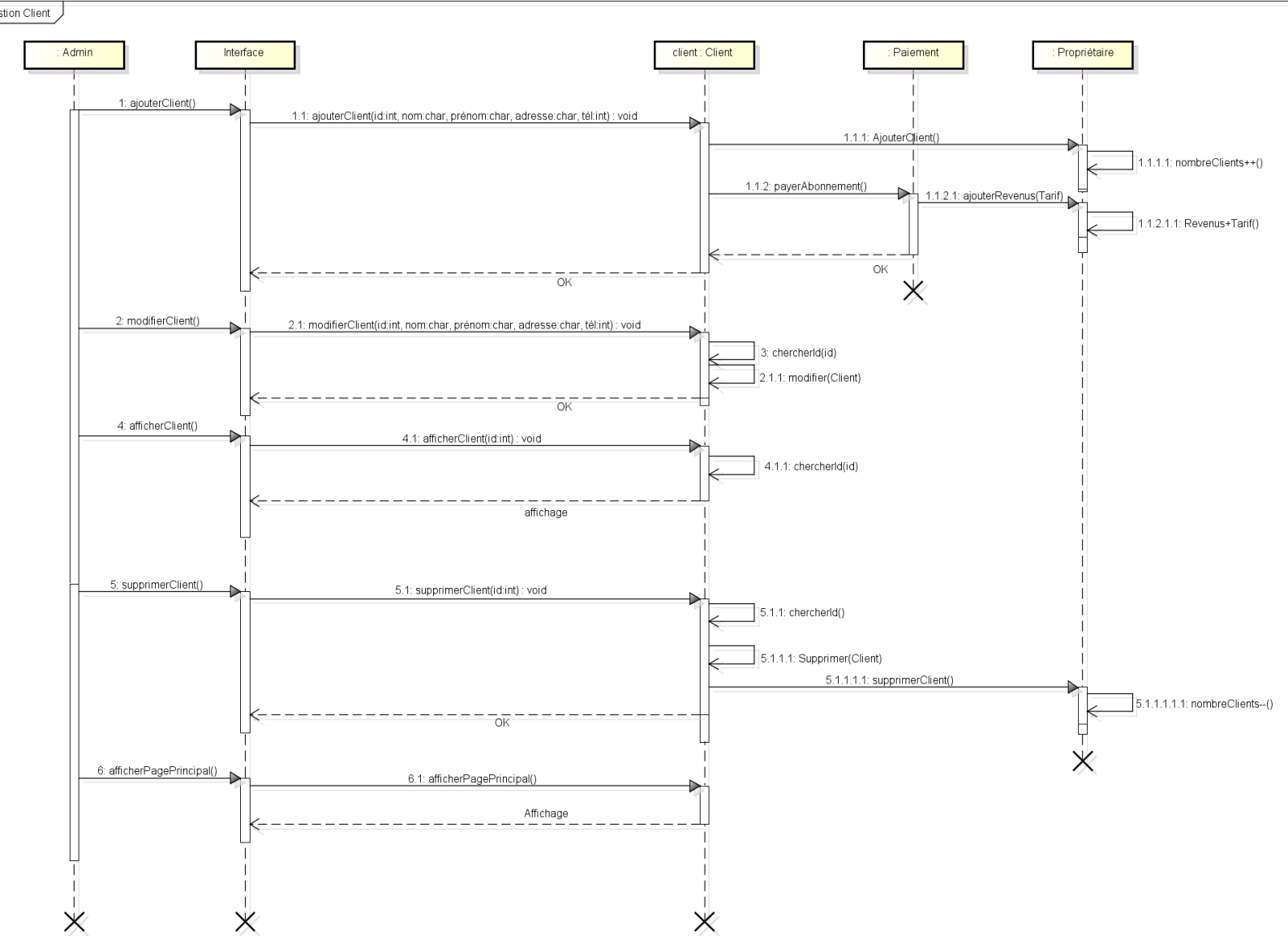
c- Diagramme de séquence :

Il représente l'interaction entre les objets d'un système dans une séquence chronologique. Ce diagramme montre comment les objets communiquent entre eux en échangeant des messages au fil du temps.

