

Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Constantine 2 – AbdelHamid Mehri



Faculté des Nouvelles Technologies de l'Information et de la Communication  
Département des Technologies des Logiciels et Systèmes d'Information

Master 2 Génie Logiciel

---

## RAPPORT MAG

En utilisant Scrum comme méthode Agile

---

# Système de Gestion d'une salle de sport

---

Dirigé par  
Dr. BOUANAKA

Réalisé par  
BOUKAHIL Kawthar  
SAHEB Abderraouf

# Table de matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Comparaison Méthodes Agile vs. Cycle en V</b>	<b>4</b>
<b>3</b>	<b>Scrum : Une Méthode Agile</b>	<b>5</b>
3.1	Principes Fondamentaux de Scrum . . . . .	5
<b>4</b>	<b>Introduction Générale Du Projet choisi</b>	<b>6</b>
<b>5</b>	<b>Méthode de Gestion de Projet</b>	<b>6</b>
5.1	TEAM . . . . .	6
5.2	L'outil choisi . . . . .	6
<b>6</b>	<b>Demarche de travail</b>	<b>7</b>
6.1	Backlog meeting . . . . .	7
6.2	Sprint planing meeting . . . . .	7
6.3	Daily meeting . . . . .	7
6.4	Sprint review meeting . . . . .	7
6.5	Sprint retrospective meeting . . . . .	7
6.6	Update product backlog . . . . .	7
<b>7</b>	<b>Backlog first meeting</b>	<b>8</b>
<b>8</b>	<b>Backlog Sprint1</b>	<b>9</b>
8.1	Analyse des besoins . . . . .	9
8.1.1	Identification des acteurs . . . . .	9
8.1.2	Les besoins fonctionnels . . . . .	9
8.1.3	Les besoins non-fonctionnels . . . . .	11
8.1.4	Diagramme de contexte statique . . . . .	12
8.1.5	Diagramme de cas d'utilisation . . . . .	12
8.2	Sprint review . . . . .	13
<b>9</b>	<b>Backlog Second meeting</b>	<b>14</b>
<b>10</b>	<b>Backlog Sprint2</b>	<b>14</b>
10.1	Analyse des besoins . . . . .	14
10.1.1	Identification des acteurs . . . . .	14
10.1.2	Les besoins fonctionnels . . . . .	14
10.1.3	Les besoins non-fonctionnels . . . . .	15
10.1.4	Diagramme de contexte statique . . . . .	16
10.1.5	Diagramme de cas d'utilisation . . . . .	17
10.2	Sprint review . . . . .	18
<b>11</b>	<b>Backlog Final Meeting</b>	<b>19</b>
<b>12</b>	<b>Backlog Sprint3</b>	<b>19</b>
12.1	Analyse des besoins . . . . .	19
12.1.1	Identification des acteurs . . . . .	19
12.1.2	Les besoins fonctionnels . . . . .	19
12.1.3	Les besoins non-fonctionnels . . . . .	21
12.1.4	Diagramme de contexte statique . . . . .	21
12.1.5	Diagramme de cas d'utilisation . . . . .	22
12.2	Sprint review . . . . .	23

<b>13 Resultat de travail (livraison)</b>	<b>24</b>
<b>14 L'application "Gym Management System"</b>	<b>25</b>
<b>15 Définition des grands choix techniques</b>	<b>32</b>
15.1 Langages Utilisés : . . . . .	32
15.2 Base de Données : . . . . .	32
15.3 Serveur de Développement : . . . . .	32
15.4 Outils de Développement : . . . . .	32
15.5 Outils de Collaboration : . . . . .	32
15.6 Outils de documentation : . . . . .	32
<b>16 Conclusion</b>	<b>32</b>

## Table de figures

1	Comparaison Méthodes Agile vs. Cycle en V . . . . .	4
2	Logo du Trello . . . . .	6
3	Backlog Product . . . . .	8
4	Les besoins fonctionnels d'acteur user . . . . .	10
5	Comment realiser une tache avec details . . . . .	10
6	Les priorites de chaque tache "red=high","blue=normal" . . . . .	11
7	Digramme de contexte statique sprint1 . . . . .	12
8	Digramme de cas d'utilisation sprint1 . . . . .	12
9	Sprint review . . . . .	13
10	Les besoins fonctionnels d'acteur admin . . . . .	15
11	Diagramme de contexte statique sprint 2 . . . . .	16
12	Diagramme de cas d'utilisation sprint 2 . . . . .	17
13	Sprint review . . . . .	18
14	ajouter le dernier besoin fonctionnel . . . . .	20
15	Diagramme de contexte statique final . . . . .	21
16	Diagramme de cas d'utilisation final . . . . .	22
17	Sprint review . . . . .	23
18	terminer les sprints . . . . .	24
19	La page d'accueil du systeme . . . . .	25
20	La page du login d'utilisateur . . . . .	25
21	La page d'inscription d'utilisateur . . . . .	26
22	La page d'acceuil d'utilisateur . . . . .	26
23	Mettre a jour le profil d'utilisateur . . . . .	27
24	Changer le mot de passe d'utilisateur . . . . .	27
25	Le login de l'admin . . . . .	28
26	La page d'acceuil d'admin . . . . .	28
27	Mettre a jour le profile de l'admin . . . . .	29
28	Changer le mot de passe de l'admin . . . . .	29
29	Ajouter une categorie . . . . .	30
30	Ajouter un package . . . . .	30
31	Gerer les packages . . . . .	31
32	gerer les reservations . . . . .	31

## 1 Introduction

Contrairement aux méthodes traditionnelles telles que le modèle en cascade (Waterfall), l'approche Agile se distingue par sa grande flexibilité et son aptitude à offrir une visibilité accrue dans la gestion de projets. À une époque où la personnalisation est devenue cruciale, cette méthodologie gagne en popularité. L'Agilité s'adapte particulièrement bien aux projets exigeant une rapidité d'exécution ou évoluant constamment dans des domaines en perpétuel changement. Elle constitue la structure idéale pour les projets susceptibles de subir des ajustements et des optimisations en cours de route, tout en ne compromettant pas la fluidité de l'organisation des équipes. Cette approche souple favorise l'agilité et l'adaptabilité, ce qui la rend de plus en plus attrayante pour les gestionnaires de projets modernes.

## 2 Comparaison Méthodes Agile vs. Cycle en V

La gestion de projet présente diverses approches, parmi lesquelles la méthode Agile et le cycle en V se distinguent par leurs philosophies et leurs méthodes d'exécution. L'Agilité privilégie la flexibilité, l'adaptabilité et la collaboration continue avec les parties prenantes, ce qui permet de répondre aux besoins changeants des projets de manière proactive. En revanche, le cycle en V repose sur une planification minutieuse en amont, suivie de phases séquentielles de développement et de validation. Bien que cette approche puisse convenir à des projets avec des exigences stables, elle peut entraîner des découvertes tardives d'erreurs et est moins adaptée aux projets dynamiques. Le choix entre Agile et le cycle en V dépend des spécificités du projet et de la nécessité de flexibilité ou de conformité rigide.

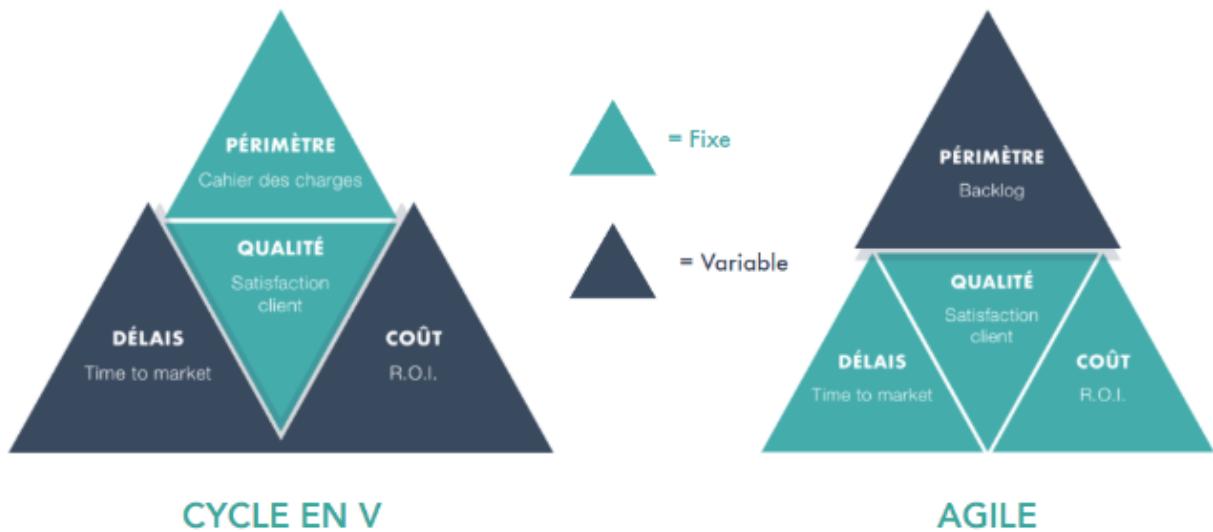


Figure 1: Comparaison Méthodes Agile vs. Cycle en V

### 3 Scrum : Une Méthode Agile

Scrum est largement reconnu comme la méthode agile la plus utilisée. Comme d'autres méthodes agiles, Scrum repose sur une approche de gestion de projet qui place le client ou l'utilisateur au centre de l'équipe chargée du développement. À l'origine, Scrum était principalement utilisé dans le domaine de l'informatique, en particulier pour le développement d'applications. Cependant, au fil du temps, Scrum a trouvé sa place dans d'autres secteurs de l'ingénierie produit.

#### 3.1 Principes Fondamentaux de Scrum

Scrum définit trois rôles clés au sein de l'organisation d'un projet agile :

1. **Product Owner** : Le Product Owner est la personne qui incarne la vision du produit à créer. Il s'agit généralement d'un expert dans le domaine métier. Le Product Owner travaille en étroite collaboration avec l'équipe de développement, ayant la responsabilité de gérer le backlog produit et de déterminer les priorités des "user stories" à réaliser. Bien que le Product Owner puisse être soit interne, soit externe à l'organisation, il est généralement le client.
2. **Scrum Master** : Le Scrum Master est un membre actif de l'équipe projet. Sa maîtrise des principes de Scrum lui permet de s'assurer que la méthodologie est correctement mise en œuvre. Il ne doit pas être confondu avec un chef de projet traditionnel, car son rôle consiste à faciliter la communication et le travail entre les différentes parties prenantes, garantissant ainsi la pleine productivité de l'équipe. Il agit en tant que coach, tant vis-à-vis du Product Owner que de l'équipe de développement. Il doit être un excellent communicateur et démontrer des compétences pédagogiques pour résoudre d'éventuels conflits qui pourraient surgir au cours du projet. Le Scrum Master anime généralement diverses réunions, telles que la réunion quotidienne Scrum, la revue de sprint et la rétrospective. Il est courant que le rôle de Scrum Master soit tournant au sein de l'équipe projet.
3. **L'équipe de Développement** : Généralement composée de 4 à 6 membres, l'équipe de développement est chargée de concrétiser les besoins exprimés par le Product Owner sous forme de "user stories" en fonctionnalités réelles, opérationnelles et utilisables. L'équipe est souvent constituée de divers profils, allant au-delà des développeurs, et peut inclure des architectes, des DBA (Administrateurs de Bases de Données), des graphistes, des ergonomes, des ingénieurs système ou réseau, en fonction des besoins spécifiques du projet.

