

République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Tunis El Manar
Faculté des Sciences de Tunis



RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention du
Diplôme National d'Ingénieur en Informatique
Spécialité : Ingénierie des Systèmes Embarqués et Mobiles

Par

Adam ABIDI

Mise en place d'une application mobile intégrant le principe "Pay As You Go" et d'un module de recommandation interactif

Encadrant professionnel :

Madame Wala Kasmi

Encadrant académique :

Madame Soukeina Ben Chikha

Réalisé au sein de WeCode Land

wecodeTM

Dédicaces

Je dédie ce travail à :

A la plus chère, à ma mère Samira pour ses sacrifices et son dévouement. A mon père Mongi qui a fait de moi l'homme que je suis aujourd'hui. A mes sœurs Balkis et Hela , mon frère Ilyes qui n'ont cessé de m'encourager tout au long de ma vie. A tous ceux qui me sont chers...

A vous tous, je dédie ce travail.

Adam ABIDI

Remerciements

Ce travail ne pouvait être mené à terme, sans la précieuse collaboration d'un certain nombre de personnes à qui je tiens à rendre hommage. Je voudrais adresser mes sincères remerciements à tous ceux qui ont fait de ce stage une expérience enrichissante et agréable, notamment :

A **Madame** Wala Kasmi, cofondatrice et directrice générale de WeCode, mon tuteur de stage de m'avoir bien accueilli au sein de son équipe. Je vous remercie pour vos conseils et votre aide tout au long de la période de stage.

A toute l'équipe du WeCode, ma deuxième famille, pour leur assistance, conseils et tout leur dévouement et persévérance. Je tiens à remercier particulièrement **Monsieur** Koussay Landolsi mon collègue pour son aide précieuse.

Je tiens à remercier particulièrement mon encadrant **Madame** Soukeina Ben Chikha pour sa disponibilité, ses remarques pertinentes et ses conseils judicieux. Je tiens à lui exprimer tout mon respect et ma reconnaissance.

J'exprime également mes sincères remerciements aux membres du jury pour avoir accepté d'évaluer ce travail.

Table des matières

Introduction générale	1
1 Cadre du projet	3
1.1 Présentation de l'organisme d'accueil	4
1.2 Contexte du projet	5
1.3 Problématique	6
1.4 Objectifs du projet	7
1.5 Méthodologie adoptée	8
1.5.1 Les méthodologies existantes	8
1.5.2 La méthodologie eXtreme Programming (XP)	9
1.5.3 User Stories du produit	12
2 Analyse et conception	14
2.1 Objectifs de l'application	15
2.2 Analyse des besoins	17
2.2.1 Identifications des acteurs	17
2.2.2 Besoins fonctionnels	17
2.2.3 Besoins non-fonctionnels	19
2.2.4 Spécification des besoins	20
2.3 Conception détaillée	28
2.3.1 Méthodologie de conception adoptée : UML	28
2.3.2 Logiciel de modélisation utilisé	29
2.3.3 Diagramme de classe	29
2.3.4 Diagrammes de séquences	32
2.4 Conclusion	37
3 Conception et spécification d'un système de recommandation	38
3.1 Les systèmes de recommandations	39
3.1.1 Principe des systèmes de recommandations	39
3.1.2 Filtrage de données : Comparer un système de recommandation à un moteur de recherche	40

3.1.3	Filtrage de données : Comparer un système de recommandation aux requêtes de bases de données	41
3.2	Les différents types des systèmes de recommandations	41
3.2.1	Filtrage collaboratif (Collaborative Filtering)	42
3.2.2	Les approches basées sur le contenu (content based)	43
3.2.3	Etude comparative entre les différents types de systèmes de recommandations	44
3.3	Mesure de similarité	47
3.4	Évaluation d'un système de recommandation	47
3.5	Proposition d'un système de recommandation à intégrer dans l'application Spory	48
3.5.1	Principe	48
3.5.2	Calcul de similarité : Similarité cosinus	49
3.5.3	Démarche	49
4	Implementation et réalisation	54
4.1	Environnement de développement	56
4.1.1	Introduction NoSQL	56
4.1.2	Mongo DB	57
4.1.3	Mongo DB Atlas	58
4.1.4	Déploiement continu	59
4.1.5	Heroku : Déploiement sur un serveur	60
4.1.6	Développement des applications hybrides	60
4.1.7	Technologies Front-End / Back-end	61
4.2	Architecture	62
4.2.1	Architecture matérielle	62
4.2.2	Architecture logicielle	63
4.3	Interfaces	65
4.3.1	Interfaces d'accueil	65
4.3.2	Scénario : Utilisation de l'application par l'adhérant	68
4.3.3	Scénario : utilisation de l'application par le gérant	74
4.3.4	Scénario : utilisation de l'application par l'administrateur	77
4.4	Conclusion	79
Conclusion générale		80

Table des figures

2.1	Exemple d'un QR code	16
2.2	Diagramme de cas d'utilisation général	21
2.3	Diagramme de cas d'utilisation «Consultation des centres»	22
2.4	Diagramme de cas d'utilisation « Réservation des séances»	23
2.5	Diagramme de cas d'utilisation « Ajout séance»	25
2.6	Diagramme de cas d'utilisation «Vérification facture»	27
2.7	Diagramme de classe	31
2.8	Diagramme de séquence « Utiliser GPS »	32
2.9	Diagramme de séquence «Réserver des séances»	34
2.10	Diagramme de séquence «Valider QR code»	36
3.1	Matrice de correspondance utilisateur-élément.	42
3.2	Schéma d'un système de recommandation collaboratif.	43
3.3	Schéma d'un système de recommandation basé sur le contenu.	44
3.4	Exemple de matrice de similarité cosinus.	51
3.5	Matrice de similarité cosinus des centre du catalogue donné par le tableau 3.3.	52
4.1	Coût de stockage des données par rapport au temps	57
4.2	Graphe du score des bases des données NoSQL	58
4.3	Interface de Github contenant les projets	59
4.4	phase de test et automatisation de Github	59
4.5	Interface de l'aperçu d'une application déployée sur Heroku	60
4.6	Logo d'Ionic	61
4.7	Logo de Node JS	62
4.8	Logo de Python	62
4.9	Architecture matérielle 3 tiers	63
4.10	Architecture logicielle détaillée	64
4.11	Page d'accueil	65
4.12	Slide d'accueil : présentation de geolocalisation	65
4.13	Slide d'accueil : présentation du calendrier de réservation	66
4.14	Slide d'accueil : présentation du QR code	66

4.15 Slide d'accueil : présentation d'autres services	67
4.16 Slide d'accueil : Invitation de passer à l'authentification	67
4.17 Interface d'authentification	68
4.18 le mail de vérification de l'adresse électronique	68
4.19 Carte initiée	69
4.20 GPS tracé	70
4.21 Interface Carte contenant la liste des salles	70
4.22 Historique des achats	71
4.23 Les réservations à venir	71
4.24 Page officielle du centre vu par l'adhérent	72
4.25 Interface de réservation	72
4.26 Interface Découvrir	73
4.27 Interface Découvrir	73
4.28 Page officielle vue par le gérant	74
4.29 Page officielle contentant l'option d'ajout des sessions	74
4.30 Interface des statistiques dynamiques	75
4.31 Interface de scan des codes QR	75
4.32 Process réel de scan exécuté sur mobile	76
4.33 Liste des adhérents pour une date fixée	76
4.34 Interface de validation des centres	77
4.35 Interface de blocage des centres	77
4.36 Interface de gestion des adhérents	78
4.37 Interface de gestion des commentaires	78

Liste des tableaux

1.1	Comparaison entre la méthodologie classique et la méthodologie agile.	9
1.2	La liste des User Stories du projet	13
2.1	Liste des besoins fonctionnels par acteur	18
2.1	Liste des besoins fonctionnels par acteur	19
2.2	Description textuelle du cas d'utilisation «Consultation des centres»	23
2.3	Description du diagramme de cas d'utilisation « Réservations des séances»	24
2.4	Description textuelle du diagramme de cas d'utilisation «Ajout séance»	26
2.5	Description textuelle de diagramme de cas d'utilisation « Vérification d'une facture» .	28
3.1	Comparaison entre système de recommandation, moteur de recherche et requêtes bases de données.	41
3.2	Tableau comparatif entre les différents types de systèmes de recommandations	45
3.2	Tableau comparatif entre les différents types de systèmes de recommandations	46
3.3	Exemple d'un catalogue de centres de remise en forme.	50
3.4	Vectorisation du catalogue des centres de remise en forme, donné par le tableau 3.3. .	52
3.5	Exemple de centres de remise à forme à recommander, selon la matrice de similarité donnée par la figure 3.5.	53

Introduction générale

De nos jours, le digital dans la société de consommation a pris une place considérable, il représente une nouvelle façon d'attirer de la clientèle mais aussi une nouvelle méthode de consommation de plus en plus répandue.

En effet, la consommation digitale constitue aujourd'hui un moyen rapide et pratique aussi bien pour effectuer des achats mais aussi pour exprimer différents avis de satisfaction ou de mécontentement vis-à-vis des produits de la société. Nous avons d'une part une évolution continue des technologies, mais aussi des clients dont les attentes se veulent plus strictes du fait de la concurrence et de la Digitalisation qui se répond de plus en plus.

Cette révolution digitale a sans nul doute conféré du pouvoir aux consommateurs qui dispose de son coté de la possibilité d'interagir avec l'entreprise. La communication est devenue bilatérale et interactive, puisque ce dernier a la possibilité de répondre, de réagir, de parler, d'exprimer un avis, une opinion ou une insatisfaction.

Il est aussi important de signaler que l'essor des technologies de l'information et de la communication a modifié la manière dont les gens communiquent. Le consommateur d'aujourd'hui est de plus en plus informé, grâce aux informations disponibles sur le web et d'autres supports, grâce à la tribune que lui donne l'accès aux réseaux sociaux et aux communautés virtuelles, il a le pouvoir de nuire à la réputation d'une entreprise ou d'une marque, il a le pouvoir de donner des recommandations, des prescriptions et d'influencer les décisions de ceux qui font partie de son entourage, et qui peuvent lui faire confiance plus que n'importe quel autre message à caractère publicitaire.

De ce fait, les entreprises investissent de plus en plus dans le digital en raison de son augmentation exponentielle, de ses aspects pratiques et de son impact conséquent.

Chaque société se doit donc de favoriser les exigences et les attentes croissantes de ses clients afin de les fidéliser au risque de les voir migrer vers la concurrence et de voir sa réputation ruinée.

On constate donc l'importance des plateformes digitales qui émergent comme des acteurs économiques incontournables, transformant profondément les industries au sein desquelles elles jouent le rôle d'intermédiaire entre les différentes parties prenantes d'un même écosystème tout en favorisant les échanges économiques. Ces plateformes vont pouvoir s'adapter aux besoins et exigences des consommateurs tout en leur proposant des services adaptés à leurs préférences d'une part mais aussi en leur conférant le pouvoir d'interagir en faisant part de leur satisfaction ou de leur mécontentement.

C'est dans ce contexte, que se situe notre projet. Il s'agit d'une application digitale, appelée **Spory**, qui a pour objectif d'intégrer le marketing digital dans le secteur sportif (centres de sport et de remise en forme). Notre application adopte le principe « *Pay As You Go* » : Pas d'abonnement, le client paye ce qu'il consomme. Autrement dit, les centres de remise en forme s'adaptent aux exigences des clients, où le choix des centres, des offres et l'horaire est fixé par le client. L'application à développer présente son aspect mobile favorisant l'accessibilité pour tout le monde, à n'importe quel moment et par n'importe quel périphérique. L'application offrira en particulier trois fonctionnalités clés à savoir : *un système de géolocalisation*, *un système de scan QR-CODE* et *un Système de recommandation*. L'application devra aussi présenter des modules d'aide à la décision qui permettront aux différents gérants d'avoir une visibilité globale sur le pourcentage de présences ainsi que des plages horaires les plus fréquentées par les clients.

Afin d'illustrer la démarche de notre travail, nous présentons dans ce qui suit la décomposition du rapport. Le rapport s'articule autour de quatre chapitres :

- Dans le premier chapitre, nous exposons le contexte général de notre projet, ainsi que l'organisme d'accueil *WeCode Land*. Nous détaillons aussi la méthodologie adoptée.
- Dans le deuxième chapitre, nous spécifions et analysons les besoins de notre application. Par la suite, nous détaillons les exigences fonctionnelles et non-fonctionnelles et nous présentons les différents aspects conceptuels de l'application.
- Dans le troisième chapitre, nous présentons la conception et la spécification d'un système de recommandation intégré au sein de notre application.
- Dans le quatrième chapitre, nous détaillons les technologies utilisées et l'architecture choisie pour assurer le fonctionnement de notre application. Nous exposons également la partie réalisation du projet ainsi que le résultat final de l'application avec des captures d'écran.

Nous terminons par une conclusion générale qui résume notre travail et expose quelques perspectives pour notre application.

CADRE DU PROJET

Plan

1	Présentation de l'organisme d'accueil	4
2	Contexte du projet	5
3	Problématique	6
4	Objectifs du projet	7
5	Méthodologie adoptée	8
1.5.1	Les méthodologies existantes	8
1.5.2	La méthodologie eXtreme Programming (XP)	9
1.5.2.1	Présentation de méthodologie XP	9
1.5.2.2	Principe de la méthodologie XP	10
1.5.2.3	Pratiques de base de la méthodologie XP	10
1.5.3	User Stories du produit	12

Introduction

Ce premier chapitre a pour objectif de situer le projet dans son cadre général. De ce fait, nous allons commencer par une introduction de l'organisme d'accueil WeCode Land, ses produits ainsi que ses activités. Par la suite, nous présentons le contexte du projet ainsi que la problématique identifiée. Enfin, nous clôturons ce chapitre en présentant la méthodologie de travail adoptée.

1.1 Présentation de l'organisme d'accueil

WeCode Land a commencé comme une idée en 2014 pour fournir des formations en économie numérique et aider les jeunes économiquement inactifs à rejoindre le marché du travail en obtenant des emplois numériques ou en lançant leurs entreprises numériques.

