

**Rapport de Projet Fédérée**

**Spécialité : Sciences de l’Informatique**

**Conception et développement d’une solution "Management de Salle de Sport"**

**République Tunisienne**

**Ministère de l’Enseignement Supérieur**

**Et de la Recherche Scientifique**

**Université de Carthage**

**Institut Supérieur des Technologies de**

**L’Information et de la Communication**

Présenté Par :

Zarrouki Oussema

Ghileb Rayen

Saidani Youssef

Encadré Par :

**Dr. Tebourski Wafa**

Année Universitaire 2023 – 2024

**Contenu**

[**Introduction générale** 3](#_Toc164604072)

[**Chapitre 1: Spécification des besoins** 4](#_Toc164604073)

[1. Introduction 4](#_Toc164604074)

[2. Identification des besoins fonctionnels 4](#_Toc164604075)

[3. Identification des besoins non fonctionnels 4](#_Toc164604076)

[4. Identification des acteurs 5](#_Toc164604077)

[5. Diagramme de cas d’utilisation globale 5](#_Toc164604078)

[6. Backlog de produit 6](#_Toc164604079)

[7. Environnement de travail 6](#_Toc164604080)

[a. Méthodologie de conception 6](#_Toc164604081)

[b. Environnement logiciel 6](#_Toc164604082)

[8. Conclusion 7](#_Toc164604083)

[**Chapitre 2 : Sprint 0** 8](#_Toc164604084)

[1. Introduction 8](#_Toc164604085)

[2. Backlog de sprint 0: 8](#_Toc164604086)

[3. Raffinement du sprint 0: 9](#_Toc164604087)

[4. Conception du sprint 0 : 13](#_Toc164604088)

[**5.** **Realisation de Sprint 0 :** 21](#_Toc164604089)

[6. Conclusion 26](#_Toc164604090)

[**Chapitre 2 : Sprint 1** 27](#_Toc164604091)

[1. Introduction 27](#_Toc164604092)

[2. Backlog de sprint 1: 27](#_Toc164604093)

[3. Conception du sprint 1 : 28](#_Toc164604094)

[4. Raffinement du sprint 1: 34](#_Toc164604095)

[5. Realisation 37](#_Toc164604096)

[6. Conclusion 39](#_Toc164604097)

[**Conclusion Générale** 40](#_Toc164604098)

[**Bibliographie:** 41](#_Toc164604099)

**Introduction générale**

Avec l'arrivée des nouvelles technologies, la manière dont on gère nos activités sportives a changé. Beaucoup de salles de sport utilisent des systèmes informatiques, mais souvent seulement le personnel peut y accéder.

Dans ce cadre notre projet permet de créer un site web pour gérer les salles de sport, accessible à tout le monde, n'importe où.

Les clients pourront voir leurs abonnements, réserver les cours et voir les offres spéciales. En revanche, Cela va aussi faciliter la vie du personnel en leur permettant de gérer les données des clients et les emplois du temps de cours, même en dehors de la salle.

En plus, c'est plus sûr parce que les données sont stockées en ligne, donc on ne risque pas de les perdre en cas de problème avec l'ordinateur.

Pour réaliser ce projet de manière efficace, nous avons choisi d'adopter la méthodologie agile Scrum. Cette approche nous permettra de travailler de manière collaborative et de livrer des fonctionnalités de haute qualité de manière itérative.

***Notre rapport est organisé en trois chapitres :***

Pour le Chapitre 1 sur les Spécifications de besoins, nous allons définir en détail les exigences fonctionnelles et non fonctionnelles de notre application, identifier les acteurs clés et établir un backlog de produit pour guider le développement.

Dans le Chapitre 2, consacré à la Gestion de la salle de sport du côté du staff, nous allons aborder les fonctionnalités spécifiques nécessaires à la gestion interne de la salle, telles que l'authentification, la gestion des abonnements, des clients, des personnels et des offres, ainsi que la planification des cours.

En ce qui concerne le Chapitre 3 sur la Gestion de la salle de sport du côté du client, notre objectif sera de mettre en place les fonctionnalités permettant aux clients d'accéder facilement à leur compte, de consulter les offres, les cours disponibles, de faire des réclamations, et plus encore.

**Chapitre 1: Spécification des besoins**

# Introduction

Dans ce premier chapitre, nous définissons les besoins fondamentaux pour notre application de gestion de salles de sport. Nous identifions les aspects fonctionnels et non fonctionnels, ainsi que les acteurs clés du système. À travers un diagramme de cas d'utilisation, nous visualisons les interactions entre les utilisateurs et le système. Ensuite, nous détaillons le backlog de produit, mettant en avant les fonctionnalités prioritaires. Enfin, nous présentons la méthodologie de conception et l'environnement logiciel utilisé pour le développement.

# Identification des besoins fonctionnels

Les besoins fonctionnels définissent les actions spécifiques que l'application doit permettre aux utilisateurs d'accomplir, telles que consulter les abonnements, réserver des cours et gérer les offres. Ils représentent les fonctionnalités essentielles pour répondre aux besoins des utilisateurs.

Pour garantir le bon fonctionnement de notre application web de gestion de salles de sport, nous devons identifier les besoins fonctionnels essentiels. Voici les principaux besoins du système :

Gérer abonnements : Les clients doivent pouvoir consulter leurs abonnements.

Consulter cour : Les utilisateurs doivent avoir la possibilité de voir les cours en ligne.

Consulter offres : Les clients peuvent consulter les offres disponibles ainsi que leurs prix.

Gestion des données client : Le personnel doit pouvoir gérer les informations des clients de manière sécurisée et organisée.

Consultation du calendrier des cours : Les clients et le personnel doivent avoir accès au calendrier des cours par semaine.

# Identification des besoins non fonctionnels

Les besoins non fonctionnels décrivent les exigences de performance, de sécurité et d'utilisabilité de l'application. Cela inclut des aspects tels que la sécurité des données, la rapidité de réponse et la convivialité de l'interface. Ces critères sont essentiels pour garantir une expérience utilisateur optimale.

Pour assurer la réussite de notre application web de gestion de salles de sport, nous nous engageons à respecter les besoins non fonctionnels suivants :

Sécurité des données : Nous devons nous assurer que les informations des utilisateurs sont bien protégées. Cela signifie que nous utilisons des mesures de sécurité avancées, comme des mots de passe forts et le cryptage des données.

Performance : Notre application doit être rapide et disponible en permanence, même lorsque beaucoup de personnes l'utilisent en même temps. Cela garantit une expérience utilisateur fluide.

Compatibilité multiplateforme : Notre application doit pouvoir être utilisée sur différents appareils, tels que des ordinateurs, des tablettes ou des smartphones. Peu importe le dispositif utilisé, l'expérience utilisateur doit être cohérente et facile.

Facilité d'utilisation : L'interface de notre application doit être simple et intuitive, avec des instructions claires pour faciliter la navigation et l'utilisation pour tous les utilisateurs.

# Identification des acteurs

Dans la conception de systèmes, un acteur est toute entité qui interagit avec le système. Cela peut être un utilisateur humain, un autre système informatique ou même un processus externe. Les acteurs sont identifiés en fonction de leur rôle ou de leur responsabilité dans le système et sont essentiels pour déterminer les besoins et les fonctionnalités du système.

Pour notre application de gestion de salles de sport, nous identifions trois acteurs principaux :

Personnel de la salle de sport : Ce sont les membres du personnel chargés de gérer les opérations internes de la salle de sport. Ils ont des autorisations pour ajouter, modifier ou supprimer des membres du personnel, ainsi que pour gérer les offres, les abonnements et le calendrier des cours.

Clients : Ce sont les personnes qui utilisent les services de la salle de sport. Ils ont accès à l'application pour consulter les offres disponibles, les cours proposés, leurs abonnements et également pour soumettre des réclamations.

Administrateur : C'est un utilisateur spécial ayant des privilèges étendus. Son rôle principal est de gérer les membres du personnel en ajoutant ou en supprimant des membres du staff.

# ****Diagramme de cas d’utilisation globale****

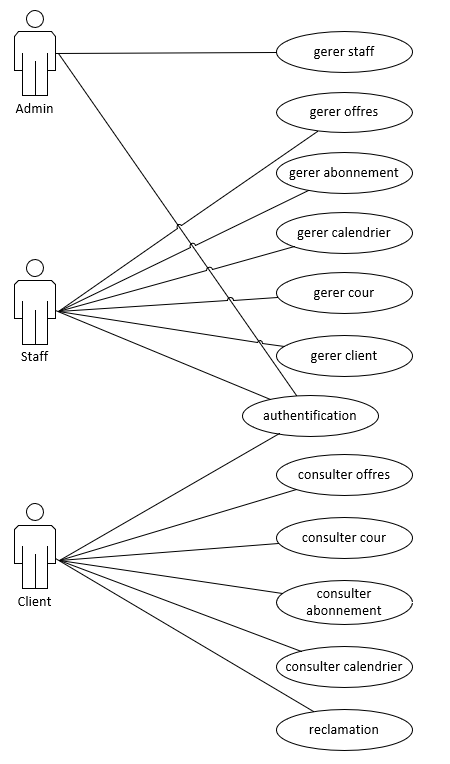


Figure 1 : diagramme de cas d’utilisation globale

|  |  |  |
| --- | --- | --- |
| Element de Backlog | Priorité | Estimation |
| En tant que Admin, je peux gérer le personnel | 1 | Elevée |
| En tant que Personnel, je peux gérer les offres | 1 | Elevée |
| En tant que Personnel, je peux gérer les abonnements. | 1 | Elevée |
| En tant que Personnel, je peux gérer le calendrier. | 1 | Elevée |
| En tant que Personnel, je peux gérer les clients. | 1 | Elevée |
| En tant que Personnel, je peux gérer les cours. | **1** | Elevée |
| En tant que Personnel, je peux m'identifier. | 1 | Elevée |
| En tant que Client, je peux m'identifier. | 2 | Moyene |
| En tant que Client, je peux consulter les offres. | 2 | Moyene |
| En tant que Client, je peux consulter le calendrier. | 2 | Moyene |
| En tant que Client, je peux consulter les cours. | 2 | Moyene |
| En tant que Client, je peux consulter mon abonnement. | 2 | Moyene |
| En tant que Client, je peux faire une réclamation. | 3 | Faible |

# ****Backlog de produit****

Figure 2 : backlog de produit

# Environnement de travail

Enfin, nous décrivons l'environnement de travail nécessaire pour le développement de l'application. Cela comprend la méthodologie de conception que nous utiliserons, ainsi que les outils logiciels nécessaires pour la mise en œuvre du projet.

1. Méthodologie de conception

La méthode **SCRUM** est une m´méthode agile créée en 2002, dont le nom est un terme emprunté au rugby qui signifie ”la mêlée”. Elle s’appuie sur le découpage des projets en itérations nommées ”Sprints”.

1. Environnement logiciel

Dans notre projet de développement d'une application web de gestion de salles de sport, nous avons adopté une méthodologie agile et sélectionné des outils logiciels appropriés pour assurer une conception efficace et une implémentation réussie. Voici notre environnement de travail simplifié :

* **HTML :**

HTML (pourHyperText Markup Language, qu'on peut traduire en « langage de balisage hypertexte ») est le langage utilisé pour structurer une page web et son contenu. On peut par exemple organiser le contenu en un ensemble de paragraphes, une liste d'éléments, utiliser des images ou des tableaux de données.

* **CSS :**

CSS (Cascading Style Sheets ou en français "Feuilles de style en cascade") est utilisé pour styliser et mettre en page des pages Web - par exemple, pour modifier la police, la couleur, la taille et l'espacement de votre contenu, le scinder en plusieurs colonnes ou ajouter des animations et autres éléments décoratifs.

* **JavaScript :**

JavaScript est un langage de programmation utilisé par les développeurs pour concevoir des sites web interactifs. Les fonctions JavaScript peuvent permettre d'améliorer l'expérience utilisateur d'un site web, de la mise à jour des flux de médias sociaux à l'affichage d'animations et de cartes interactives. En tant que langage de script côté client, c'est l'une des principales technologies du web. Lors de la navigation sur Internet, à tout moment vous pouvez par exemple voir un carrousel d'images, un menu déroulant « Cliquer pour afficher » ou le changement dynamique de la couleur des éléments d'une page web. Tout cela est possible grâce à JavaScript.

* **Angular :**

Angular est un framework open source JavaScript développé par Google. Ce framework est utilisé pour développer des applications web et mobile. Avec cette technologie, on réalise des interfaces de type monopage ou “one page” qui fonctionnent sans rechargement de la page web.

* **PHP :**

Hypertext Preprocessor, plus connu sous son sigle PHP (sigle auto-référentiel), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur web, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale.

* **VSCode :**

Visual Studio Code (ou VSCODE) est un éditeur de code gratuit et multiplateforme qui intègre de nombreuses fonctionnalités et permet d'en intégrer de nouvelles grâce à un large catalogue de plugins. Disponible sous Windows, Linux et macOS, et sur le web.

* **XAMPP :**

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

* **FIGMA :**

Figma est une application spécialisée dans la conception des interfaces Web et mobiles. Cet outil a été conçu dans l'esprit d'un Design System, partageable par une équipe de designers. En ce sens, c'est un outil puissant et complètement dans les tendances actuelles.

* **MySql :**

MySQL est un système de base de données d'Oracle, utilisé dans le monde entier, et qui permet de gérer des bases de données. Il est basé sur l'algèbre relationnelle et se retrouve en premier lieu utilisé pour le stockage de données de différents services Web.

# Conclusion

Dans ce premier chapitre, nous avons défini les besoins essentiels de notre application de gestion de salles de sport, couvrant à la fois les aspects fonctionnels et non fonctionnels. Nous avons également identifié les principaux acteurs du système et présenté notre méthodologie de conception basée sur Scrum, ainsi que notre environnement logiciel. Ces éléments fournissent une base solide pour le développement futur de notre application.

**Chapitre 2 : Sprint 0**

# ****Introduction****

Dans ce deuxième chapitre, nous nous focalisons sur la gestion pratique de la salle de sport du point de vue du personnel. Nous débutons en récapitulant le sprint 0, notre première phase cruciale. Ensuite, nous détaillons l'identification du backlog, le raffinement des éléments du sprint 0, la conception et la réalisation des fonctionnalités essentielles. Enfin, nous évaluons notre progression et planifions les prochaines étapes. Ce chapitre offre une vue claire de la gestion de la salle de sport côté personnel, en mettant en avant les processus et fonctionnalités principaux.

# ****Backlog de sprint 0:****

Figure 3 :Backlog du sprint 0

|  |  |  |
| --- | --- | --- |
| Element de Backlog | Priorité | Estimation |
| En tant que Admin, je peux m'identifier. | 1 | Elevée |
| En tant que Admin, je peux gérer le personnel | 1 | Elevée |
| En tant que Personnel, je peux gérer les offres | 1 | Elevée |
| En tant que Personnel, je peux gérer les abonnements. | 1 | Elevée |
| En tant que Personnel, je peux gérer le calendrier. | 1 | Elevée |
| En tant que Personnel, je peux gérer les clients. | 1 | Elevée |
| En tant que Personnel, je peux gérer les cours. | **1** | Elevée |
| En tant que Personnel, je peux m'identifier. | 1 | Elevée |

# ****Raffinement du sprint 0:****

Dans cette section, nous passons en revue les diverses rafinnement des cas d'utilisation envisagées pour le premier sprint (sprint 0).

**rafinnement de cas d’utilisation <<s’authentifier>>**

|  |  |
| --- | --- |
| C u | authentification |
| Acteur | Admin |
| Precondition | Avoir un compte |
| Postcondition | Connecté, echoué |
| Description de Scénarii P | L’admin saisit son identifiant (email ou nom d'utilisateur) et son mot de passe.  Le système vérifie les informations d'identification du client.  Si les informations sont correctes, le client est redirigé vers son espace personnel.  Sinon, un message d'erreur est affiché, invitant le client à réessayer. |
|  |  |

Figure 4: table de rafinnement de cas d’utilisation <<s’authentifier>>

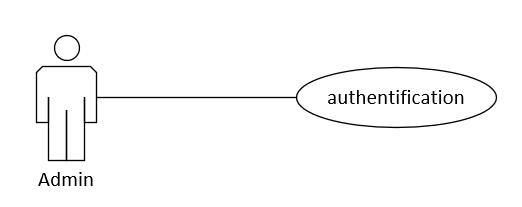


Figure 5: rafinnement de cas d’utilisation <<s’authentifier>>

**rafinnement de cas d’utilisation <<gerer personnel>>**

|  |  |
| --- | --- |
| C u | gerer personnel |
| Acteur | Admin |
| Precondition | authentication |
| Postcondition | Personnel gérée |
| Description de Scénarii Principal | Le système affiche l'interface contenant la liste du personnel.  L'administrateur sélectionne l'opération de gestion du personnel.  Le système affiche l'interface correspondant au choix de l'administrateur. |
|  |  |

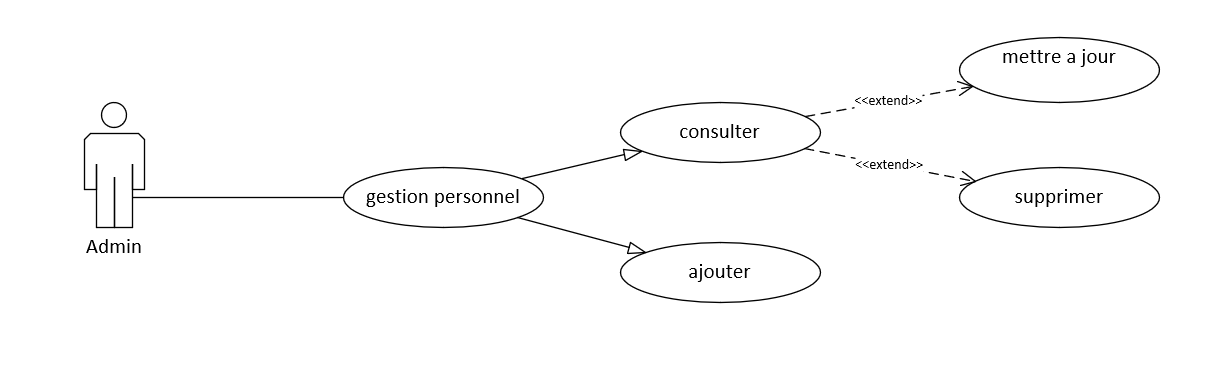
Figure 6: table de rafinnement de cas d’utilisation <<gérer personnel>>

Figure 7: rafinnement de cas d’utilisation << gérer personnel >>

**rafinnement de cas d’utilisation <<s’authentifier>>**

|  |  |
| --- | --- |
| C u | authentification |
| Acteur | Personnel |
| Precondition | Avoir un compte |
| Postcondition | Connecté, echoué |
| Description de Scénarii P | Le personnel saisit son identifiant (email ou nom d'utilisateur) et son mot de passe.  Le système vérifie les informations d'identification du client.  Si les informations sont correctes, le client est redirigé vers son espace personnel.  Sinon, un message d'erreur est affiché, invitant le client à réessayer. |
|  |  |