Scrum repose sur ces concepts fondamentaux pour favoriser la collaboration, la communication, et la réalisation de produits de haute qualité de manière itérative et incrémentielle.

## 4 Introduction Générale Du Projet choisi

Notre projet porte sur la conception d'un système de gestion complet dédié aux installations de gymnas-tique. Ce système est composé de deux modules interconnectés, visant à simplifier et à améliorer la gestion globale de ces établissements.

L'objectif de ce projet est de fournir des outils efficaces, à la fois pour les utilisateurs et l'admin, afin de faciliter la gestion quotidienne des activités d'une salle de gym. Ce système vise à optimiser l'expérience des utilisateurs, qu'il s'agisse de clients en quête de réservations de forfaits ou d'administrateur supervisant l'ensemble des opérations.

Dans les sections suivantes, nous explorerons en détail les différents modules et fonctionnalités de ce système de gestion de gym, en mettant en avant leur importance et leur contribution à l'amélioration de l'efficacité opérationnelle.

## 5 Méthode de Gestion de Projet

Pour la gestion de notre projet, nous avons opté pour la méthodologie Scrum. Après avoir établi les concepts fondamentaux de Scrum, notre équipe Scrum pour ce projet est composée de :

### 5.1 TEAM

- **Product Owner** : Directeur du gym
- **Scrum Master** : SAHEB Abderraouf
- **Scrum Team** : BOUKAHIL Kawthar , SAHEB Abderraouf

### 5.2 L'outil choisi

Nous avoins choisi **Trello** comme un outil.

En bref, Trello est un outil de gestion de projet en ligne basé sur des tableaux, des listes et des cartes, qui facilite la visualisation, le suivi et la collaboration sur les tâches. Il peut être utilisé en combinaison avec la méthodologie Scrum pour organiser et gérer efficacement les projets en équipe, en particulier en utilisant des tableaux Kanban pour suivre les étapes du processus de développement.



Figure 2: Logo du Trello

## **6 Demarche de travail**

La méthode Scrum constitue un processus récurrent et itératif centré sur un sprint. Dans les prochains paragraphes, nous détaillerons la procédure et les étapes à suivre au cours d'un sprint. Ce cycle se réitère jusqu'à la finalisation ou l'interruption du projet.

### **6.1 Backlog meeting**

Au cours de cette réunion, le Product Owner assume la direction en réunissant toute l'équipe Scrum. L'objectif est de formuler les user stories, permettant ainsi l'introduction exhaustive des besoins fonctionnels du projet. Cette phase est communément désignée comme le "backlog project" dans le cadre de la méthodologie Scrum.

### **6.2 Sprint planning meeting**

La "Sprint Planning Meeting" (Réunion de planification de sprint) est un événement clé dans la méthodologie Scrum. Elle se déroule au début de chaque sprint et réunit l'équipe Scrum, le Scrum Master et, généralement, le Product Owner. L'objectif principal de cette réunion est de planifier le travail qui sera réalisé pendant le sprint.

### **6.3 Daily meeting**

Au cours de cette réunion, durant une plage horaire de 10 à 15 minutes, nous discutons des obstacles que nous avons rencontrés et partageons nos expériences tout en explorant des solutions possibles. Cette pratique, bien que spécifique à notre équipe et distincte de la méthode Scrum, nous offre l'opportunité de planifier les tâches à accomplir pour la journée, structurées sous forme de liste de tâches (todo list). Cette approche permet une quantification du travail effectué, prévenant ainsi tout égarement dans des fonctionnalités non essentielles qui pourraient potentiellement retarder l'avancement global du projet.

### **6.4 Sprint review meeting**

Au cours de cette réunion, nous présentons ce que nous avons accompli durant le sprint, mettant en avant les nouvelles fonctionnalités développées.

### **6.5 Sprint retrospective meeting**

Cette réunion est orchestrée par le Scrum Master, se tient à la clôture de chaque sprint. Elle offre à nous l'occasion de discuter des événements survenus pendant le sprint écoulé et d'identifier des ajustements potentiels visant à accroître la productivité du prochain sprint. Cette pratique, cruciale pour l'évolution constante tout au long du projet, encourage l'expression ouverte, honnête et constructive d'opinions par tous les acteurs, y compris l'équipe, le propriétaire du produit et le Scrum Master. La rétrospective aide également la direction à recueillir des commentaires sur le travail et les progrès du projet.

### **6.6 Update product backlog**

Ici, nous réintégrons dans le backlog les tâches qui n'ont pas été réalisées ou qui ont généré des problèmes, et nous sélectionnons de nouvelles tâches dans le backlog projet. Cela vise à établir la planification du prochain sprint. Dans cette démarche, le processus reprend à partir de la Sprint Planning Meeting (étape 2) de manière itérative jusqu'à l'achèvement ou l'arrêt du projet.

## 7 Backlog first meeting

Après la réunion avec le Product Owner, le focus initial a été défini sur les besoins de l'acteur utilisateur/client. La priorité a été accordée aux fonctionnalités essentielles, notamment l'inscription, la connexion au compte, le changement de mot de passe, la mise à jour du profil, et l'accès à l'historique des réservations de forfaits de gym. Ces fonctionnalités ont été identifiées comme des éléments clés pour offrir une expérience utilisateur initiale et complète.

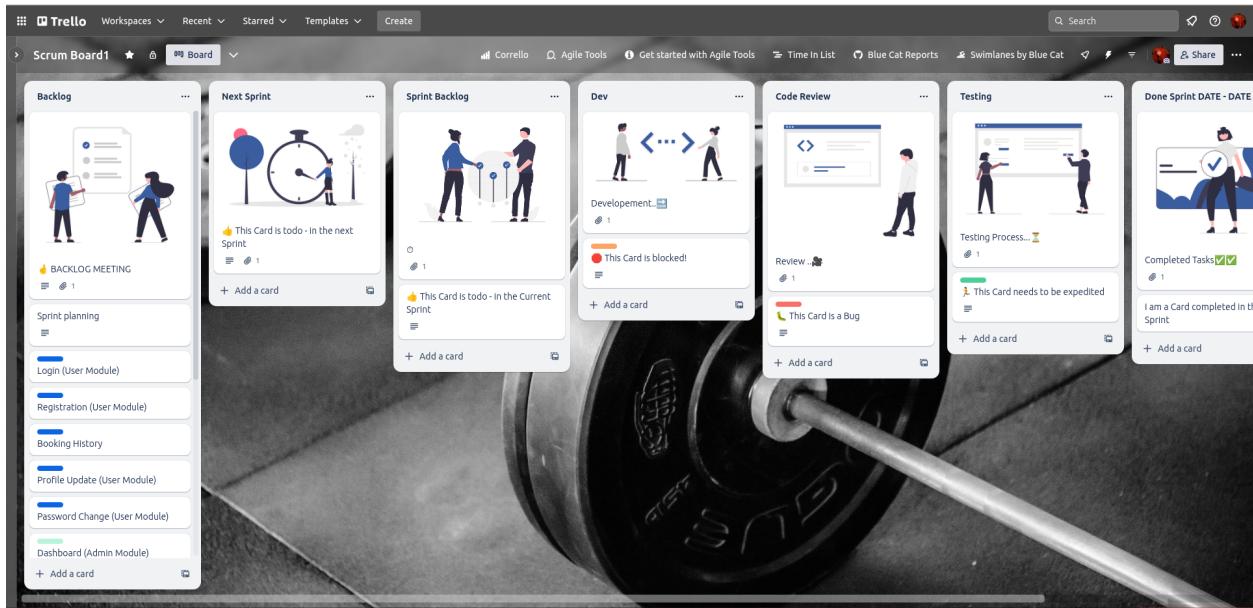


Figure 3: Backlog Product

## 8 Backlog Sprint1

Suite à la décision conjointe avec le Product Owner, il a été convenu que ce sprint serait étendu sur une période de 4 semaines afin de développer les fonctionnalités requises, tant du côté backend que frontend. Afin d'assurer une compréhension claire et détaillée du travail à réaliser, nous allons élaborer un cahier des charges exhaustif.

### 8.1 Analyse des besoins

#### 8.1.1 Identification des acteurs

**Clients/Utilisateurs :** Ce sont les membres de la salle de sport qui utilisent le système pour réserver des forfaits, consulter leur historique de réservation et mettre à jour leur profil.

#### 8.1.2 Les besoins fonctionnels

- **Inscription :**

- Les utilisateurs peuvent s'inscrire sur le site web.
- Les utilisateurs fournissent des informations telles que leur nom, leur adresse e-mail, etc.
- Les informations d'inscription sont stockées dans la base de données.

- **Connexion :**

- Les utilisateurs inscrits peuvent se connecter à leur compte.

- **Historique des Réservations :**

- Les utilisateurs ont accès à un historique de leurs réservations de forfaits de gym.
- L'historique inclut des détails sur les forfaits réservés, les dates de réservation et les paiements effectués.

- **Mise à Jour du Profil :**

- Les utilisateurs peuvent mettre à jour leur profil, y compris des informations telles que l'adresse, le numéro de téléphone, etc.
- Les modifications sont enregistrées dans la base de données.

- **Changement de Mot de Passe :**

- Les utilisateurs peuvent changer leur propre mot de passe.
- Le processus est sécurisé et nécessite une authentification.