« WeCode Land » est une start-up qui a une ambition de soutenir la transition numérique dans les pays émergents en offrant des formations à l'économie numérique aux jeunes. A ce jour, « WeCode Land » a servi plus de 4000 jeunes en Tunisie et est présente dans 4 pays différents de la région MENA.

En 2020, «WeCode Land» annonce un partenariat international avec UNITY Technologies, leader mondial du développement de logiciels 3D, réalité virtuelle et réalité augmentée. WeCode Land représente désormais les formations et les certifications UNITY Technologies dans la région et offre tout le catalogue des formations en temps réel de la plateforme UNITY. Par ce partenariat, « WeCode Land » confirme sa position de leader de la transformation digitale en Tunisie et dans la région et consolide ainsi ses projets de développement axés sur les opportunités offertes aux jeunes de rejoindre ou de s'intégrer aux nouveaux marchés du travail, par les métiers du digital et même en créant leur propre entreprise digitale.

WeCode Land gère également plusieurs autres projets dans divers secteurs d'activités, à savoir :

- Le coding : WeCode accompagne les jeunes en les formant les bases de développement informatique notamment le web, mobile, data et même les jeux vidéos.
- Le conseil et l'assistance.
- La refonte et l'optimisation des processus métiers.
- Le support aux grands projets de transformation.
- Le marketing digital.
- La communication

WeCode Land a participé dans divers concours et a remporté des différents prix telque :

- Digital Woman of the Year Award for the Founder of WeCode Land 2017.
- Ashoka Fellowship 2017.
- Tunisian Youth Ministry Honor 2016.
- Tunisian Parliament Honor 2016.
- Orange Most Innovative Social Entrepreneur 2015.
- French President François Holland Award for Social

1.2 Contexte du projet

Avec l'explosion digitale au cours ces vingt dernières années, le marketing digital s'est imposé comme LA discipline actuelle pour les entreprises de toute taille et toute branche d'activité, par le fait des différents avantages qu'il génère pour les entreprises mais aussi du fait qu'il apporte une réponse sur-mesure aux nouveaux comportements des consommateurs actuels [1]. Chaque jour, nous apprenons et nous échangeons, via internet, nos smartphones, nos tablettes et autres objets connectés. En effet, plus de 4,5 milliards de personnes utilisent aujourd'hui Internet, tandis que le nombre d'utilisateurs de médias sociaux a franchi la barre des 3,8 milliards [2]. Près de 60% de la population mondiale est désormais en ligne. Plus de 5,19 milliards de personnes dans le monde utilisent un téléphone portable, un nombre d'utilisateurs en hausse de 124 millions (2,4%) depuis l'année dernière. Cette tendance n'échappe pas à la Tunisie qui compte 7,55 millions d'internautes en janvier 2020, avec 7,30 millions d'utilisateurs des réseaux sociaux et 17,77 millions de connexions mobiles [3]. D'ailleurs le nombre de connexions mobiles en janvier 2020 équivalait à 151% de la population tunisienne totale.

Avec ce passage obligé, les sites Internet, qui ont été considérés comme un outil indispensable pour être visible et pour se conformer aux attentes des consommateurs, ne suffisent plus aux entreprises pour se faire connaître sur le marché et cibler le consommateur. Des stratégies se sont ainsi développées au fur et à mesure, englobées dans ce qu'on nomme le marketing digital [4]. Ce nouveau concept de marketing est souvent confondu avec le webmarketing [5], car il s'appuie essentiellement sur le web. Cependant, le marketing digital est en réalité désigné par l'ensemble des pratiques qui ont pour but de promouvoir une entreprise, de capter des leads et de gagner de nouveaux clients via différents supports numériques, à savoir [6] :

- Le web : sites corporate et sites marchands, blogs.
- Les réseaux sociaux.
- Le SEO : référencement naturel sur les moteurs de recherche.

- Le SMO : référencement naturel sur les plateformes social media.
- Le marketing mobile : applications, m-couponing, notifications push géolocalisées, etc.
- Les campagnes d'emailings.
- Le Big Data : l'exploitation des données des internautes.
- Le marketing local : annonces publicitaires géolocalisées sur les moteurs de recherche, réseaux sociaux, etc.

Cette nouvelle tendance de marketing permet de coller au plus près des habitudes et des exigences des consommateurs actuels en créant une relation de proximité et de confiance avec eux. Le client peut interagir à tout moment avec l'entreprise, qui détient à son tour une foule d'informations à l'égard du client et peut adapter son offre pour lui proposer le produit ou le service qui lui correspond exactement. La relation entreprise-client a totalement évolué : les entreprises connaissent finement leurs consommateurs et bâtiennent ainsi une relation durable et de qualité avec eux, ce qui favorise le taux de fidélisation et de satisfaction.

1.3 Problématique

Le marché des centres de sport et de remise en forme voit sa croissance augmenter en Tunisie. En effet, ces dernières années on a pu voir la naissance de plusieurs centres, avec une concurrence exacerbée entre les clubs low cost, le premium et les établissements plus traditionnels. Malgré que ce secteur demeure un domaine avec un grand potentiel et de forte croissance, il reste néanmoins mal exploité numériquement.

En effet, au jour d'aujourd'hui, en Tunisie , il n'existe toujours pas de plateforme digitale regroupant tous les centres de loisirs, ni d'un catalogue numérique unifié affichant les offres de divertissement. A ce jour, le consommateur tunisien est obligé de parcourir les salles de sport et de remise en forme manuellement afin de pouvoir juger lui-même la globalité de l'offre. De plus, il est difficile de prendre des avis et des retours des clients, puisque la digitalisation est négligée dans ce domaine.

En plus presque la totalité des propositions sont classiques et statiques, tous les détails sont fixes, le local et la date sont permanents. Cela assigne que la flexibilité est quasiment inexistante. Le client est obligé d'assister à un et un seul centre de remise en forme pendant une période fixe, sans oublier que les jours non consommées sont inclus dans le contrat de l'inscription, dans ce genre de cas de figure le consommateur paie donc plus.

Le digital fait évoluer l'ensemble des secteurs économiques, le secteur sportif ne doit pas résister à cette révolution numérique. Intégrer le marketing digital dans sa stratégie d'acquisition client est

aujourd’hui indispensable par les centres de remise en forme. En effet, les canaux digitaux sont ceux privilégiés par les clients pour chercher les offres adaptées à leurs besoins. Adopter une stratégie digitale constitue donc un moyen de se conformer aux pratiques des clients. Pour acquérir de nouveaux clients grâce au marketing digital, il faut optimiser la visibilité de l’entreprise et de ses produits sur le Web. Cette action est possible grâce à l’utilisation des leviers comme la création de contenu, la publicité payante mais surtout par des tunnels de vente. C’est dans ce contexte que se situe notre projet, qui a pour objectif de proposer une plateforme digitale, appelée Spory, permettant de regrouper les centres de remises en forme et de s’incliner aux besoins et aux exigences des consommateurs, à savoir une flexibilité de paiement, de temps et de lieu.

1.4 Objectifs du projet

Notre application digitale Spory adopte le principe « *Pay As You Go* » : Pas d’abonnement, le client paye ce qu’il consomme. Autrement dit, les centres de remise en forme s’adaptent aux exigences des clients, où le choix des centres, des offres et l’horaire est fixé par le client. L’application à développer présente son aspect mobile favorisant l’accessibilité pour tout le monde à n’importe quel moment.

Notre idée encourage de remplacer la démarche classique avec un système de réservation dynamique. Elle consiste à proposer un abonnement au client lui donnant l’accès aux différents centres de remise en forme partenaires sans engagement, sans contrainte de temps et avec des tarifs flexibles. L’utilisateur a la possibilité de visiter n’importe quel endroit de divertissement tant qu’il a, à sa disposition, son abonnement numérique.

Pour suivre la tendance de la digitalisation, notre objectif est de proposer à l’utilisateur des recommandations personnalisées de centres de remise en forme, c'est-à-dire qu'elles sont établies en fonction de l'utilisateur qui en est le bénéficiaire et non pour l'ensemble des utilisateurs. Pour aboutir à ce but, l'application intègre un système de recommandations. Ainsi, en accumulant une quantité de données sur l'utilisateur (son comportement d'achat et de navigation, ses caractéristiques) et en les croisant, par traitement algorithmique, notre application sera en mesure de prédire les biens qui seront consommés par cet utilisateur.

En plus, notre application a pour objectif d'aider l'utilisateur à choisir le centre le plus adéquats à ses contraintes, en intégrant un système de géolocalisation, permettant de déterminer la position précise de l'utilisateur et de lui afficher les centres de remise en forme situés dans sa zone géographique et de calculer leurs distances.

L’application Spory permet également aux clients de donner leurs avis sur les centres visités. Ces

avis sont sous forme de commentaires rédigés ou des notations avec des nombres d'étoiles. Notre objectif est d'aider d'une part les clients à choisir leurs destinations, mais aussi aider les responsables des centres de remise en forme à avoir un feed-back sur les prestations proposées.

En outre, notre application Spory introduit le principe de dématérialisation des abonnements, permettant aux usagers d'acheter et de gérer leurs abonnements depuis leur smartphone. Plus besoin de carte d'abonnement ou de justificatif papier, le smartphone suffit ! Pour ce faire nous proposons de remplacer la carte d'abonnement (ou la facture) par un code-barres à deux dimension sécurisé, de type QR Code qui est affiché directement sur l'écran du smartphone. La dématérialisation des abonnements permet au client d'être :

- Plus léger : Le client n'a plus besoin d'avoir sa carte d'abonnement, elle est accessible en un clic sur son smartphone.
- Plus simple : Tous les documents d'abonnement (réservation, carte de réduction, etc) sont enregistrés sur le smartphone.
- Plus tranquille : Au moment d'accès au centre, plus de risque d'oubli ou de perte, il lui suffit de présenter le QR code de sa réservation sur son smartphone.

1.5 Méthodologie adoptée

Les méthodologies de développement logiciel ne cessent d'évoluer. L'objectif est de s'adapter aux contraintes de l'environnement à la fois internes et externes qui s'imposent au projet. Afin de répondre aux spécifications de notre projet, nous présentons les différentes méthodologies existantes. Nous définissons par la suite la méthodologie de travail finalement adoptée pour le développement de ce projet.

1.5.1 Les méthodologies existantes

La méthode classique se caractérise par la succession de certaines étapes. La première étape consiste à recueillir des besoins, suivie d'une définition du produit. Ensuite vient, le développement et les tests avant la livraison finale [7]. Les méthodes agiles [8] quant à elles se caractérisent par le développement itératif. Il consiste à découper le projet en plusieurs étapes appelées itérations.

Une comparaison des ces deux méthodologies est exposée dans le tableau 1.1.

Nous avons opté pour les méthodes agiles puisque ces dernières offrent une meilleure adaptabilité, visibilité et gestion des risques.

	Méthodologie Classique	Méthodologie Agile
Cycle de vie	Phases séquentielles, sans rétro-action possible.	Itératif et incrémental.
Documentation	Documentation en quantité importante pour tout le projet.	Réduit au strict nécessaire pour chaque incrément.
Changement	Résistance aux changements.	Adaptation aux changements.
Qualité	Contrôle qualité à la fin du cycle de développement.	Contrôle qualité permanent.
Satisfaction client	Risque d'insatisfaction du client car il découvre le produit en intégralité qu'à la fin du projet.	Satisfaction client par la livraison de valeur ajoutée.

Tableau 1.1: Comparaison entre la méthodologie classique et la méthodologie agile.

Les méthodes agiles sont des groupes de pratiques pour la conduite de projet informatiques [9]. Ces méthodes visent à impliquer le client de plus en plus dans le but d'offrir une grande réactivité du projet face à ses demandes et ses exigences. Les différentes méthodes agiles consistent à diviser le projet en un ensemble d'étapes correspondant à des fonctionnalités à développer. Ces systèmes permettent à chaque étape d'avoir un rendu que nous pouvons tester et revoir dans le but de l'améliorer. Les méthodes Agiles disponibles sont nombreuses, parmi lesquelles on peut citer la méthodologie Scrum [10], qui est la méthodologie la plus utilisée dans les grands projets qui nécessitent la collaboration de plusieurs membres de l'équipe. Dans le cadre de notre projet, nous avons opté pour la méthodologie eXtreme Programming. C'est une méthode agile plus récente basée sur le rassemblement de bonnes pratiques déjà connues et utilisées. Cette méthodologie de développement est basée sur les valeurs, les principes et les pratiques de l'agilité : petites livraisons et itérations. Cette méthodologie est souvent utilisée dans des projets développés par des petites équipes (12 personnes maximum).

1.5.2 La méthodologie eXtreme Programming (XP)

La méthodologie eXtreme Programming ou XP est une méthode de gestion de projet qui applique à l'extrême les principes du développement agile [11].

1.5.2.1 Présentation de méthodologie XP

La méthodologie XP est issue d'un projet de la société Chrysler, qui au milieu des années 90, a souhaité remettre totalement à jour son système de paie, en s'attaquant à des fonctionnalités touchant près de 10 000 salariés.

Le principe de cette méthodologie se résume principalement en 3 points à savoir :

- Identifier les facteurs déterminants d'un projet.

- Mettre en place un développement itératif.
- Appliquer l'intégration continue.

La méthode agile XP a été élaborée à l'origine pour le secteur informatique, mais aujourd'hui elle est très populaire car elle fonctionne pour tous types de projets, de toutes tailles et de tous secteurs confondus. Cette méthodologie est idéale pour de petites et moyennes équipes.

1.5.2.2 Principe de la méthodologie XP

La méthode eXtreme Programming s'appuie sur :

- Une forte réactivité au changement des besoins du client;
- Un travail d'équipe;
- La qualité du travail fourni;
- La qualité des tests effectués au plus tôt.