Figure 8: table de rafinnement de cas d’utilisation << s’authentifier >>

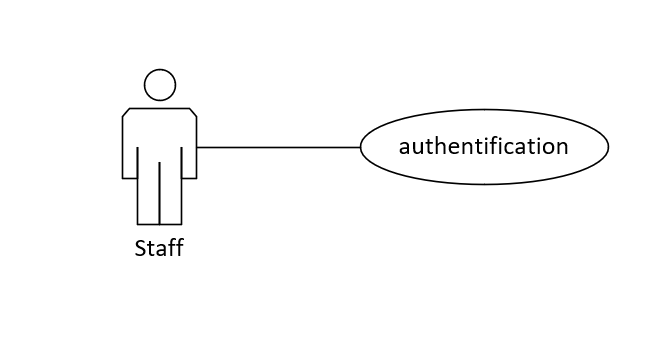


Figure 9: rafinnement de cas d’utilisation << s’authentifier >>

**rafinnement de cas d’utilisation <<gerer offres>>**

|  |  |
| --- | --- |
| C u | Gerer offres |
| Acteur | Personnel |
| Precondition | authentication |
| Postcondition | Offre gérée |
| Description de Scénarii Principal | Le système affiche l'interface contenant la liste des offres.  L'personnel sélectionne l'opération de gestion des offres.  Le système affiche l'interface correspondant au choix du personnel. |
|  |  |

Figure 10: table de rafinnement de cas d’utilisation << Gérer offres >>

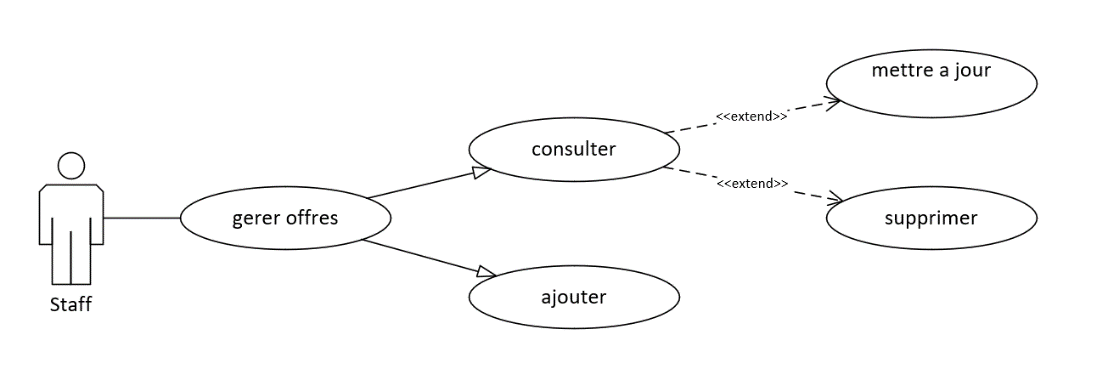


Figure 11: rafinnement de cas d’utilisation << Gérer offres >>

**rafinnement de cas d’utilisation <<gerer abonnement>>**

|  |  |
| --- | --- |
| C u | Gerer abonnement |
| Acteur | Personnel |
| Precondition | authentication |
| Postcondition | Abonnement gérée |
| Description de Scénarii Principal | Le système affiche l'interface contenant la liste des abonnements.  L'personnel sélectionne l'opération de gestion des abonnements.  Le système affiche l'interface correspondant au choix du personnel. |
|  |  |

Figure 12: table de rafinnement de cas d’utilisation << Gérer abonnement >>

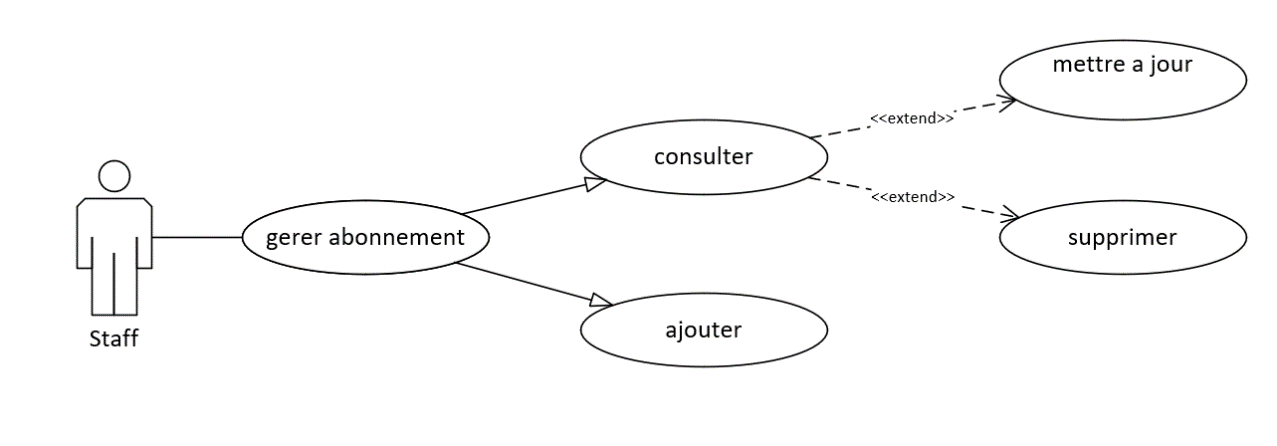


Figure 13: rafinnement de cas d’utilisation << Gérer abonnement >>

**rafinnement de cas d’utilisation <<gerer calendrier>>**

|  |  |
| --- | --- |
| C u | Gerer calendrier |
| Acteur | Personnel |
| Precondition | authentication |
| Postcondition | Calendrier gérée |
| Description de Scénarii Principal | Le système affiche l'interface contenant le calendrier.  L'personnel sélectionne l'opération de gestion du calendrier.  Le système affiche l'interface correspondant au choix du personnel. |
|  |  |

Figure 14: table de rafinnement de cas d’utilisation << Gérer calendrier >>

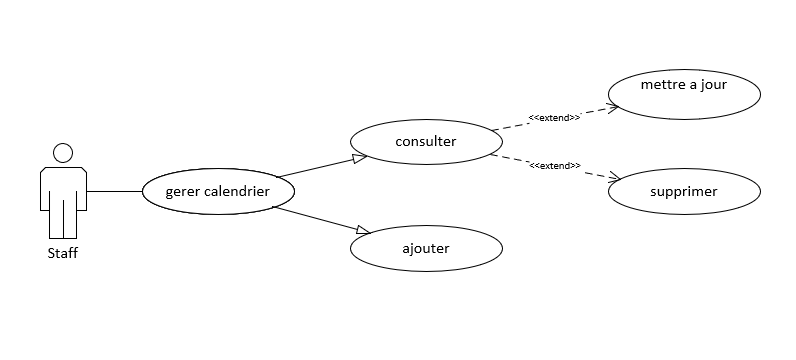


Figure 15: rafinnement de cas d’utilisation << Gérer calendrier >>

**rafinnement de cas d’utilisation <<gerer cour>>**

|  |  |
| --- | --- |
| C u | Gerer cour |
| Acteur | Personnel |
| Precondition | authentication |
| Postcondition | cour gérée |
| Description de Scénarii Principal | Le système affiche l'interface contenant la liste des cours.  L'personnel sélectionne l'opération de gestion des cours.  Le système affiche l'interface correspondant au choix du personnel. |
|  |  |

Figure 16: table de rafinnement de cas d’utilisation << Gérer cour >>

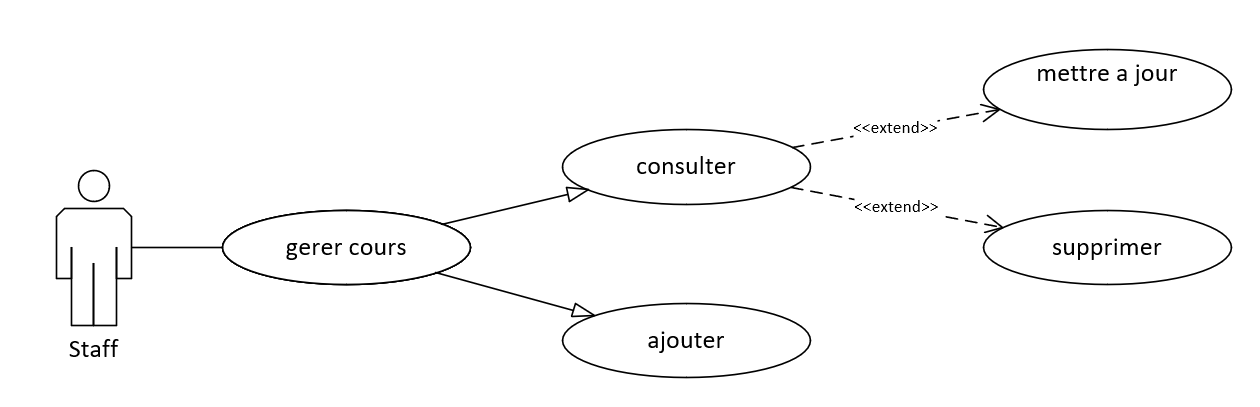


Figure 17: rafinnement de cas d’utilisation << Gérer cour >>

**rafinnement de cas d’utilisation <<gerer client>>**

|  |  |
| --- | --- |
| C u | Gerer client |
| Acteur | Personnel |
| Precondition | authentication |
| Postcondition | client gérée |
| Description de Scénarii Principal | Le système affiche l'interface contenant la liste des clients.  L'personnel sélectionne l'opération de gestion des clients.  Le système affiche l'interface correspondant au choix du personnel. |
|  |  |

Figure 18: table de rafinnement de cas d’utilisation << Gérer client >>

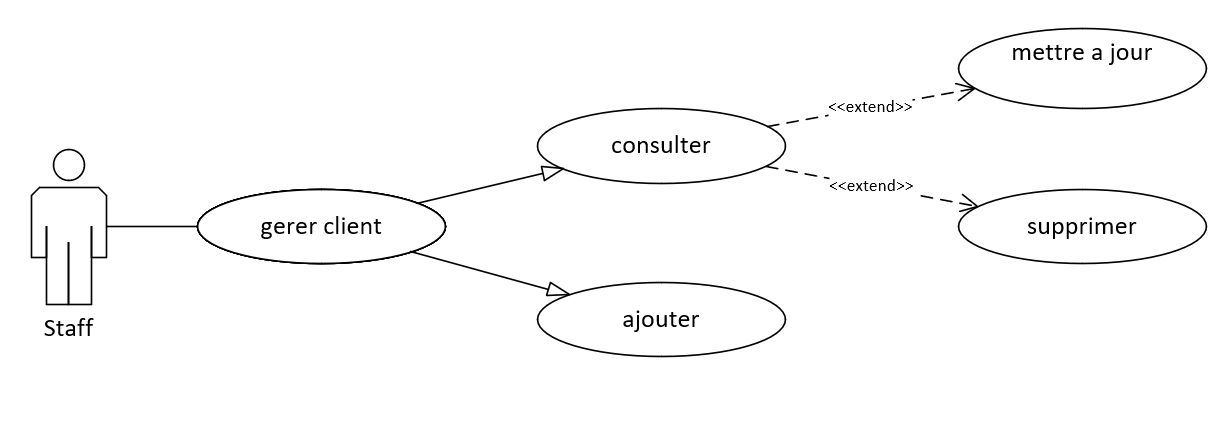


Figure 19: rafinnement de cas d’utilisation << Gérer client >>

# ****Conception du sprint 0 :****

Dans cette section, nous allons présenter la conception des différents cas d'utilisation, ce qui implique les diagrammes de séquence et de classe relatif associés à chaque cas d'utilisation.

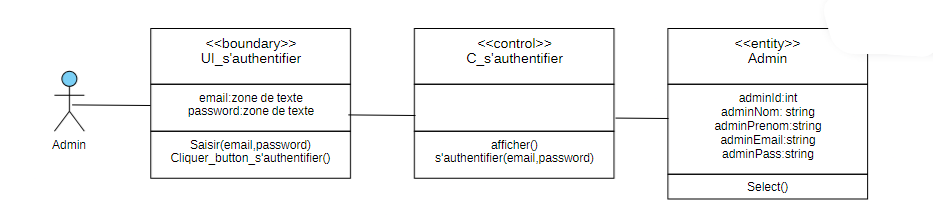
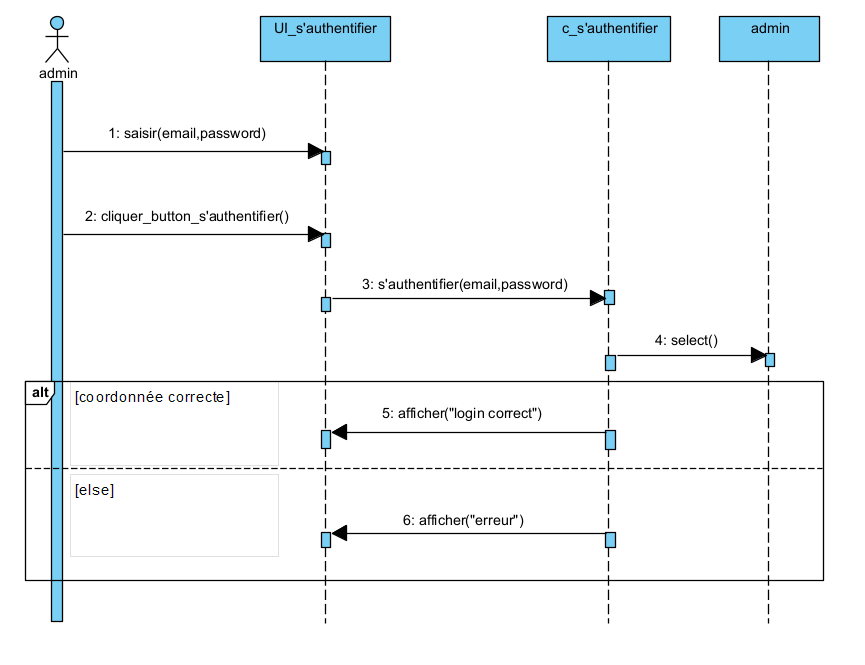
**Conception du cas d’utilisation « s’authentifier »**

Figure 20 : Diagramme de classe relatif au cas d’utilisation « s’authentifier » 

- Figure 21 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « s’inscrire »

**Conception du cas d’utilisation « s’authentifier »**

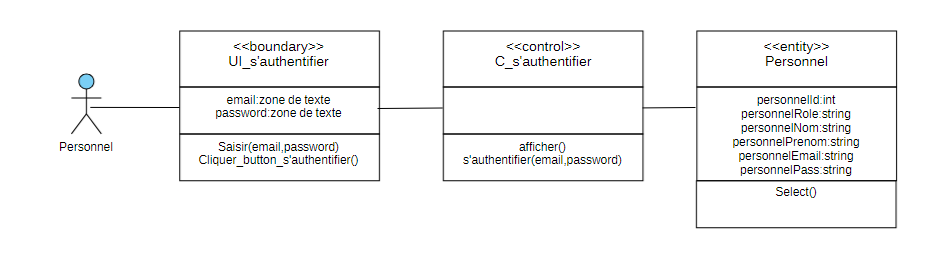
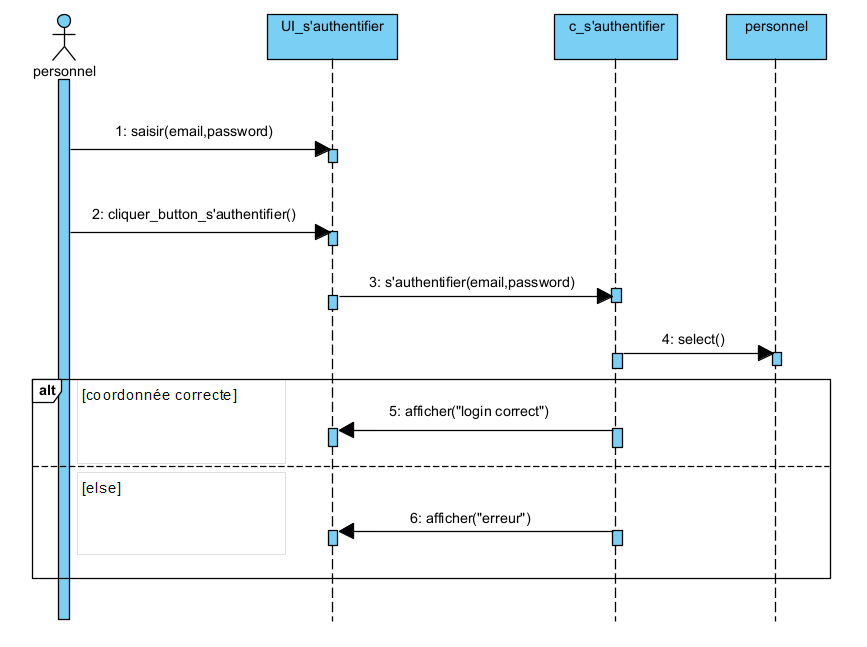


Figure 22 : Diagramme de classe relatif au cas d’utilisation « s’authentifier » 

- Figure 23 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « s’inscrire »