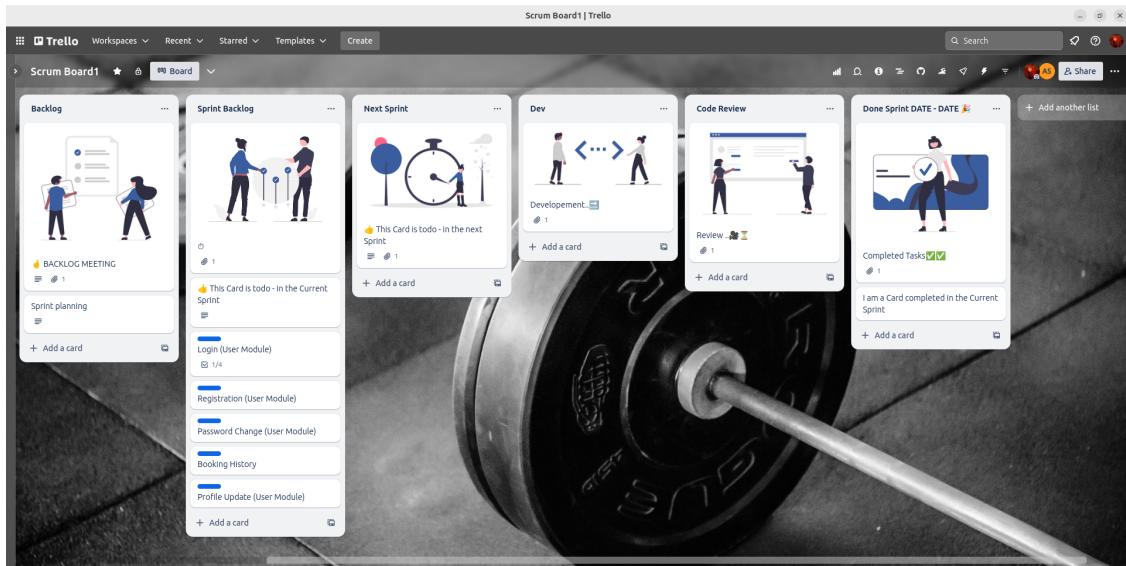


Figure 4: Les besoins fonctionnels d'acteur user

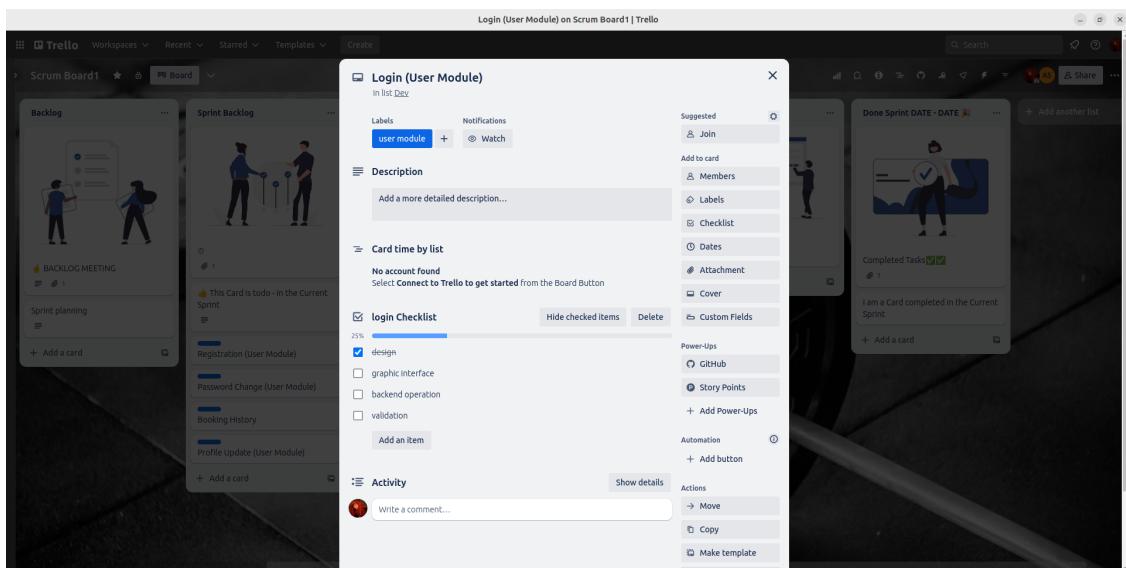


Figure 5: Comment realiser une tache avec details

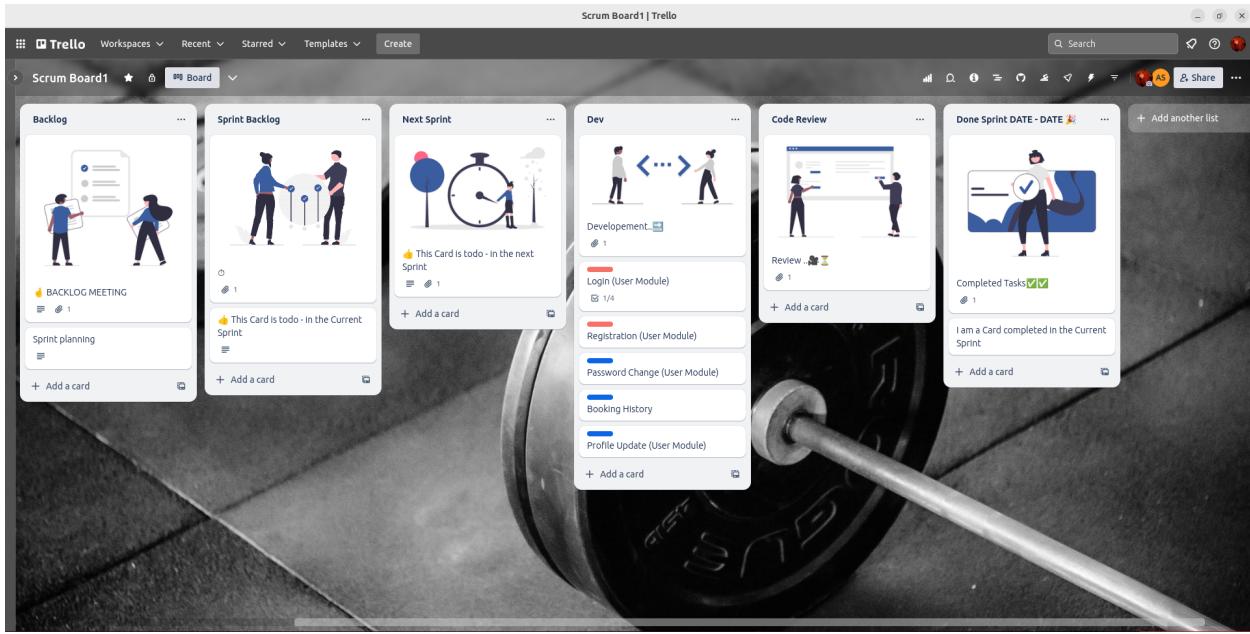


Figure 6: Les priorités de chaque tache "red=high", "blue=normal"

### 8.1.3 Les besoins non-fonctionnels

- **Performance** : Assurer une réponse rapide aux utilisateurs, même en cas d'utilisation intensive.
- **Sécurité** : Protéger les données des utilisateurs contre les accès non autorisés.
- **Convivialité** : Offrir une interface utilisateur facile à utiliser et compréhensible.
- **Disponibilité** : Garantir un accès 24/7 au système.
- **Extensibilité** : Permettre des modifications et des extensions sans perturber le fonctionnement.
- **Conformité aux normes** : Respecter les réglementations et normes applicables.
- **Sauvegarde et récupération des données** : Protéger les données et permettre leur récupération en cas de problème.