Le déroulement d'un projet en Extreme Programming se base sur des cycles de développement rapides afin d'avoir une évolution itérative du produit final. La première étape de la méthode agile XP consiste à définir les spécifications. En effet, selon la méthodologie XP il faut se concentrer sur les fonctionnalités les plus importantes au lieu de spécifier tout le projet dès le début. Cela demande naturellement de l'expérience et une analyse assez fine du besoin. Ainsi, le projet va être découpé en modules et sous-modules. Les développements vont avoir lieu lors d'itérations pendant des courtes périodes, de deux à cinq semaines maximum. Une livraison est effectuée à l'issue de chaque itération et elle est testée. Si les tests sont concluants et validés, on passe à l'itération suivante. Sinon, l'itération recommence jusqu'à acceptation. Le suivi du projet se fait au fur et à mesure de l'avancement des itérations, ce qui permet de mettre en évidence les problèmes éventuels et de pouvoir prendre en compte de nouvelles demandes du client.

La relecture et l'amélioration du code, ainsi que des tests systématiques tout au long du processus de développement permet de produire un système constamment opérationnel. Toute anomalie est rapidement repérée et corrigée et l'équipe produit un code de qualité supérieure.

1.5.2.3 Pratiques de base de la méthodologie XP

Les cycles de développement dans la méthodologie XP sont rapides. Ce sont des itérations courtes sauf la première itération qui contient les fonctionnalités de base, elle représente le produit minimum viable (Minimum Viable Product, MVP). Les quatre phases principales de chaque cycle sont les suivantes [12] :

- Une phase d'exploration qui détermine les scénarios fournis par le client pendant l'itération.
- Une phase de planning qui transforme les scénarios en des tâches.
- Une phase de développement qui se compose de plusieurs phases de programmation et de test.
- Une phase de livraison vers le client.

Pour bien mener le développement d'un produit, la méthodologie XP est fondée sur douze règles de base :

- Jeu du planning : le planning est réalisé en collaboration avec le client. Ce dernier contribue en créant des scénarios pour les fonctionnalités qu'il souhaite obtenir. Par la suite, l'équipe évalue le temps nécessaire pour les mettre en oeuvre. Ensuite, le client sélectionne ensuite les scénarios en fonction des priorités et du temps disponible.
- Client sur site : le client doit être représenté sur place pendant toute la durée du projet. Ce représentant doit avoir une vision globale du résultat à obtenir et être disponible pour répondre aux questions de l'équipe.
- Intégration continue : une fois une tâche est terminée, elle est intégrée dans le produit complet. Cela permet d'éviter la surcharge de travail due à l'intégration de tous les éléments avant la livraison.
- Petites livraisons : les livraisons doivent être les plus fréquentes possible afin que le client évalue le livrable et donne son avis Ainsi, les modifications sont rapidement prises en compte par l'équipe.
- Rythme soutenable (40 heures par semaine) : Ne pas travailler plus de 40 heures par semaine. L'équipe ne fait pas d'heures supplémentaires. Si le cas se présente, il faut revoir le planning. Un collaborateur fatigué travaille mal et fait plus d'erreurs.
- Les tests : les programmeurs écrivent continuellement des tests unitaires, qui doivent fonctionner parfaitement pour continuer. Des tests fonctionnels doivent être établis pour vérifier les fonctionnalités.
- Fonctionnalité est requise. À chaque modification du code, tous les tests sont lancés afin d'identifier immédiatement s'il y a un problème de fonctionnement.
- Conception simple : le système doit être conçu aussi simplement que possible à un moment donné. Envisager les prochaines évolutions ferait perdre du temps sans avoir la garantie d'un gain ultérieur.
- Utilisation de métaphores : les équipes XP utilisent des métaphores pour décrire le système et son fonctionnement afin de clarifier les fonctionnalités à atteindre : Tout le monde parle le

même langage. Le fonctionnel et le technique se comprennent beaucoup mieux lorsqu'ils sont d'accord sur les termes qu'ils emploient.

- Refactoring (ou remaniement du projet) : Les programmeurs restructurent le système sans changer son comportement pour éliminer les redondances, améliorer la communication, simplifier, ou ajouter de la flexibilité.
- Appropriation collective du projet : la responsabilité du projet est collective. Chaque membre de l'équipe peut modifier toutes les portions du projet, même celles sur lesquelles il n'a pas travaillé. L'objectif est d'être efficace et rapide.
- Standards de langage : Les programmeurs écrivent tout le code en conformité avec les normes de la programmation. Il est indispensable d'établir et de respecter des normes de nommage, e.g. pour les variables, méthodes, objets, classes, fichiers.
- Travail en binôme : La programmation se fait par deux. Le premier appelé driver (ou pilote) tient le clavier. C'est lui qui va travailler sur la portion du code à écrire. Le second appelé partner (ou copilote) est là pour l'aider en suggérant de nouvelles possibilités ou en décelant d'éventuels problèmes.

1.5.3 User Stories du produit

Comme nous utilisons eXtreme Programming comme méthodologie, les besoins de notre client sont exprimés sous formes de User Stories dans la phase exploration de chaque itération. Les besoins de base sont exprimés durant la première itération.

Pour accélérer le travail, XP se diffère de la méthodologie Scrum. Les équipes Scrum ne permettent pas de changements dans leurs sprints. Une fois la réunion de planification de sprint est terminée, l'ensemble des tâches reste inchangé jusqu'à la fin du sprint. Les équipes XP sont beaucoup plus susceptibles de changer durant leurs itérations. Tant que l'équipe n'a pas commencé à travailler sur une fonctionnalité particulière, une nouvelle fonctionnalité détaillée équivalente peut être échangée dans l'itération avec la fonctionnalité non entamée. Comme XP donne la priorité à l'exécution technique, les rôles de conception et de développement sont fusionnés, i.e. les concepteurs sont les développeurs. Extreme Programming estime que la partie la plus importante de la conception est le code. En conséquence, le code source est écrit pour exprimer la conception clairement.

Le tableau 1.2 représente le user stories de notre projet. Pour chaque besoin, nous devons mentionner :

- Le rôle du demandeur ;
- Les niveaux de priorité qui varient entre 1 le plus prioritaire à 4 le moins prioritaire.

ID	User story	Priorité
1	En tant qu'utilisateur non identifié, je dois pouvoir créer un compte adhérent ou gérant. Je dois pouvoir récupérer mon mot de passe en cas d'oubli.	1
2	En tant qu'adhérent ou invité, je peux voir la carte géographique me positionner, localiser les centres et les consulter.	2
3	En tant que adhérent ou invité je peux tracer le plus court chemin en utilisant le GPS temps réel.	4
4	En tant que gérant de centre je peux gérer mon compte.	2
5	En tant que gérant de centre je peux gérer mes sessions.	1
6	En tant qu'adhérent je peux réserver des sessions.	1
7	En tant qu'adhérent je peux gérer mes réservations.	2
8	En tant qu'adhérent ou invité je peux voir les nouveautés, les sessions et les centres les plus populaires.	3
9	En tant qu'adhérent je peux voir les recommandations selon mon historique d'achats.	2
10	En tant que gérant de centre je peux scanner le QR code du client.	1
11	En tant que gérant de centre je peux consulter mes statistiques.	3
12	En tant que gérant de centre je peux consulter la liste des adhérents journalière.	3
13	En tant que administrateur je peux gérer les centres, les utilisateurs et les avis	1
14	En tant que utilisateur je peux utiliser l'application par internet à n'importe quel moment (hébergement serveur)	2

Tableau 1.2: La liste des User Stories du projet

Conclusion

Dans ce chapitre, nous avons brièvement présenté l'organisme d'accueil Wecode Land, les produits et les services qu'elle développe. Nous avons aussi introduit le contexte et la problématique de notre travail ainsi que la méthodologie utilisée. Enfin, nous avons planifié notre projet selon la méthodologie choisie.

ANALYSE ET CONCEPTION

Plan

1	Objectifs de l'application	15
2	Analyse des besoins	17
2.2.1	Identifications des acteurs	17
2.2.2	Besoins fonctionnels	17
2.2.3	Besoins non-fonctionnels	19
2.2.4	Spécification des besoins	20
2.2.4.1	Diagramme de cas d'utilisation général	20
2.2.4.2	Diagrammes de cas d'utilisation raffinés	22
3	Conception détaillée	28
2.3.1	Méthodologie de conception adoptée : UML	28
2.3.2	Logiciel de modélisation utilisé	29
2.3.3	Diagramme de classe	29
2.3.4	Diagrammes de séquences	32
2.3.4.1	Diagramme de séquence « Utiliser GPS »	32
4	Conclusion	37

Introduction

Dans ce chapitre, nous allons d'abord commencer par présenter l'ensemble des fonctionnalités que doit satisfaire l'application à développer, ainsi que les différentes contraintes auxquelles elle doit se soumettre. Nous introduisons aussi l'ensemble des acteurs susceptibles d'interagir avec notre application. Nous consacrons la deuxième partie de ce chapitre à la conception détaillée de notre solution, en présentant en premier lieu la méthodologie de conception adoptée. Ensuite, nous détaillons la conception de notre application en utilisant différents diagrammes.

2.1 Objectifs de l'application

Notre projet consiste à développer une application mobile et puissante pour la réservation des sessions des centres de loisirs destinée aux utilisateurs tunisiens.

Nous avons choisi « *Spory* » comme nom de l'application. Nous avons pensé à ce qu'il soit composé d'un seul mot du dialecte tunisien, simple à retenir et qui résume le sujet du site. Outre le fait d'être accessible par n'importe quel périphérique mobile (Smartphone, tablette....) et à tout moment via une simple connexion internet, « *Spory* » devra regrouper toutes les fonctionnalités nécessaires pour la gestion complète des séances de sport.

L'application devra permettre de publier / consulter des sessions, réserver en avance, effectuer le paiement en ligne, rechercher des centres par nom ou par localisation géographique ainsi qu'évaluer le centre en laissant un commentaire.

L'application regroupera toutes les fonctionnalités techniques telles que : s'inscrire, s'authentifier, se déconnecter, gérer son compte, modifier ses informations personnelles, gérer le compte du gérant, gérer les produits, *etc*. Elle proposera en particulier un module d'administration qui permettra d'administrer l'application et de valider les différents mouvements effectués par les clients/gérants (Inscriptions, commentaires, etc).

Spory offrira aussi un espace de commentaires qui permettra à ses membres de partager leurs avis à propos de leurs expériences.

L'application offrira en particulier trois fonctionnalités clés à savoir :

1. **Un système de géolocalisation** qui permettra de détecter la position de l'utilisateur et de lui afficher à l'ouverture de l'application une carte géographique avec les centres disponibles dans sa zone mais aussi le positionnement de tous les centres disponibles sur la carte. L'affichage pourra être paramétré comme une liste déroulante accessible par un simple clic ou filtre effectué par l'abonné. On mettra à disposition une barre de recherche qui effectuera un filtre automatiquement

selon la recherche effectuée.

Une fois le centre sélectionné, l'application remontera le chemin le plus court avec le détail de la distance à parcourir ainsi qu'un traçage sur le map.

2. **Un système de scan QR Code :** lorsqu'un client s'inscrit à un court (séance), il n'aura pas besoin d'un abonnement physique, il suffira de montrer son GSM le QR Code de sa réservation. En effet, un QR Code est un code-barres à deux dimensions qui permet d'encoder des données, comme le montre la figure 2.1. Il permet d'effectuer des interactions avec les Smartphones.



Figure 2.1: Exemple d'un QR code

D'un point de vue fonctionnel, le QR Code permet d'étendre l'information présente sur tout support physique et de donner une dimension numérique aux campagnes marketing. Cette technologie accélère et incite l'usage des services Internet mobile : c'est un véritable outil interactif. Dans notre application, le système QR Code permettra de stocker toutes les informations relatives à l'abonné ainsi que celle du centre et de l'abonnement, ceci aidera à dématérialiser le format papier et la facture.

Le propriétaire du centre aura juste à installer une simple application afin de lire le QR Code et d'avoir tout ce qui est en rapport avec l'abonnement de l'abonné. Une fois le QR Code scanné, il perdra sa validité puisque l'application générera un QR Code spécifique par séance.

3. **Un Système de recommandation** qui présentera les centres les plus visités et les meilleures ventes via un filtrage au niveau de la base de données d'une part. Il permet aussi de proposer des recommandations *personalisées* via un algorithme développé dans l'application. Ce système permettra de remonter des informations adaptées aux choix et à la tendance de l'abonné, selon des informations pertinentes qui caractérisent chaque centre d'une part (la localisation, la note moyenne, le nombre de votes, la nature des séances, le nombre d'activités), mais aussi selon l'historique de l'adhérent.

Autrement dit, le système de recommandation a pour but de comprendre le comportement (les choix) de chaque utilisateur, afin de lui proposer via une interface les offres les plus intéressantes, et aussi de trouver des offres qu'il est susceptible d'aimer mais auxquels il n'aurait pas forcément

fait attention.

L'application devra aussi présenter des modules d'aide à la décision qui permettront aux différents gérants d'avoir une visibilité globale sur le pourcentage de présences ainsi que des plages horaires les plus fréquentées.

Chaque gérant pourra voir les différents avis des adhérents sans aucune possibilité de suppression, notre application se veut la plus crédible possible.

2.2 Analyse des besoins

Dans cette section, nous identifions en premier lieu les différents acteurs et nous exposons la liste des besoins fonctionnels et non fonctionnels du projet.

2.2.1 Identifications des acteurs

Un acteur est un élément externe qui interagit avec le système étudié en prenant des décisions et en effectuant des tâches. Dans le cadre de notre projet, les acteurs à prendre en considération sont :

- **Un administrateur** : Cet acteur est responsable de la gestion de l'application : suppression, blocage et validations des clients et gérants ainsi que la suppression des commentaires.
- **Un gérant** : Cet acteur s'occupe de tous les processus internes du centre à savoir la gestion du local et la programmation des sessions d'activités sportives qu'il mettra à disposition.
- **Un adhérent (client)** : Cet acteur peut interagir avec l'application en réservant les séances et consulter les profils des clubs. Il peut effectuer la gestion et le paramétrage de son profil et de ses préférences et pourra surtout donner son avis en laissant un commentaire et une note qui seront contrôlés par l'administrateur afin de ne pas pouvoir impacter subjectivement les réputations de chaque centre (un avis par utilisateur, possibilité de suppression des commentaires non respectueux).
- **Un utilisateur mode invité (sans compte)** : Cet acteur aura accès à des fonctionnalités simplifiées permettant d'avoir une visibilité globale de l'application.