**Conception du cas d’utilisation « gérer personnel »**

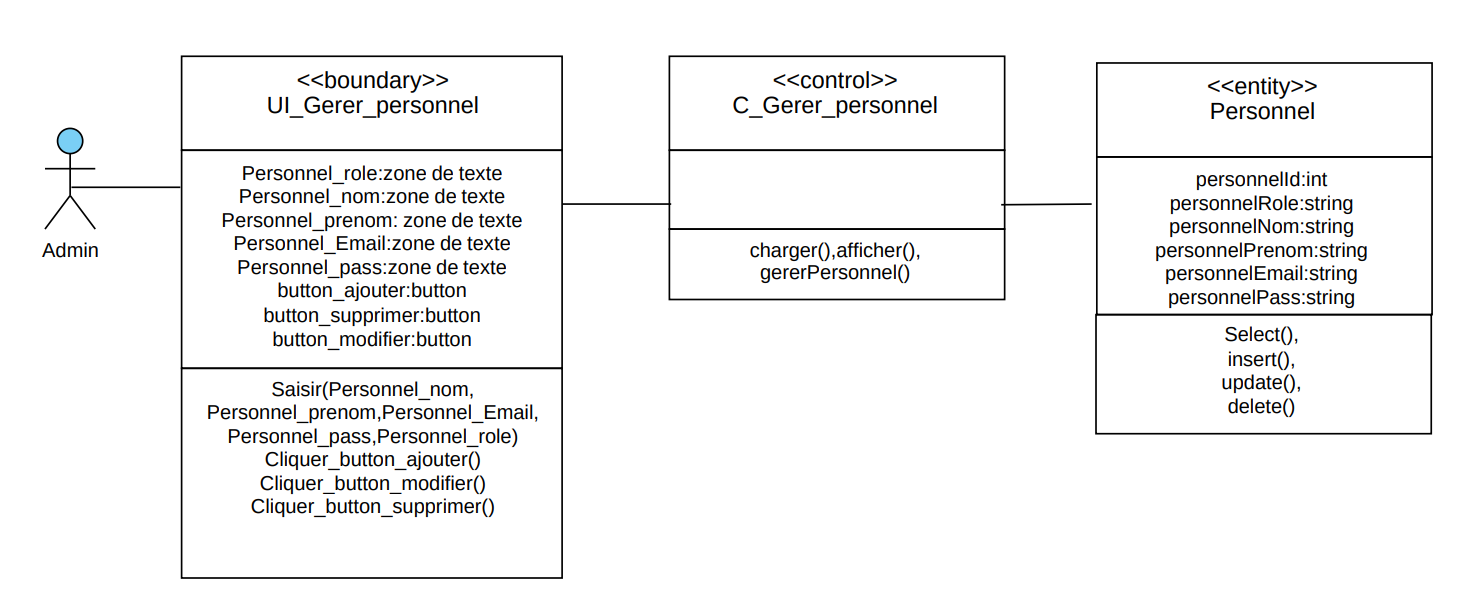
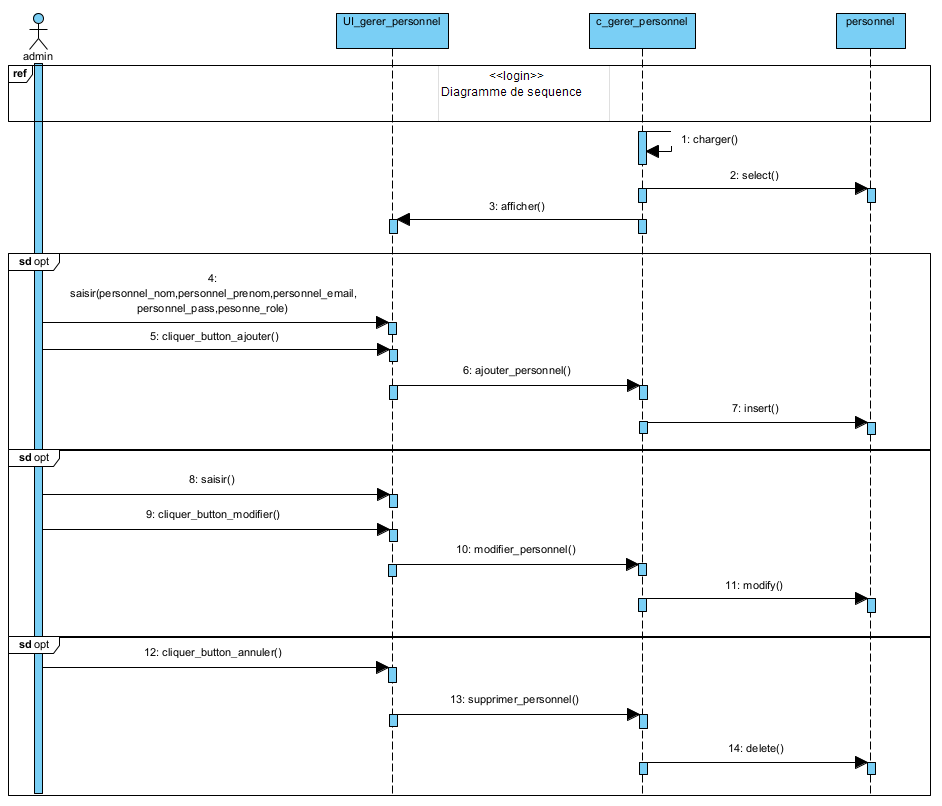


Figure 24 : Diagramme de classe relatif au cas d’utilisation « gérer personnel » 

- Figure 25 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « gérer personnel »

**Conception du cas d’utilisation « gérer offres »**

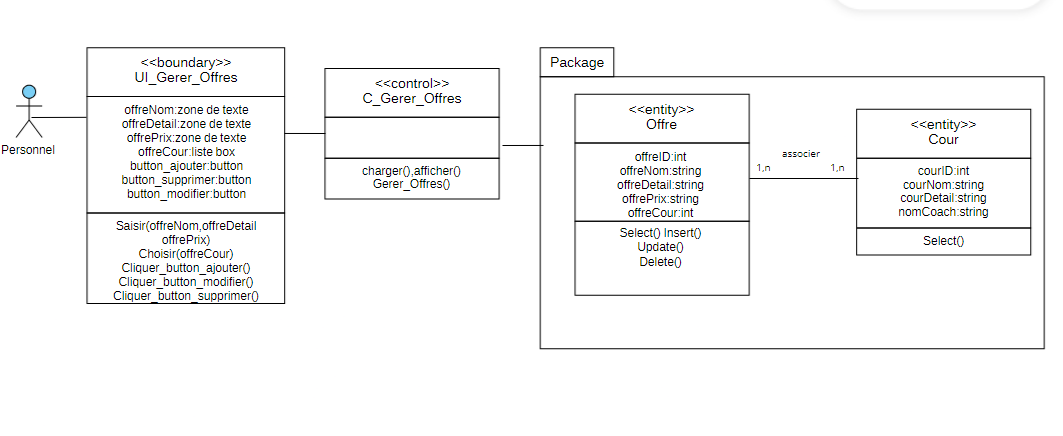
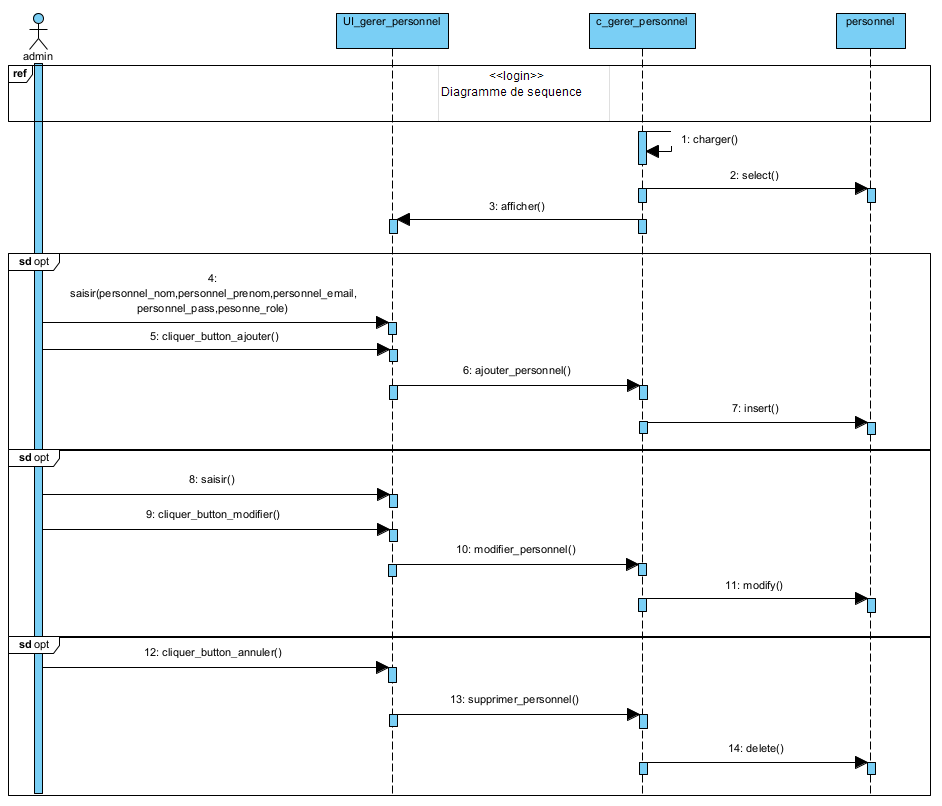


Figure 26 : Diagramme de classe relatif au cas d’utilisation « gérer offres »



- Figure 27 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « gérer offres »

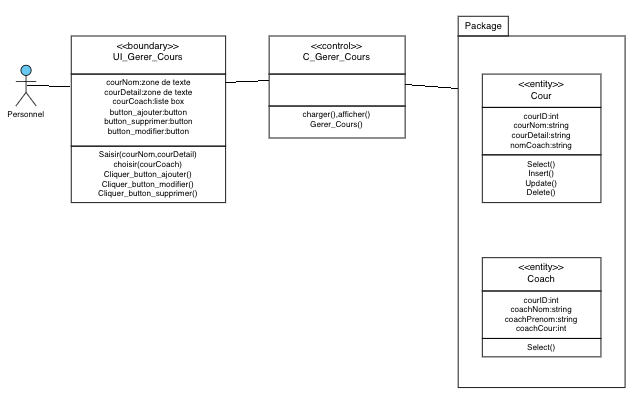
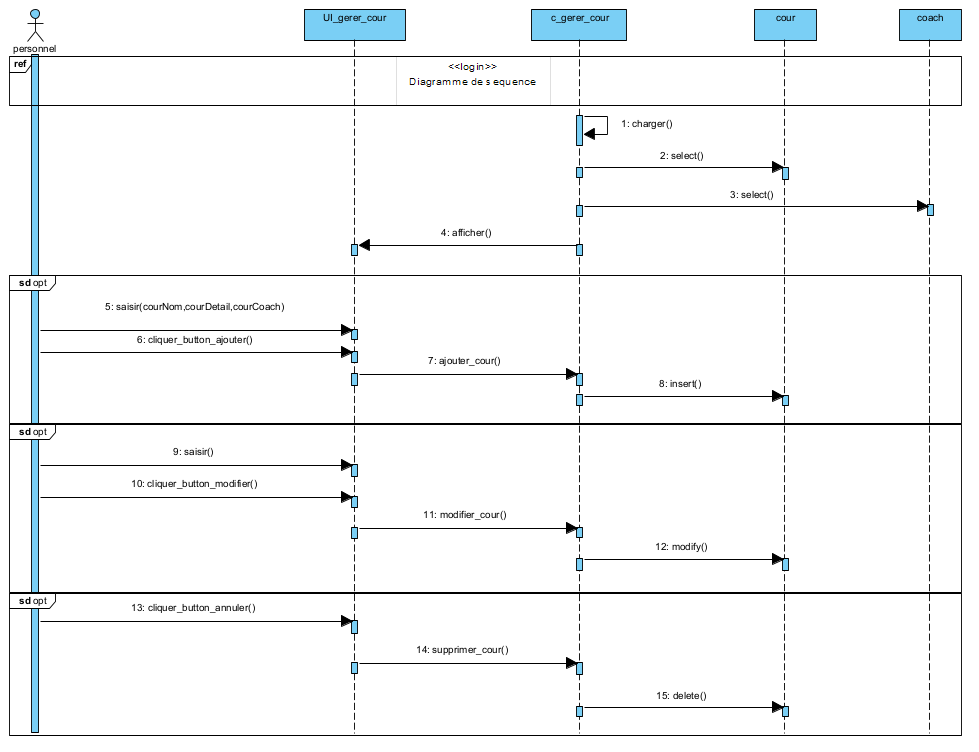
**Conception du cas d’utilisation « gérer cours »**

Figure 28 : Diagramme de classe relatif au cas d’utilisation « gérer cours »

- Figure 29 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « gérer cours »

**Conception du cas d’utilisation « gérer clients »**

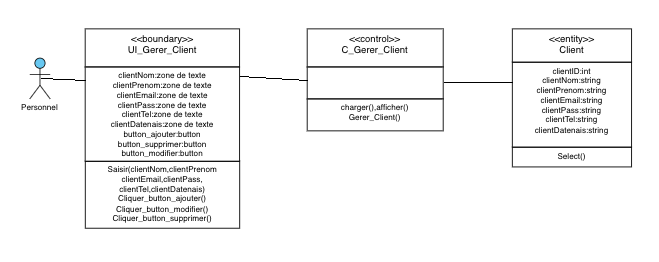
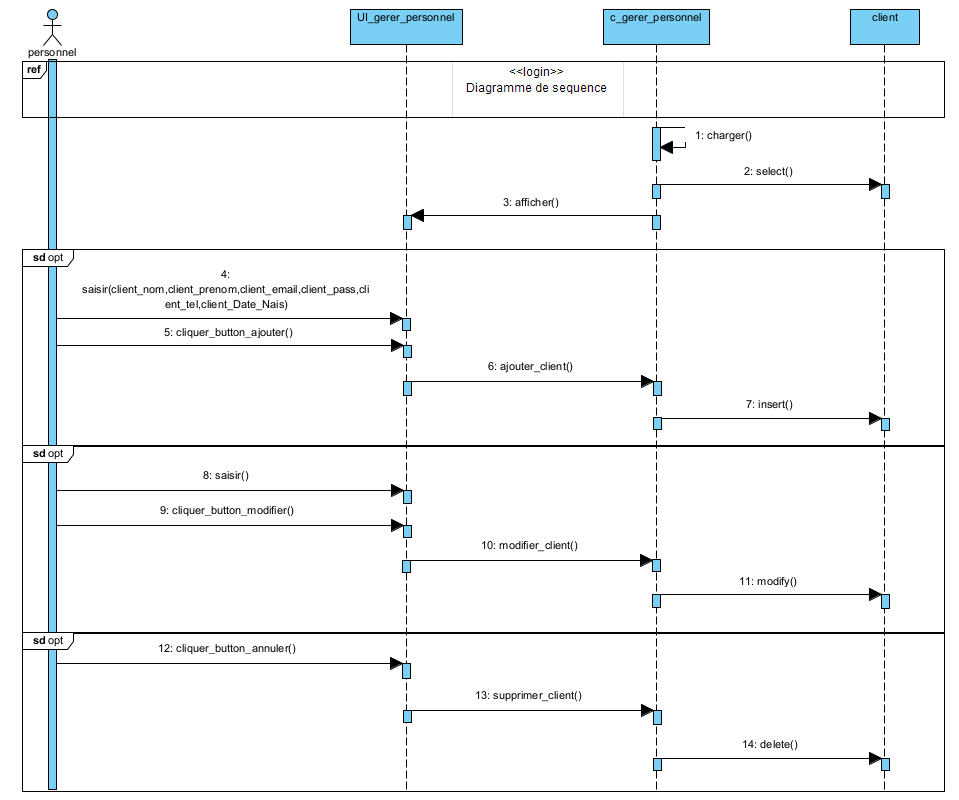


Figure 30 : Diagramme de classe relatif au cas d’utilisation « gérer clients »



- Figure 31 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « gérer clients »

**Conception du cas d’utilisation « gérer abonnements »**

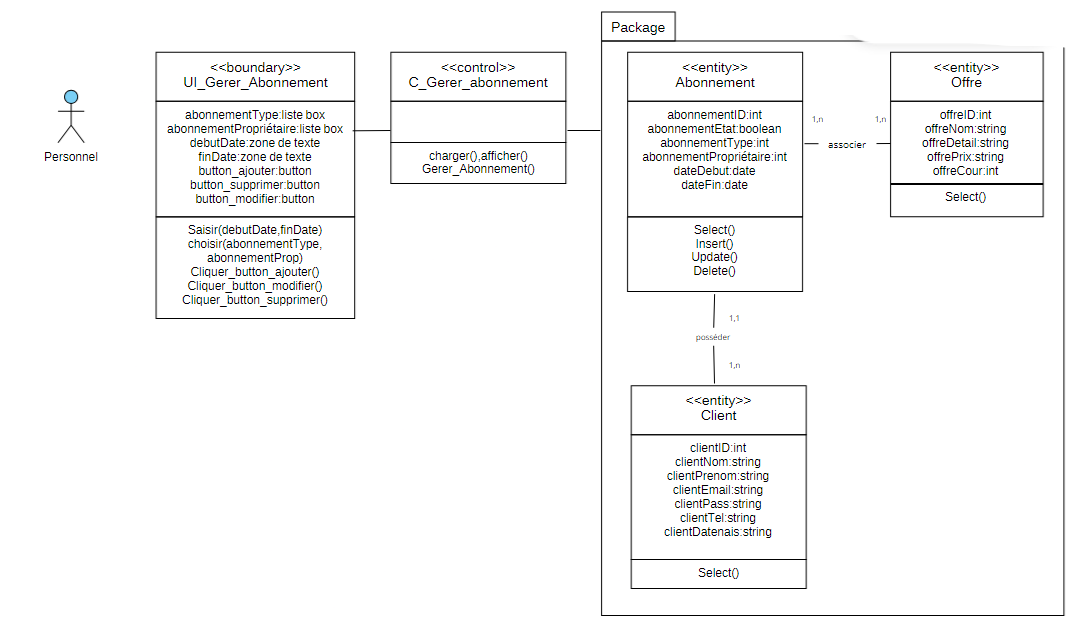
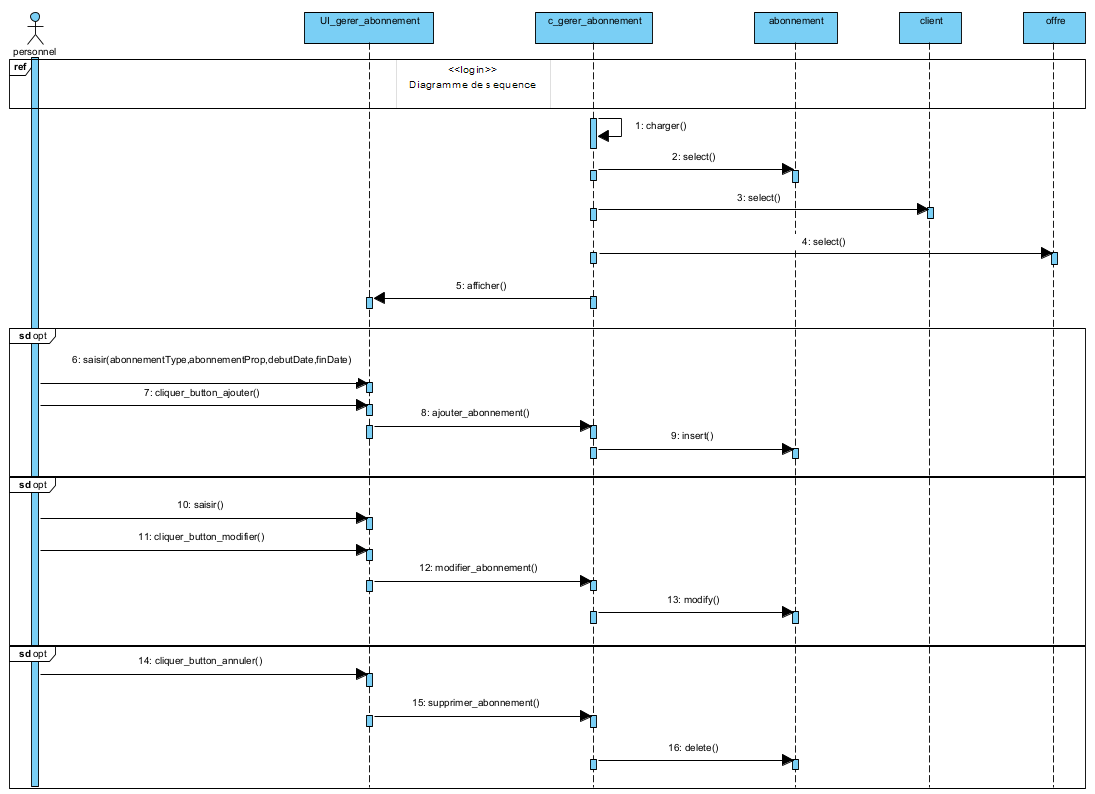