#### 8.1.4 Diagramme de contexte statique

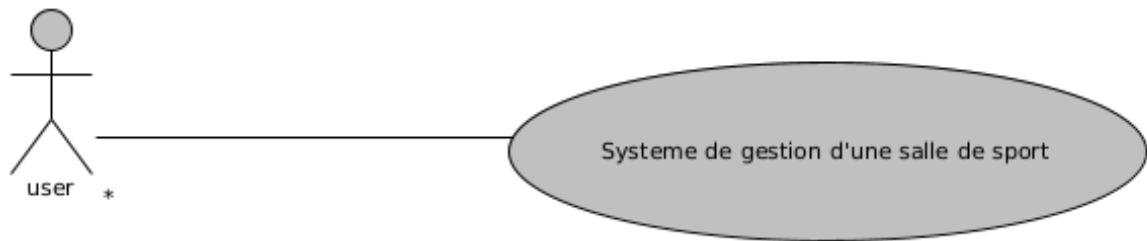


Figure 7: Diagramme de contexte statique sprint1

#### 8.1.5 Diagramme de cas d'utilisation

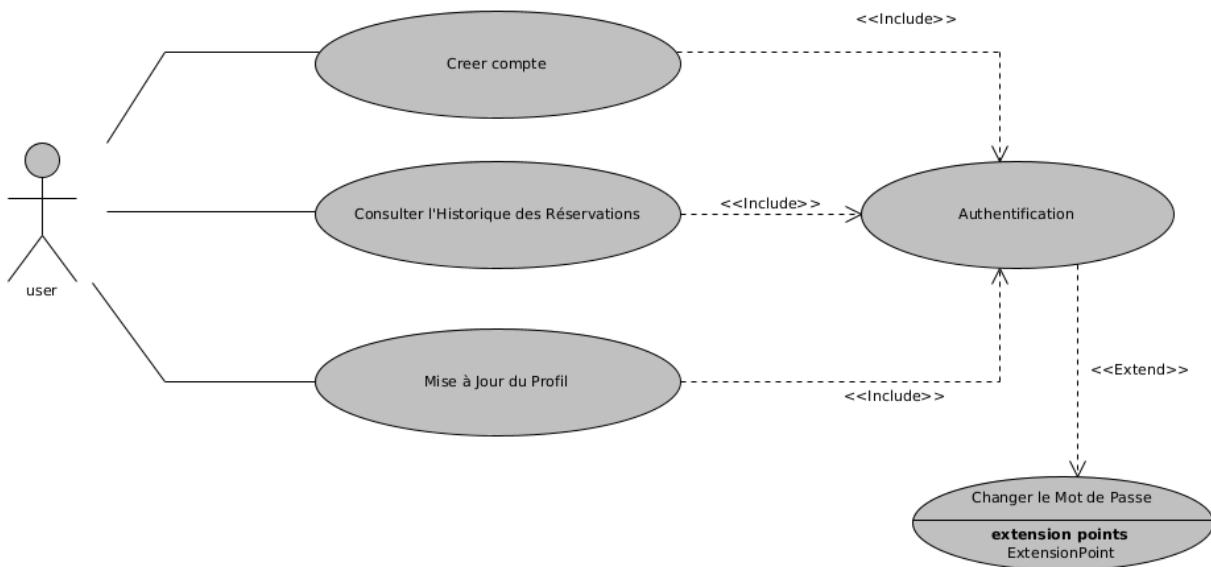


Figure 8: Diagramme de cas d'utilisation sprint1

## 8.2 Sprint review

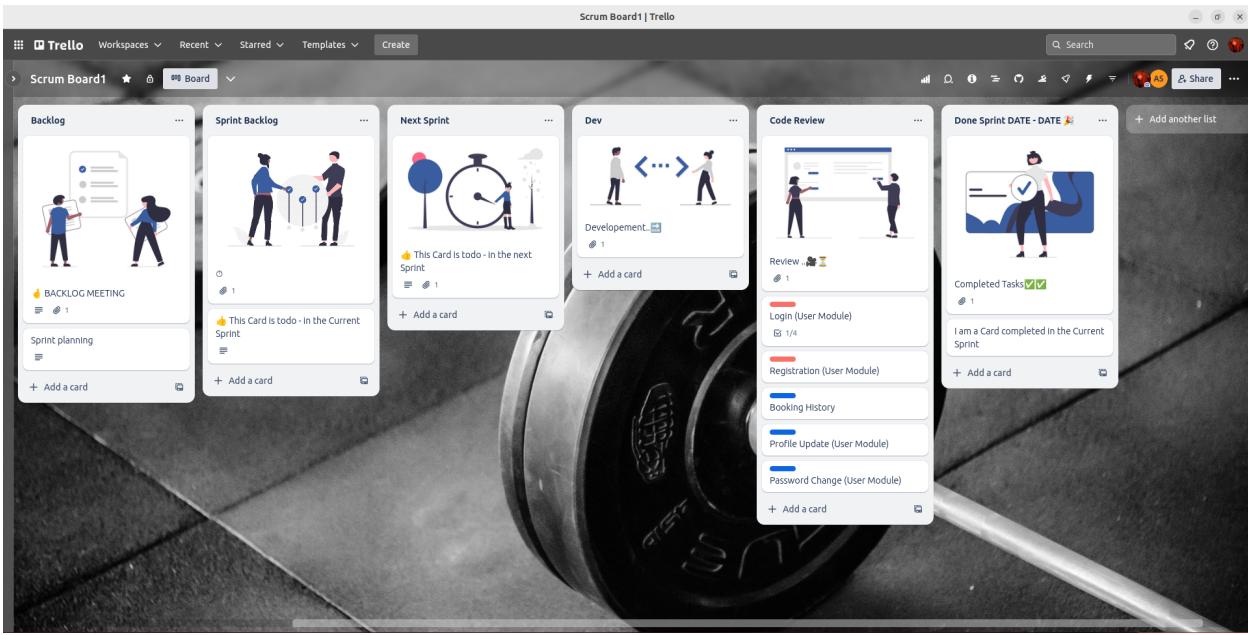


Figure 9: Sprint review

## 9 Backlog Second meeting

Dans le contexte du "backlog second meeting", nous avons tenu une deuxième réunion avec le Product Owner. Il nous a recommandé d'ajouter un nouvel acteur, l'"admin", afin de bien gérer le système, y compris la gestion des forfaits, des catégories, des réservations, etc.

## 10 Backlog Sprint2

Suite à la décision conjointe avec le Product Owner, il a été convenu que ce sprint serait étendu sur une période de 4 semaines afin de développer les fonctionnalités requises, tant du côté backend que frontend. Afin d'assurer une compréhension claire et détaillée du travail à réaliser, nous allons élaborer un cahier des charges exhaustif.

### 10.1 Analyse des besoins

#### 10.1.1 Identification des acteurs

**Admin :** L'admin est le responsable de la gestion globale du système.

#### 10.1.2 Les besoins fonctionnels

- Tableau de Bord :

- L'administrateur peut accéder à un tableau de bord affichant une vue d'ensemble des réservations, des forfaits, des catégories et des types de forfaits.

- Gestion des Catégories :

- L'administrateur peut ajouter et supprimer des catégories pour les forfaits de gym.

- Gestion des Forfaits :

- L'administrateur peut ajouter et supprimer des types de forfaits (par exemple, forfait mensuel, forfait annuel, etc.).
  - L'administrateur peut ajouter et éditer des forfaits, y compris leur description, leur prix, etc.

- Gestion des Réservations :

- L'administrateur peut vérifier les nouvelles réservations ainsi que les réservations avec paiement partiel/complet. Ici, l'administrateur peut également mettre à jour les détails de paiement pour une réservation particulière.

- Génération de Rapports :

- Dans cette section, l'administrateur peut générer des rapports entre des dates spécifiques pour les réservations et les utilisateurs inscrits.

- Gestion du Profil et des Mots de Passe de l'Administrateur :

- L'administrateur peut mettre à jour son propre profil, changer son mot de passe et récupérer son mot de passe en cas d'oubli.

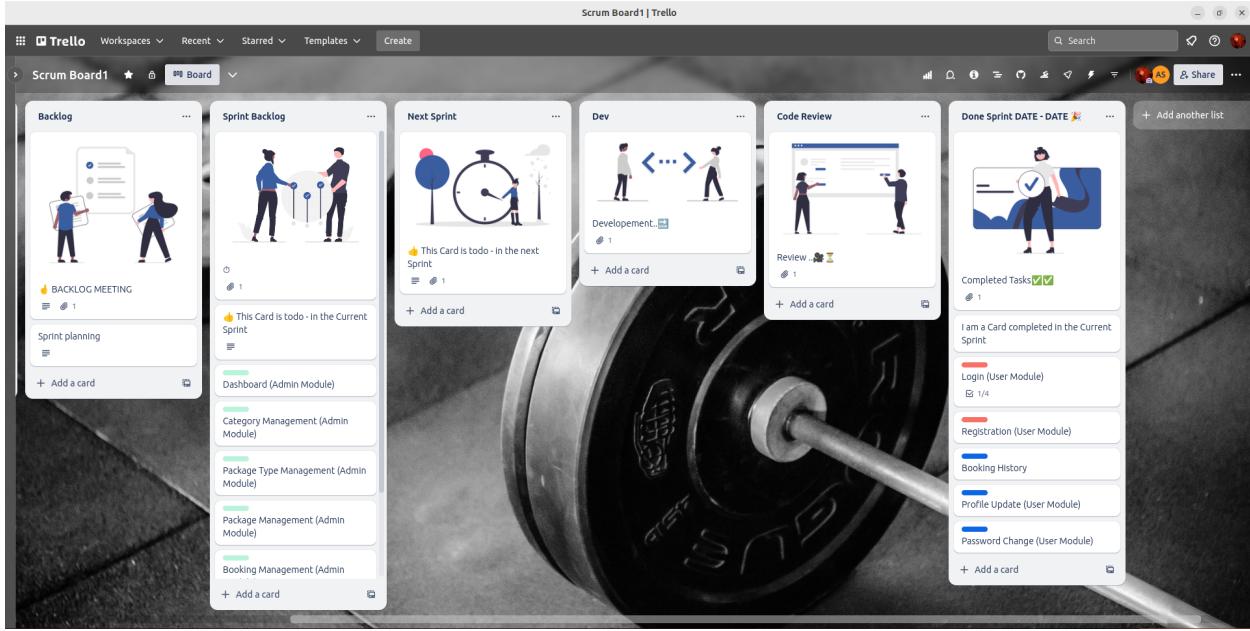


Figure 10: Les besoins fonctionnels d'acteur admin

#### 10.1.3 Les besoins non-fonctionnels

- **Interface Utilisateur Intuitive :**

- L'interface utilisateur du module admin doit être intuitive, facilitant la navigation et l'utilisation.

- **Sécurité :**

- Assurer la sécurité des informations sensibles, telles que les détails des utilisateurs et les transactions financières.

- **Performances du Système :**

- Garantir des performances optimales du système, permettant une réponse rapide aux requêtes de l'administrateur.

- **Disponibilité du Système :**

- Assurer une disponibilité continue du système, permettant à l'administrateur d'accéder aux informations à tout moment.

- **Extensibilité :**

- Concevoir le système de manière à permettre des extensions futures, facilitant l'ajout de nouvelles fonctionnalités.

- **Conformité aux Normes de Sécurité :**

- Respecter les normes de sécurité établies pour la protection des données et des transactions.

- **Documentation Complète :**

- Fournir une documentation complète pour le module admin, facilitant la compréhension et la gestion du système.

- **Notifications et Rapports en Temps Réel :**

- Mettre en place des notifications en temps réel pour les nouvelles réservations et générer des rapports actualisés selon les besoins de l'administrateur.

- **Compatibilité du Navigateur :**

- Assurer la compatibilité du module admin avec les navigateurs couramment utilisés, tels que Mozilla, Google Chrome, et Internet Explorer.

