UNIVERSITE de la POLYNÉSIE FRANCAISE
DEPARTEMENT SCIENCE TECHNOLOGIE ET SANTÉ

Informatique

Semestre 5

GENIE LOGICIEL

Rapport :

Gestion de réservation d'une salle de sport

De :
Manuel Aie
Legrand Pauline
Bambridge Mary Ann

Rédigé par :
Manuel Aie

Année universitaire 2024-2025

Table des matières

Table des figures

1 Contexte

Dans le cadre de notre Licence Informatique, nous avons travaillé en groupe de trois personnes sur un projet donné dans l'unité d'enseignement Génie Logiciel. Le sujet portait sur la création d'un Système de Gestion de Réservation pour une Salle de Sport. L'objectif était de concevoir, modéliser et implémenter un programme permettant de gérer les réservations pour différentes activités sportives.

Le programme devait répondre à plusieurs fonctionnalités clés :
- Consultation des créneaux disponibles pour des activités telles que le tennis, le basketball, et le fitness.
- Réservation d'un créneau en fournissant les informations nécessaires (nom, activité, date et heure).
- Paiement d'une cotisation annuelle obligatoire par l'utilisateur afin de pouvoir faire une réservation.
- Mise à jour du créneau lors de la réservation pour une activité.
- Gestion des annulations, avec des pénalités en cas d'annulation tardive (moins de 24 heures avant l'activité).

Nous avons donc dû réfléchir à un système qui intègre ces contraintes tout en respectant les horaires d'ouverture de la salle (8h à 21h). Ce projet nous a permis d'aborder toutes les étapes d'un développement logiciel : analyse des besoins, modélisation UML, et développement d'un prototype fonctionnel.

L'absence de détails techniques précis dans le sujet initial nous a amenés à prendre des décisions pour définir les aspects pratiques du système. Ces choix nous ont également guidés dans la répartition des tâches au sein de l'équipe, où chaque membre s'est vu attribuer une partie de la conception ou du développement.

2 Répartition des tâches et choix initiaux

Dès notre première réunion, nous avons pris le temps d'analyser le sujet pour comprendre ses enjeux. Nous avons décidé de commencer par la réalisation de plusieurs diagrammes UML afin de poser une base solide pour la conception et l'implémentation du projet. Ces diagrammes incluent le diagramme de cas d'utilisation, le diagramme de classe, le diagramme de séquence et le diagramme d'activité.

2.1 Répartition des tâches

Pour assurer une organisation claire et efficace, nous avons réparti les tâches de la manière suivante :
- Pauline : Diagramme de cas d'utilisation et rédaction de la partie correspondante dans le rapport.
- Aie : Diagramme de classe et vérification de sa cohérence avec les autres diagrammes.
- Mary Ann : Diagramme de séquence et participation à la mise en relation avec le diagramme de classe et d'activité.

2.2 Choix initiaux

Nous avons fait le choix de travailler en PHP car c'est un langage principalement utilisé côté serveur ce qui répondait exactement à nos besoins. Au départ, l'objectif était de développer

une application en utilisant Android Studio et Java, mais après réflexion, nous avons remis en question cette approche, ce qui nous a amenés à changer de stratégie. Pour l'environnement de développement, nous avons utilisé VS Code, car il est facilement personnalisable et adapté pour la gestion de projets en équipe. La gestion des versions a été assurée via un dépôt GitHub, où nous avons partagé les fichiers et suivi les modifications en temps réel.

3 Conception : Réalisation des diagrammes UML

3.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation a été réalisé par Pauline. Il nous a permis de définir les différentes interactions possibles entre les utilisateurs (clients, modérateurs) et le système. Ce diagramme met en évidence les fonctionnalités principales comme la consultation des créneaux, la réservation, le paiement de la cotisation, et la gestion des annulations.

Acteurs

- Client
  - Consulter les créneaux disponibles.
  - Réserver une activité (après vérification de la disponibilité et paiement de la cotisation annuelle).
  - Annuler une réservation (avec une pénalité pour les annulations tardives).
- Modérateur
  - Gérer les créneaux et activités (création et modification).