2.2.2 Besoins fonctionnels

Le tableau 2.1 liste les différents besoins fonctionnels de chaque acteur.

Acteur	Besoins fonctionnels	Description
Administrateur	<ul style="list-style-type: none"> — Valider les clients; — Valider les centres; — Bloquer les clients; — Bloquer les centres; — Supprimer les clients; — Supprimer les centres; — Contrôler les avis. 	L'administrateur dispose d'un rôle indispensable permettant de veiller au bon fonctionnement de l'application sans favoriser aucun des centres adhérents. Il pourra ainsi gérer les commentaires et les différentes demandes d'adhésion aussi bien des gérants que des clients.
Gérant	<ul style="list-style-type: none"> — Inscription; — Authentification; — Déconnexion; — Modifier profil; — Publier des programmes (séances) sportifs; — Modifier des séances; — Supprimer des séances; — Visualiser de statistiques dynamiques; — Visualiser les activités quotidiennes. 	Le gérant constitue un point clé de l'application puisqu'il permettra d'alimenter la base de données avec ses informations qui seront traitées et présentées par l'application. Le gérant pourra ainsi rentrer les différents programmes (séances) proposées ainsi que les différents détails de facturation lié à son centre. En outre il aura une visibilité synthétisée de différentes statistiques liée à la fréquentation des programmes proposés.

Tableau 2.1: Liste des besoins fonctionnels par acteur

Acteur	Besoins fonctionnels	Description
Client	<ul style="list-style-type: none"> — Incription; — Authentification; — Déconnexion; — Modifier son profil; — Consulter les programmes (séances); — Consulter profil de la salle; — Chercher les centres sur une map ou bien sur une liste déroulante; Utiliser GPS pour tracer le chemin vers un centre; — Visualiser les offres réservées; Modifier les offres réservées; — Evaluer un centre. 	Le client est le centre primordial de l'application puisqu'il permettra d'alimenter en premier lieu le système de recommandation via les commentaires et les retours vis-à-vis des centres sportifs. Il aura accès à une panoplie de fonctionnalités standards qui lui permettront de paramétrier son profil d'un façon complète mais aussi de gérer ses abonnements d'une façon très ergonomique.
Utilisateur mode invité (sans compte)	<ul style="list-style-type: none"> — Visualiser les centres; — Utiliser GPS pour tracer le chemin d'un centre; — Consulter profil d'une salle. 	Un utilisateur en mode invité aura accès à des fonctionnalités simplifiées mais qui restent tout de même très pratiques pour avoir une visibilité globale des différentes salles de sport.

Tableau 2.1: Liste des besoins fonctionnels par acteur

2.2.3 Besoins non-fonctionnels

L'application devra satisfaire des exigences non fonctionnelles de qualité, d'ergonomie, de performance et de sécurité.

- **Exigences d'ergonomie :** Notre application s'adresse à un public jeune et sportif. Elle doit

offrir une interface graphique simple d'utilisation. Les différentes fonctionnalités devront être bien présentées et simple d'accès.

- **Exigences de qualité** : L'application doit offrir des données fiables. L'utilisateur doit avoir accès à une panoplie de données qui ont été au préalable bien synthétisés, contrôlés et validés.
- **Exigences de performance** : L'application doit pouvoir supporter un bon nombre de connexions simultanées. Lors de la saisie des différents paramètres de filtrage, elle devra répondre au besoin rapidement, sans latence et sans bugs d'accès concurrentiels.
- **Exigences de sécurité** : L'application doit garantir la confidentialité des informations personnelles de ses membres ainsi que la sécurisation des moyens de paiement.

2.2.4 Spécification des besoins

Un cas d'utilisation est un ensemble d'actions qui correspondent à une fonctionnalité dont certains acteurs ont besoin. Il s'agit d'une utilisation particulière de l'application. Le diagramme des cas d'utilisation explique donc de quelle manière les acteurs vont pouvoir interagir avec l'application.

2.2.4.1 Diagramme de cas d'utilisation général

Le diagramme de cas d'utilisation général permet d'avoir une vue globale du fonctionnement du système. Il décrit les interactions du système à concevoir avec ses utilisateurs. Ce diagramme est illustré dans la figure 2.2 (page 21).

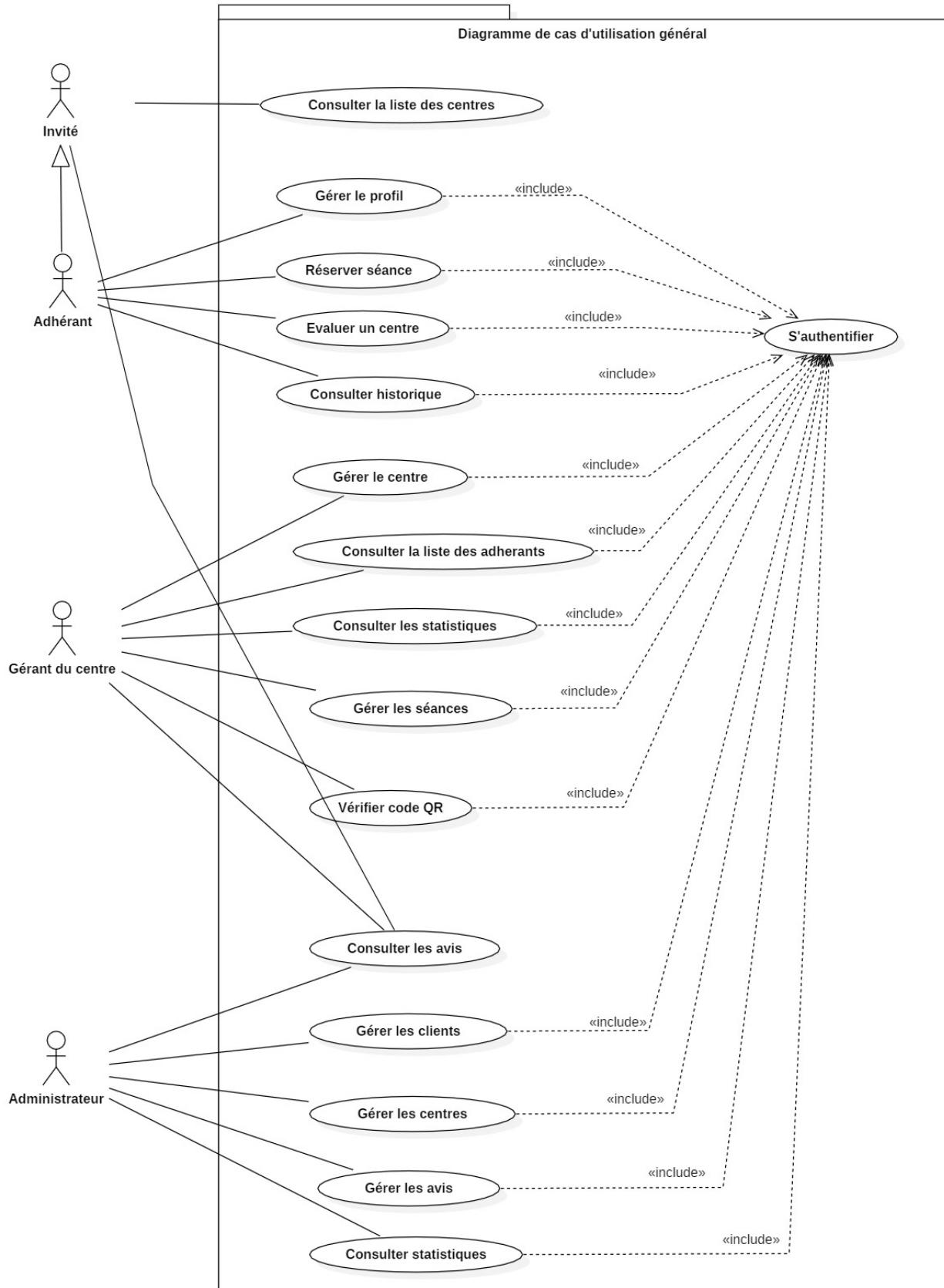


Figure 2.2: Diagramme de cas d'utilisation général

2.2.4.2 Diagrammes de cas d'utilisation raffinés

Dans cette partie nous allons détailler les cas d'utilisation «Consultation des centres», «Consultation des centre», «Réservation des séances», «Vérification facture» et «Ajout séance» de notre système dans le but de décrire le fonctionnement de chaque cas.

Cas d'utilisation «Consultation des centres»

La figure 2.3 (page 22) illustre les détails du cas d'utilisation «Consultation des centres». La description textuelle de ce diagramme est donnée par le tableau 2.2 (page 23).

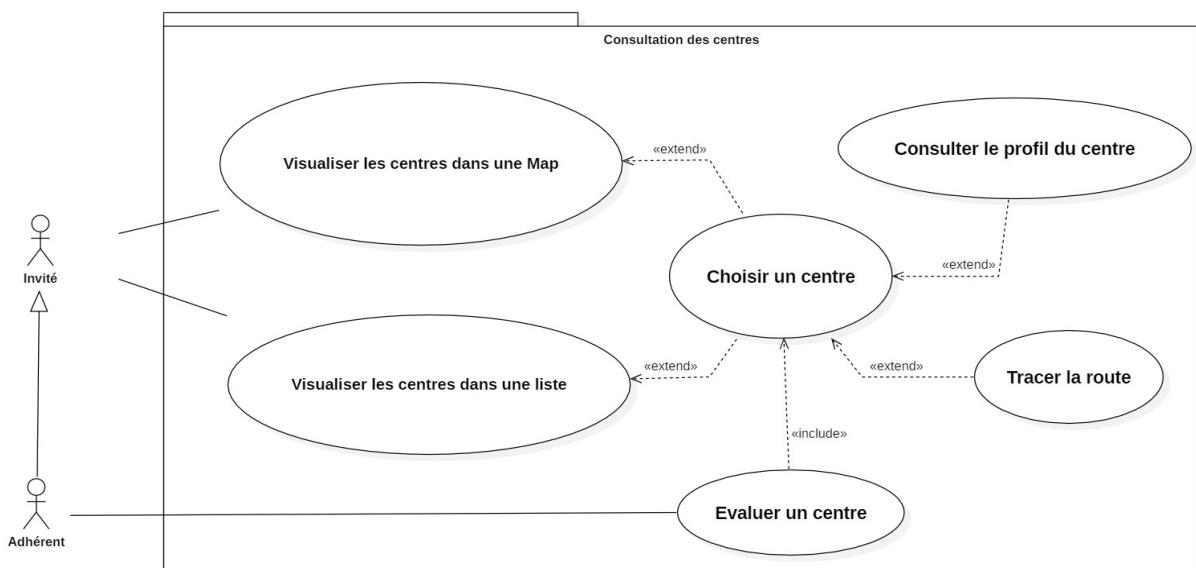


Figure 2.3: Diagramme de cas d'utilisation «Consultation des centres»

Cas d'utilisation	Consultation des centres
Acteurs	Invité / Adhérent
Pré-condition	Certaines tâches «Evaluer un centre» exigent l'authentification.
Post-condition	Une session sera ouverte pour cet utilisateur.
Scénario nominale	<ol style="list-style-type: none"> 1. L'utilisateur entre son login et mot de passe. 2. Le système charge les données des centres. 3. L'utilisateur a le choix de visualiser les centres sur une carte géographique (map) ou bien sur une liste. 4. L'utilisateur choisit un centre. 5. L'utilisateur peut tracer la route ou bien consulter le profil du centre et donner son avis.
Scénario alternatif	<ol style="list-style-type: none"> 1. L'utilisateur passe comme un invité. 2. Le système charge les données des centres. 3. L'utilisateur a le choix de visualiser les centres sur une carte géographique ou bien sur une liste. 4. L'utilisateur choisit un centre. 5. L'utilisateur peut tracer la route ou bien consulter le profil du centre.

Tableau 2.2: Description textuelle du cas d'utilisation «Consultation des centres»

Cas d'utilisation «Réservation des séances»

La figure 2.4 (page 23) schématise le diagramme du cas d'utilisation «Réservation des séances».

Pour pouvoir effectuer cette tâche, il faut être un adhérent. Le tableau 2.3 (page 24) détaille les fonctionnalités de ce diagramme.

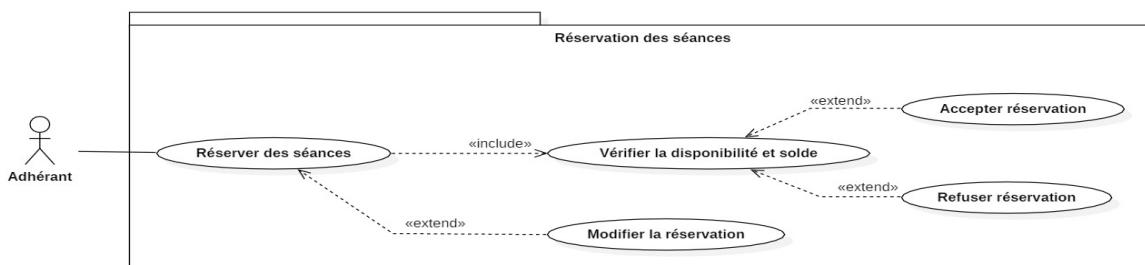


Figure 2.4: Diagramme de cas d'utilisation « Réservation des séances »

Cas d'utilisation	Réservation des séances
Acteurs	Adhérent
Pré-condition	S'authentifier
Post-condition	Une session sera ouverte pour l'adhérent.
Scenario nominal	<ol style="list-style-type: none"> 1. Un utilisateur saisit ses identifiants (login et mot de passe). 2. Le système charge les données des sessions du centre choisi. 3. Le système affiche les séances. 4. L'utilisateur choisit le type, la date et les séances. 5. Le système vérifie la disponibilité et le crédit de l'adhérent au fur et à mesure. 6. L'utilisateur envoie sa demande. 7. Le système met à jour la base des données.
Scenario alternatif	<ol style="list-style-type: none"> 1. Un utilisateur saisit son login et son mot de passe. 2. Le système charge les données des sessions du centre choisi. 3. Le système affiche les séances. 4. L'utilisateur choisit le type, la date et les séances. 5. Le système vérifie la disponibilité et le crédit de l'adhérent au fur et à mesure. 6. L'utilisateur envoie sa demande. 7. Le système retourne un message d'erreur indiquant que la séance choisie est déjà prise ou le solde est insuffisant.