#### 10.1.4 Diagramme de contexte statique

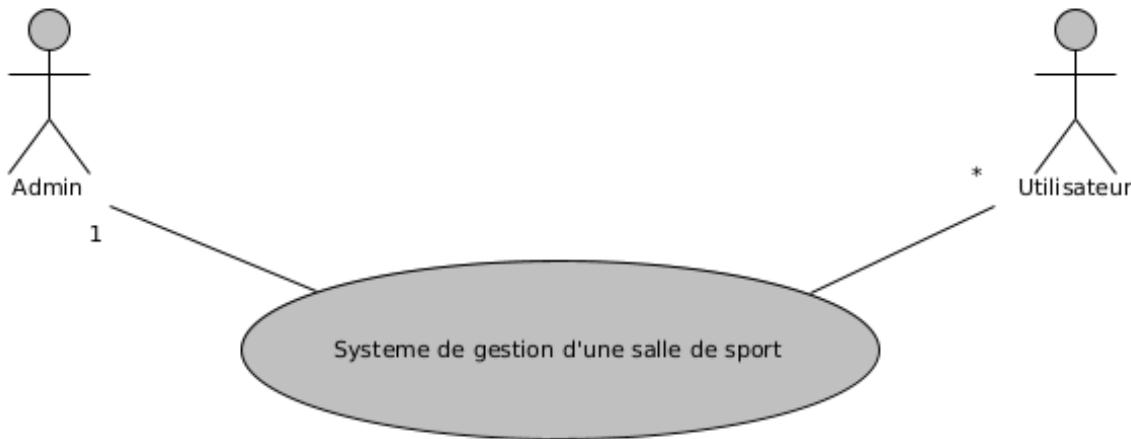


Figure 11: Diagramme de contexte statique sprint 2

### 10.1.5 Diagramme de cas d'utilisation



Figure 12: Diagramme de cas d'utilisation sprint 2

## 10.2 Sprint review

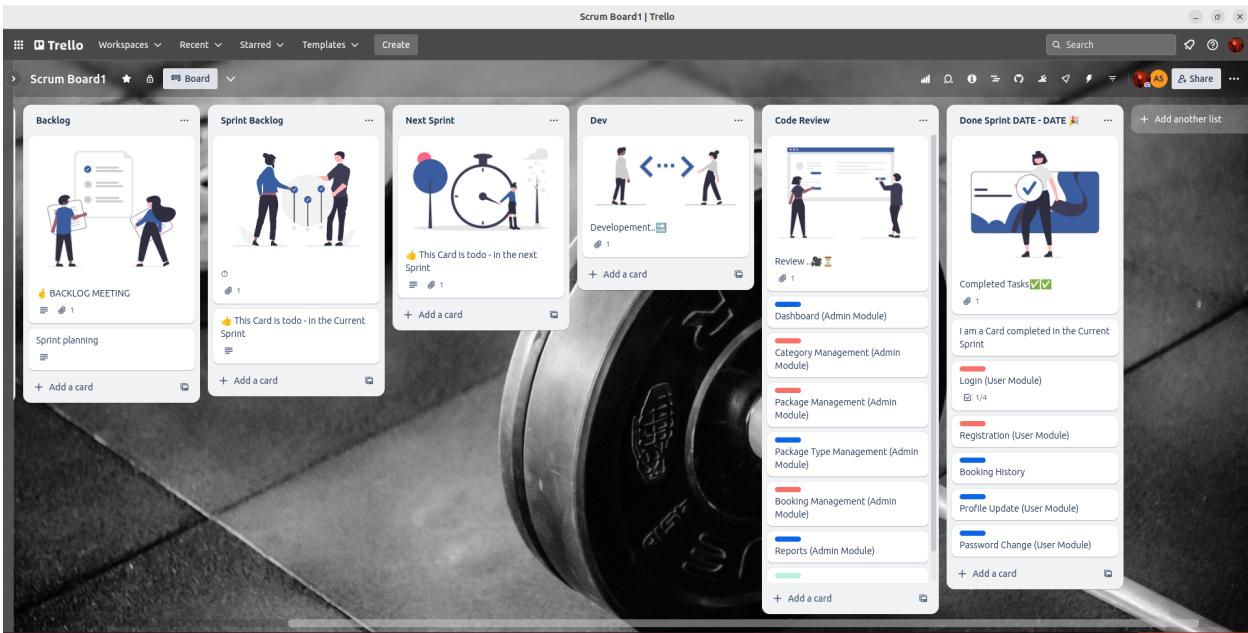


Figure 13: Sprint review

## 11 Backlog Final Meeting

Après avoir développé toutes les fonctionnalités demandées pour les acteurs admin et user, nous avons organisé une autre réunion, qui sera la dernière, afin de finaliser le travail ou d'ajouter des éléments supplémentaires. Le Product Owner a exprimé le désir d'apporter une dernière touche distinctive au système. Il souhaite intégrer une fonctionnalité permettant à l'utilisateur de choisir un repas depuis un restaurant partenaire de sa salle de sport. Le repas sera ensuite livré au client.

## 12 Backlog Sprint3

Suite à la décision conjointe avec le Product Owner, il a été convenu que ce sprint serait étendu sur une période de 2 semaines afin de développer cette fonctionnalité, tant du côté backend que frontend. Afin d'assurer une compréhension claire et détaillée du travail à réaliser, nous allons élaborer un cahier des charges exhaustif.

### 12.1 Analyse des besoins

#### 12.1.1 Identification des acteurs

**Clients/Utilisateurs :** Ce sont les membres de la salle de sport qui utilisent le système pour réserver des forfaits, consulter leur historique de réservation et mettre à jour leur profil.

**Admin :** L'admin est le responsable de la gestion globale du système.

#### 12.1.2 Les besoins fonctionnels

**Les besoins fonctionnels de l'utilisateur :**

- **Inscription :**

- Les utilisateurs peuvent s'inscrire sur le site web.
- Les utilisateurs fournissent des informations telles que leur nom, leur adresse e-mail, etc.
- Les informations d'inscription sont stockées dans la base de données.

- **Connexion :**

- Les utilisateurs inscrits peuvent se connecter à leur compte.

- **Historique des Réservations :**

- Les utilisateurs ont accès à un historique de leurs réservations de forfaits de gym.
- L'historique inclut des détails sur les forfaits réservés, les dates de réservation et les paiements effectués.

- **Mise à Jour du Profil :**

- Les utilisateurs peuvent mettre à jour leur profil, y compris des informations telles que l'adresse, le numéro de téléphone, etc.
- Les modifications sont enregistrées dans la base de données.

- **Changement de Mot de Passe :**

- Les utilisateurs peuvent changer leur propre mot de passe.
- Le processus est sécurisé et nécessite une authentification.

- l'utilisateur peut choisir un repas depuis un restaurant partenaire de sa salle de sport.

## Les besoins fonctionnels de l'admin

- **Tableau de Bord :**

- L'administrateur peut accéder à un tableau de bord affichant une vue d'ensemble des réservations, des forfaits, des catégories et des types de forfaits.

- **Gestion des Catégories :**

- L'administrateur peut ajouter et supprimer des catégories pour les forfaits de gym.

- **Gestion des Forfaits :**

- L'administrateur peut ajouter et supprimer des types de forfaits (par exemple, forfait mensuel, forfait annuel, etc.).
- L'administrateur peut ajouter et éditer des forfaits, y compris leur description, leur prix, etc.

- **Gestion des Réservations :**

- L'administrateur peut vérifier les nouvelles réservations ainsi que les réservations avec paiement partiel/complet. Ici, l'administrateur peut également mettre à jour les détails de paiement pour une réservation particulière.

- **Génération de Rapports :**

- Dans cette section, l'administrateur peut générer des rapports entre des dates spécifiques pour les réservations et les utilisateurs inscrits.

- **Gestion du Profil et des Mots de Passe de l'Administrateur :**

- L'administrateur peut mettre à jour son propre profil, changer son mot de passe et récupérer son mot de passe en cas d'oubli.

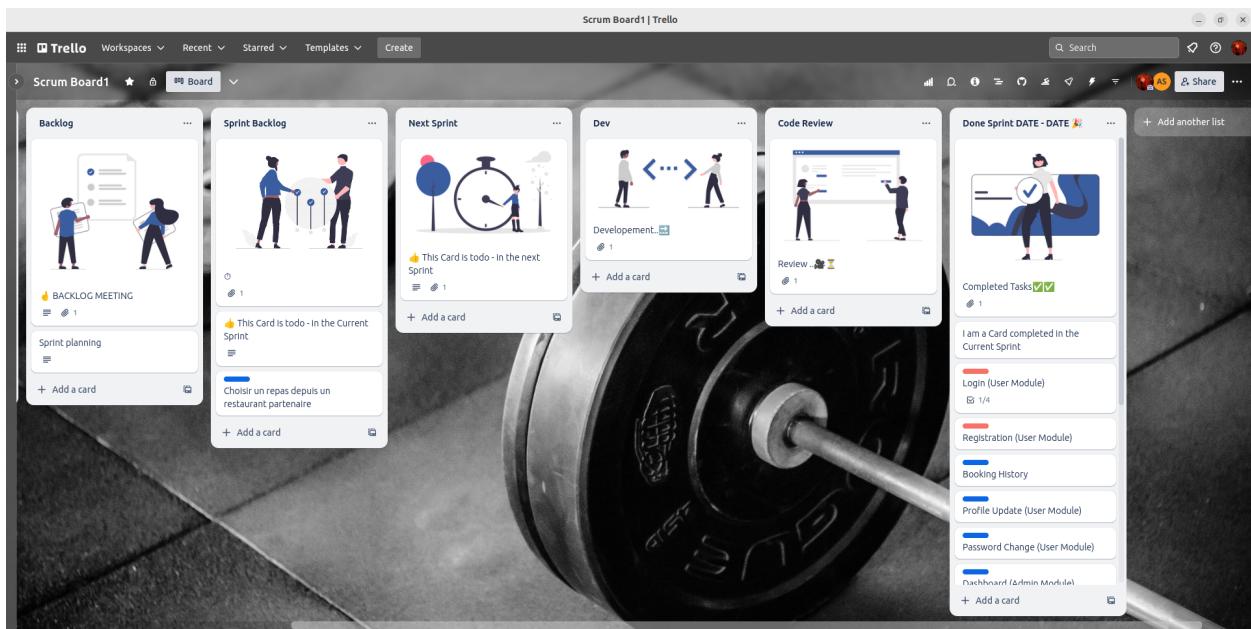


Figure 14: ajouter le dernier besoin fonctionnel

### 12.1.3 Les besoins non-fonctionnels

- **Interface Utilisateur Intuitive** : Faciliter la navigation et l'utilisation de l'interface utilisateur du module admin.
- **Sécurité** : Assurer la sécurité des informations sensibles, notamment les détails des utilisateurs et les transactions financières.
- **Performances du Système** : Garantir des performances optimales pour assurer une réponse rapide aux requêtes de l'administrateur.
- **Disponibilité du Système** : Assurer une disponibilité continue du système, permettant à l'administrateur d'accéder aux informations à tout moment.
- **Extensibilité** : Concevoir le système pour permettre des extensions futures sans perturber le fonctionnement.
- **Documentation Complète** : Fournir une documentation complète pour le module admin, facilitant la compréhension et la gestion du système.
- **Compatibilité du Navigateur** : Assurer la compatibilité du module admin avec les navigateurs couramment utilisés, tels que Mozilla, Google Chrome, et Internet Explorer.