Cas d'utilisation principaux

- Consulter les créneaux : Permet de visualiser les horaires pour diverses activités.
- Réserver un créneau : Inclut la vérification de la disponibilité et le paiement de la cotisation annuelle.
- Payer une cotisation annuelle : Nécessaire pour effectuer des réservations.
- Annuler une réservation : Pénalité appliquée si l'annulation est effectuée moins de 24 heures avant.
- Mettre à jour les réservations : Libération automatique des créneaux après usage.

Relations

- Inclusion («include») : Vérification de la disponibilité et paiement inclus dans la réservation.
- Extension («extend») : Pénalité ajoutée en cas d'annulation tardive.

3.2 Diagramme de classe

Le diagramme de classe, réalisé par Aie, est le centre de notre projet. Il regroupe les différentes classes du système, telles que Utilisateur, Activité, Creneau, et leurs relations. Ce diagramme a été révisé à plusieurs reprises pour garantir sa cohérence avec les autres diagrammes et pour intégrer les modifications apportées au projet en cours de route. Le diagramme a donc été réalisé par l'ensemble du groupe afin qu'il soit bien en accord avec le projet.

Principales Classes et Justification

- Personne (classe abstraite) : Représente l'entité de base pour les utilisateurs, avec des méthodes communes comme la connexion. Les classes dérivées Utilisateur et Modérateur héritent des fonctionnalités spécifiques (ex. : gestion des créneaux pour les modérateurs).
- BaseDeDonnees : Centralise la gestion des données pour toutes les opérations, assurant une communication fluide avec les autres classes.

— Reservation : Gère les réservations en vérifiant leur disponibilité, leur confirmation et leur annulation.
— Activité et Créneau : Modélisent les différentes activités proposées et leurs créneaux horaires associés, permettant une gestion flexible des réservations. Nous avons également décidé que les activités pouvaient avoir des durées différentes
— Cotisation : Assure le suivi des paiements annuels obligatoires avant toute réservation.
— GestionUtilisateur et GestionSuperUtilisateur : Fournissent des fonctionnalités avancées pour la gestion des utilisateurs et des activités respectivement, en assurant une séparation claire des responsabilités.

Relations entre Classes

— RIBEntreprise : développée rapidement, ce qui a conduit Pauline à oublier d'ajouter l'héritage avec la classe RIB. Dans le diagramme de classes, nous avons relié la classe RIBEntreprise aux classes associées à RIB, alors qu'une relation d'héritage aurait été plus appropriée.
— GestionUtilisateur interagit directement avec Reservation et Activité pour coordonner les réservations.
— Cotisation est associée à Utilisateur, indiquant qu'un utilisateur doit payer pour accéder aux fonctionnalités.
— L'interface Paiement assure l'extensibilité en prenant en charge différents modes de paiement.

La réalisation de ce diagramme s'est avérée complexe en raison des liens à établir entre toutes les classes. Notre problème avec le premier jet résidait dans le fait que chaque créneau devait correspondre à une activité spécifique, ce qui entraînait un nombre excessif d'entrées dans la base de données. Nous avons donc décidé de simplifier l'organisation en répartissant les créneaux de manière uniforme sur tous les jours.(Voir README du dépôt git)

3.3 Diagramme d'activité

Vue générale du diagramme d'activités :

Le diagramme d'activités général du système de gestion des réservations pour une salle de sport, fait par Pauline, illustre les étapes et décisions clés dans l'interaction utilisateur (client ou modérateur) avec le système. Il inclut les actions principales telles que l'inscription, le paiement, la réservation, et la gestion des activités. Ce schéma permet de comprendre le flux logique des actions, tout en intégrant des exceptions comme les paiements refusés ou les annulations tardives.