Tableau 2.3: Description du diagramme de cas d'utilisation « Réservations des séances»

Cas d'utilisation «Ajout séance»

Le diagramme de cas d'utilisation «Ajout séance» est illustré dans la figure 2.5 (page 25) et développé dans le tableau 2.4 (page 26).

Pour pouvoir effectuer ces tâches, il faut être un gérant.

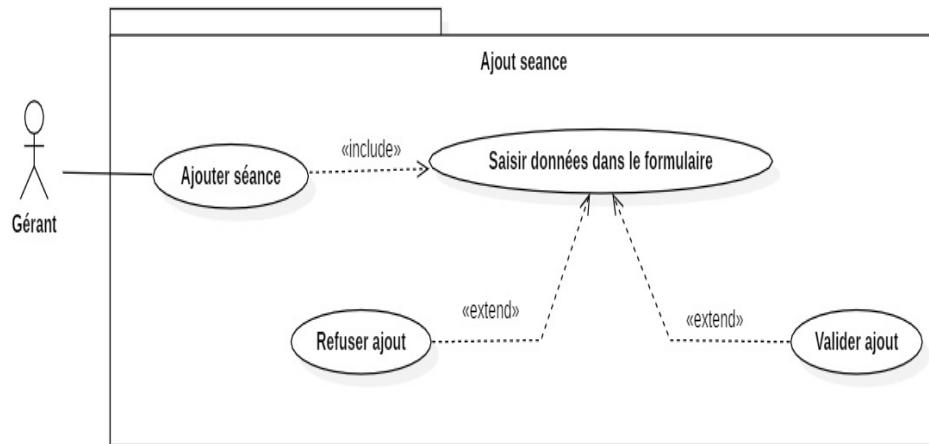


Figure 2.5: Diagramme de cas d'utilisation « Ajout séance»

Cas d'utilisation	Ajout séance
Acteurs	Gérant du centre
Pré-condition	S'authentifier
Post-condition	Une séance sera ouverte pour un gérant
Scenario nominal	<ol style="list-style-type: none"> 1. Le gérant saisit ses identifiants (login et mot de passe). 2. Le système charge les données du centre. 3. Le gérant spécifie les propriétés de la séance : <ul style="list-style-type: none"> — Le type et la description de la séance. — La date de début et de fin de la séance (durée). — Les jours de la semaine disponibles ainsi que l'horaire de chaque jour. — Possibilité d'être suivi par un coach personnel. — Le nombre maximal d'adhérents. — Le prix. 4. Le gérant confirme l'ajout. 5. Le système valide et met à jour la base des données.
Scenario alternatif	<ol style="list-style-type: none"> 1. Le gérant entre son login et son mot de passe. 2. Le système charge les données du centre. 3. Le gérant spécifie les propriétés de la séance : <ul style="list-style-type: none"> — Le type et la description de la séance. — La date de début et de fin de la séance (durée). — Les jours de la semaine disponibles ainsi que l'horaire de chaque jour. — Possibilité d'être suivi par un coach personnel. — Le nombre maximal d'adhérents. — Le prix. 4. Le gérant confirme l'ajout. 5. Le système trouve une duplication de la séance et retourne un message d'erreur.

Tableau 2.4: Description textuelle du diagramme de cas d'utilisation «Ajout séance»

Cas d'utilisation «Vérification facture»

Le diagramme de cas d'utilisation «Vérification facture» est illustré par la figure 2.6 (page 27) et décrit par le tableau 2.5 (page 28). Pour pouvoir effectuer ces tâches, il faut être un gérant.

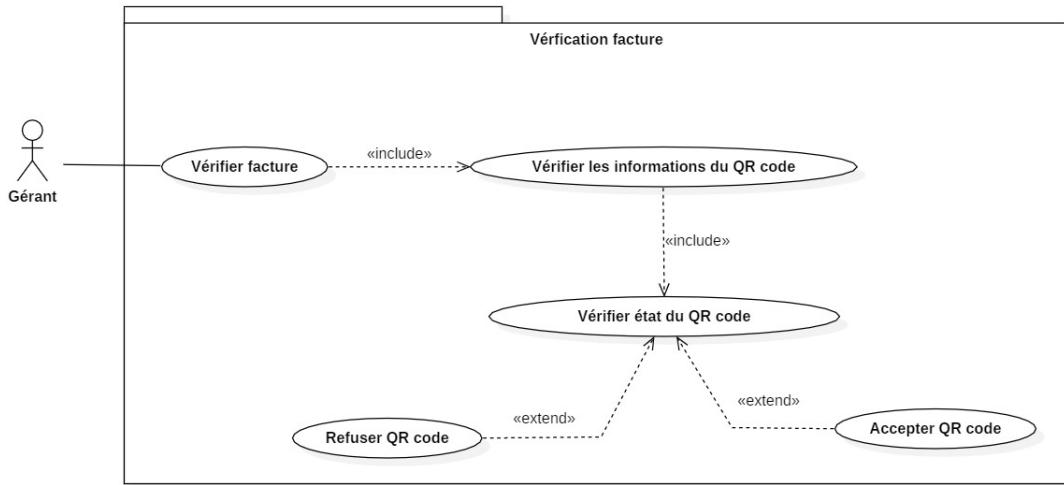


Figure 2.6: Diagramme de cas d'utilisation «Vérification facture»

Cas d'utilisation	Vérification d'une facture
Acteurs	Gérant
Pré-condition	S'authentifier
Post-condition	Une session est ouverte pour le gérant
Scenario nominal	<ol style="list-style-type: none"> 1. Le gérant saisit son login et son mot de passe. 2. Le gérant scanne le QR code du client. 3. Le système vérifie les données : date / id du centre / token de la facture et la validation de la facture. 4. Le système valide la facture. 5. Le système met à jour la base de données.
Scenario alternatif	<ol style="list-style-type: none"> 1. Le gérant entre son login et mot de passe. 2. Le gérant scanne le QR code du client. 3. Le système vérifie les données : date / id du centre / token de la facture et la validation de la facture. 4. Le système repère une erreur au niveau des données (date / id centre) ou bien détecte que la facture est déjà scannée. 5. Le système indique un message spécifique à l'erreur.

Tableau 2.5: Description textuelle de diagramme de cas d'utilisation « Vérification d'une facture»

2.3 Conception détaillée

Dans cette partie nous présentons la méthodologie de conception adoptée, ainsi que la conception détaillée de notre application en exposant le diagramme de classe et les diagrammes de séquence.

2.3.1 Méthodologie de conception adoptée : UML

La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. En effet, pour que la conception soit efficace, elle doit suivre une méthodologie rationnelle et lisible par les différents acteurs qui interviennent tout au long du cycle de vie d'une application. Dans notre cas, la méthode UML nous a apparu la plus adéquate.

UML est un langage visuel, sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions [13]. Le principal avantage d'UML c'est qu'il est devenu le standard en terme de modélisation objet, universellement reconnu celui-ci est polyvalent et performant. Le méta modèle d'UML en fait un langage formel possédant les caractéristiques

suivantes [14] :

- Un langage universel pouvant servir de support pour tout langage orienté objet.
- Un moyen de définir la structure d'un programme.
- Une représentation visuelle permettant la communication entre les acteurs d'un même projet.
- Il permet de représenter l'aspect traitement du système aussi bien que l'aspect données.
- Une notation graphique simple, compréhensible même par des non informaticiens.
- Il représente un standard de modélisation, une référence.

2.3.2 Logiciel de modélisation utilisé

Pour modéliser les diagrammes UML de la conception de notre application, nous avons choisi StarUML [15]. StarUML est un logiciel de modélisation UML, open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux comme Rational Rose ou Together. Étant simple d'utilisation, nécessitant peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une modélisation de notre application.

2.3.3 Diagramme de classe

Le diagramme de classe représente les classes nécessaires pour assurer le bon fonctionnement du système à mettre en œuvre.

Les classes de notre système sont illustrées dans la figure 2.7 (page 31) et qui sont les suivantes :

- **Membre** : le membre peut être un administrateur de l'application, un gérant de centre ou bien un utilisateur, nous avons donc trois sous classes qui héritent de la classe membre à savoir :
 - **Un Administrateur** : utilisateur qui devra administrer l'application.
 - **Un Gérant** : cette classe est en relation directe avec la classe **Session**, puisque c'est lui qui doit en créer.
 - **Un Adhérent** : est aussi en relation avec les sessions puisqu'il devra en réserver.
- **Centre** : Un centre appartient à un seul gérant. Un gérant peut posséder plusieurs centres. Un utilisateur peut interagir avec plusieurs centres en donnant son avis.
- **Avis** : Cette classe listera les différents avis remontés par les adhérents vis-à-vis de chaque Centre.
- **Session** : Tout centre contient un ensemble de sessions (séances). Chaque session appartient à un seul centre. Un utilisateur peut réserver une séance qui devra être validée par le gérant.

- **Réservation** : cette classe est une classe d’association entre l’adhérent, le gérant et la session.
- **Statistiques** : Cette classe représente une table calculée, elle comportera des mesures relatives à des indicateurs clés en relation avec chaque Centre. Les indicateurs calculés sont :
 - Montant des réservations par centre par mois.
 - Total du montant des réservations par centre.
 - Le nombre de fréquentations par plage horaire par mois.
 - Le total du nombre de fréquentations par plage horaire.
 - Les plage horaire calculées sont les suivantes : [4h-10h], [10H-16H] et [>16H].
 - Le nombre de fréquentations par genre par mois.
 - Total du nombre de fréquentations par genre.
 - Nombre de réservation (ou taux de fréquentation) par plage d’âge par mois.
 - Total du nombre de réservations par plage d’âge.
 - Les plages d’âge définies sont les suivantes :[18 - 35], [35 - 45] et [>45].

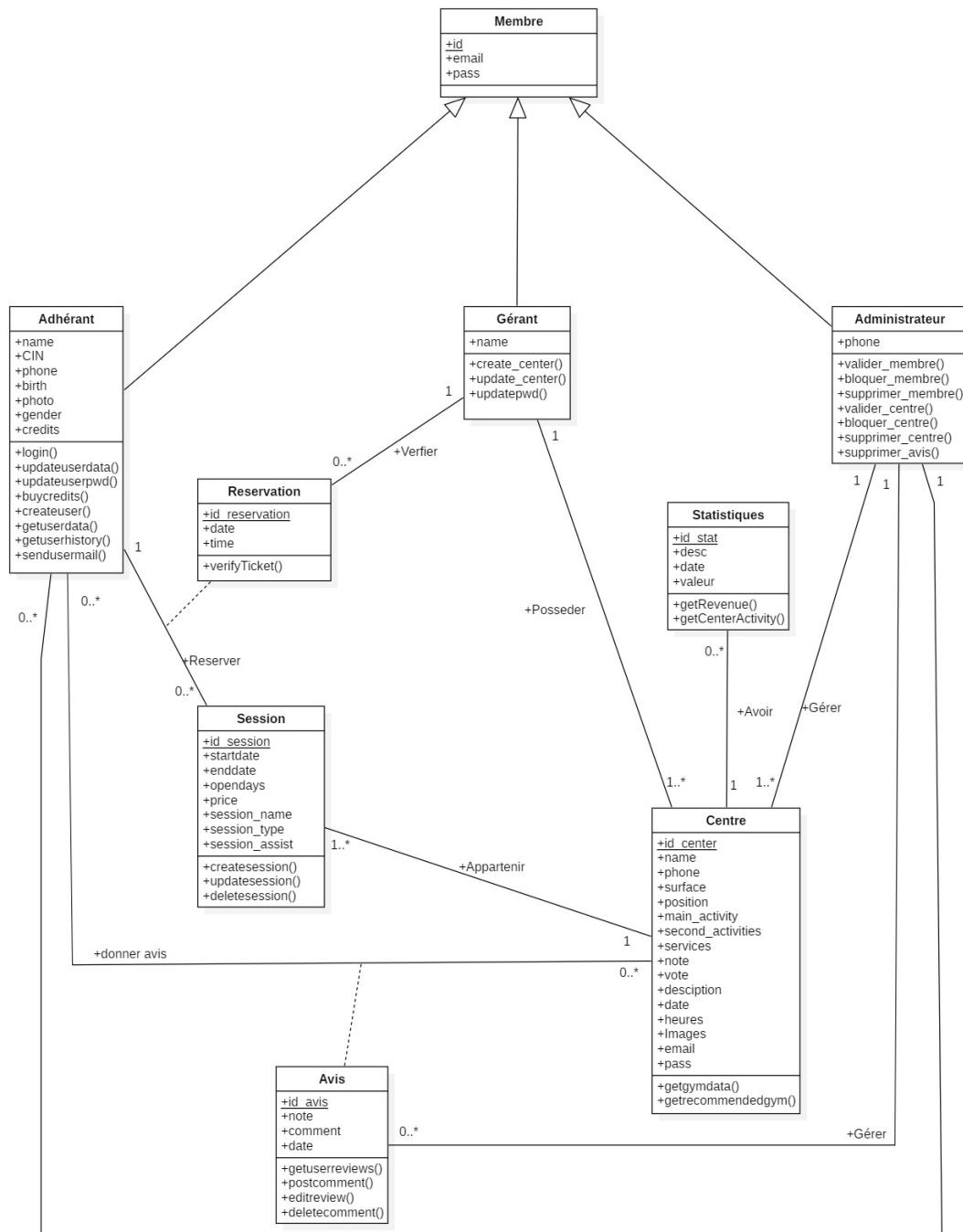


Figure 2.7: Diagramme de classe

2.3.4 Diagrammes de séquences

Un diagramme de séquence permet de décrire un scénario d’interactions ordonnées dans le temps entre les acteurs et le système ou entre les objets internes du système et ses utilisateurs.