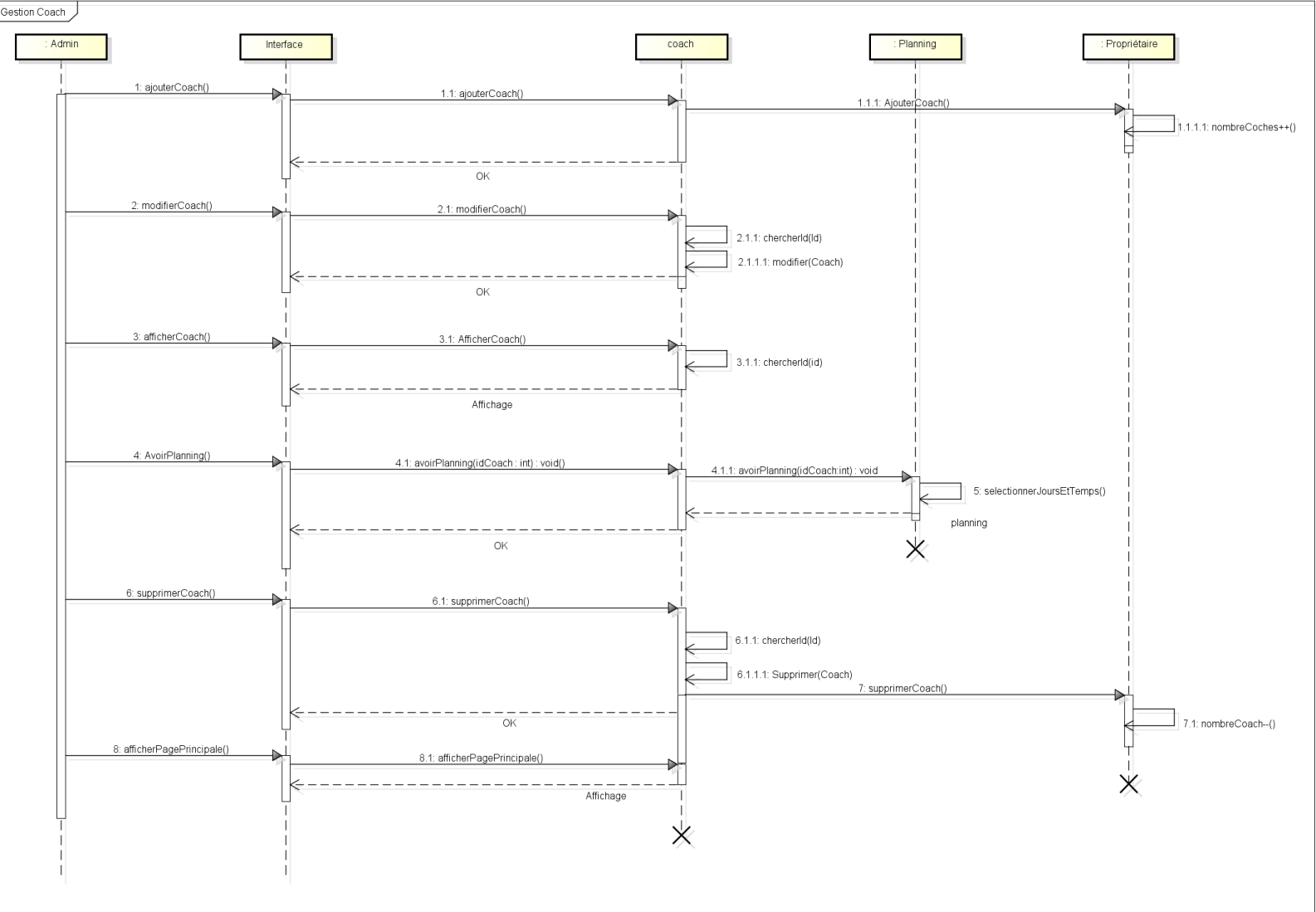
Veillez trouver ci-dessous le **diagramme de séquence de l'administrateur** :