### 12.1.4 Diagramme de contexte statique

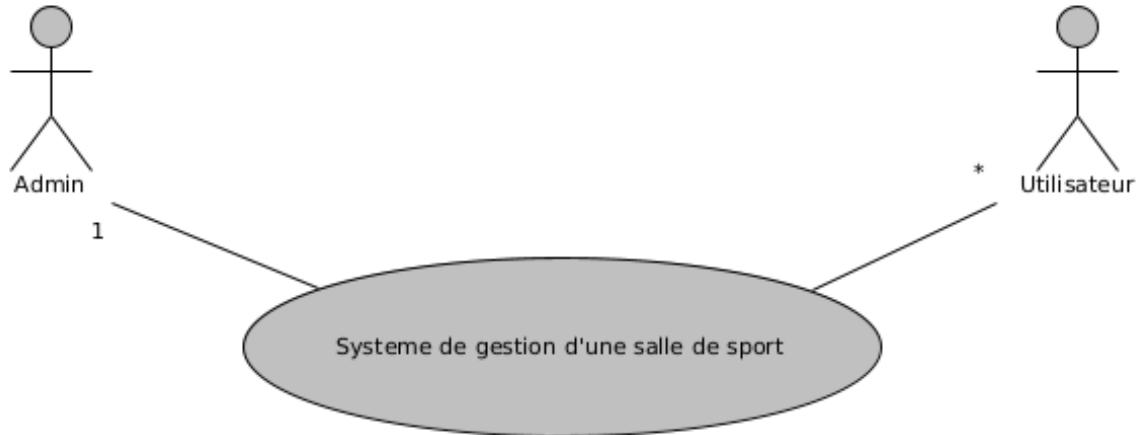


Figure 15: Diagramme de contexte statique final

### 12.1.5 Diagramme de cas d'utilisation

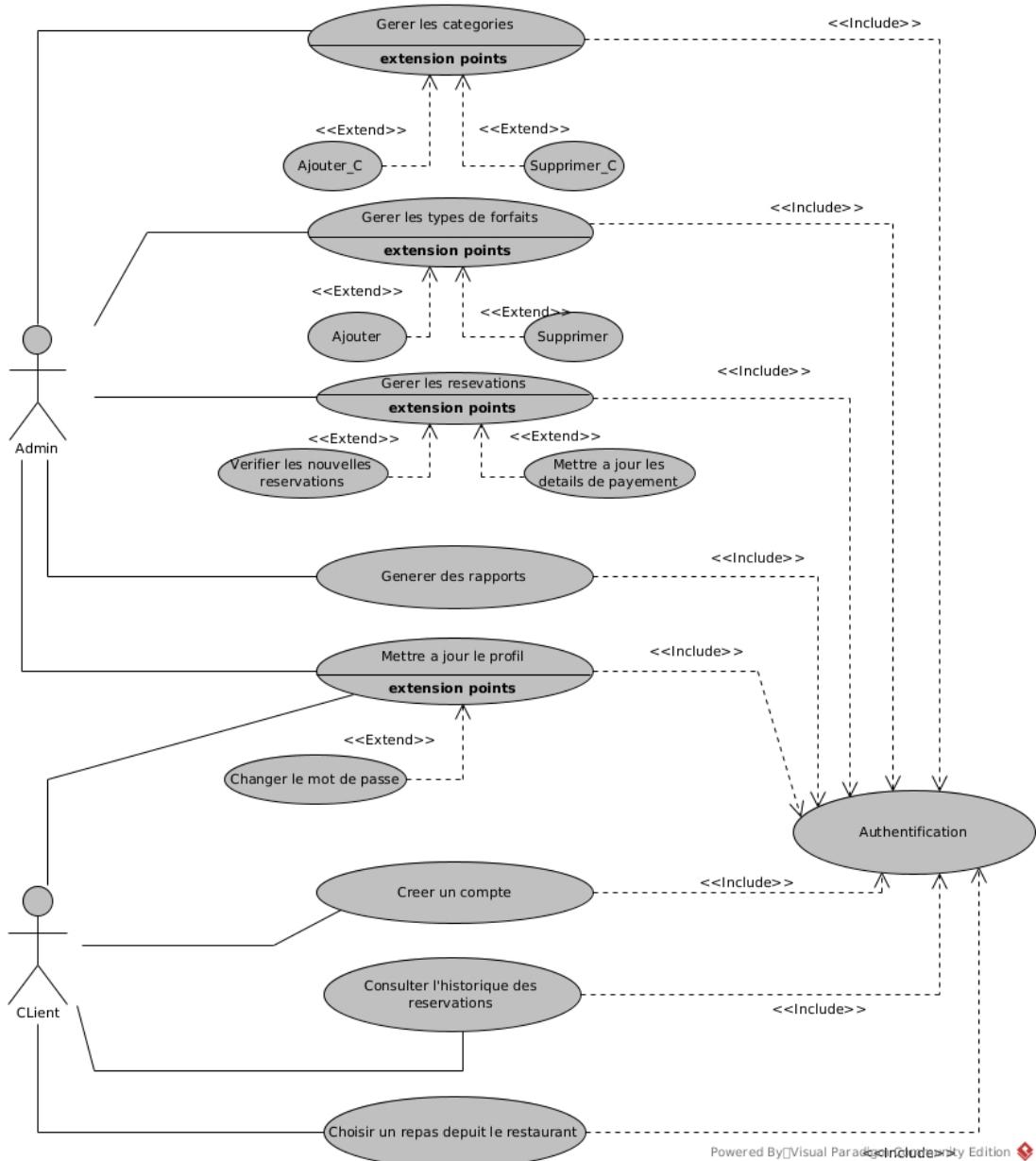


Figure 16: Diagramme de cas d'utilisation final

## 12.2 Sprint review

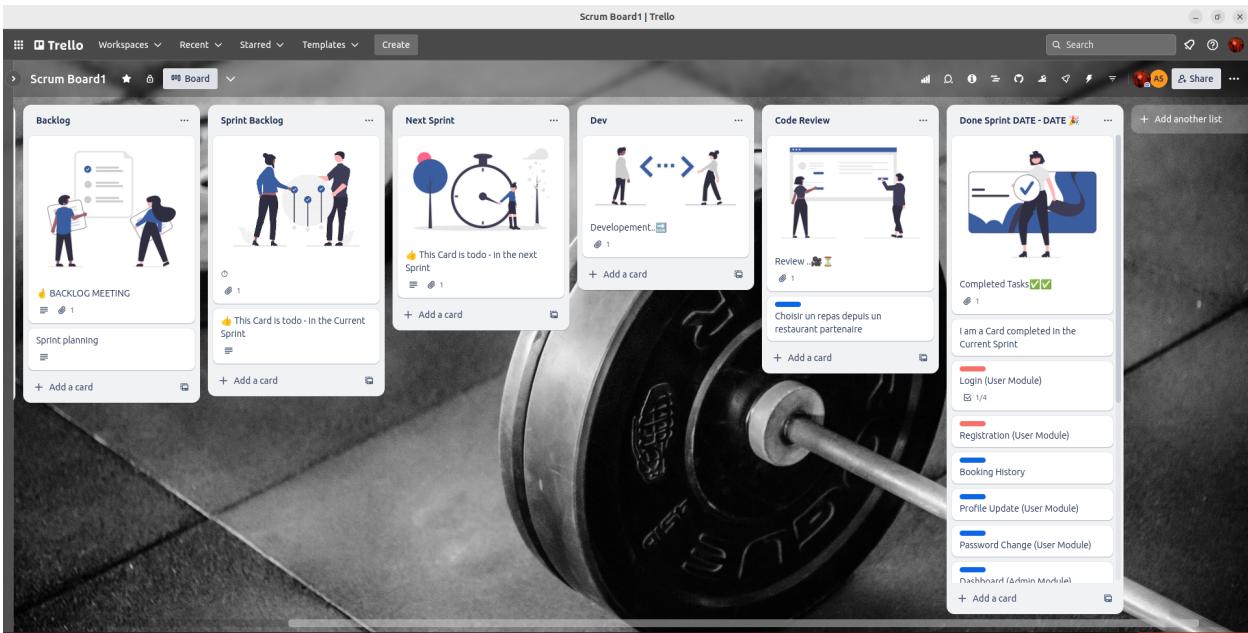


Figure 17: Sprint review

## 13 Résultat de travail (livraison)

Nous avons réalisé ce projet en appliquant la méthode agile Scrum, structurant notre travail sur trois sprints. Les deux premiers sprints ont été planifiés sur une période de quatre semaines chacun, tandis que le dernier sprint a été condensé sur une durée de deux semaines. Cette approche itérative nous a permis de maintenir un rythme de développement efficace et de garantir une flexibilité dans l'adaptation aux éventuels ajustements et défis rencontrés tout au long du processus. Grâce à cette méthodologie, nous avons pu atteindre nos objectifs dans un délai de deux mois et demi, assurant ainsi le succès de la réalisation de notre projet.