A travers ce qui suit, nous allons présenter les diagrammes de séquence système correspondants aux cas d'utilisation déjà détaillés dans la partie précédente.

Dans cette partie nous allons détailler les diagrammes de séquences « Utiliser GPS », « Réserver des séances », « Valider QR code » de notre système dans le but de décrire le fonctionnement de chaque cas.

2.3.4.1 Diagramme de séquence « Utiliser GPS »

Le diagramme « Utiliser GPS » illustre les détails des différentes séquences effectuées lors de l’utilisation du GPS. Le processus de ces séquences est schématisé dans la figure 2.8 (page 32).

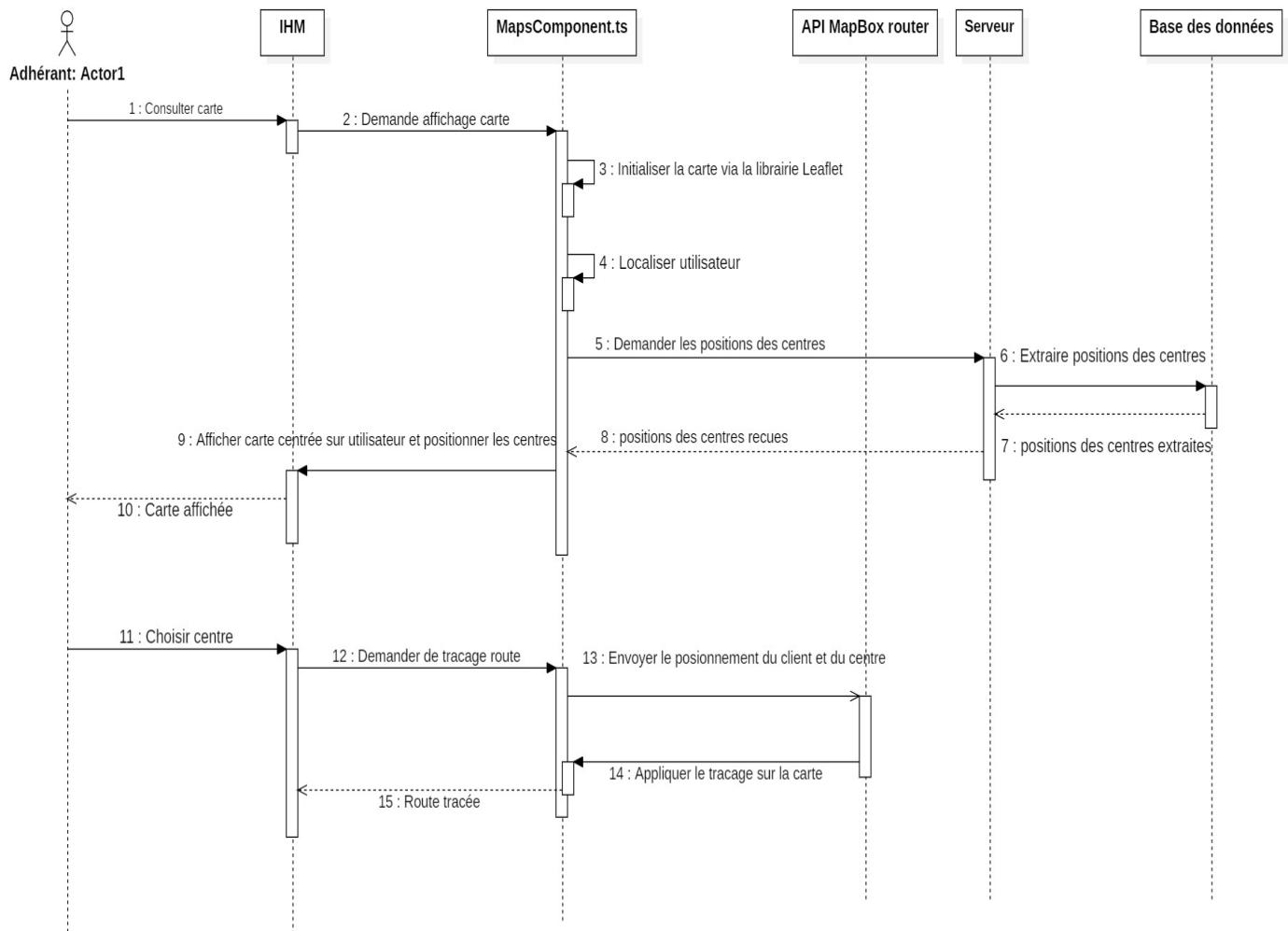


Figure 2.8: Diagramme de séquence « Utiliser GPS »

Description

- Le composant componentMaps.ts :
 - Initialise la carte géographique, en utilisant une librairie appelée Leaflet, sur l'Interface Homme Machine (IHM).
 - Localise le client et le positionne sur la carte.
 - Reçoit les positions des centres et les place sur la carte ou bien sur la liste.
- Le client :
 - Choisit un centre en cliquant sur la carte.
 - Interagit avec l'IHM.
- Le composant componentMaps.ts :
 - Utilise l'API MapBox Router en envoyant les positions du client et du centre choisi.
 - Le composant applique le résultat obtenu sur la carte.

Diagramme de séquence « Réserver des séances »

Le diagramme de séquence, illustré dans la figure 2.9 (page 34) les différentes séquences relatives à une réservation de séance.

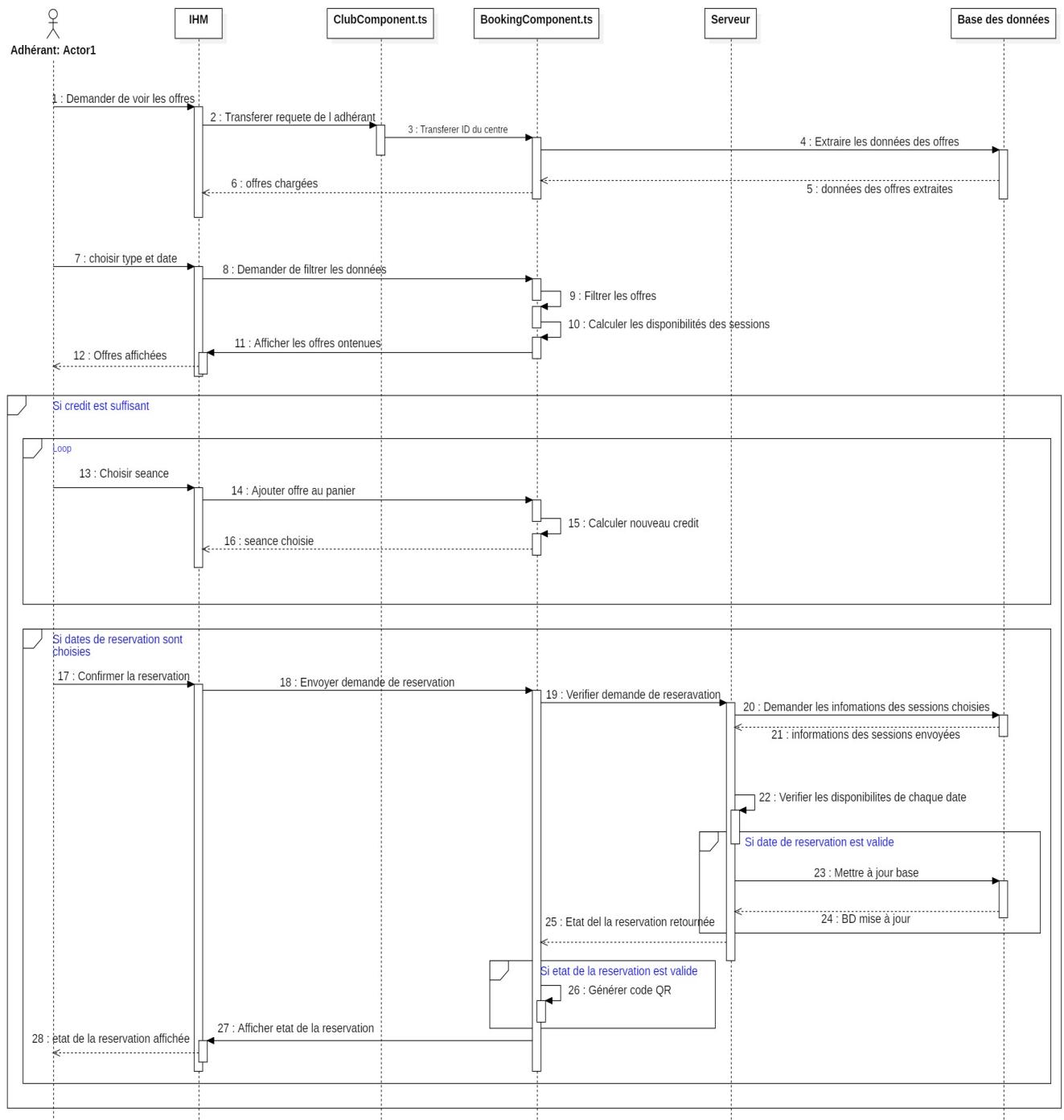


Figure 2.9: Diagramme de séquence «Réserver des séances»

Description

- Le composant componentMaps.ts :
 - Initialise la carte en utilisant la librairie Leaflet sur l'IHM.
 - Localise le client et le positionne sur la carte.
 - Reçoit les positions des centres et les placer sur la carte ou bien sur la liste.
- Le client :
 - Visualise la liste des centres, qui sont affichés soit sur une map ou bien sur une liste déroulante.
 - Choisit un centre.
- Le composant componentMaps.ts :
 - Envoie l'identifiant du centre au composant componentClub.ts.
 - demande de recevoir les informations générales du centre d'après le serveur en envoyant l'identifiant.
 - Affiche le profil du centre.
- Le client demande de consulter les offres.
- Le composant componentClub.ts :
 - Envoie l'identifiant du centre au composant componentBooking.ts
- Le composant componentBooking.ts :
 - Demande de recevoir les informations des offres envoyant l'identifiant au serveur.
- Le client choisit le type et la date.
- Le composant componentBooking.ts :
 - Filtre les données selon type et date et calcule les places vacantes des séances présentes.
 - Si le crédit du client est valide, l'utilisateur peut sélectionner les séances.
- Le composant componentBooking.ts :
 - Calcule le montant restant au fur et à mesure
- Le client envoie sa demande de réservation.
- Le composant componentBooking.ts :
 - Envoie au serveur les dates et les séances choisies.

- Le serveur vérifie une autre fois le montant s'il est suffisant, afin de contourner toute possibilité d'arnaque.
- Le serveur vérifie les places disponibles des séances pendant les dates choisies
- En cas d'erreur le serveur envoie un message d'erreur.
- Sinon il met à jour la base des données.
- Le composant componentBooking.ts :
 - Affiche un message d'indication.

Diagramme de séquence «Valider QR code»

Le diagramme «Valider QR code», donné par la figure 2.10 (page 36), illustre les séquences relatives à la validation du QR code.

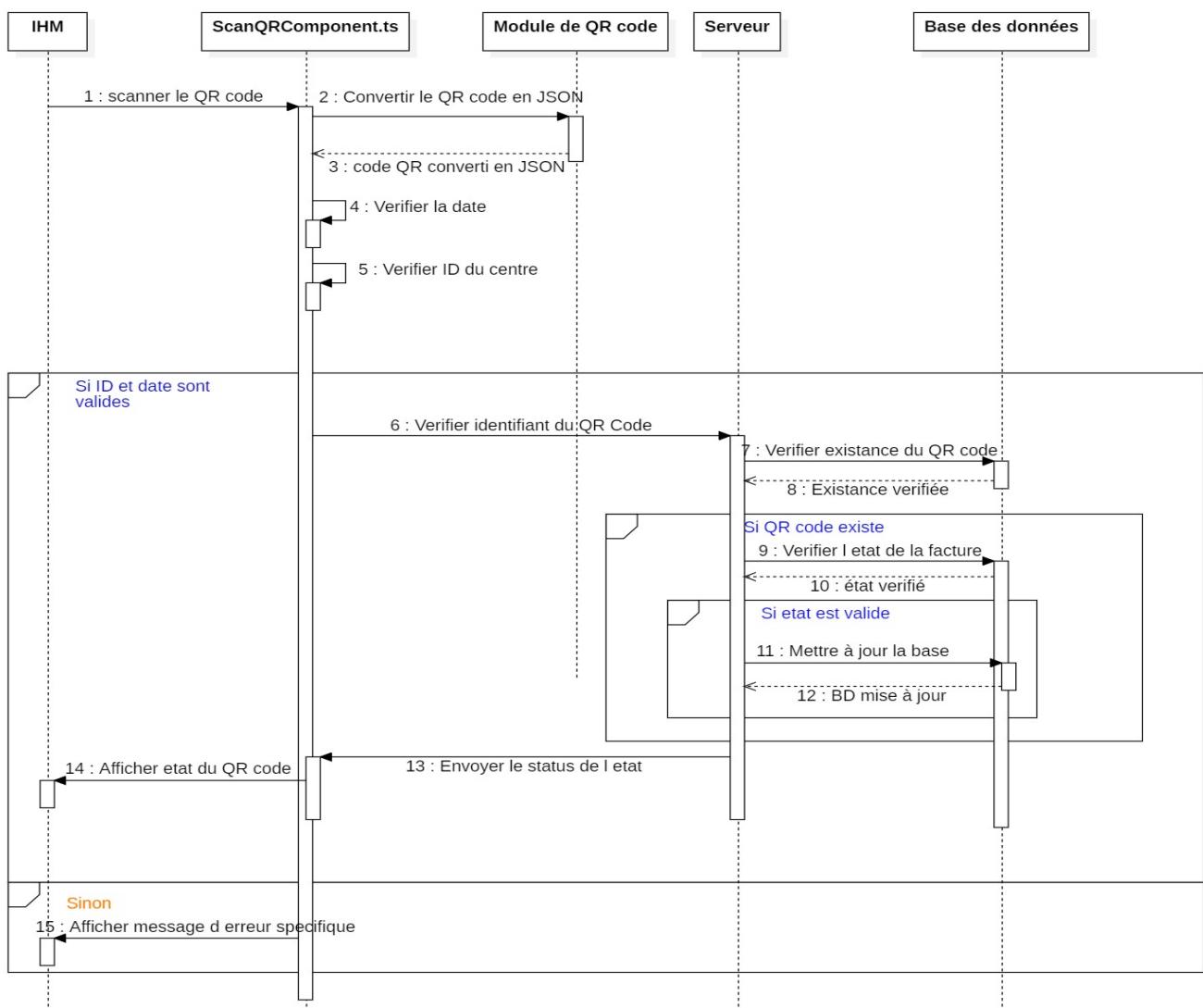


Figure 2.10: Diagramme de séquence «Valider QR code»

Description

- Le gérant appuie sur le bouton « Scanner QR code »
- Le composant componentScan.ts :
 - Convertit le QR code du client en JSON.
 - Extrait les informations du résultat obtenu.
 - Vérifie la validité de la date extraite
 - Vérifie l'identifiant du centre extrait et celui de son club
 - En cas d'erreur, Le composant componentScan.ts retourne un message d'erreur à l'IHM.
 - Sinon, Le composant componentScan.ts envoie au serveur le token de la facture pour vérifier son existence et sa validation (si elle est scannée auparavant ou non).
 - En cas d'erreur, Le composant componentScan.ts retourne un message d'erreur à l'IHM
- Le serveur met à jour la base des données et retourne un message d'indication si l'opération ne retourne pas d'erreurs.