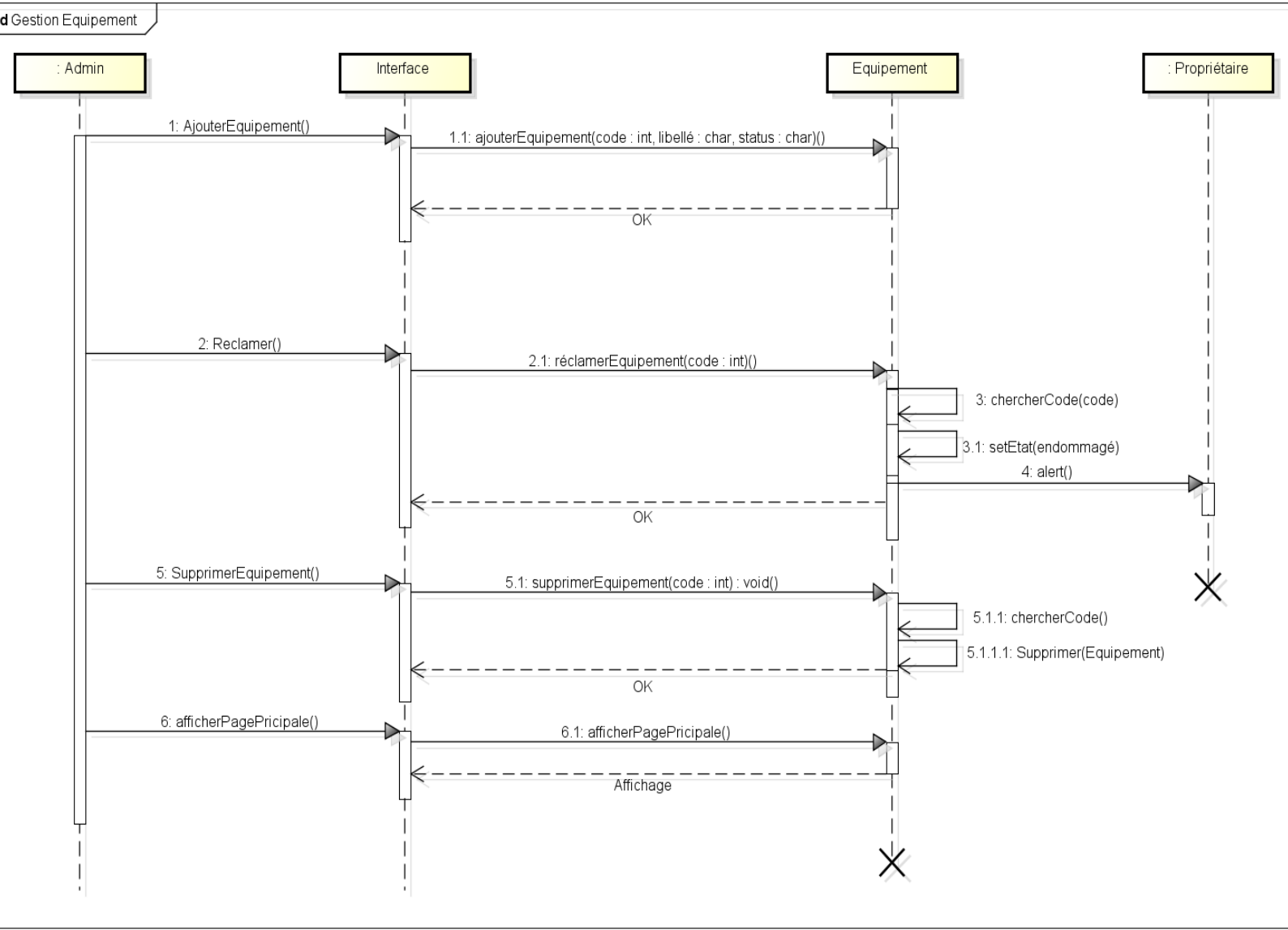
Veillez trouver ci-dessous le diagramme de séquence référencé de la partie **Gestion client** :