Figure 32 : Diagramme de classe relatif au cas d’utilisation « gérer abonnements »



- Figure 33 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « gérer abonnements »

**Conception du cas d’utilisation « gérer calendrier »**

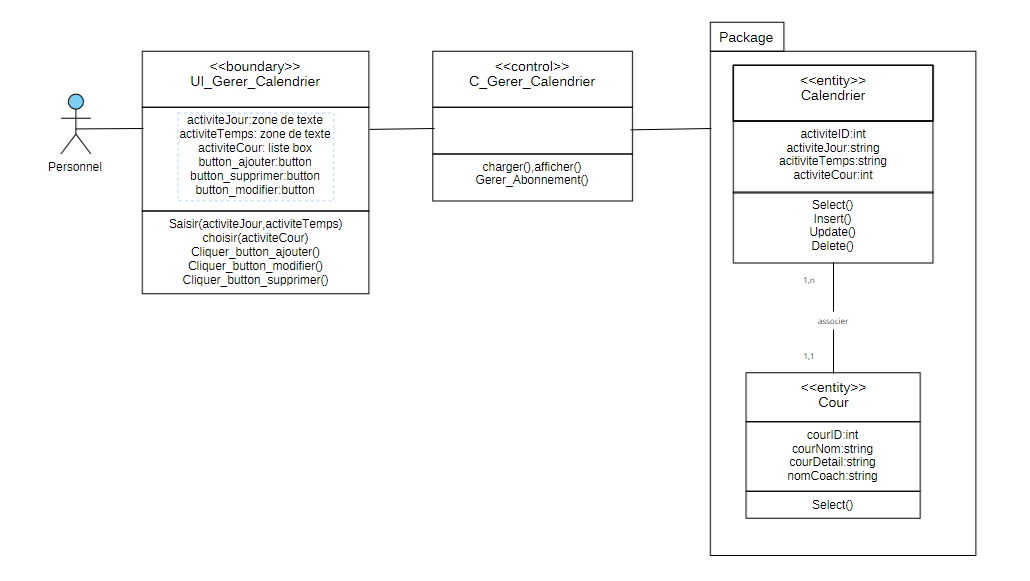
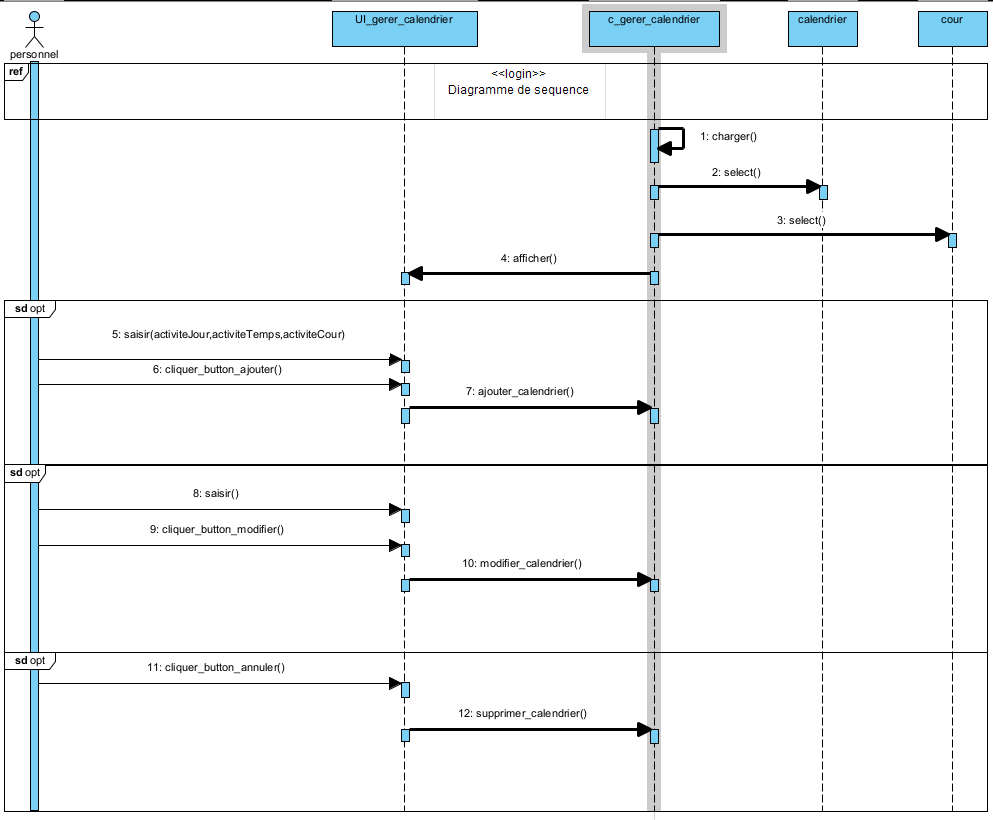
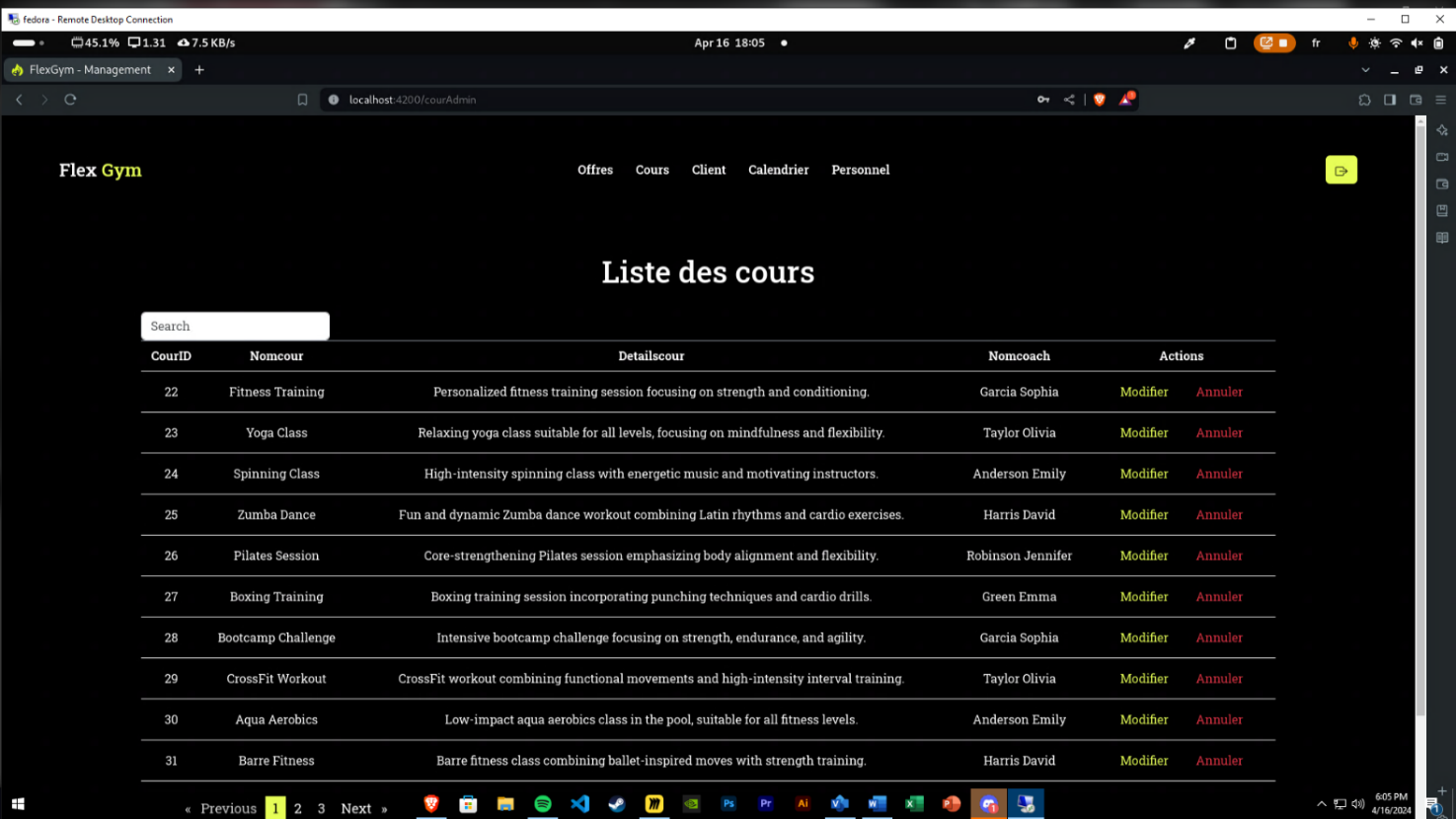
-Diagramme de classe : cette figure représente le diagramme de classe relatif au cas d’utilisation « gérer calendrier »

Figure 34 : Diagramme de classe relatif au cas d’utilisation « gérer calendrier »

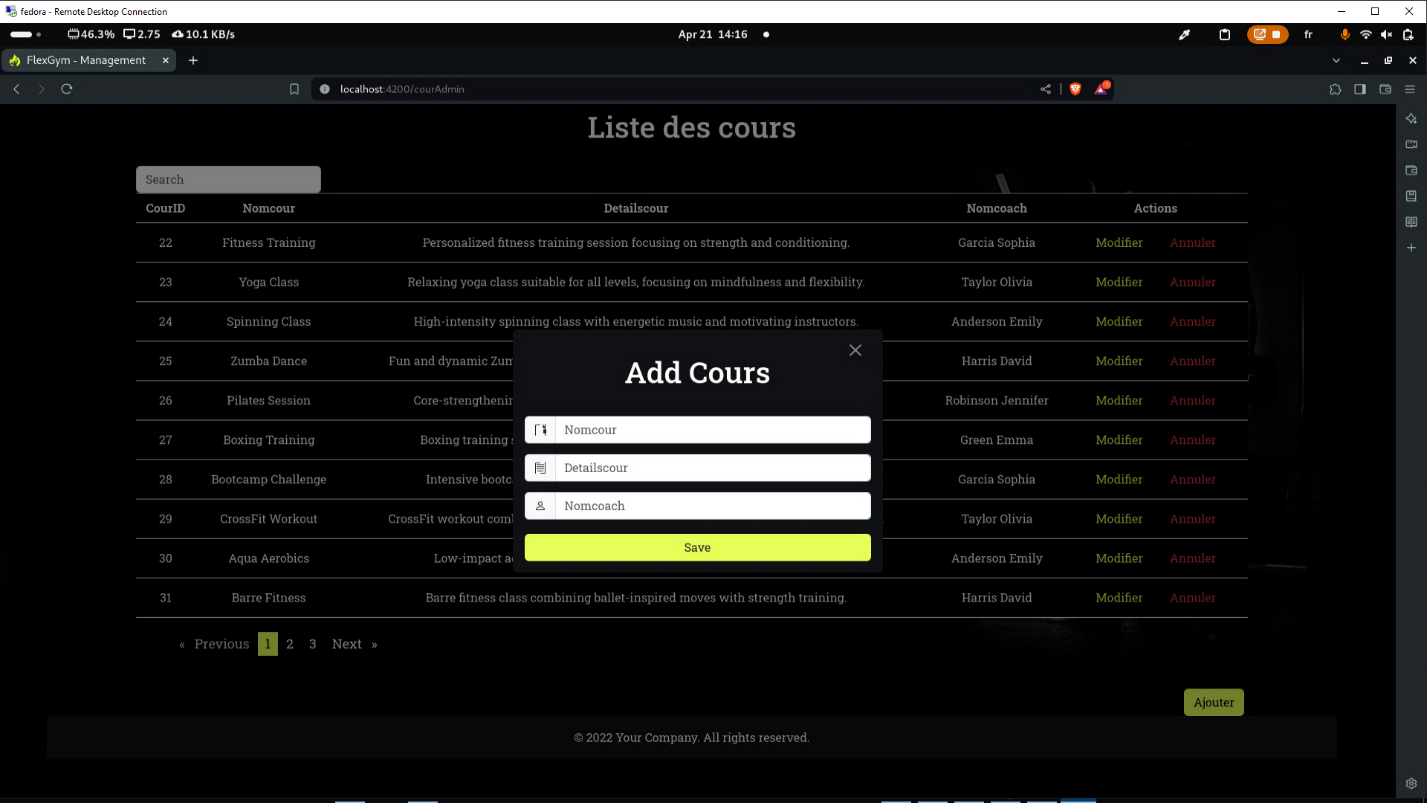
- Figure 35 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « gérer calendrier »

1. **Realisation de Sprint 0 :**

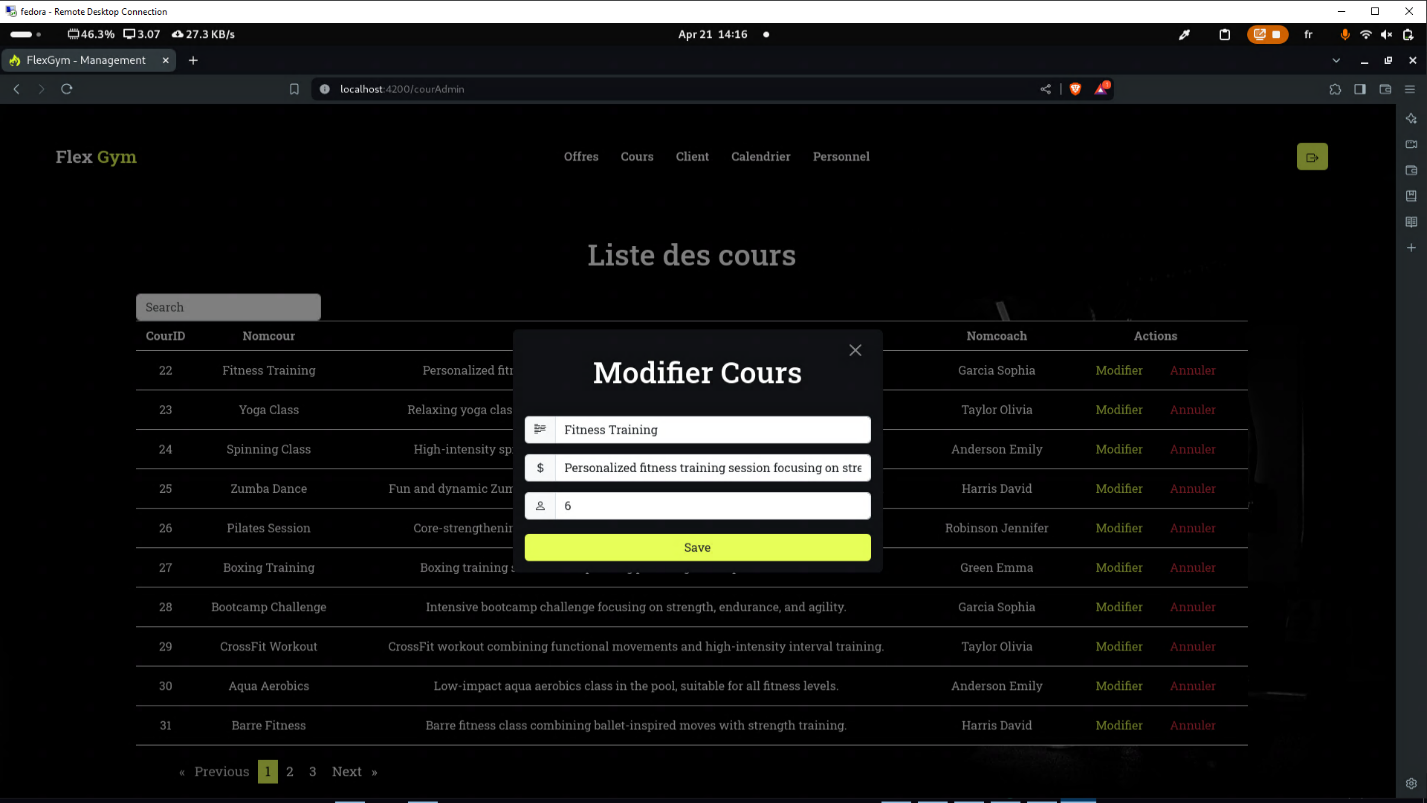
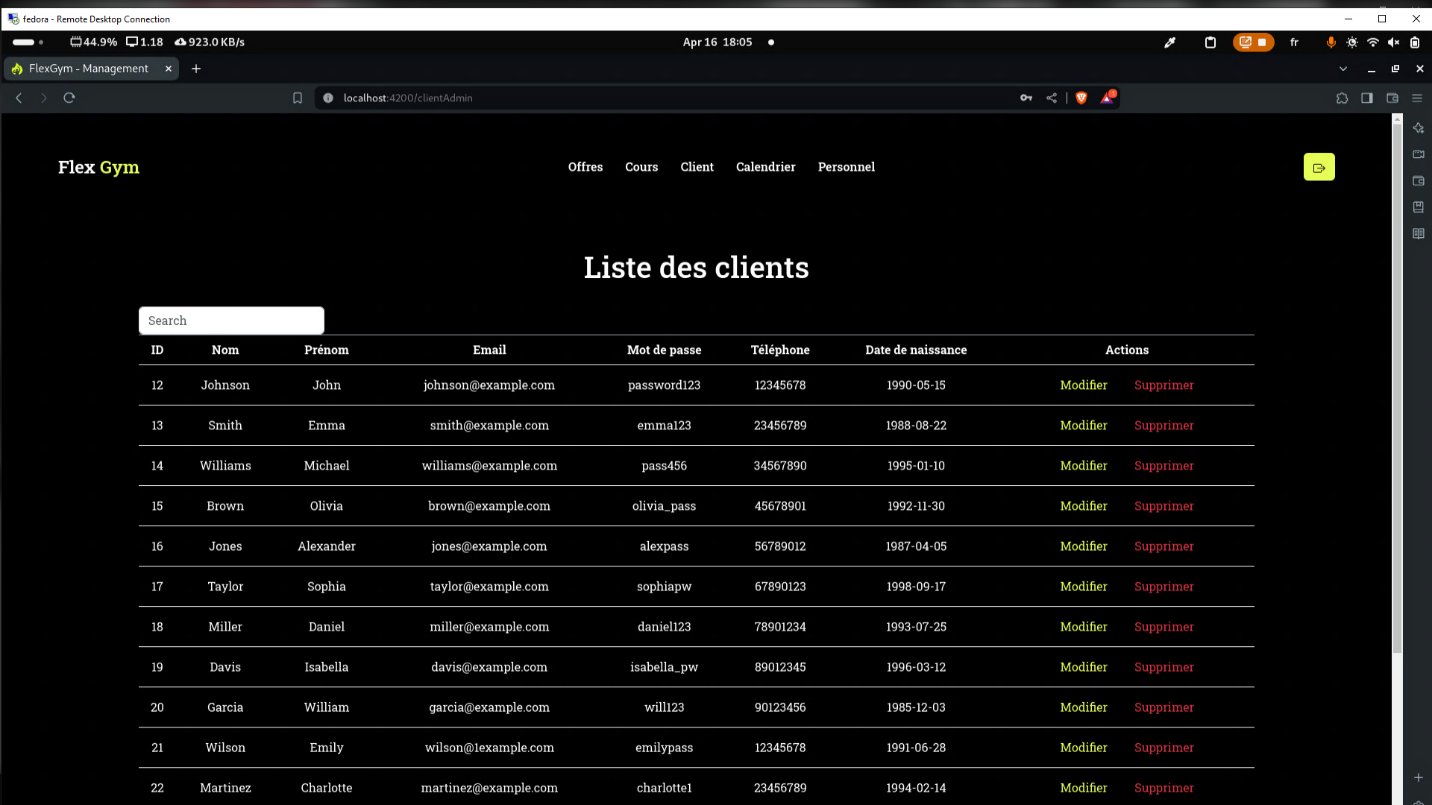
**-Figure 36 : Gérer cours**