— Inscription et Connexion : Les utilisateurs doivent s'inscrire ou se connecter pour accéder aux fonctionnalités du système.
— Paiement de la cotisation annuelle : Cette étape conditionne l'accès à la réservation. Le paiement est validé ou refusé en fonction des informations fournies.
— Réservation et gestion des créneaux : Les utilisateurs peuvent réserver ou annuler des créneaux, tandis que les modérateurs gèrent les disponibilités et créent/modifient les activités.
— Mise à jour après l'activité : Le système libère les créneaux automatiquement après la fin des activités réservées.

Ce diagramme n'était pas prévu initialement, mais nous avons jugé qu'il serait plus judicieux d'en créer un pour obtenir une meilleure visibilité de notre projet. Cela s'est avéré très utile et a grandement contribué à la gestion et à la clarté de notre travail.

Diagramme d'activité pour le paiement de la cotisation annuelle lors de l'inscription :

Ce diagramme, réalisé par Aie, détaille les étapes suivies pour permettre au client de payer sa cotisation annuelle. L'objectif est de garantir que le paiement soit correctement effectué avant de débloquer l'accès aux autres fonctionnalités.

— Inscription dans le système : L'utilisateur s'inscrit pour accéder à son tableau de bord.
— Passage au paiement

Diagramme d'activité pour réservé un créneau :

Ce diagramme, réalisé par Aie, illustre les étapes pour réserver un créneau d'activité dans le système :

— Choix de l'activité : L'utilisateur sélectionne une activité parmi celles proposées.
— Consultation des créneaux disponibles : Il consulte les créneaux associés à l'activité choisie.
— Sélection du créneau : L'utilisateur choisit un créneau spécifique.
— Vérification de la disponibilité : Le système s'assure que le créneau est toujours disponible et que l'utilisateur n'a pas déjà réservé une autre activité sur le même créneau.
  — Si le créneau n'est pas disponible, l'utilisateur peut en choisir un autre.
  — Si le créneau est disponible, la réservation est validée.
— Réservation confirmée : Le créneau est réservé avec succès, et l'utilisateur est notifié.

Ce processus garantit que seules les options disponibles sont accessibles, optimisant la gestion des créneaux.

3.4 Diagramme de séquence

Mary Ann a réalisé des diagrammes de séquence pour modéliser les échanges entre les différentes entités du système. Par exemple, ils montrent comment un utilisateur effectue une réservation en passant par différentes étapes (vérification de la disponibilité, validation de la réservation, paiement).

4 Étape d'implémentation

4.1 Organisation du code

Nous avons structuré notre projet en plusieurs fichiers PHP, chaque fichier correspondant à une classe ou une fonctionnalité spécifique. Par exemple :

- utilisateur.php
- activite.php
- reservation.php

4.2 Logiciels et outils utilisés

- Langage : PHP(version 8.3.9)
- IDE : Visual Studio Code
- Système de gestion de base de données : MySQL, pour gérer les données des utilisateurs, créneaux et réservations.
- GitHub : Pour assurer le suivi des versions et faciliter la collaboration en équipe, nous avons créé différentes branches dédiées à chaque classe, ainsi que des branches supplémentaires pour la gestion de nos documents.
- Lucidchart : Logiciel qui nous a permis de partager en temps réel les diagrammes
- Messenger : Nous avons créé un groupe Messenger pour faciliter le partage d'informations, comme les dates de réunion et les différentes communications importantes.

Nous avons travaillé selon un cycle itératif : chaque membre développait les classes ou fonctionnalités qui lui étaient attribuées avec les tests correpondants, puis nous les testions en équipe pour vérifier leur bon fonctionnement et leur cohérence avec le reste du système.

5 Implémentation des tests

Les tests étaient une partie cruciale pour garantir la qualité et la fiabilité de notre application. Nous avons mis en place trois catégories principales de tests :

- Tests unitaires : Vérification des méthodes de chaque classe (par exemple, la validation d'un créneau disponible).
- Tests d'intégration : Vérification des interactions entre les classes (par exemple, entre la classe Utilisateur et la classe Reservation).
- Tests fonctionnels : Simulation des scénarios d'utilisation réels, comme la réservation d'un créneau ou l'annulation d'une activité.