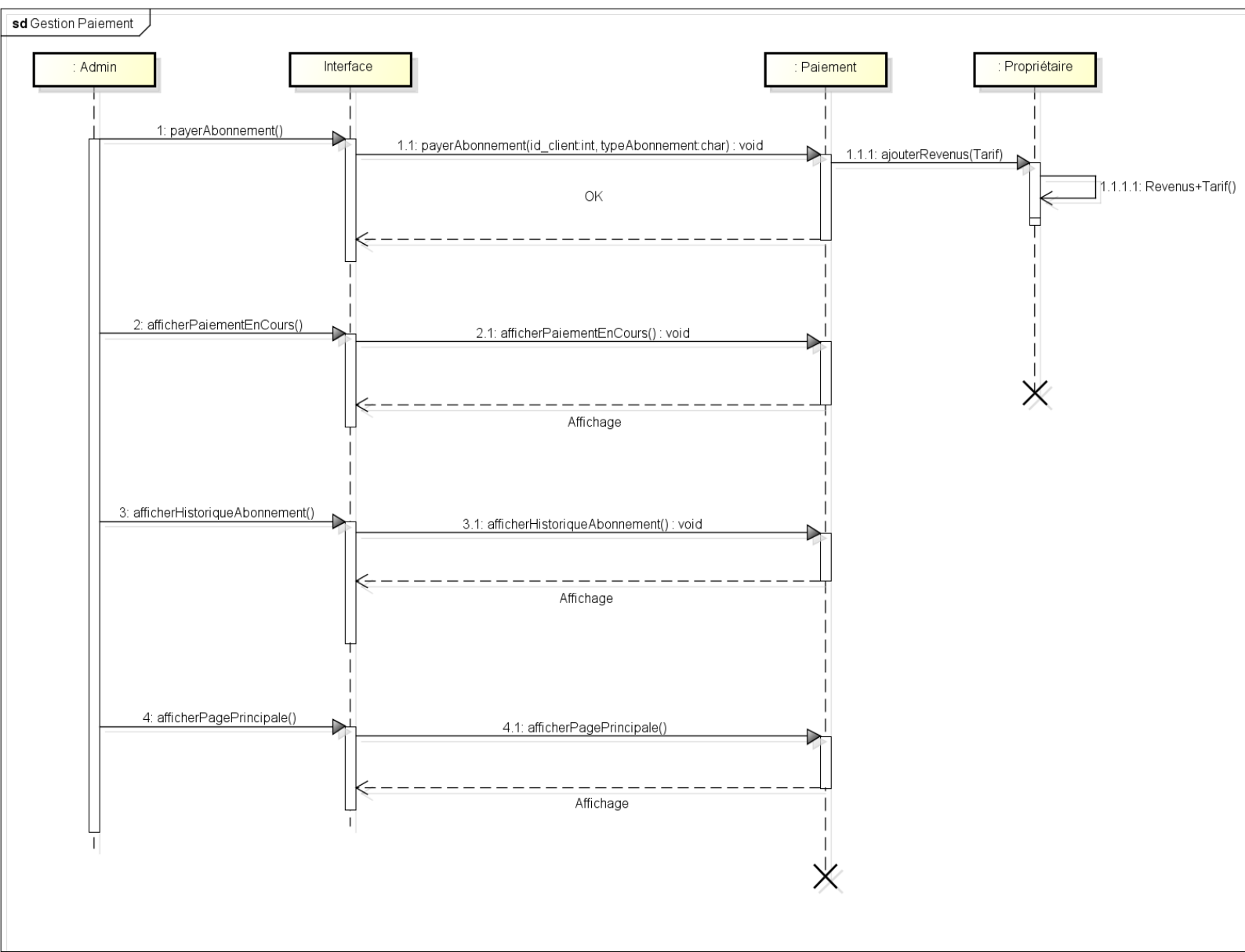
Le diagramme de séquence référencé de la partie **Gestion coach** :