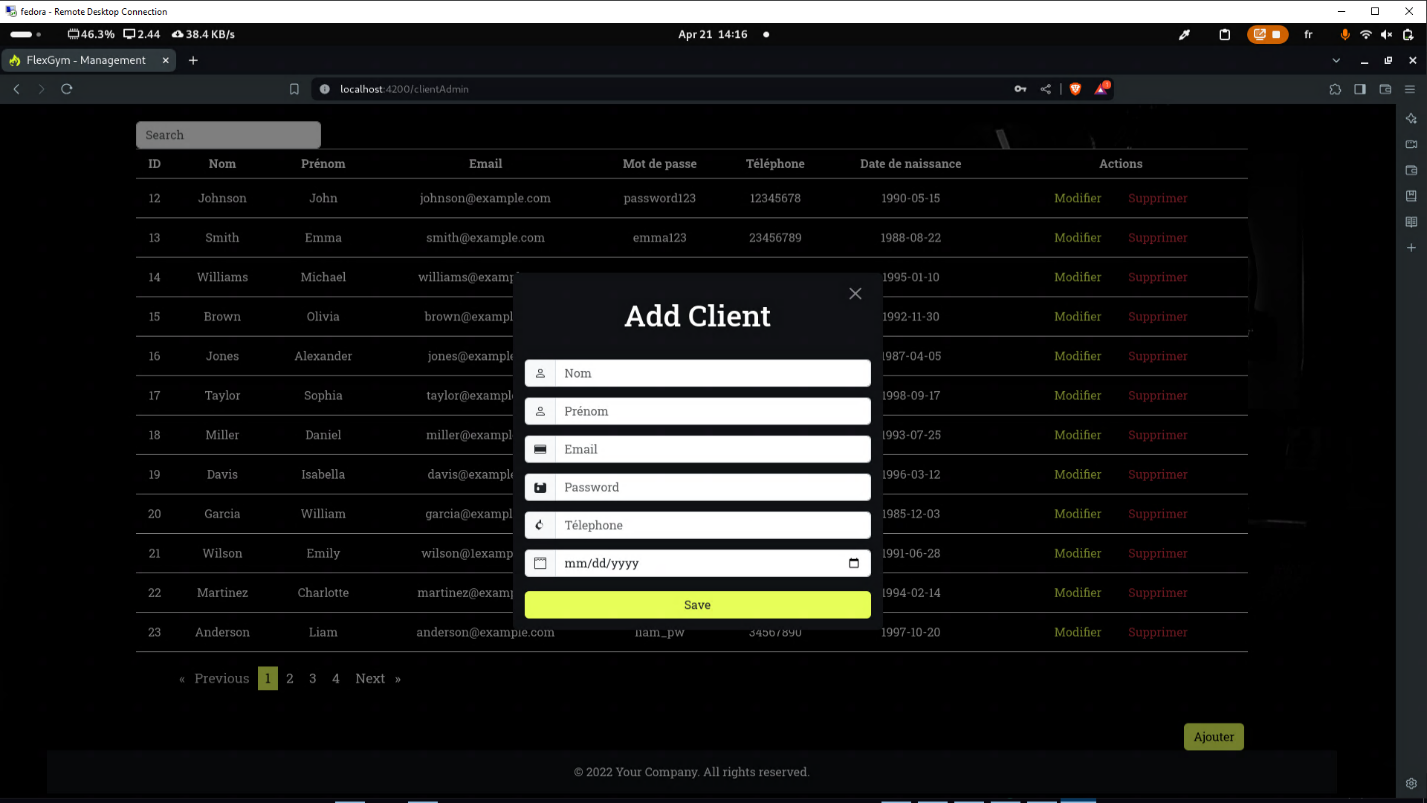
**-Figure 37 : Ajouter Cour**

****

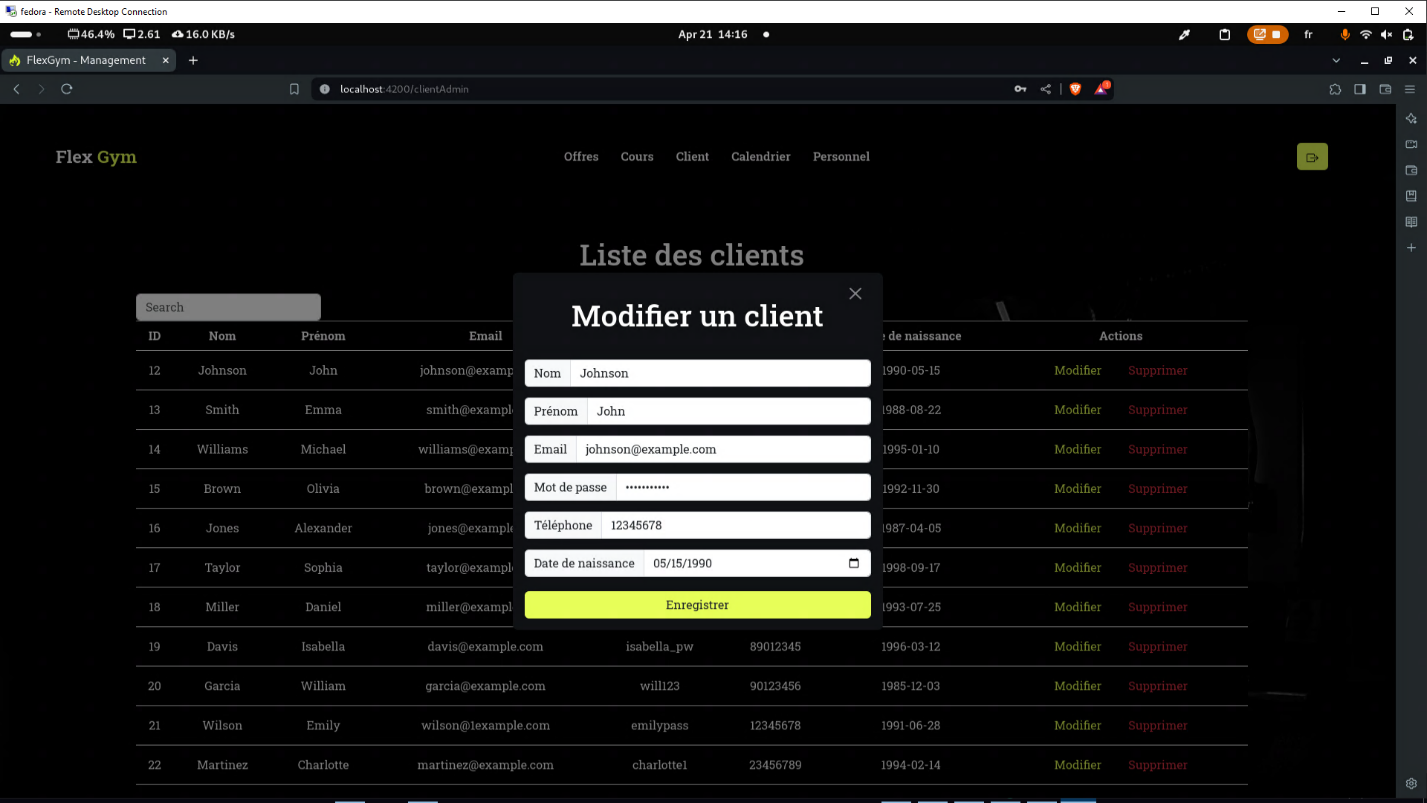
**-Figure 38 : Modifier Cour**

**-Figure 39 : Gérer clients**

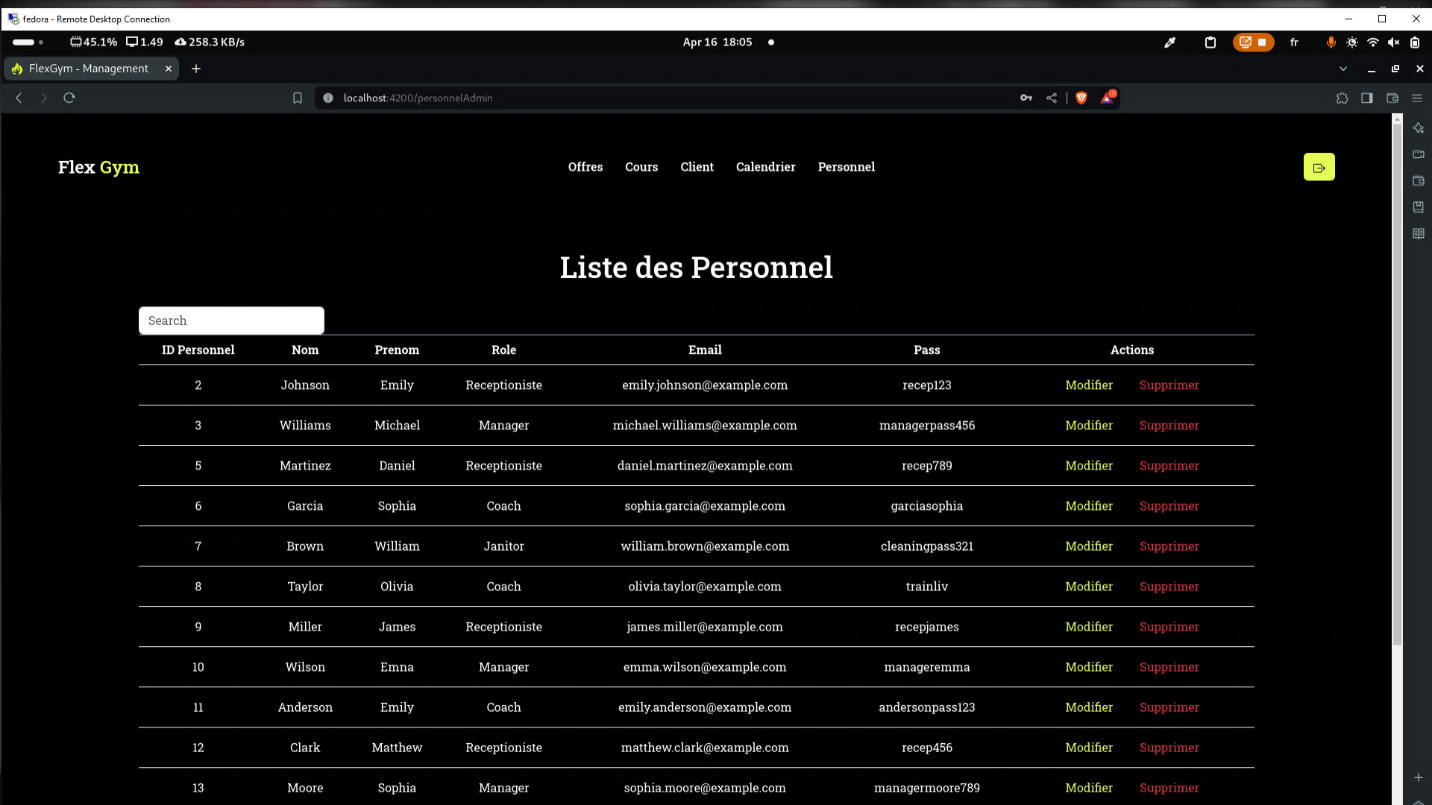
**-Figure 40 : Ajouter Client**

****

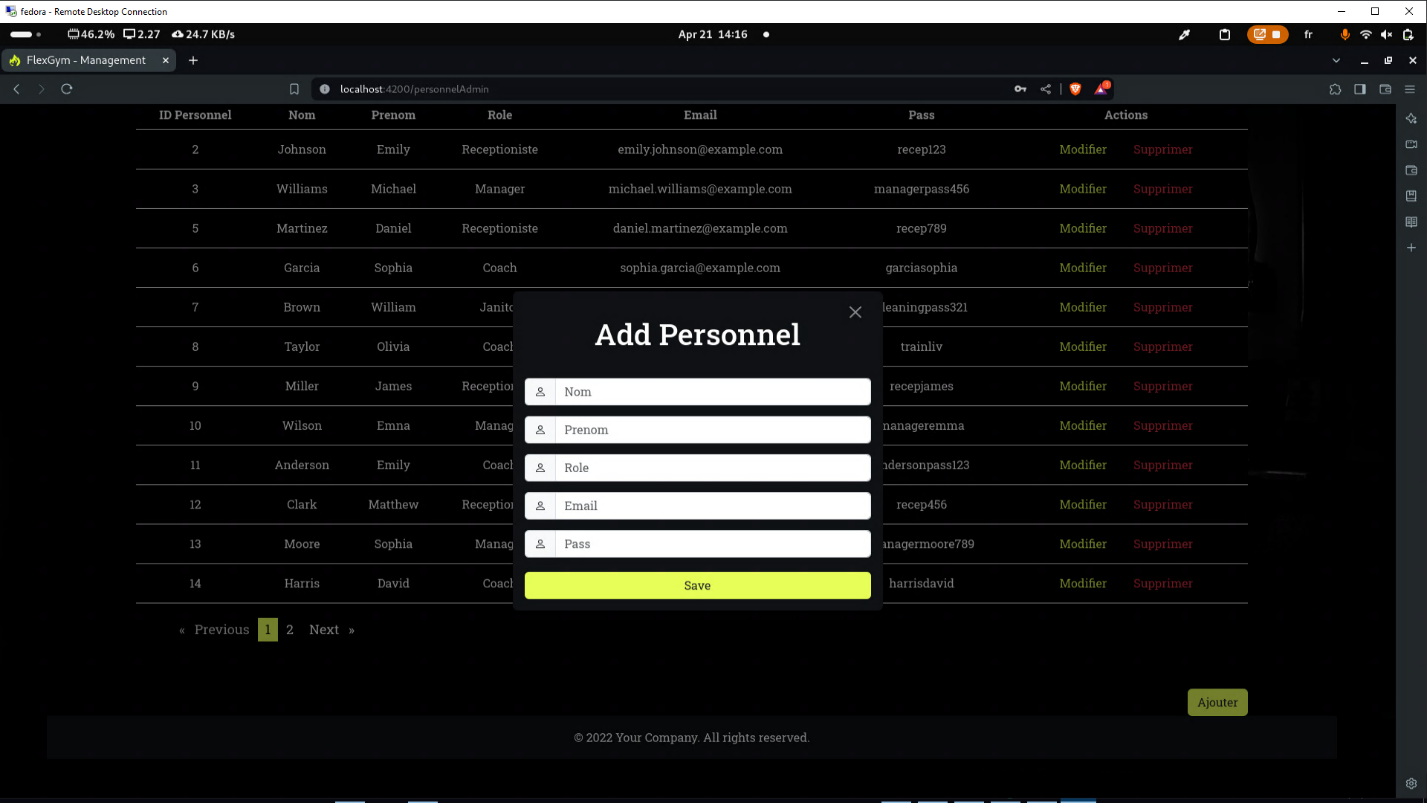
**-Figure 41 : Modifier Client**

****

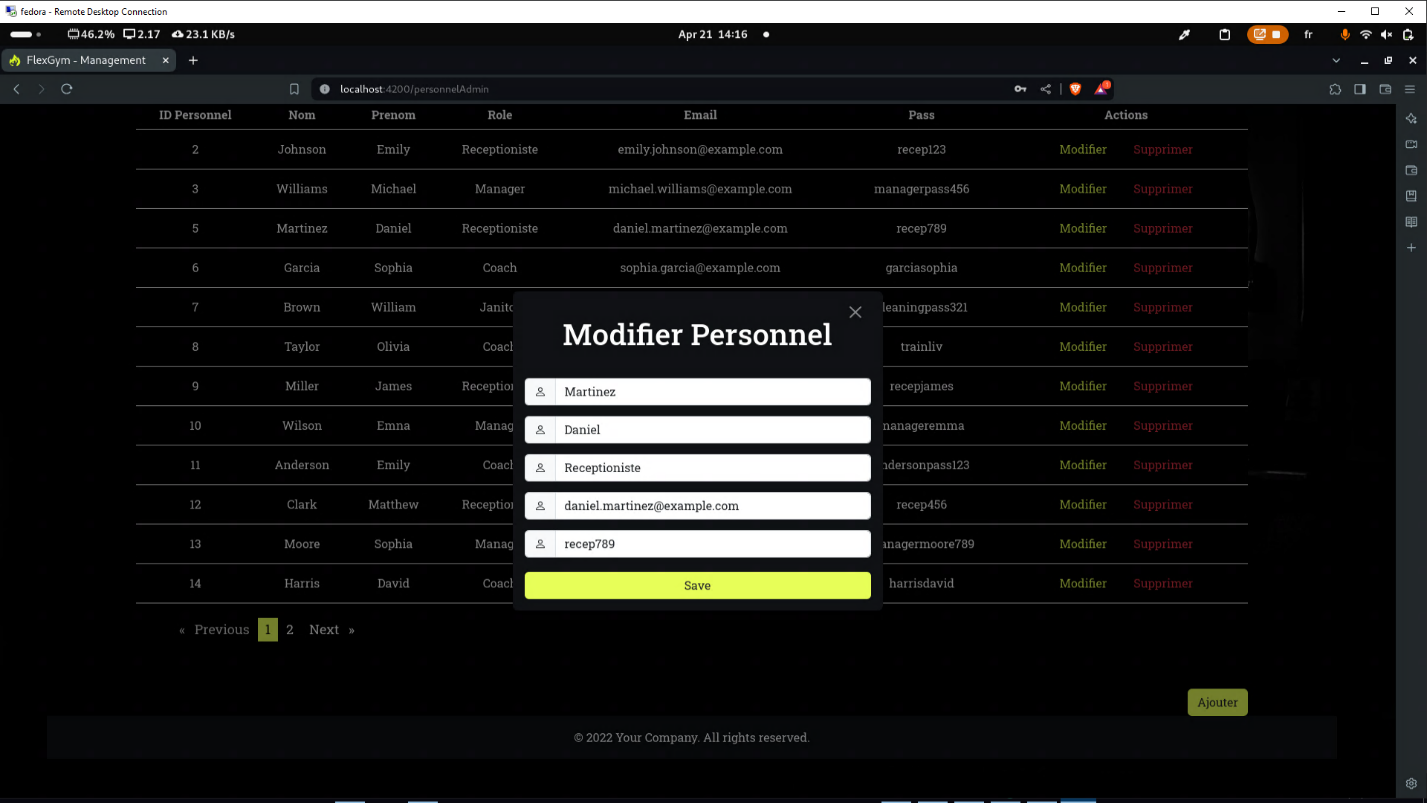
**-Figure 42 : Gérer Personnel**



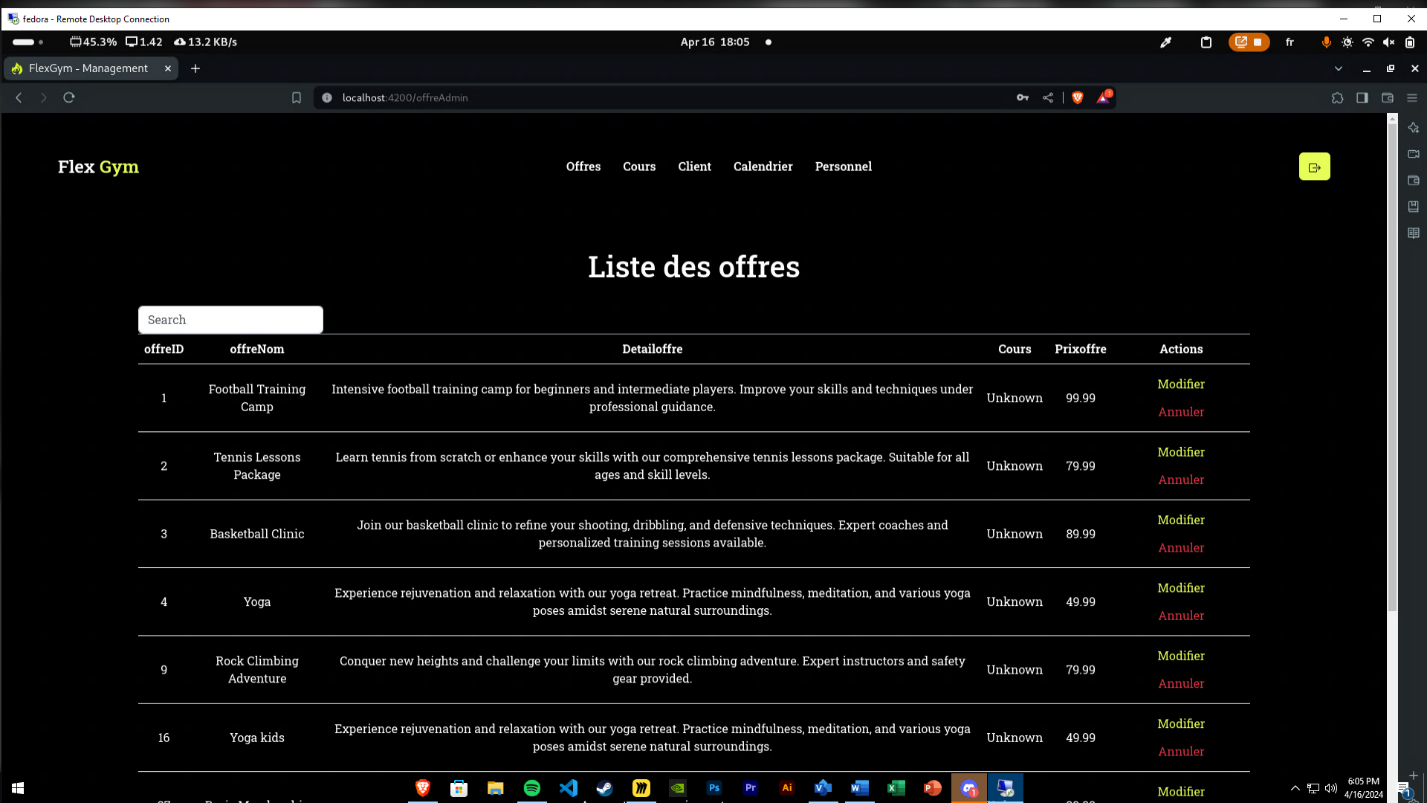
**-Figure 43 : Ajouter Personnel**

****

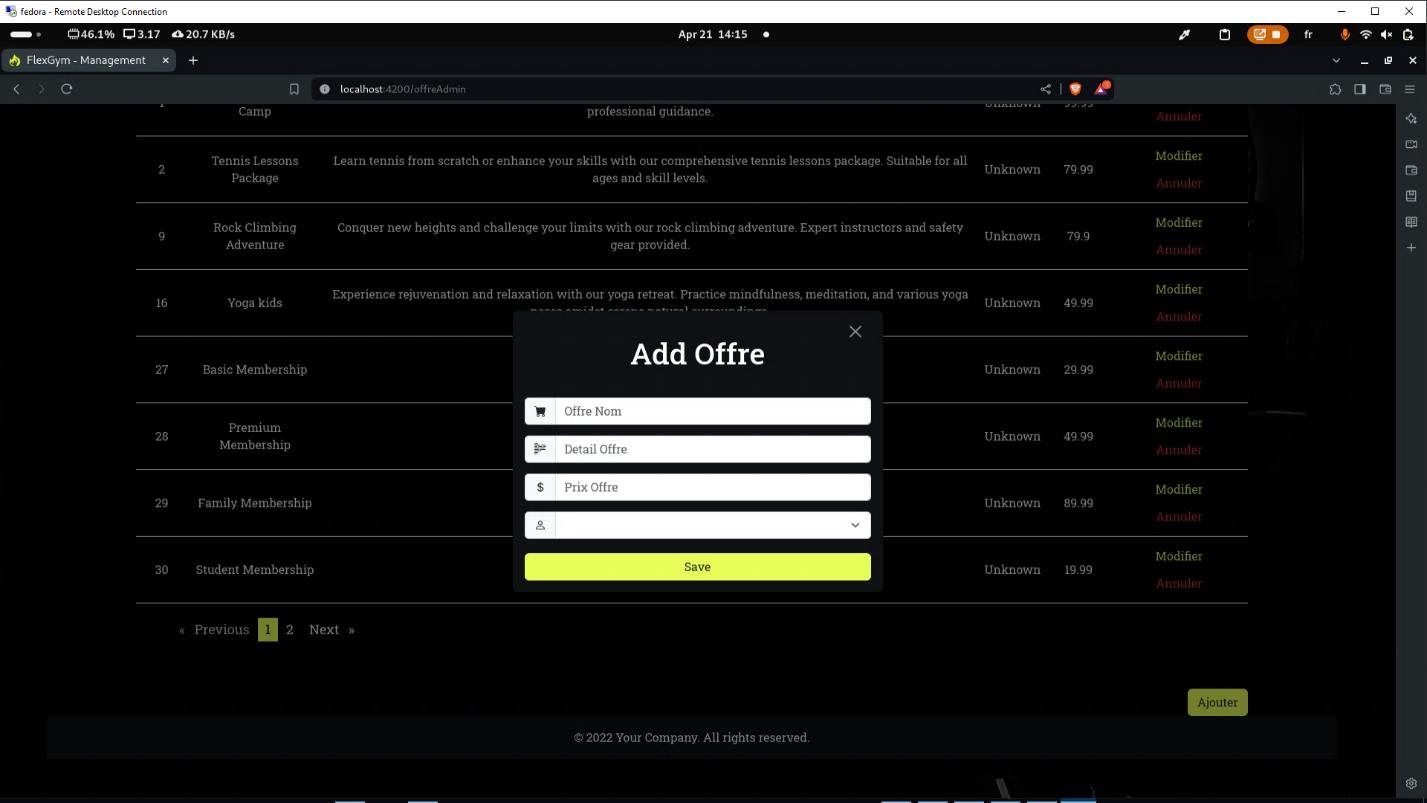
**-Figure 44 : Modifier Personnel**

****

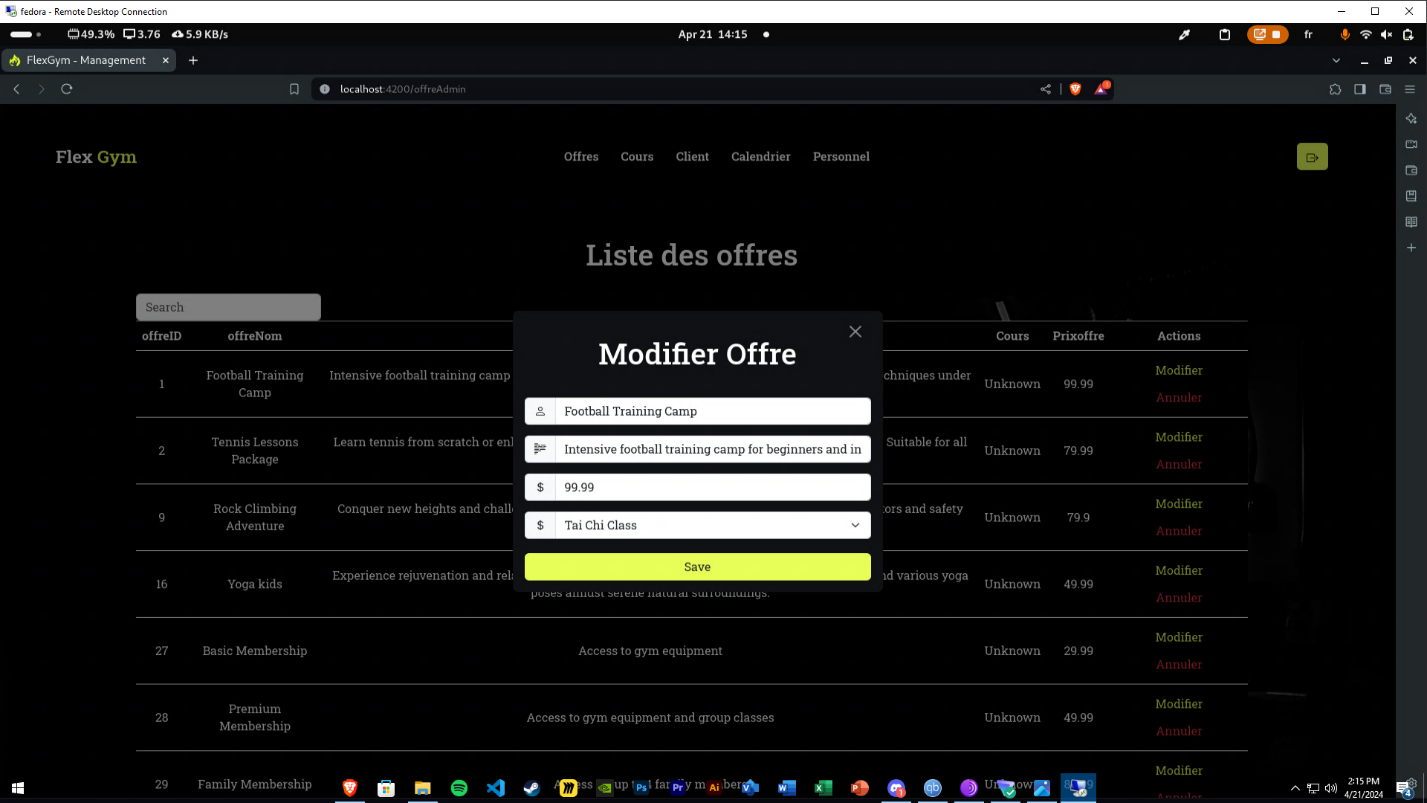
**-Figure 45 : Gérer offres**

****

**-Figure 46 : Ajouter Offre**

****

**-Figure 47 : Modifier Offre**

****

# ****Conclusion****

Dans ce deuxième chapitre, nous avons exploré le Sprint 0 de notre projet de gestion de salle de sport du côté du personnel. Nous avons identifié le backlog de Sprint 0, affiné les éléments, conçu et réalisé les premières étapes de développement. Ce sprint initial a jeté les bases essentielles de notre projet, nous préparant ainsi à aborder la gestion de la salle de sport du côté du client dans le prochain chapitre.