2.4 Conclusion

Dans ce chapitre, nous avons abordé les besoins fonctionnels ainsi que les besoins non fonctionnels de notre projet. Nous avons aussi présenté le diagramme de cas d'utilisation global ainsi que certains diagrammes de cas d'utilisation détaillés, enfin nous avons détaillé quelques diagrammes de séquence ainsi que le diagramme de classes.

CONCEPTION ET SPÉCIFICATION D'UN SYSTÈME DE RECOMMANDATION

Plan

1	Les systèmes de recommandations	39
3.1.1	Principe des systèmes de recommandations	39
3.1.2	Filtrage de données : Comparer un système de recommandation à un moteur de recherche	40
3.1.3	Filtrage de données : Comparer un système de recommandation aux requêtes de bases de données	41
2	Les différents types des systèmes de recommandations	41
3.2.1	Filtrage collaboratif (Collaborative Filtering)	42
3.2.2	Les approches basées sur le contenu (content based)	43
3.2.3	Etude comparative entre les différents types de systèmes de recommandations	44
3	Mesure de similarité	47
4	Évaluation d'un système de recommandation	47
5	Proposition d'un système de recommandation à intégrer dans l'application Spory .	48
3.5.1	Principe	48
3.5.2	Calcul de similarité : Similarité cosinus	49
3.5.3	Démarche	49
3.5.3.1	Phase 1 : Préparation du catalogue	49
3.5.3.2	Phase 2 : Construction de la matrice de similarité	51
3.5.3.3	Phase 3 : Filtrage des centres à recommander à chaque utilisateur . .	52

Introduction

Dans ce chapitre nous allons présenter notre modèle de système de recommandation qui sera intégré dans l'application développée. Notre système a pour objectif de comprendre le comportement de l'utilisateur et de lui proposer des recommandations et des suggestions des centres qui peuvent l'intéresser et auxquels il n'aurait pas spontanément prêtés attention.

Nous allons d'abord commencer par introduire le principe des systèmes de recommandations. Nous présenterons, par la suite, les principaux types des systèmes existants et nous proposerons une étude comparative de ces systèmes. Enfin, nous exposerons le modèle de système de recommandation développé, nous détaillerons les différentes phases de notre modèle et nous présenterons un exemple explicatif pour mieux illustrer les étapes.

3.1 Les systèmes de recommandations

3.1.1 Principe des systèmes de recommandations

Nous sommes fréquemment confrontés à faire des choix de type : *Quel article acheter ?, Quel film regarder ?, Dans quel restaurant diner ?, etc.* La liste des possibilités qui s'offrent à nous est en général de très grande taille, l'évaluation de ces possibilités pour trouver ce qui nous convient le plus est une tâche difficile et peut consommer beaucoup de notre temps. Les systèmes de recommandation sont apparus dans le début des années 1990 pour répondre à ce problème de surcharge d'information et de choix [16].

Les systèmes de recommandation peuvent être vus comme une réponse donnée aux utilisateurs ayant des difficultés à prendre une décision dans le cadre d'utilisation d'un système de recherche d'information "classique".

Les systèmes de recommandation sont essentiels dans certaines industries, ils peuvent générer une énorme quantité de revenus lorsqu'ils sont efficaces et constituer aussi un moyen de se démarquer de manière significative des concurrents. Citons par exemple Netflix qui est une success story basée sur des algorithmes de recommandation. En effet, Netflix, la petite entreprise de location de DVD qui est devenue le leader de la VOD dans le monde. Netflix compte aujourd'hui plus de 167 millions d'abonnés et continue de dépasser année après année son record de recrutement de nouveaux abonnés. Pour avoir un maximum d'abonnés le spécialiste de la VOD doit à la fois trouver de nouveaux clients, mais aussi, et surtout garder ceux qui ont déjà souscrit à une offre. Avoir un taux de désabonnement faible constitue donc un des principaux objectifs de Netflix. C'est là qu'interviennent les algorithmes

de recommandations qui ont fait le succès planétaire de Netflix.

Les systèmes de recommandations deviennent indispensables dans de nombreux autres domaines, notamment dans les domaines des industries culturelles. On peut citer :

- Le e-commerce (Amazon.com),
- Le tourisme (Tripadvisor.com, Airbnb),
- La vidéo à la demande (Youtube.com).
- La musique (spotify)

Ces domaines d'application sont différents mais partagent tous un même problème : ils offrent un choix de possibilités trop important aux utilisateurs, chacun d'eux ayant par ailleurs ses propres préférences. Ainsi, un moteur de recommandation qui permet de proposer à un utilisateur donné, de manière personnalisée, le sous-ensemble d'éléments qui l'intéresse, devient très utile, cela permettra de réduire de manière considérable l'effort que doit fournir l'utilisateur pour accéder à ce qui l'intéresse et participe ainsi à sa satisfaction et à sa fidélisation.

Un système de recommandation doit donc avoir accès à un historique des données qui peut être sous plusieurs formes : des notes, des achats, des clics sur des pages web, des historiques de navigation, etc. A partir de ces informations, le système de recommandation sera en mesure d'adapter la réponse à l'utilisateur. Ce type de système est considéré donc comme un outil de *filtrage* de l'information offrant à un usager l'assistance à la sélection personnalisée. Pour faire simple, un système de recommandation évalue le niveau d'intérêt d'un utilisateur pour certains produits, en fonction de son comportement antérieur à un catalogue d'éléments. L'utilité d'un élément est généralement représentée par un score qui indique comment un utilisateur lambda a aimé un élément en particulier.

3.1.2 Filtrage de données : Comparer un système de recommandation à un moteur de recherche

Le besoin des systèmes de recommandations est semblable à celui des moteurs de recherche mais diffère dans sa conception. En effet, un moteur de recherche reçoit une requête de la part de l'utilisateur et fournit comme résultat une liste ordonnée d'éléments (pages web, images, vidéos, *etc*). Cela a pour but de permettre à l'utilisateur d'accéder rapidement à un contenu considéré comme pertinent par le système par rapport à sa recherche parmi le très grand nombre d'informations disponibles sur Internet.

A l'inverse, un système de recommandation ne reçoit pas de requête directe de la part de l'utilisateur, mais doit lui fournir des ressources pertinentes en fonction de ses préférences. Cela permettra à

l'utilisateur de réduire son temps de recherche mais de recevoir également des suggestions de la part du système auxquelles il n'aurait pas spontanément prêtées attention.

3.1.3 Filtrage de données : Comparer un système de recommandation aux requêtes de bases de données

Le système de recommandation, tout comme les requêtes en base de données, accèdent à de très nombreux éléments ou données pour effectuer un filtrage. Ces deux outils permettent à l'utilisateur de trouver l'élément ou l'information d'un élément parmi une multitude de données. L'avantage d'un système de recommandation est d'automatiser l'analyse et la sélection des éléments les plus pertinents pour l'utilisateur.

Le système de recommandation identifie d'abord les besoins, ou plutôt les préférences, de l'utilisateur, puis filtre les éléments selon différentes approches. Tandis que, lors d'une requête en base de données, l'utilisateur doit spécifier et identifier ses besoins et renseigner les attributs qui l'intéressent dans le système. Le système se contente ainsi d'afficher les éléments correspondants aux attributs entrés par l'utilisateur.

Le tableau 3.1 récapitule la différence entre un filtrage d'un système de recommandation et un filtrage d'un moteur de recherche ou d'une requête sur une BD.

	Système de recommandation	Requêtes en BD / Moteur de recherche
Accès	Accès à de nombreux items	Accès à de nombreux items
Processus	Analyse et Sélection des items les plus pertinents pour l'utilisateur par le système	<ol style="list-style-type: none">1. L'utilisateur identifie les besoins et écrit les requêtes.2. Le système affiche les éléments.

Tableau 3.1: Comparaison entre système de recommandation, moteur de recherche et requêtes bases de données.

Dans ce qui suit, nous allons présenter les différents types de systèmes de recommandations ainsi que leurs fonctionnements.

3.2 Les différents types des systèmes de recommandations

Il existe plusieurs approches des systèmes de recommandations qui visent à faire le lien entre les utilisateurs et les éléments (les objets ou items) à recommander [17]. Nous nous intéressons particulièrement aux deux principaux types présents dans la littérature, à savoir les systèmes de

recommandations basés sur un « **Filtrage Collaboratif** » et les systèmes de recommandations basés sur le « **Contenu** ».

3.2.1 Filtrage collaboratif (Collaborative Filtering)

Les méthodes collaboratives sont des méthodes qui sont basées uniquement sur l'historique des interactions enregistrées entre les utilisateurs et les éléments afin de produire de nouvelles recommandations [18]. Ce type de méthodes procède en calculant la similarité entre les préférences d'un utilisateur et celles des autres. Les interactions utilisateurs-items sont stockées dans une structure de données appelée «matrice d'interactions utilisateur-élément», illustrée dans la figure 3.1.

		M items			
		Image 1	Image 2	Image 3	Image 4
N users	User 1	5		1	
	User 2			3	
	User 3		5		
	User 4			2	
	User 5				5

Figure 3.1: Matrice de correspondance utilisateur-élément.

L'idée principale qui régit les méthodes collaboratives est que les interactions «utilisateur-élément» passées sont suffisantes pour détecter des utilisateurs et des éléments similaires et générer ainsi des prédictions basées sur ces valeurs estimées. De tels systèmes ne tentent pas d'analyser ou de comprendre le contenu des éléments à recommander. La méthode consiste à faire des prévisions automatiques ciblant les intérêts d'un utilisateur en collectant un nombre important d'avis. D'une manière générale, l'hypothèse sous-jacente de cette approche, illustrée par la figure 3.2 (page 43), est que les utilisateurs qui ont aimé un élément spécifique dans le passé auront tendance à aimer cet élément spécifique, ou un autre très « proche », à nouveau dans l'avenir.

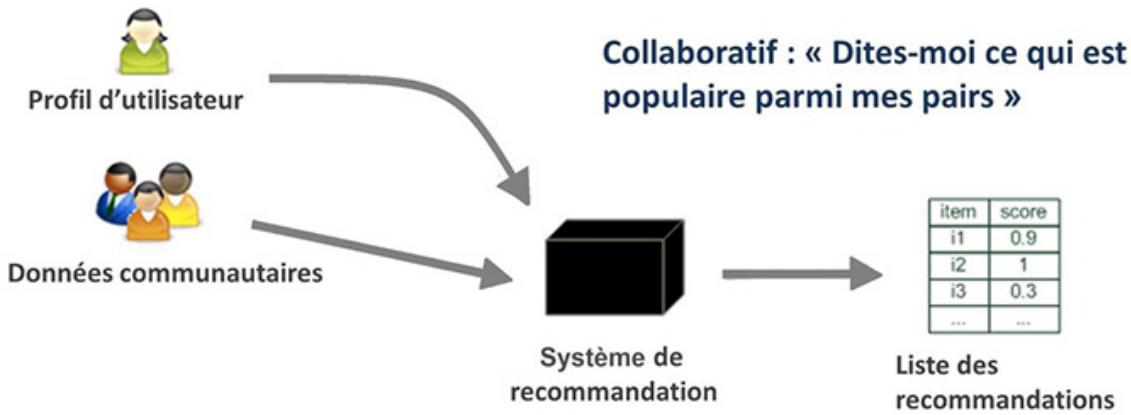


Figure 3.2: Schéma d'un système de recommandation collaboratif.

L'idée des approches collaboratives est d'essayer de prédire le choix d'un utilisateur vis à vis les différents éléments. La recommandation est basée sur les choix précédents de l'utilisateur, mais également sur une mesure de similarité avec d'autres utilisateurs.

Ce type de système de recommandation procède selon les principales étapes suivantes :

1. Enregistrer de nombreuses préférences d'utilisateurs.
2. Repérer un sous-groupe d'utilisateurs dont les préférences sont similaires à celles de l'utilisateur qui cherche la recommandation.
3. Calculer une moyenne des préférences pour ce groupe.
4. Utiliser une fonction de préférence qui en résulte pour recommander des éléments à l'utilisateur.