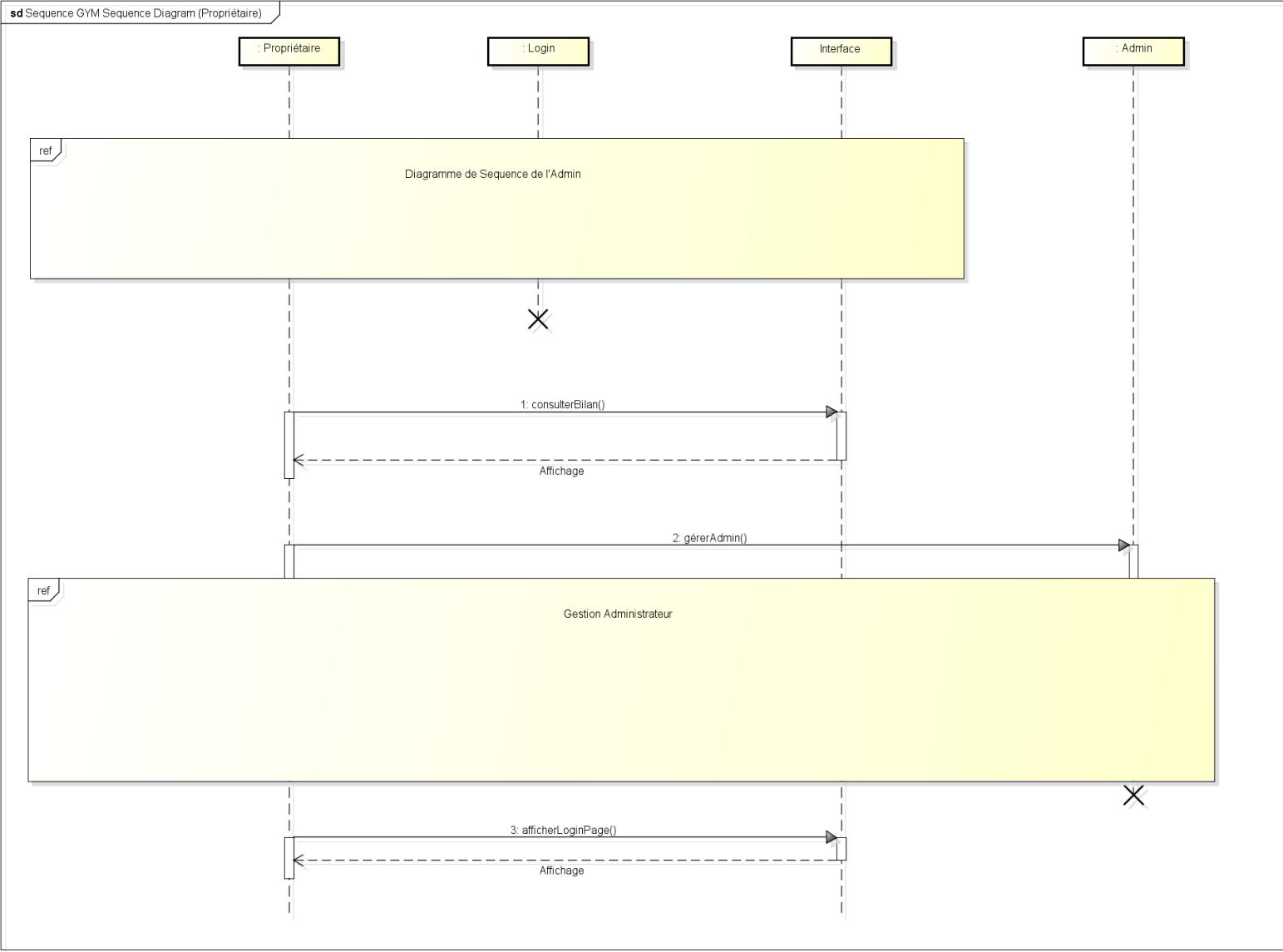
Veillez trouver ci-dessous le diagramme de séquence référencé de la partie **Gestion équipement** :