Lors de la phase tests, nous avons rencontré des difficultés liées à l'installation des outils nécessaires et à la comptabilité des environnements. Bien que le processus ait fonctionné pour Pauline, qui utilise MacOS, elle n'a pas pu nous aider, Mary Ann et moi, car nous sommes sous Windows. Nous avons utilisé la unitTest de PHP pour l'automatisation des tests. Les résultats ont permis de corriger plusieurs bugs, notamment dans la gestion des annulations et dans les mises à jour de l'état des créneaux.

6 Reflexion sur le projet

En travaillant sur ce projet, nous avons réalisé l'importance d'une communication claire et d'une organisation rigoureuse :

- Nous avons organisé deux réunions par semaine pour assurer un suivi régulier de l'avancement du projet et coordonner efficacement les tâches entre les membres de l'équipe.

— Nous avons parfois perdu du temps en raison d'incompréhensions autour du sujet initial, ce qui a nécessité des ajustements dans les diagrammes et les fonctionnalités.
— La répartition des tâches a été efficace dans l'ensemble, mais l'absence de certains membres lors de réunions importantes a retardé certaines étapes, notamment la cohérence entre les diagrammes.
— Nous avons passé plus de temps à debuger qu'à coder.
— Nous avons sous estimer l'importance d'une bonne base de données.
— Les tests sont de super outils.
— Pauline a pris en charge les directives de codage, en tirant parti de la flexibilité de PHP pour ne pas spécifier les types des paramètres. Cette approche a permis de mieux identifier les erreurs potentielles. Elle a également assuré la création de la classe BaseDeDonnees, essentielle pour la gestion des interactions avec la base de données.

6.1 Avis critique du groupe

L'avis général est que nous avons sous-estimé le temps nécessaire pour réaliser ce projet, ce qui a eu un impact sur notre progression. De plus, l'engagement de chaque membre n'a pas été homogène, entraînant des disparités dans la charge de travail et l'implication de chacun. Bien que nous ayons eu des discussions, elles n'ont pas été suffisamment efficaces pour faire avancer le projet comme il le fallait. Ceci est notamment visible dans nos notes de réunions, où il est apparu que les objectifs restaient les mêmes sur plusieurs réunions, ce qui ne devrait pas être le cas. Nous avons également passé trop de temps sur la partie théorique, alors que nous aurions dû commencer à coder plus tôt pour tendre plus rapidement vers des améliorations efficaces. Cela aurait permis d'obtenir un rendu plus proche de ce que nous avons finalisé et d'économiser un temps précieux. De plus, la coordination de notre travail de développement a été insuffisante, ce qui a entraîné des incohérences importantes. Pour résoudre cela, nous avons dû confier le code à une seule personne, ce qui a retardé notre avancement. En ce qui concerne la répartition des tâches, nous pouvons critiquer le fait que nous aurions probablement dû organiser le travail par catégories plutôt que par parties. Par exemple, désigner une personne pour superviser les graphiques, une autre pour le front-end, et une autre pour le back-end, aurait peut-être permis de mieux gérer les problèmes de cohérence et de réduire le nombre de versions. Si une personne s'était consacrée exclusivement au diagramme de classes et avait supervisé le travail des autres, elle aurait pu identifier plus rapidement les incohérences et potentiellement éviter de devoir y revenir autant de fois.

7 Conclusion et rétrospective

Ce projet nous a permis d'appliquer les concepts de Génie Logiciel à un cas concret. Nous avons produit un système fonctionnel répondant aux exigences principales du sujet. Au début, nous avons eu des difficultés à comprendre le fonctionnement du diagramme de classes, notamment en ce qui concerne les relations entre les entités. De plus, nous avons commencé les TD de réalisation un peu tard, ce qui nous a fait perdre du temps et nous a contraints à reprendre notre travail depuis le début. Ces défis ont impacté notre progression, mais nous avons pu ajuster notre approche pour avancer de manière plus efficace. Lors de la conception du site au début, nous avons choisi de développer sans tester l'ensemble des fonctionnalités. Cette approche nous a conduit à des blocages, notamment au niveau de la gestion des créneaux et des activités, entrainant une perte de temps. Nous aurions dû mettre davantage l'accent sur les tests dès le début pour éviter ces problèmes et optimiser notre processus de développement. De plus, nous n'avons pas réussi à suivre le plan que nous nous étions fixé ni à respecter les deadlines. Ce manque d'organisation a créé du retard et nous a mis beaucoup de pression à la fin.