**Chapitre 2 : Sprint 1**

# ****Introduction****

Dans ce deuxième chapitre, nous nous focalisons sur la gestion pratique de la salle de sport du point de vue du personnel. Nous débutons en récapitulant le sprint 0, notre première phase cruciale. Ensuite, nous détaillons l'identification du backlog, le raffinement des éléments du sprint 0, la conception et la réalisation des fonctionnalités essentielles. Enfin, nous évaluons notre progression et planifions les prochaines étapes. Ce chapitre offre une vue claire de la gestion de la salle de sport côté personnel, en mettant en avant les processus et fonctionnalités principaux.

# ****Backlog de sprint 1:****

|  |  |  |
| --- | --- | --- |
| Element de Backlog | Priorité | Estimation |
| En tant que Client, je peux m'identifier. | 2 | Moyene |
| En tant que Client, je peux consulter les offres. | 2 | Moyene |
| En tant que Client, je peux consulter le calendrier. | 2 | Moyene |
| En tant que Client, je peux consulter les cours. | 2 | Moyene |
| En tant que Client, je peux consulter mon abonnement. | 2 | Moyene |

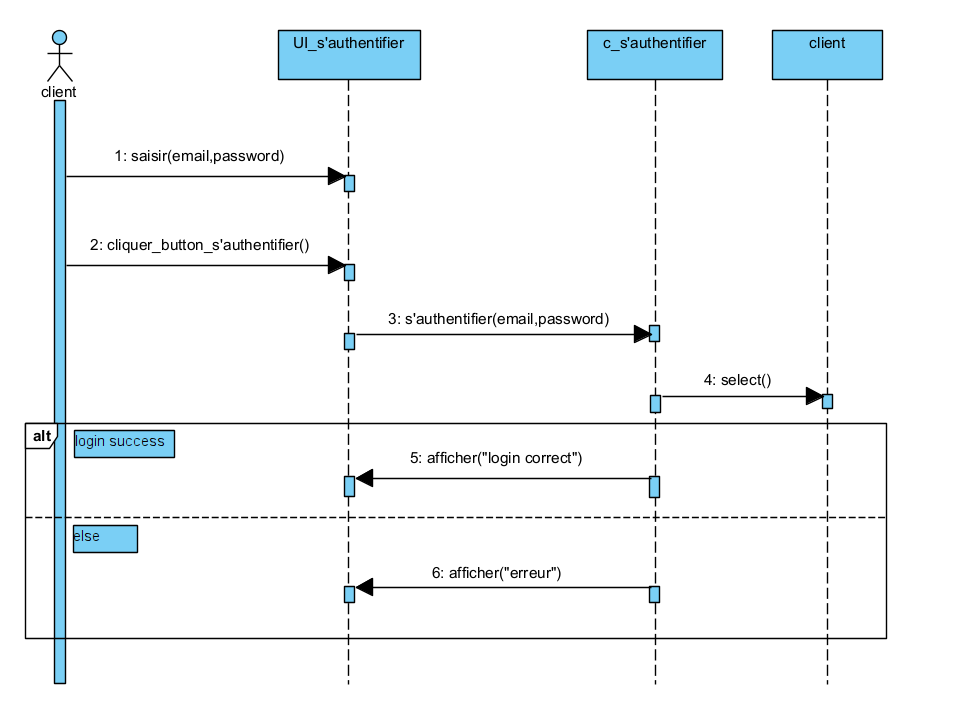
Figure 48 : Backlog du sprint 1

# ****Conception du sprint 1 :****

Dans cette section, nous allons présenter la conception des différents cas d'utilisation, ce qui implique les diagrammes de séquence et de classe relatif associés à chaque cas d'utilisation.