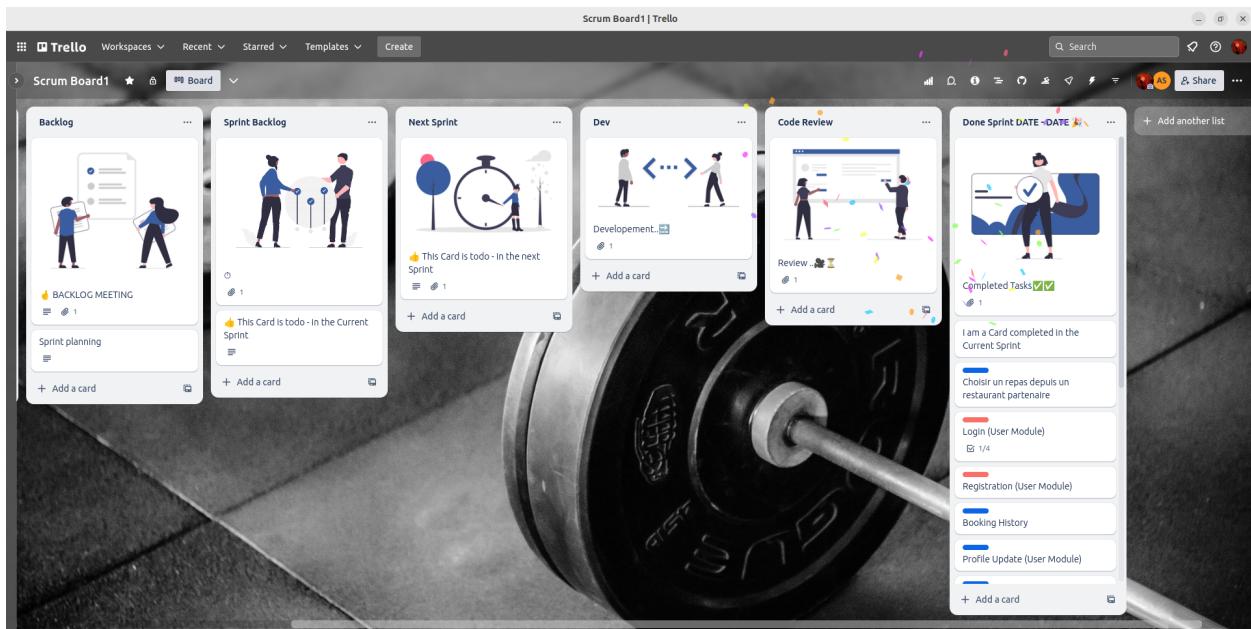


Figure 18: terminer les sprints

## 14 L'application "Gym Management System"

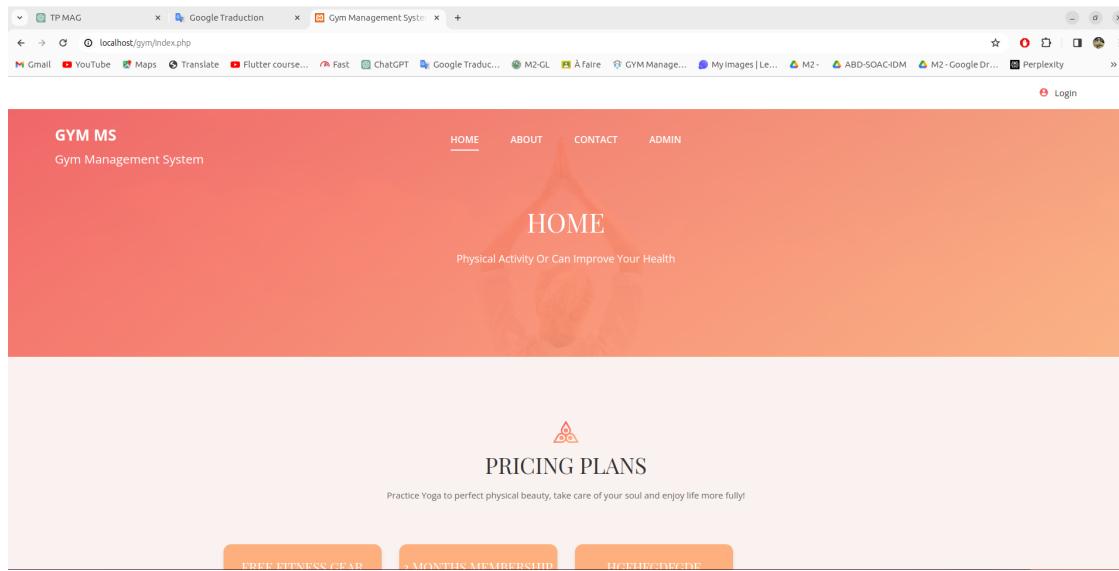


Figure 19: La page d'accueil du systeme

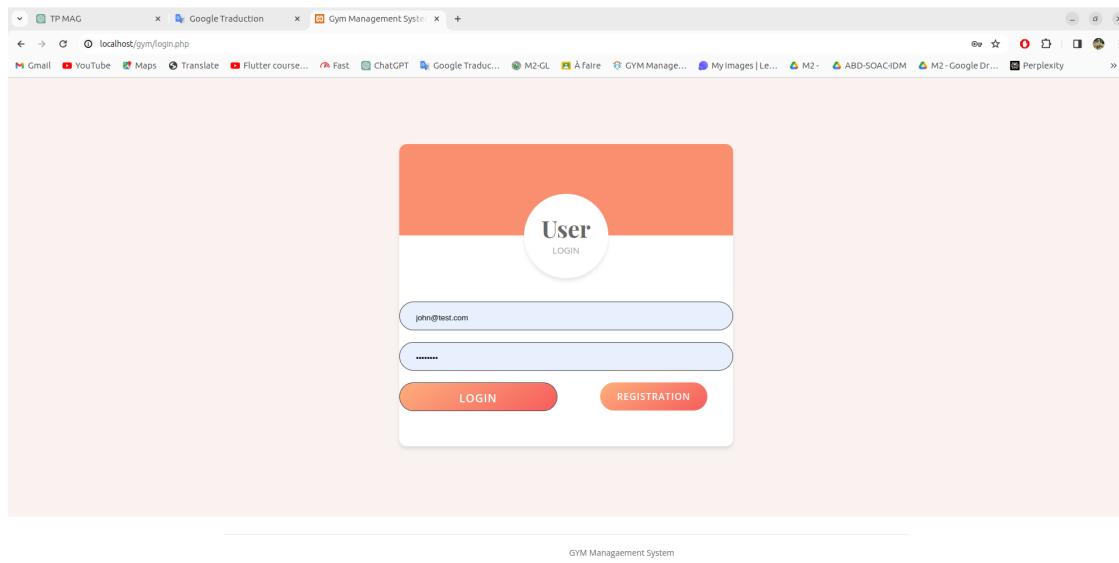


Figure 20: La page du login d'utilisateur

The screenshot shows a web browser window with the title "Gym Management System". The page has a red header with the word "REGISTRATION". Below the header is a form with six input fields arranged in two rows of three. The first row contains "First Name" and "Last Name" in separate input fields. The second row contains "Your Email" and "Mobile Number" in separate input fields. The third row contains "Your State" and "Your City" in separate input fields. Below the form is a red button labeled "REGISTER NOW".

---

Figure 21: La page d'inscription d'utilisateur

The screenshot shows a web browser window with the title "Gym Management System". The page has a red header with the word "HOME". Below the header is a slogan "Physical Activity Or Can Improve Your Health". In the center is a section titled "PRICING PLANS" with a small icon above it. Below the title is a quote "Practice Yoga to perfect physical beauty, take care of your soul and enjoy life more fully!". There are three pricing plans shown in orange boxes:

- FREE FITNESS GEAR PACKAGE** (3 MONTH) - Price: **600**
- 3 MONTHS MEMBERSHIP PACKAGE** (6 MONTH) - Price: **800**
- HGFHFGDFGDF** (4 MONTH) - Price: **12000**