8 Bilan et perspective

Bien que la répartition du travail ait été bien définie sur le papier, elle n'a pas été rigoureusement respectée en pratique.
Pour améliorer le projet, plusieurs pistes pourraient être explorées :

- Implémentation de notifications par e-mail : Les utilisateurs pourraient recevoir des confirmations de réservation, des rappels avant une activité ou des notifications en cas de modification.
- Ajout de différents modes de paiement : Actuellement, les paiements sont simplifiés. Une extension permettant de gérer les paiements en espèces ou par d'autres moyens serait un plus.
- Revoir la rigueur du code : Certains aspects du code pourraient être refactorisés pour être plus lisibles, maintenables et performants.
- Créer une application mobile : Une version mobile du système offrirait une meilleure accessibilité et répondrait davantage aux besoins des utilisateurs modernes.

A ANNEXES

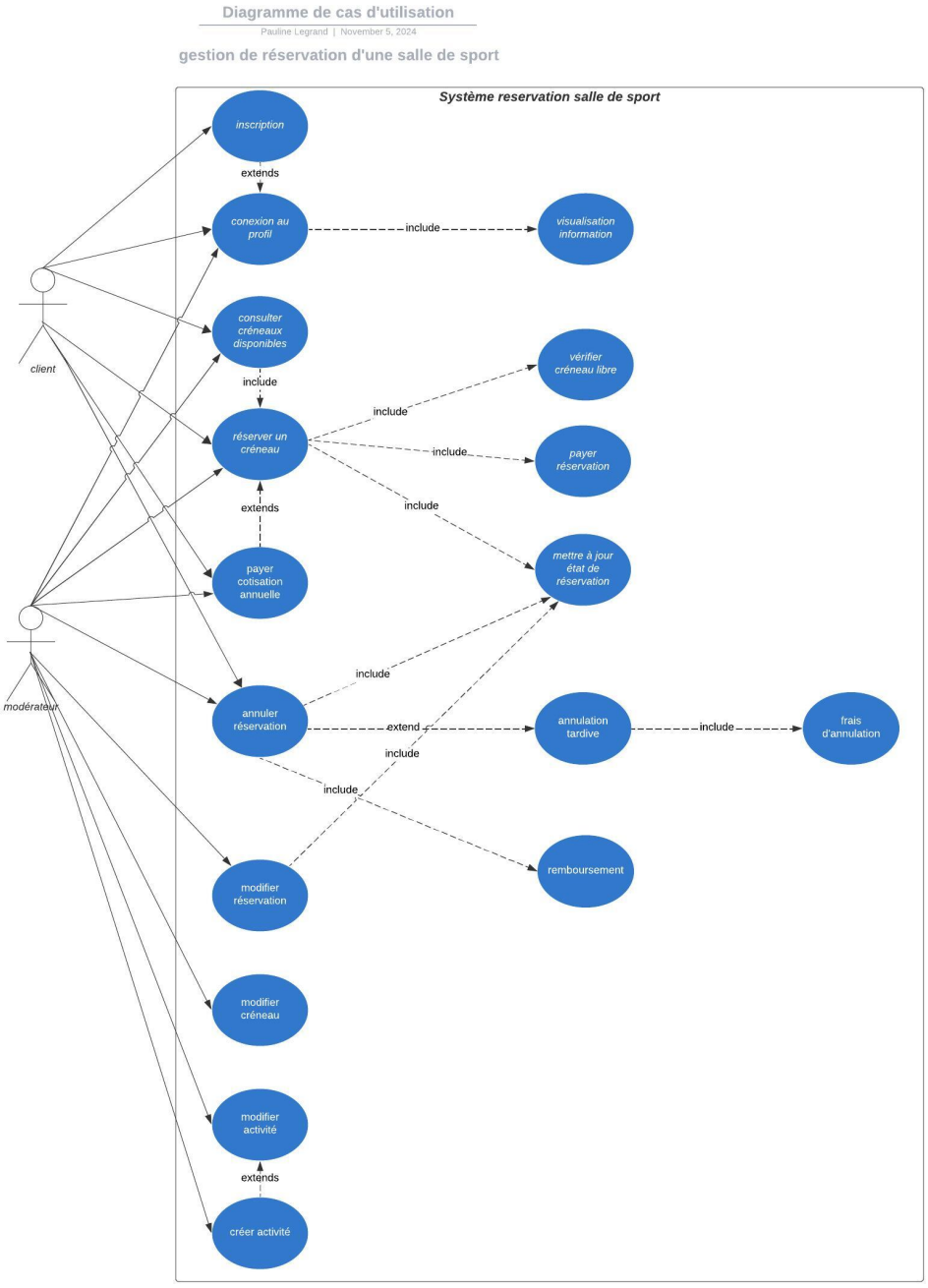

FIGURE 1 – Diagramme de cas d'utilisation