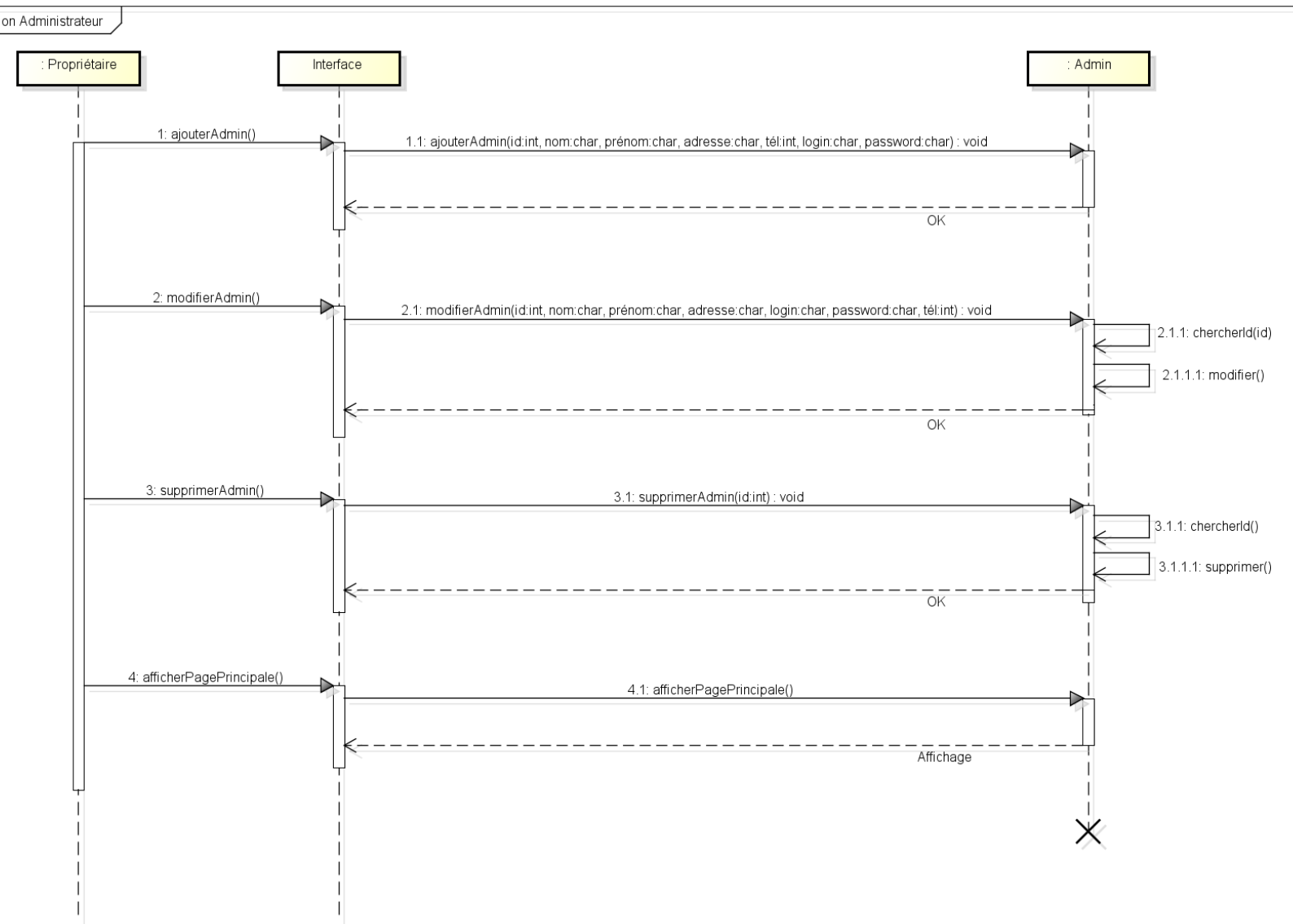
Veillez trouver ci-dessous le diagramme de séquence référencé de la partie **Gestion paiement** :



Veillez trouver ci-dessous le **diagramme de séquence du Propriétaire** :



Veillez trouver ci-dessous le diagramme de séquence référencé de la partie **Gestion Administrateur** :



CHAPITRE III :

Organisation et Bilan du Projet

1- Introduction :

Ce chapitre a pour objectif majeur de présenter le bilan qualitatif du travail et son déroulement (difficultés, points intéressants...) et l'organisation et la répartition des tâches au sein du groupe durant la durée du projet.

2- L'organisation et la répartition des tâches au sein du groupe :

Au sein de notre groupe de projet, nous avons adopté une approche collaborative pour l'organisation et la répartition des tâches. Dès le départ, nous avons identifié nos domaines d'expertise et nos intérêts spécifiques. L'un d'entre nous s'est concentré sur **la conception UML de l'interface**, en élaborant les diagrammes pour représenter la structure et le comportement de l'application. Pendant ce temps, les deux autres membres se sont attelés à la partie **développement du projet**. Ils ont travaillé en étroite collaboration pour coder les différentes fonctionnalités de l'interface en utilisant Java et les bibliothèques appropriées telles que JavaFX.

Nous avons maintenu une communication régulière tout au long du projet, partageant nos avancées, résolvant les problèmes rencontrés et harmonisant nos efforts. Nous avons également effectué des réunions périodiques pour évaluer notre progression, ajuster nos plans si nécessaire et nous assurer que nous restions alignés sur nos objectifs communs.

Cette approche de répartition des tâches nous a permis de tirer parti de nos compétences individuelles tout en travaillant en équipe. Elle nous a également permis de gagner du temps en parallélisant les activités de conception et de développement. En fin de compte, nous avons réussi à réaliser une interface fonctionnelle en respectant les délais du projet.