Figure 22: La page d'accueil d'utilisateur

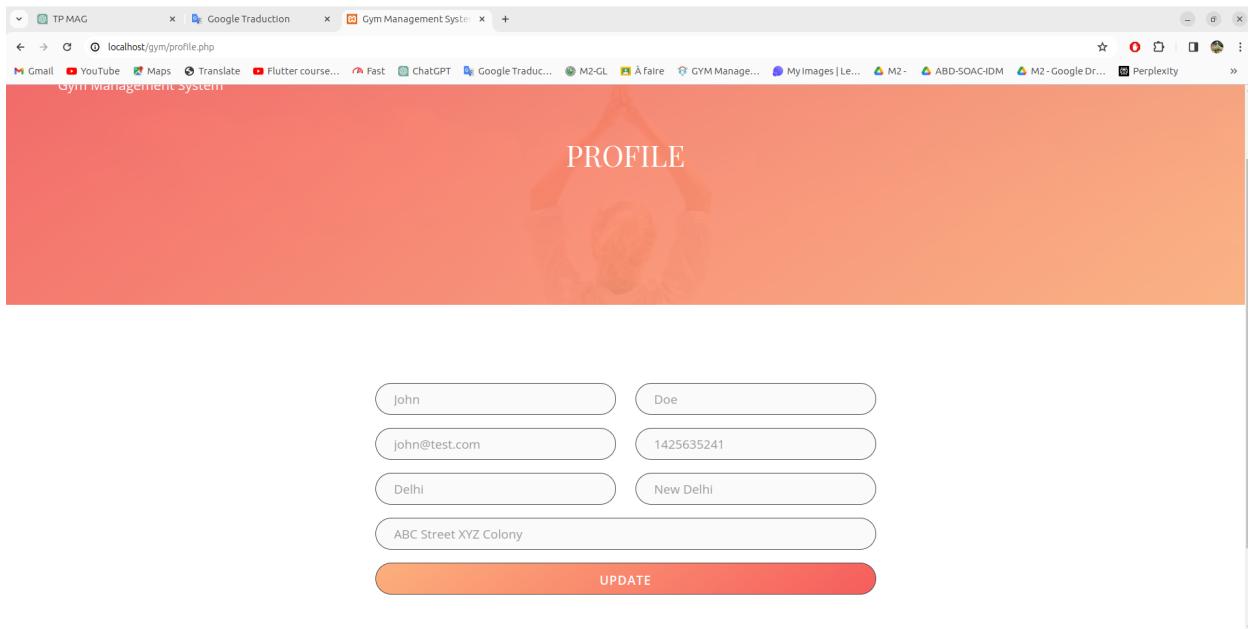


Figure 23: Mettre à jour le profil d'utilisateur

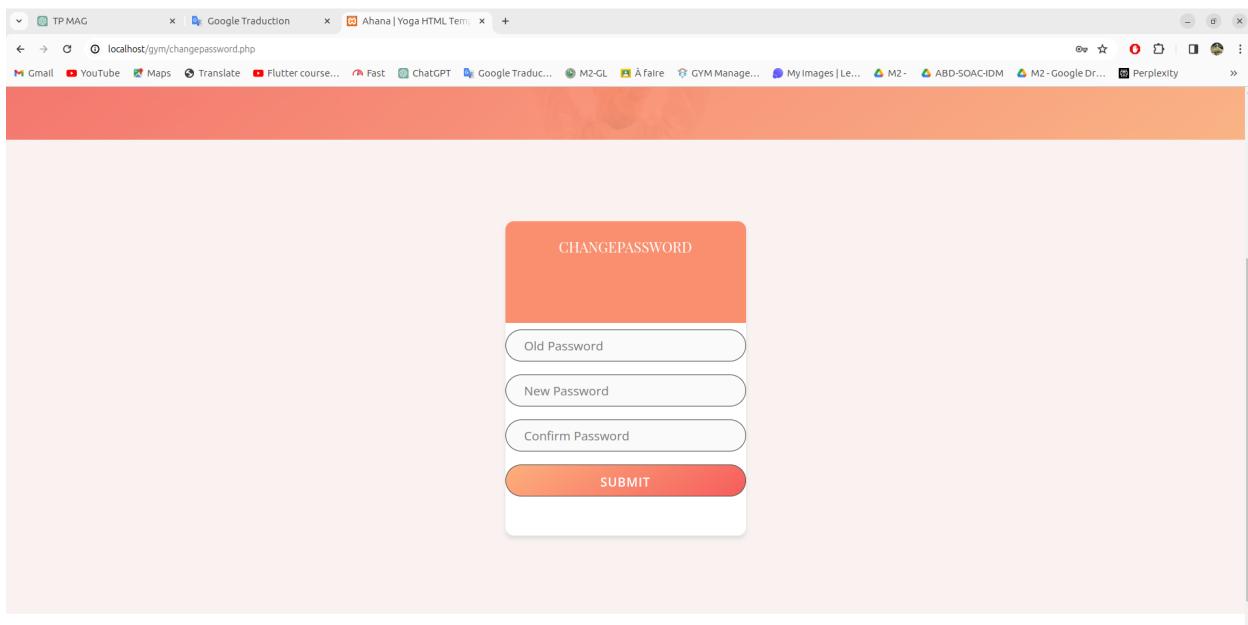


Figure 24: Changer le mot de passe d'utilisateur

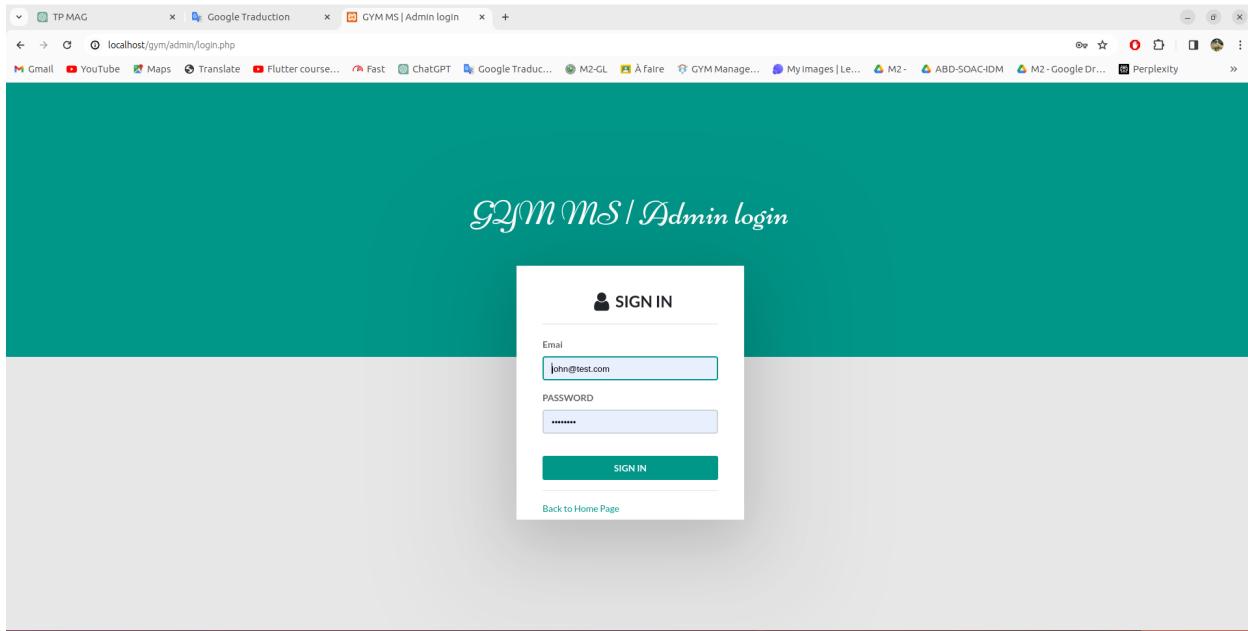


Figure 25: Le login de l'admin

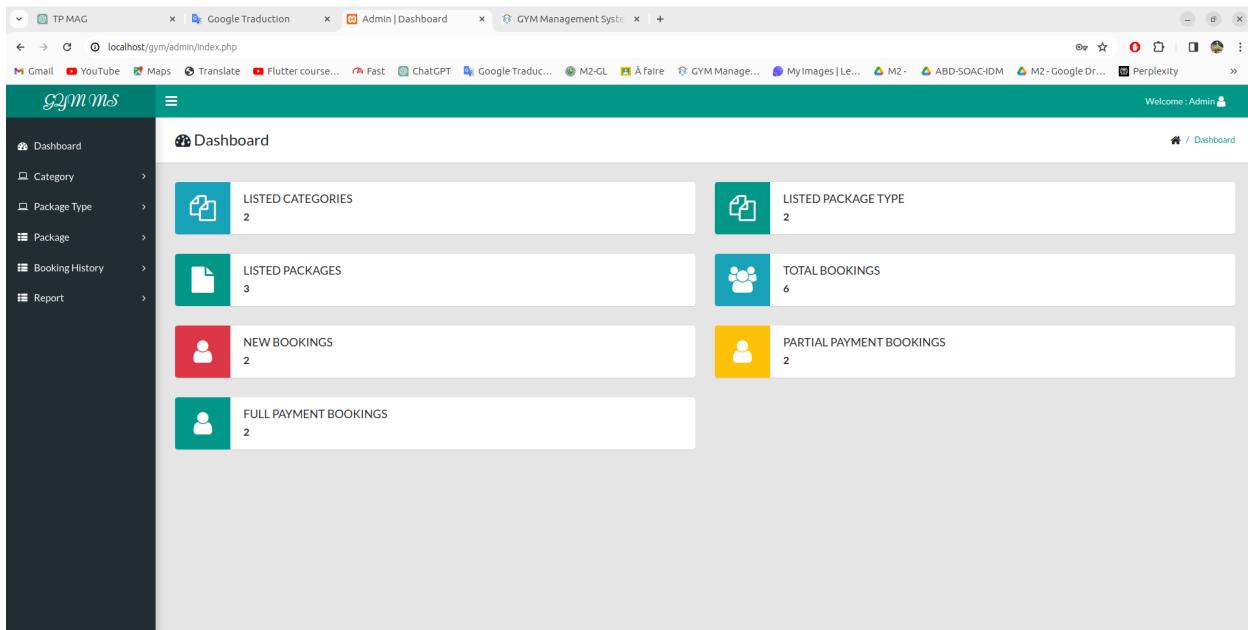


Figure 26: La page d'accueil d'admin

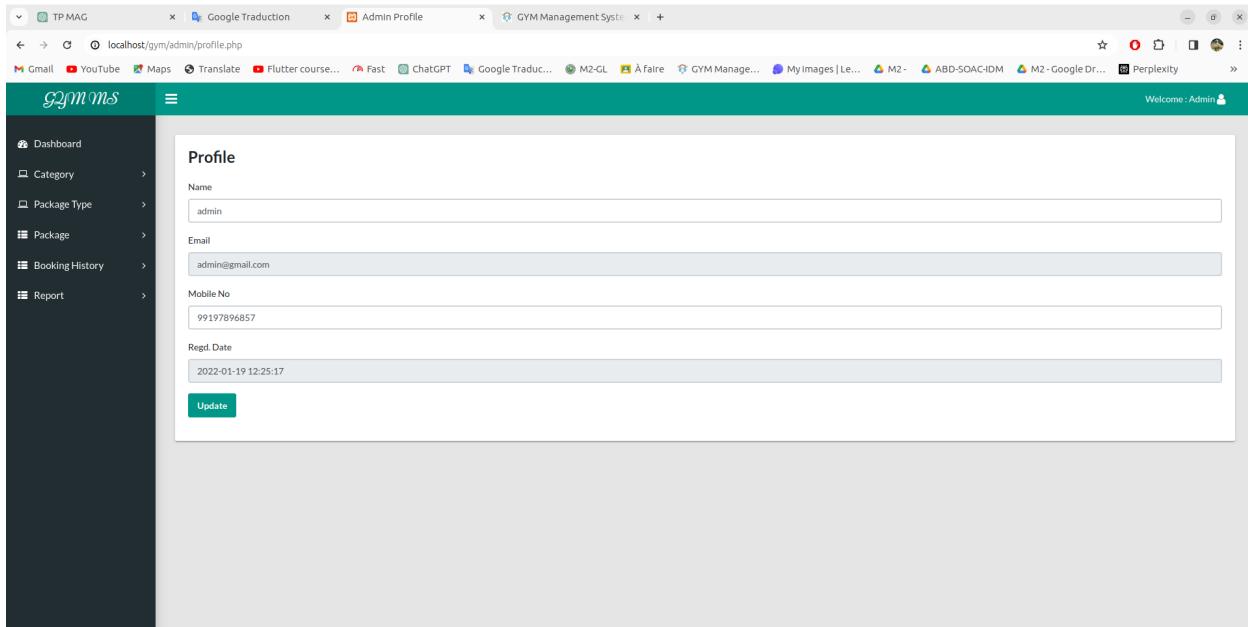


Figure 27: Mettre à jour le profil de l'admin

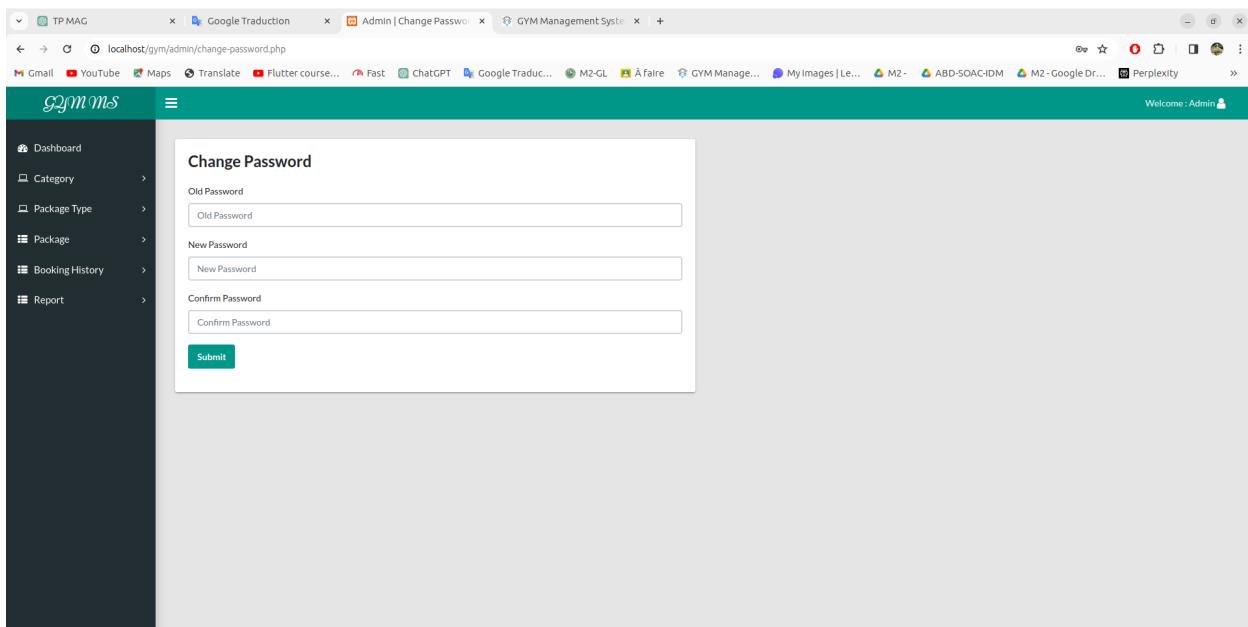


Figure 28: Changer le mot de passe de l'admin

The screenshot shows the 'Categories' section of the GYM Management System. On the left, a sidebar menu includes 'Dashboard', 'Category', 'Package Type', 'Package', 'Booking History', and 'Report'. The main area has a title 'Categories' and a sub-section 'Add Category' with a text input field 'Enter Add Category' and a 'Submit' button. Below this is a table listing categories:

Sr.No	Name	Action
1	Category1	Delete
2	Category2	Delete

At the bottom, it says 'Showing 1 to 1 of 1 entries' and has navigation buttons for 'Previous', '1', and 'Next'.

Figure 29: Ajouter une categorie

The screenshot shows the 'Add Package' page. The sidebar menu is identical to Figure 29. The main form has fields for 'Category' (dropdown), 'Title Name' (text input 'Enter your Title Name'), 'Price' (text input 'Enter your Price'), 'Package Type' (dropdown), 'Package Duration' (text input 'Enter Package Duratiobn'), and 'Description' (Rich Text Editor). A 'Submit' button is at the bottom.

Figure 30: Ajouter un package

Figure 31: Gerer les packages

Figure 32: gerer les reservations

## **15 Définition des grands choix techniques**

### **15.1 Langages Utilisés :**

- PHP : Pour la logique de gestion du système.
- HTML : Pour la structuration des pages web.
- AJAX : Pour des interactions asynchrones avec le serveur.
- JavaScript : Pour des fonctionnalités dynamiques du côté client.

### **15.2 Base de Données :**

- MySQL via phpMyAdmin : Pour le stockage et la gestion des données liées aux utilisateurs, aux réservations, et aux packages de la salle de sport.

### **15.3 Serveur de Développement :**

- XAMPP : Pour la création d'un environnement de développement local (inclus Apache, MySQL, PHP, et phpMyAdmin).

### **15.4 Outils de Développement :**

- Éditeur de Code : Utilisation de Visual Studio Code pour le développement.
- Contrôle de Version : Utilisation de Git pour la gestion de version du code source.

### **15.5 Outils de Collaboration :**

- GitHub : Pour le partage de code, la collaboration, et la gestion des problèmes.

### **15.6 Outils de documentation :**

- LaTeX : Pour la documentation du projet.

## **16 Conclusion**

En résumé, ce projet de gestion de salle de gym a été mené à bien en utilisant la méthodologie Scrum sur trois sprints. Les deux premiers sprints ont été consacrés au développement des fonctionnalités pour les utilisateurs et les administrateurs, tandis que le dernier sprint a ajouté une fonctionnalité permettant aux utilisateurs de commander des repas depuis un restaurant partenaire. Le processus Scrum, avec ses réunions régulières et son approche itérative, a permis une gestion efficace du projet, favorisant la flexibilité et l'adaptabilité tout au long du développement.