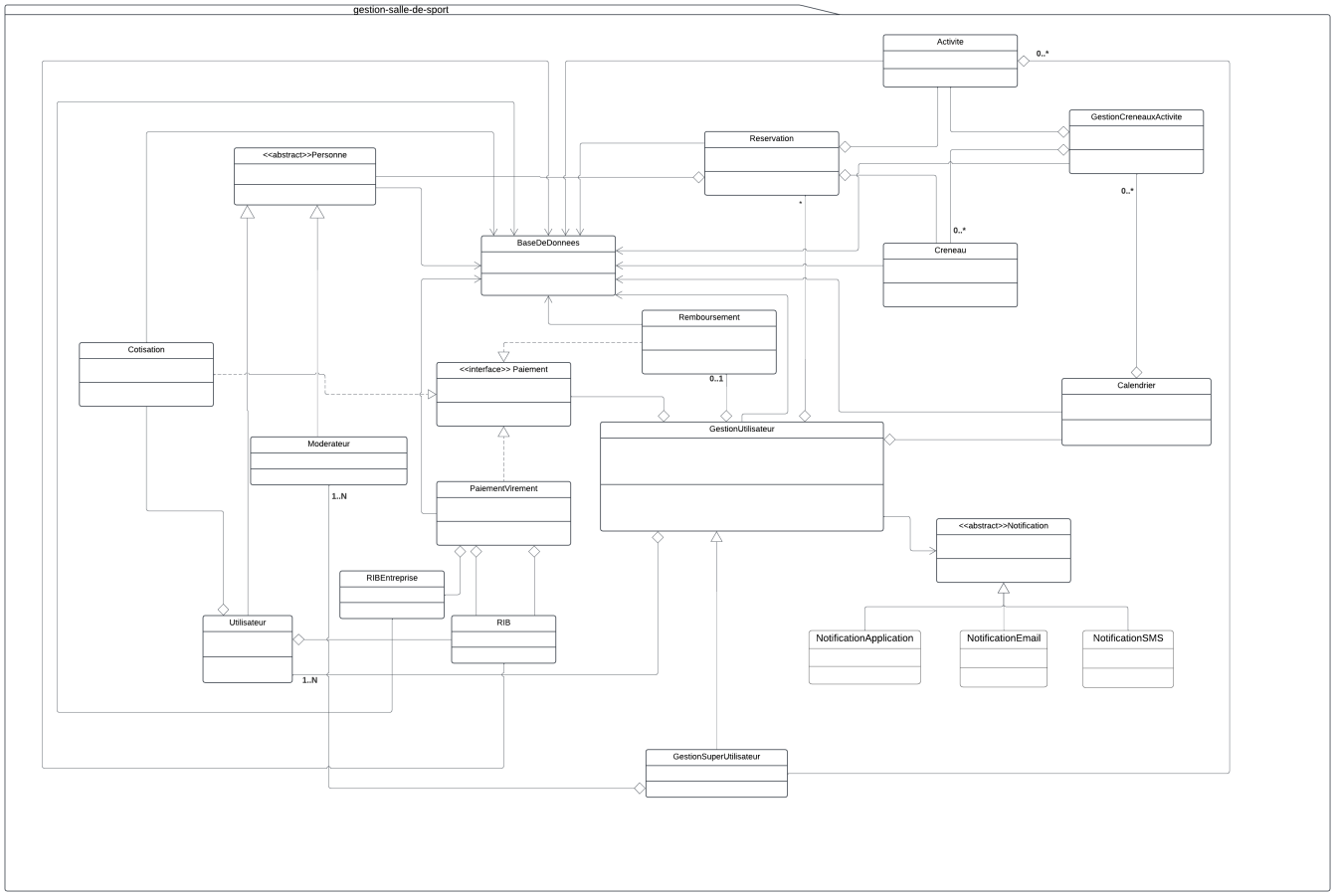

FIGURE 2 – Diagramme de classe

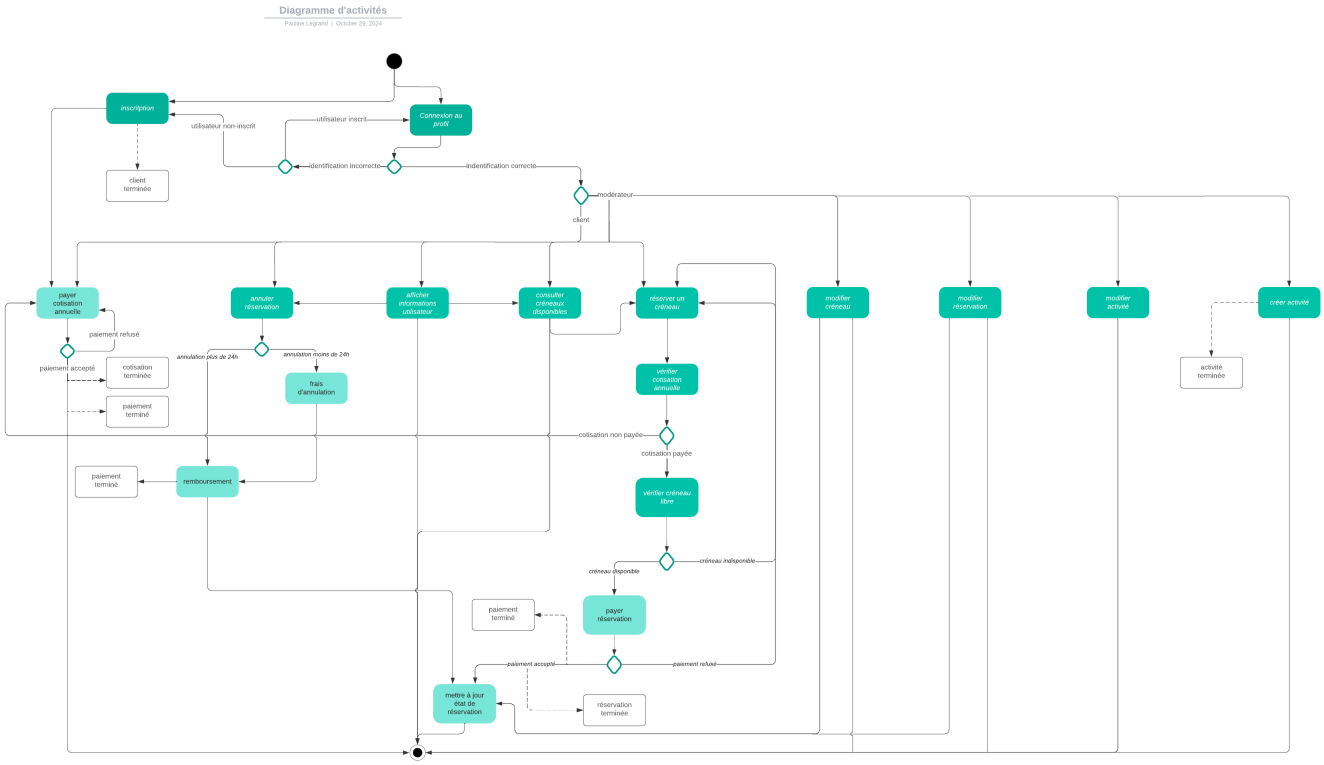

FIGURE 3 – Diagramme d'activité(vue générale)

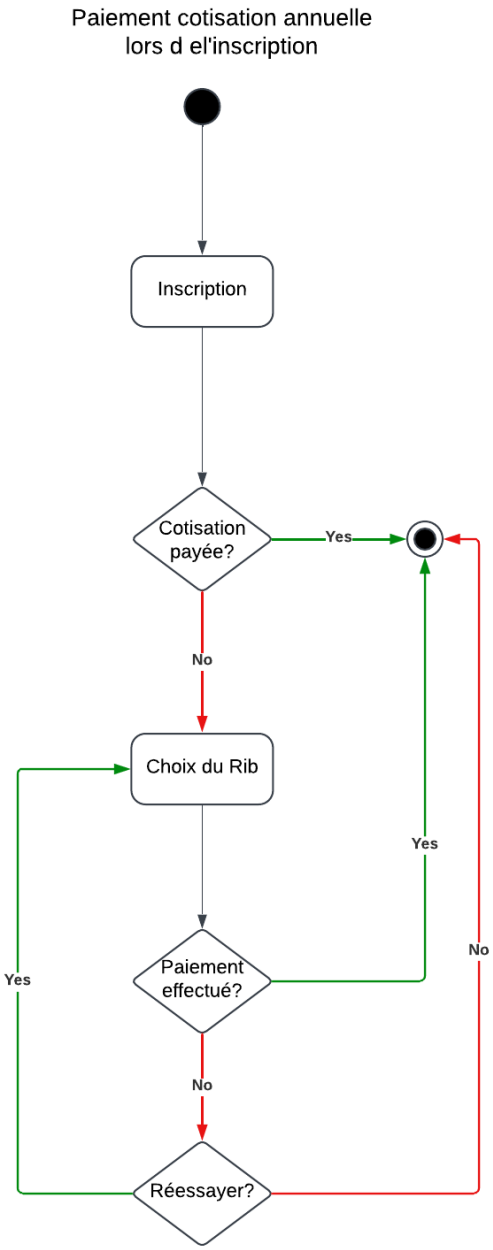

FIGURE 4 – Diagramme d'activité(Paiement Cotisation)

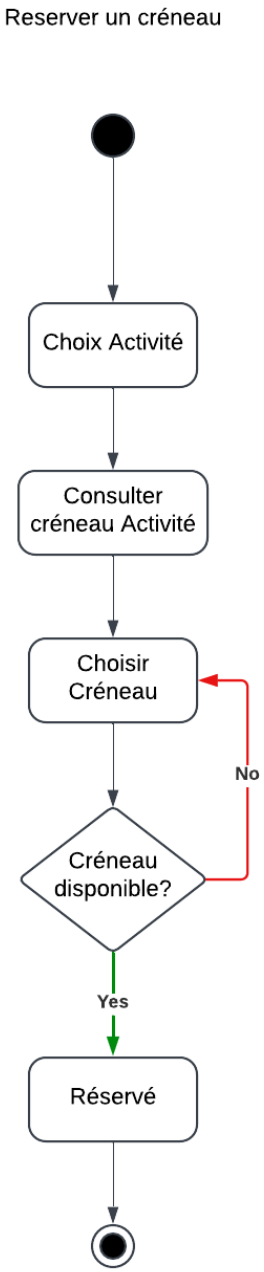

FIGURE 5 – Diagramme d'activité(Réservation d'un créneau)