Dans l'ensemble, nous avons trouvé cette expérience intéressante et enrichissante, combinant à la fois les aspects conceptuels et techniques de la réalisation d'une interface. La répartition des tâches et la collaboration étroite ont été des éléments clés de notre réussite dans la réalisation de ce projet.

2- Bilan qualitatif du travail (difficultés rencontrées, points intéressants...) :

🔧 Difficultés rencontrées :

Pendant la réalisation de ce projet, nous avons été confrontés à quelques difficultés :

- Une difficulté que nous avons rencontrée était liée à l'utilisation de JavaFX, en particulier lorsqu'il s'agissait de personnaliser l'apparence et le comportement des composants graphiques. **La documentation de JavaFX était parfois limitée** et nous avons dû faire preuve de persévérance et de recherche approfondie pour trouver des solutions aux problèmes spécifiques que nous rencontrions.
- En addition, nous avons rencontré une **difficulté particulière lors de la création des diagrammes UML**. Cette difficulté résidait dans la représentation précise et claire des relations et des interactions entre les différents éléments du système. Nous nous sommes confrontés à des choix délicats pour décrire correctement les associations, les dépendances, les héritages et les flux d'informations entre les classes, les composants ou les acteurs.

🔧 Points intéressants :

Au cours de ce projet, plusieurs aspects nous ont semblé particulièrement intéressants :

- Tout d'abord, la conception UML de l'interface nous a permis de **visualiser et de planifier efficacement les différentes** fonctionnalités de l'application. Cela nous a donné une vision d'ensemble du système et nous a aidés à mieux comprendre les interactions entre les différents composants.
- De plus, le développement de l'interface avec Java nous a permis de **renforcer nos compétences en programmation orientée objet** et d'approfondir notre compréhension de **JavaFX** en tant que framework de développement d'interfaces graphiques. Nous avons pu explorer les fonctionnalités avancées

de JavaFX, telles que la gestion des événements, la création de styles personnalisés et l'animation des composants.

- Enfin, travailler en équipe sur ce projet nous a permis **d'améliorer notre collaboration et notre communication**. Nous avons appris à partager efficacement les tâches, à coordonner nos efforts et à résoudre les problèmes ensemble. Cette expérience de collaboration nous a préparés à travailler de manière efficace et harmonieuse dans des projets futurs.

Dans l'ensemble, malgré les défis rencontrés, les points intéressants que nous avons découverts dans ce projet nous ont permis de développer nos compétences techniques, de renforcer notre compréhension de la conception d'interfaces graphiques et de renforcer notre esprit d'équipe.

Conclusion générale

Le projet que nous avons réalisé a apporté des contributions significatives sur les plans technique, scientifique et humain. Sur le plan technique, nous avons acquis une expérience approfondie dans le développement d'une application desktop avec Java et JavaFX. Nous avons renforcé nos compétences en programmation orientée objet, en manipulation d'interfaces graphiques et en utilisation de bibliothèques spécifiques.

D'un point de vue scientifique, la réalisation de ce projet nous a permis d'approfondir notre compréhension des concepts de modélisation et de conception logicielle. Nous avons utilisé des techniques telles que la conception UML et l'analyse des besoins pour élaborer une architecture robuste et fonctionnelle. Nous avons également exploré les différents diagrammes UML, tels que les diagrammes de classe, de séquence et d'état de transition, pour représenter les aspects statiques et dynamiques de notre application.

Sur le plan humain, ce projet nous a offert l'opportunité de travailler en équipe et de développer nos compétences en communication et en collaboration. Nous avons appris à répartir les tâches de manière efficace, à partager nos connaissances et à résoudre des problèmes ensemble. Nous avons également développé notre capacité à travailler dans un environnement de projet structuré, en respectant les délais et en faisant preuve de rigueur.

En conclusion, ce projet a été extrêmement bénéfique sur les plans technique, scientifique et humain. Il nous a permis d'acquérir de nouvelles compétences techniques, d'approfondir notre compréhension des principes de conception logicielle et de renforcer notre capacité à travailler en équipe. Nous sommes fiers du résultat que nous avons obtenu et sommes confiants dans notre capacité à appliquer ces connaissances et ces compétences dans nos futurs projets.