**Conception du cas d’utilisation « s’authentifier »**



Figure 49 : Diagramme de classe relatif au cas d’utilisation « s’authentifier » 

- Figure 50 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « s’authentifier »

**Conception du cas d’utilisation « consulter cours »**

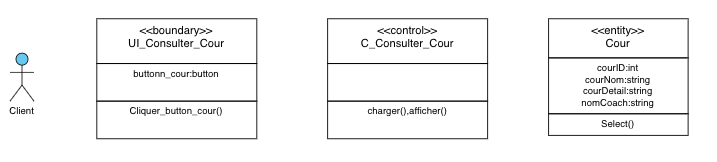
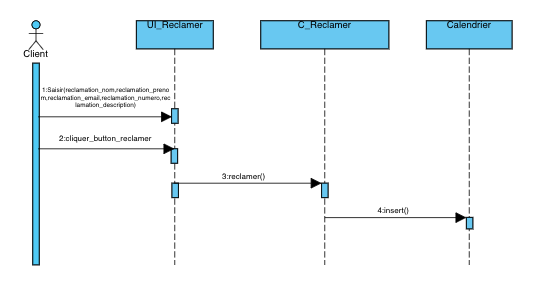


Figure 51 : Diagramme de classe relatif au cas d’utilisation « consulter cours » 

- Figure 52 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « consulter cours »

**Conception du cas d’utilisation « consulter abonnement »**

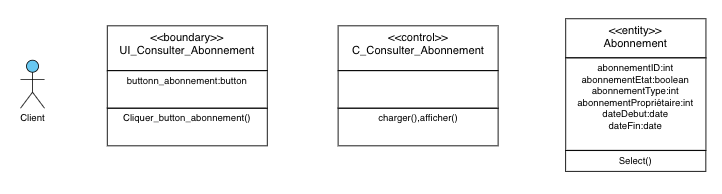
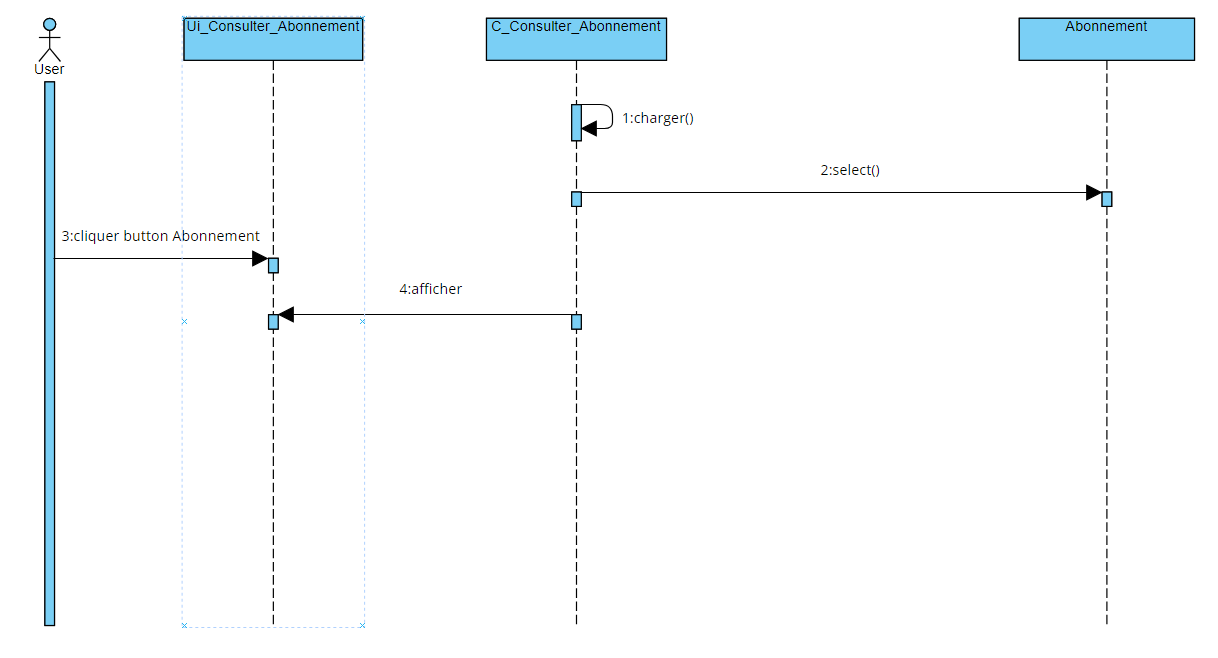


Figure 53 : Diagramme de classe relatif au cas d’utilisation « consulter abonnement » 

- Figure 54 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « consulter abonnement »

**Conception du cas d’utilisation « consulter offres »**

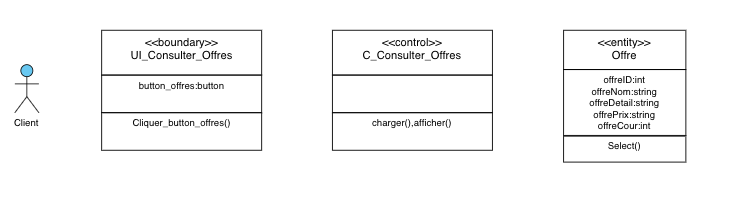
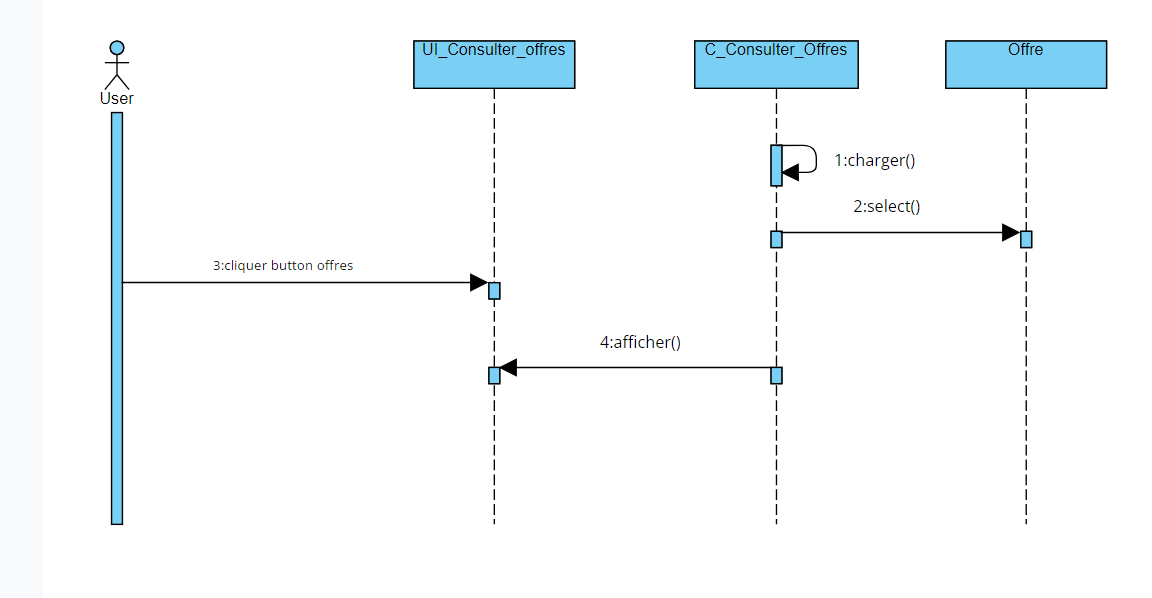


Figure 55 : Diagramme de classe relatif au cas d’utilisation « consulter offres » 

- Figure 56 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « consulter offres »

**Conception du cas d’utilisation « consulter calendrier »**

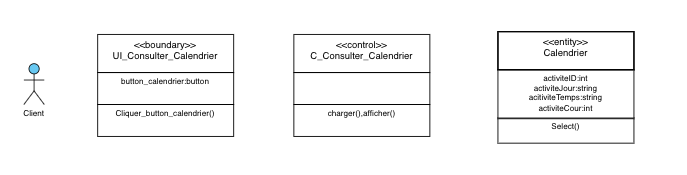
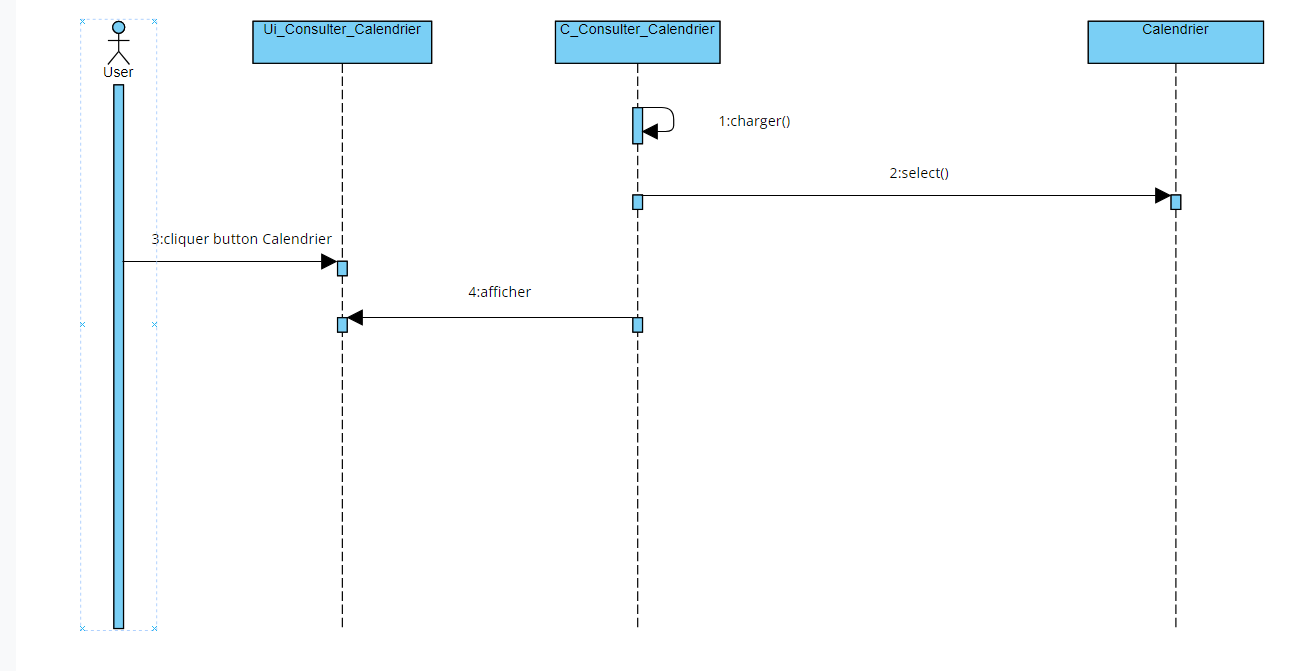


Figure 57 : Diagramme de classe relatif au cas d’utilisation « consulter calendrier » 

- Figure 58 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « consulter calendrier »

**Conception du cas d’utilisation « consulter réclamation »**

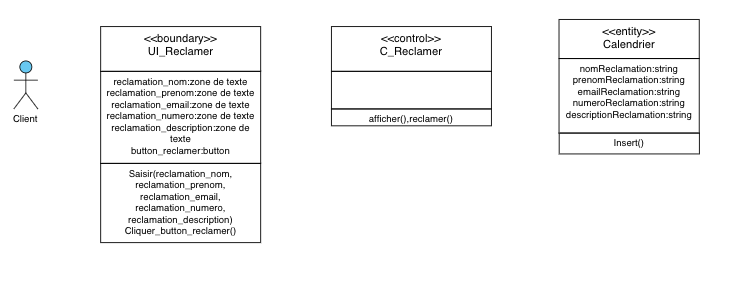
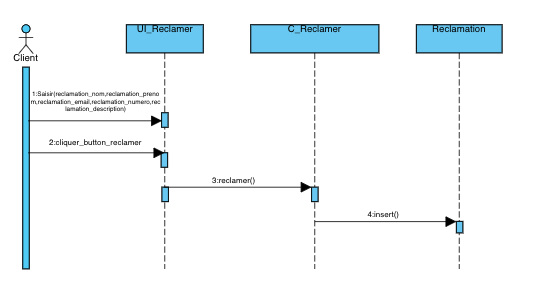


Figure 59 : Diagramme de classe relatif au cas d’utilisation « réclamation » 

- Figure 60 : Diagramme de séquence : cette figure représente le diagramme de séquence relative au cas d’utilisation « réclamation »

# 

# ****Raffinement du sprint 1:****

Dans cette section, nous passons en revue les diverses rafinnement des cas d'utilisation envisagées pour le deuxième sprint (sprint 1).

**rafinnement de cas d’utilisation << authentification>>**

|  |  |
| --- | --- |
| C u | authentification |
| Acteur | Client |
| Precondition | Avoir un compte |
| Postcondition | Connecté, echoué |
| Description de Scénarii P | Le système affiche l'interface correspondant à l’authentification.  L’utilisateur sélectionne l'opération d’authentication.  Le système affiche l'interface correspondant au type du compte authentifié. |
|  |  |

Figure 61: table de rafinnement de cas d’utilisation << authentification >>

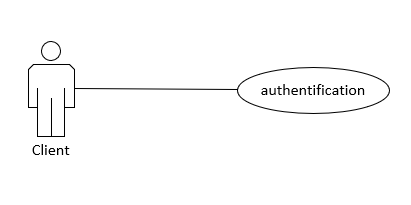


Figure 62: rafinnement de cas d’utilisation << authentification >>

**rafinnement de cas d’utilisation << consulter offres>>**

|  |  |
| --- | --- |
| C u | consulter offres |
| Acteur | Client |
| Precondition | Le client accède au site de salle de sport. |
| Postcondition | Client visualise les offres disponibles |
| Description de Scénarii P | Le client accède à la section "Offres" de la plateforme.  Le système affiche la liste des offres disponibles.  Le client peut parcourir les offres et obtenir plus de détails au besoin. |
|  |  |

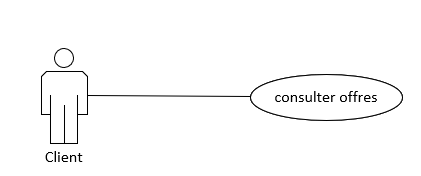
Figure 63: table de rafinnement de cas d’utilisation << consulter offres >>

Figure 64: rafinnement de cas d’utilisation << consulter offres >>

**rafinnement de cas d’utilisation << consulter cour>>**

|  |  |
| --- | --- |
| C u | consulter cour |
| Acteur | Client |
| Precondition | Le client accède au site de salle de sport. |
| Postcondition | Client visualise les offres disponibles |
| Description de Scénarii P | Le client accède à la section "Cours" de la plateforme.  Le système affiche la liste des cours disponibles.  Le client peut parcourir les cours et obtenir plus de détails au besoin. |
|  |  |

Figure 65: table de rafinnement de cas d’utilisation << consulter cour >>

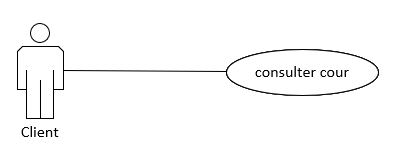


Figure 66: rafinnement de cas d’utilisation << consulter cour >>

**rafinnement de cas d’utilisation << consulter calendrier>>**

|  |  |
| --- | --- |
| C u | consulter calendrier |
| Acteur | Client |
| Precondition | Le client accède au site de salle de sport. |
| Postcondition | Client visualise le calendrier |
| Description de Scénarii P | Le client accède à la section "Calendrier" de la plateforme.  Le système affiche le calendrier.  Le client peut parcourir les activités et obtenir plus de détails au besoin. |
|  |  |

Figure 67: table de rafinnement de cas d’utilisation << consulter calendrier >>

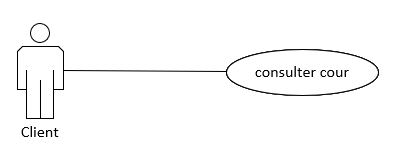


Figure 68: rafinnement de cas d’utilisation << consulter calendrier >>

**rafinnement de cas d’utilisation << consulter abonnement>>**

|  |  |
| --- | --- |
| C u | consulter abonnement |
| Acteur | Client |
| Precondition | Avoir un compte |
| Postcondition | Client visualise les offres disponibles |
| Description de Scénarii P | Le client accède à la section "Profile" de la plateforme.  Le système affiche les détails de l'abonnement du client.  Le client peut vérifier les informations relatives à son abonnement. |
|  |  |

Figure 69: table de rafinnement de cas d’utilisation << consulter abonnement >>

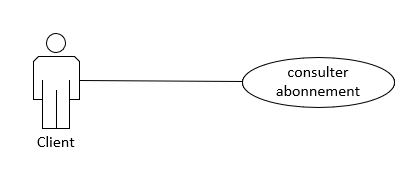


Figure 70: rafinnement de cas d’utilisation << consulter abonnement >>

**rafinnement de cas d’utilisation << reclamation >>**

|  |  |
| --- | --- |
| C u | reclamation |
| Acteur | Client |
| Precondition | Le client accède au site de salle de sport. |
| Postcondition | Client visualise les offres disponibles |
| Description de Scénarii P | Le client accède à la section "Réclamation" de la plateforme.  Le client remplit le formulaire de réclamation en fournissant les détails nécessaires.  Le système enregistre la réclamation et la transmet au personnel de la salle de sport pour traitement ultérieur. |
|  |  |

Figure 71: table de rafinnement de cas d’utilisation << reclamation >>

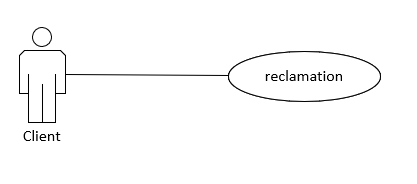
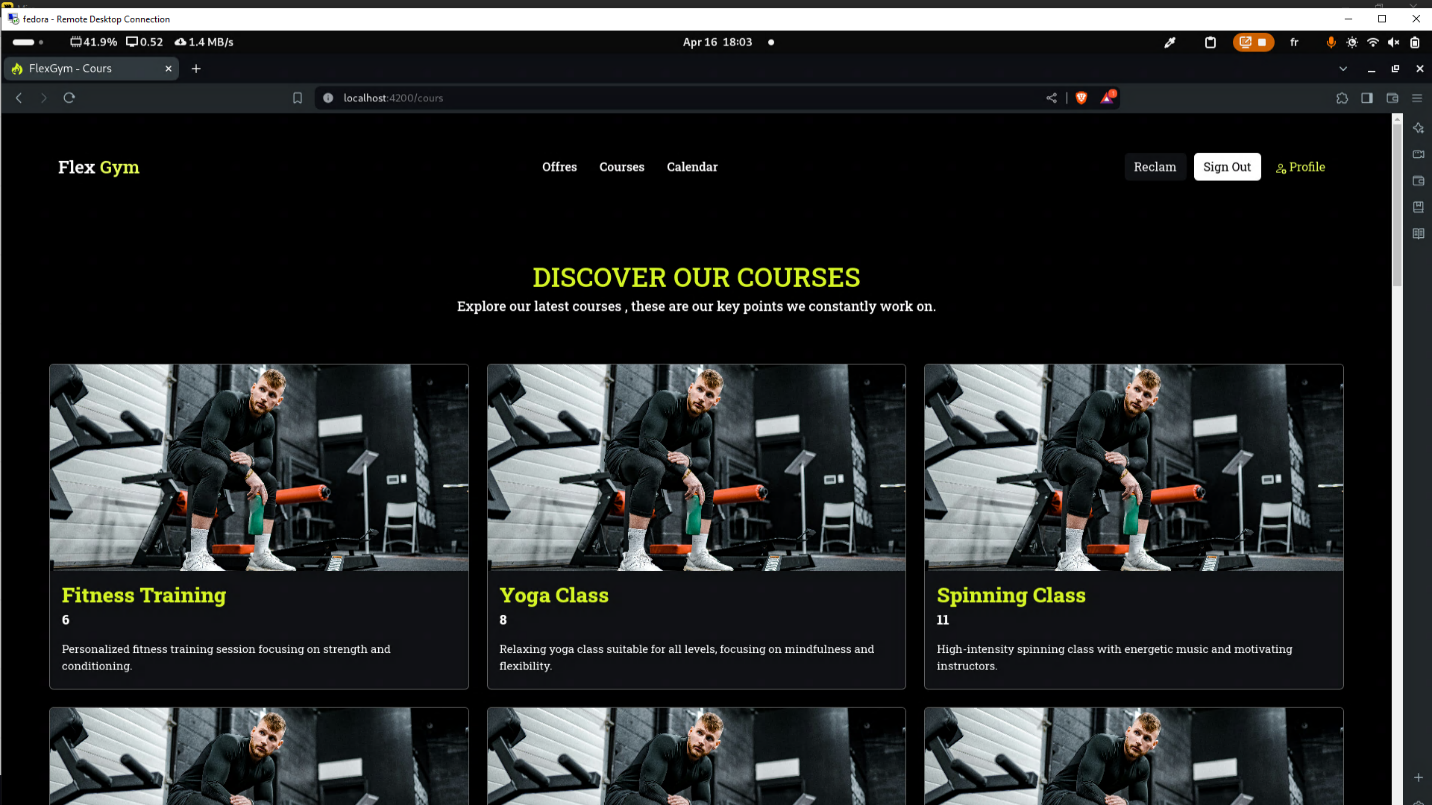
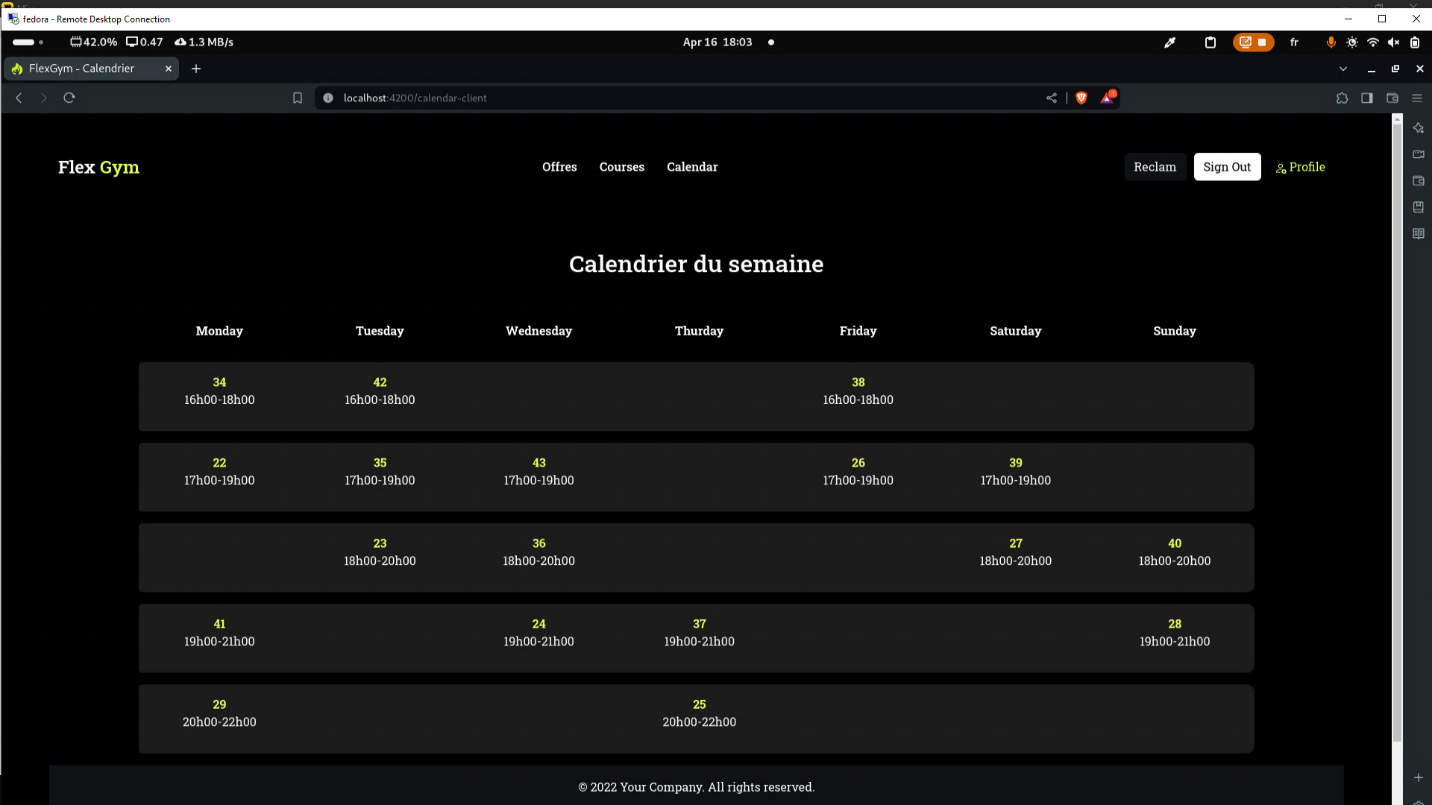


Figure 72: rafinnement de cas d’utilisation << reclamation >>

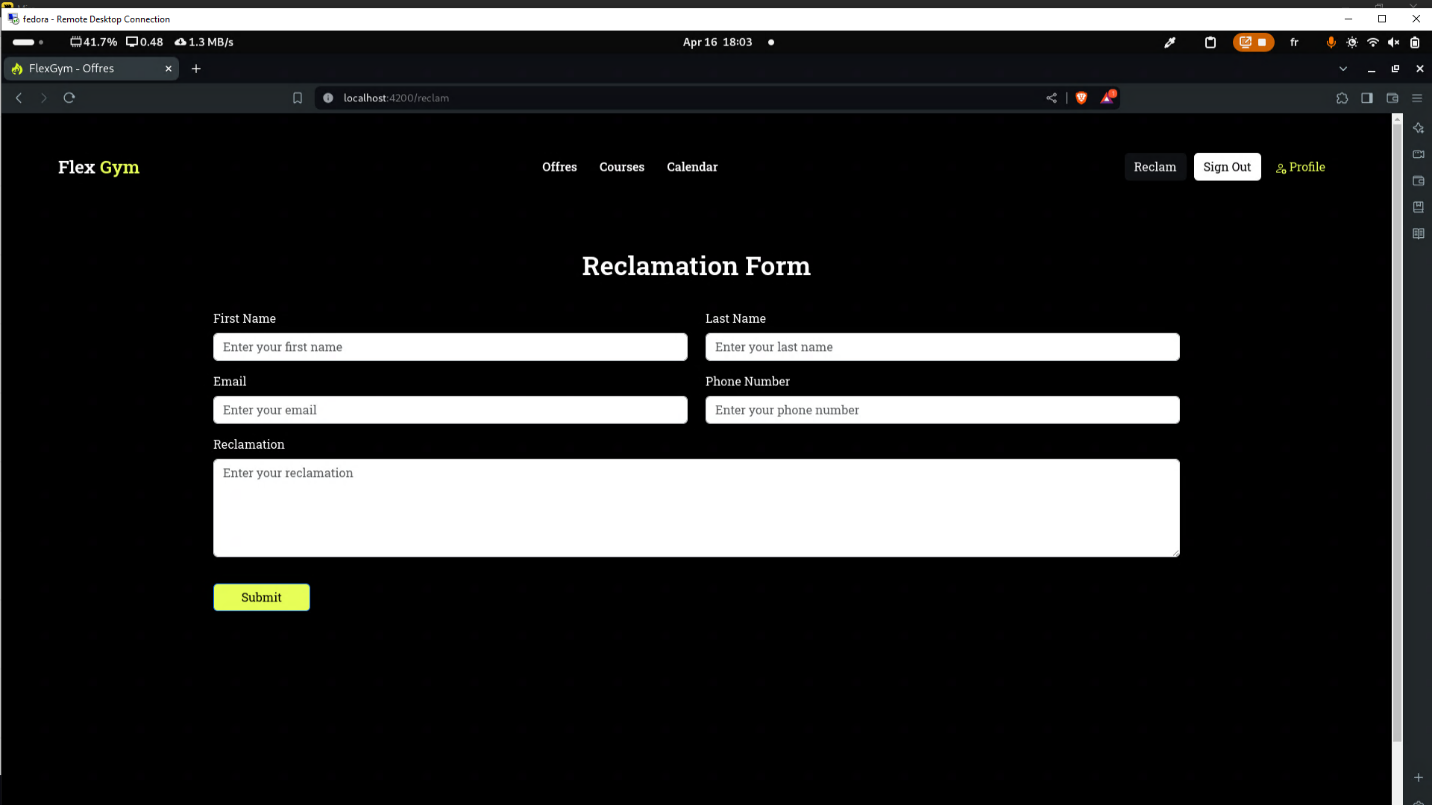
# ****Realisation****

**-Figure 73 : Consulter Cours** 

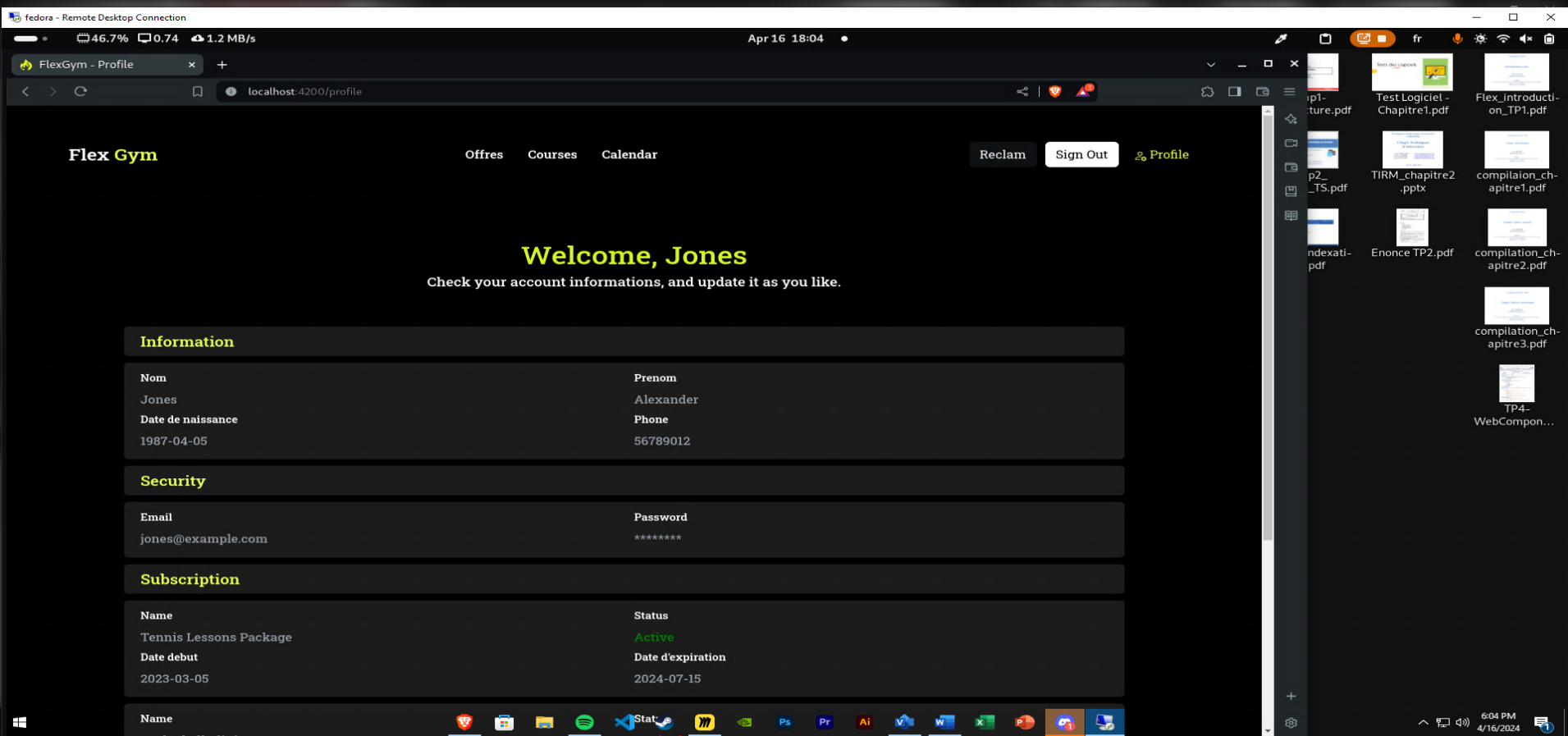
**-Figure 74 : Consulter Calendrier**



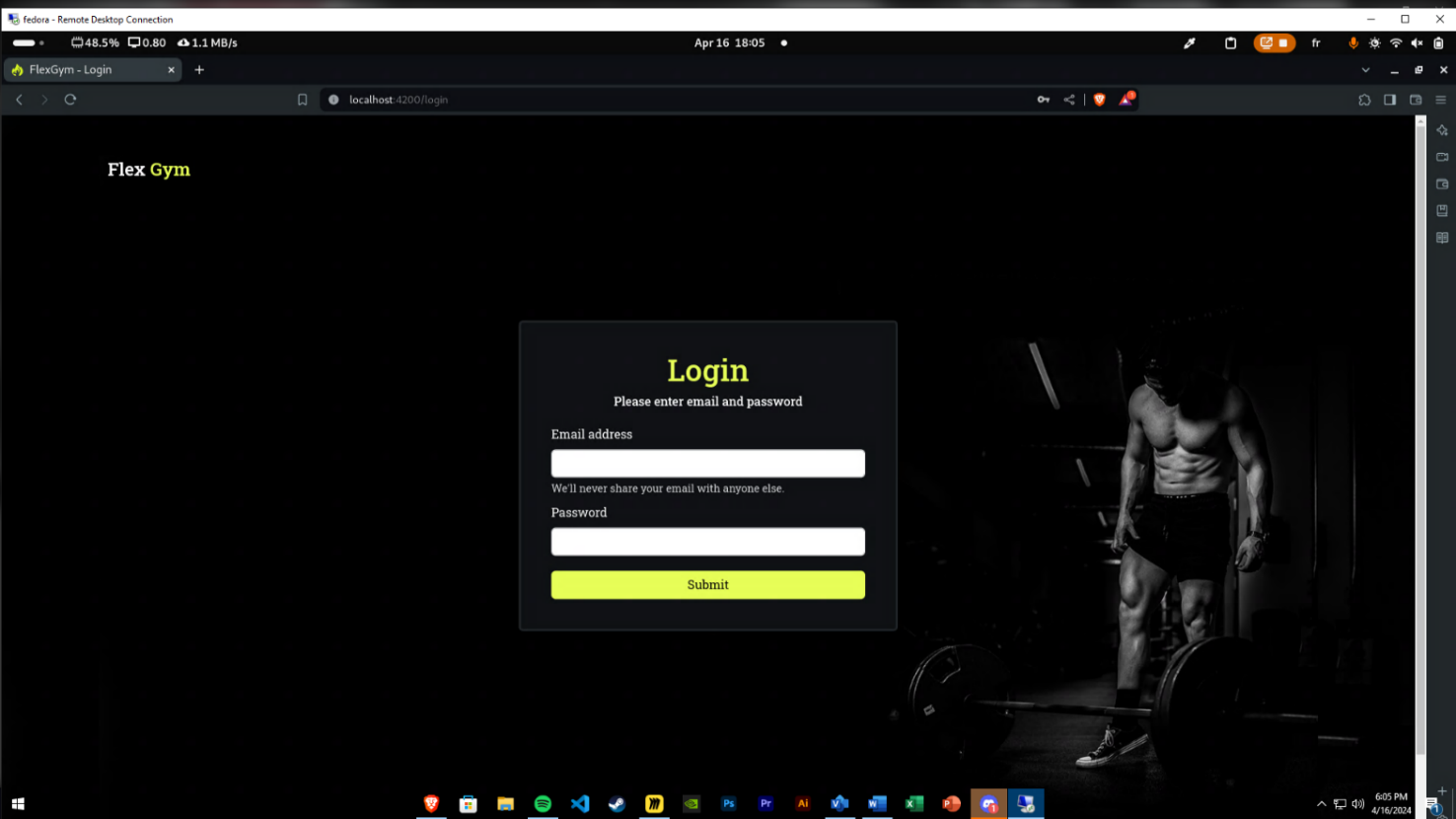
**-Figure 75 : Réclamation**



**-Figure 76 : Consulter abonnement**



**-Figure 77 : S’authentifier**



# ****Conclusion****

Dans ce troisième et dernier chapitre de développement, nous avons plongé dans le Sprint 1 de notre projet de gestion de salle de sport côté client. Après avoir identifié et raffiné le backlog de Sprint 1, nous avons conçu et réalisé les fonctionnalités essentielles. Ce sprint consolide les progrès du Sprint 0, marquant ainsi une avancée significative dans notre projet.

**Conclusion Générale**

La mise en œuvre de notre projet de gestion de salles de sport a été une expérience enrichissante, reflétant l'importance de l'adaptation aux nouvelles technologies pour répondre aux besoins changeants du secteur. À travers notre plateforme web, nous avons visé à rendre la gestion des activités sportives plus accessible et plus efficace pour les clients et le personnel des salles de sport.

En adoptant la méthodologie agile Scrum, nous avons pu développer notre application de manière itérative, en livrant des fonctionnalités de haute qualité tout au long du processus. Le premier chapitre a posé les bases en spécifiant les besoins fonctionnels et non fonctionnels, en identifiant les acteurs clés et en présentant notre méthodologie de conception. Cette approche collaborative nous a permis de rester flexibles et réactifs aux besoins changeants du projet.

Dans le deuxième chapitre, nous avons exploré la gestion de la salle de sport du côté du personnel, en nous concentrant sur le Sprint 0. Ce sprint initial a établi les fondements essentiels de notre projet, en identifiant le backlog, en raffinant les éléments et en réalisant les premières étapes de développement.

Le troisième chapitre a approfondi la gestion du côté du client, en plongeant dans le Sprint 1. Nous avons conçu et réalisé les fonctionnalités essentielles, consolidant ainsi les progrès réalisés dans le Sprint 0.

En conclusion, notre projet a réussi à créer une plateforme web intuitive et sécurisée, offrant une expérience transparente tant pour le personnel que pour les clients des salles de sport. Nous sommes convaincus que notre application contribuera à simplifier la gestion des activités sportives et à améliorer l'expérience des utilisateurs.

**Bibliographie:**

-Le cours dispensé par Dr. Tebourski Wafa porte sur la "**Méthodologie de conception orientée objet**".

-**SCRUM** :https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-French.pdf

-**HTML** :https://developer.mozilla.org/fr/docs/Learn/Getting\_started\_with\_the\_web/HTML\_basics

**CSS** :https://developer.mozilla.org/fr/docs/Learn/CSS/First\_steps

**JavaScript** :https://aws.amazon.com/fr/what-is/javascript/#:~:text=JavaScript%20est%20un%20langage%20de,animations%20et%20de%20cartes%20interactives.

**Angular** :https://www.bocasay.com/fr/developpement-application-angular/#:~:text=C'est%20quoi%20Angular%20%3F,rechargement%20de%20la%20page%20web.

**Figma** :https://www.usabilis.com/figma-outil-dexcellence-vos-design-systems/#:~:text=Figma%20est%20une%20application%20spécialisée,complètement%20dans%20les%20tendances%20actuelles

**XAMPP** :https://fr.wikipedia.org/wiki/XAMPP#:~:text=XAMPP%20est%20un%20ensemble%20de,son%20installation%20simple%20et%20rapide.

**MySql** :https://www.ionos.fr/digitalguide/serveur/know-how/quest-ce-que-mysql/#:~:text=MySQL%20est%20un%20système%20de,données%20de%20différents%20services%20Web.

**VsCode** :https://www.lesnumeriques.com/telecharger/visual-studio-code-vscode-53512#:~:text=Visual%20Studio%20Code%20(ou%20VSCODE,macOS%2C%20et%20sur%20le%20web.

**PHP**:https://fr.wikipedia.org/wiki/PHP#:~:text=PHP%3A%20Hypertext%20Preprocessor%2C%20plus%20connu,langage%20interprété%20de%20façon%20locale.