3.2.2 Les approches basées sur le contenu (content based)

Contrairement aux méthodes collaboratives qui ne dépendent que des interactions utilisateur-élément, les approches basées sur le contenu utilisent des informations supplémentaires sur les utilisateurs ou sur les éléments [18]. Pour un système de recommandation de films, par exemple, des informations supplémentaires peuvent être de type : âge, sexe, travail ou toute autre information personnelle relative aux utilisateurs ainsi que la catégorie du film, les principaux acteurs, la durée ou d'autres caractéristiques des films.

Pour ce type de système de recommandation, la tâche consiste à déterminer quels éléments du catalogue coïncident le mieux avec les préférences de l'utilisateur. Ce processus est illustré par la figure 3.3 (page 44).

Dans les méthodes basées sur le contenu, le problème de recommandation revient à un problème de classification, qui consiste à prédire si un utilisateur « aime » ou non un élément, ou en un problème

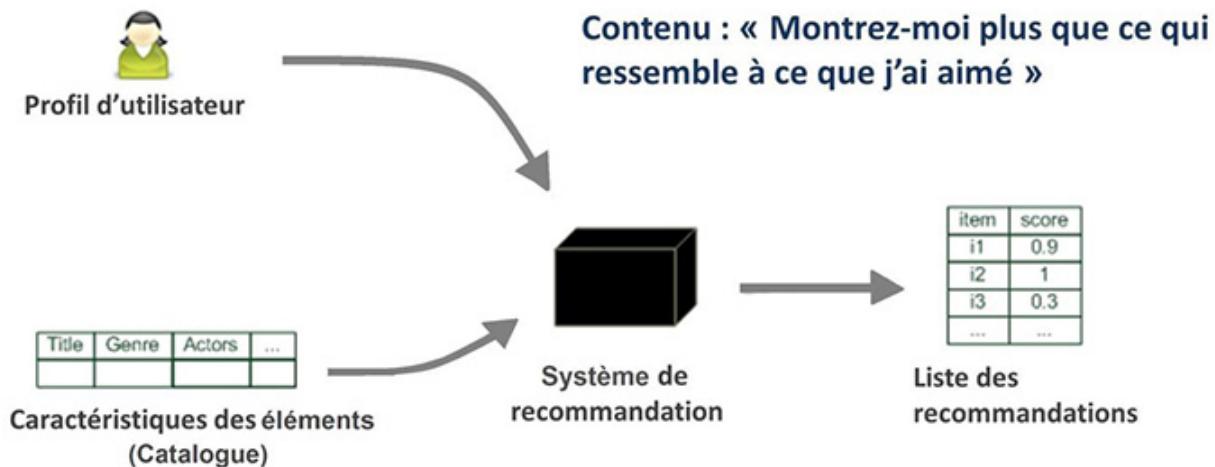


Figure 3.3: Schéma d'un système de recommandation basé sur le contenu.

de régression, qui consiste à prédire la note attribuée par un utilisateur à un élément. Dans les deux cas, il faut définir un modèle qui sera basé sur les paramètres utilisateur ou élément.

- Si le modèle est basé sur les fonctionnalités des utilisateurs, l'approche est dite centrée sur les éléments, c.à.d. la modélisation, les optimisations et les calculs peuvent être effectués «par élément». Dans ce cas, un modèle est construit par article en fonction des fonctionnalités des utilisateurs essayant de répondre à la question «*quelle est la probabilité pour chaque utilisateur d'aimer cet article ?*».
- D'un autre côté, si le modèle est basé sur les fonctionnalités d'éléments, la méthode est alors centrée sur l'utilisateur : la modélisation, les optimisations et les calculs peuvent être effectués «par utilisateur». Le modèle est implémenté par utilisateur en fonction des fonctionnalités des articles qui essaient de répondre à la question "*quelle est la probabilité pour cet utilisateur d'aimer chaque article ?*".

On peut alors attacher un modèle à chaque utilisateur formé sur ses données : le modèle obtenu est donc plus personnalisé que son homologue centré sur l'objet car il ne prend en compte que les interactions de l'utilisateur considéré.

3.2.3 Etude comparative entre les différents types de systèmes de recommandations

Le tableau 3.2 synthétise les avantages et les inconvénients de chaque modèle [19].

On peut considérer qu'il n'y a pas un modèle plus performant que l'autre, mais pour chaque situation il existe un modèle plus adéquat.

Approche	Avantages	Inconvénients
Filtrage collaboratif	<ul style="list-style-type: none"> — Il permet d'utiliser les scores d'autres utilisateur pour évaluer l'utilité des éléments. — Il permet de trouver des utilisateurs ou groupes d'utilisateurs dont les intérêts correspondent à l'utilisateur courant. — Plus il y a d'utilisateurs plus il y a de scores, donc de recommandations. 	<ul style="list-style-type: none"> — Trouver des utilisateurs ou groupes d'utilisateurs similaires est difficile. — Le système de recommandation se heurte à la faible densité de la matrice "Utilisateurs-Éléments". — Il existe le problème du démarrage à froid (cold-start problem) : lorsqu'un nouvel utilisateur utilise le système, ses préférences ne sont pas connues et lorsqu'un nouvel élément est ajouté au catalogue, personne ne lui a attribué de score. — Cette méthode ne suggère pas de recommandation de l'élément sauf si ce dernier est utilisé. — Dans les systèmes avec un grand nombre d'éléments et d'utilisateurs, le calcul croît linéairement, des algorithmes appropriés sont donc nécessaires.

Tableau 3.2: Tableau comparatif entre les différents types de systèmes de recommandations

Approche	Avantages	Inconvénients
Basée sur le contenu	<ul style="list-style-type: none"> — Permet de recommander des éléments similaires à ceux que les utilisateurs ont aimés dans le passé. — Permet de prendre en compte le profil des utilisateurs qui est la clé pour avoir les recommandations les plus pertinentes pour chacun. — Faire coïncider les préférences de l'utilisateur et les caractéristiques des éléments fonctionne pour de nombreux types de données (textuelles, numériques, etc.) puisqu'on utilise généralement des listes de mots-clés. Les données relatives aux autres utilisateurs sont inutiles. — Il n'y a pas de problème de démarrage à froid lorsqu'un nouvel élément est ajouté au catalogue ou de faible densité puisqu'il s'agit de faire coïncider les préférences de l'utilisateur et les caractéristiques des éléments. — Il est possible de faire des recommandations à des utilisateurs avec des goûts «uniques». — Il est possible de recommander de nouveaux éléments ou même des éléments qui ne sont pas populaires. 	<ul style="list-style-type: none"> — Tous les contenus ne peuvent pas être représentés avec des mots-clés (par exemple, les images). — Des éléments représentés par le même ensemble de mots-clés ne peuvent pas être distingués. Les utilisateurs ayant visualisé un très grand nombre d'éléments posent un problème (trop d'informations dans le profil de l'utilisateur à faire coïncider avec les caractéristiques des éléments). — Lorsqu'un nouvel utilisateur commence à utiliser le système, il n'existe pas d'historique. Un risque de « sur-spécialisation » apparaît, c'est-à-dire que l'on se limite aux éléments similaires et que les réponses sont trop homogènes. — Les profils des utilisateurs restent difficiles à élaborer et, en plus, il faut prendre en compte l'évolution des intérêts de l'utilisateur. — Pour que le système produise des recommandations précises, l'utilisateur doit fournir un feedback sur les suggestions retournées. — Ce type de système est entièrement basés sur les scores d'éléments et les scores d'intérêt : moins il y a de scores, plus l'ensemble de recommandations possibles est limité.

Tableau 3.2: Tableau comparatif entre les différents types de systèmes de recommandations

3.3 Mesure de similarité

On distingue principalement deux types d'approches pour définir la similitude ou la similarité :

- les approches **élément-élément** : basées sur la similarité entre les éléments. Cette approche s'adapte à un nombre très important d'utilisateurs ou d'éléments.

Exemple : Supposons que pour un système de recommandation de films, deux utilisateurs Youssef et Ali aiment « L'Âge de glace » et « Avangers ». Cela suggère que, en général, les personnes qui aiment « L'Âge de glace » aimeront aussi «Avangers», donc «Avangers» pourra être recommandé à Zeineb (qui aime aussi « L'Âge de glace »).

- Les approches **Utilisateur-Utilisateur** : basées sur la similarité entre les utilisateurs. Cette approche n'est pas adaptée à un nombre très important d'utilisateurs.

Exemple : Supposons que Youssef et Ali ont des avis similaires ; Youssef apprécie aussi « Joker », c'est donc une bonne recommandation pour Ali.

3.4 Évaluation d'un système de recommandation

Comme pour tout algorithme d'apprentissage automatique, nous devons être en mesure d'évaluer les performances de nos systèmes de recommandation afin de décider quel algorithme correspond le mieux à notre situation. Les méthodes d'évaluation des systèmes de recommandation peuvent principalement être divisées en deux ensembles : évaluation basée sur des paramètres bien définis et évaluation basée principalement sur le jugement humain et l'estimation de la satisfaction.

- **Évaluation basée sur des métriques** : Si notre système de recommandation est basé sur un modèle qui génère des valeurs numériques telles que les prévisions de cotation ou les probabilités de correspondance, nous pouvons évaluer la qualité de ces sorties de manière très classique en utilisant une mesure d'erreur telle que, par exemple, l'erreur quadratique moyenne (MSE). Dans ce cas, le modèle n'est formé que sur une partie des interactions disponibles et est testé sur les autres.
- **Évaluation basée sur l'homme** : Lors de la conception d'un système de recommandation, nous pouvons être intéressés non seulement pour l'obtention un modèle produisant des recommandations dont nous sommes très sûrs, mais nous pouvons également nous attendre à d'autres bonnes propriétés telles que la diversité et l'explication des recommandations.

Avant de commencer l'explication du contenu, nous voulons clarifier un principe que l'on utilisera après.

3.5 Proposition d'un système de recommandation à intégrer dans l'application Spory

Notre objectif est d'intégrer dans notre application un système de recommandation pour fournir à l'utilisateur des propositions de centres de remise en forme pertinentes selon ses préférences. Ce système de recommandation a pour but de proposer à l'utilisateur via une interface les offres les plus intéressantes pour lui, et aussi de trouver des offres qu'il est susceptible d'aimer mais auxquels il n'aurait pas forcément fait attention.

Dans cette section nous introduisons le principe de notre système de recommandation ainsi que la démarche adoptée.

3.5.1 Principipe

Dans notre projet nous avons opté pour un système de recommandation de type « Content-Based ». La tâche consiste donc à déterminer quels éléments du catalogue coïncident le mieux avec les préférences de l'utilisateur. Une telle approche ne requiert pas une grande communauté d'utilisateurs ou un gros historique d'utilisation du système.

Il est à noter que dans notre application, le catalogue des éléments correspond à la liste des centres de remises en formes. Tandis que la liste des utilisateurs correspond à la liste des adhérents.

Notre système de recommandation doit résoudre un problème de classification, qui consiste à prédire si un utilisateur « aime » ou non un centre (élément). Autrement dit, notre modèle doit répondre à la question suivante : "quelle est la probabilité pour cet utilisateur d'aimer chaque centre ?". Si la probabilité est supérieure à un seuil fixé, alors l'utilisateur va probablement aimer un centre qui sera recommandé par notre modèle. Notre modèle est donc centré sur l'utilisateur (adhérant).

Pour aboutir à notre objectif, nous proposons de calculer la similarité (similitude) entre les différents centres. Cette similarité est donc de type élément-élément (item-item). Par la suite, nous proposons d'extraire l'historique, c.à.d. les centres visités, d'un utilisateur pour lui recommander les centres les plus similaires à son historique : c'est l'étape de filtrage. Le fonctionnement de notre système de recommandations s'améliore avec l'enrichissement des informations des centres.

Ainsi, notre modèle est constitué principalement de trois phases :

1. Préparation du catalogue des centres
2. Calcul de similarité entre les centres (éléments).
3. Filtrage et classification des centres à recommander à chaque utilisateur

3.5.2 Calcul de similarité : Similarité cosinus

La similarité cosinus (ou cosine similarity) permet de calculer la similarité entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux [20]. Cette métrique est fréquemment utilisée en fouille de textes [21]. En effet, dans le domaine de la recherche de l'information, les modèles de l'espace vectoriel sont largement adoptés. Ces approches utilisent un vecteur caractéristique, dans un espace dimensionnel, pour représenter chaque objet et calculent la similarité en se basant sur la mesure de cosinus ou la distance euclidienne.

Soit deux vecteurs A et B, la similarité de cosinus se calcule par la formule donnée par l'équation 3.1.

$$\text{similarity}(A, B) = \frac{AB}{\|A\|\|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (3.1)$$

La similarité cosinus est fréquemment utilisée en tant que mesure de ressemblance entre deux documents. Cependant, la mesure d'angle entre deux vecteurs ne pouvant être réalisée qu'avec des valeurs numériques, il faut dans ce cas trouver un moyen de convertir les mots d'un document en nombres. La forme la plus simple pourrait être de compter le nombre d'occurrences des mots dans les documents.

En se basant sur ce principe, nous avons supposé qu'un centre peut être considéré comme un document, étant donné qu'un centre est caractérisé par un ensemble de champs numériques et textuels. Ainsi, pour mesurer les similarités entre les centres, il faudra représenter les centres par des vecteur n-dimensionnel, où n correspond au nombre d'attributs des centres. Ensuite, la similarité entre deux centres est mesurée en termes de "mesure cosinus".

3.5.3 Démarche

Rappelons que notre modèle de système de recommandation est composé de trois phases successives. Il commence d'abord par construire le catalogue utilisateur – élément. Ensuite, il mesure la similarité entre les centres. Enfin, il applique un filtrage des centres à recommander pour chaque utilisateur.

3.5.3.1 Phase 1 : Préparation du catalogue

Cette phase consiste à collecter les données afin de générer la matrice d'interactions utilisateur-élément (adhérant-centre), en utilisant un fichier plat, qu'on nomme F_{centre} . Ce fichier contient les informations du centre, il est créé au moment de la configuration préliminaire de l'application et est sauvegardé coté serveur. Il est à noter que ce fichier est incrémenté à chaque inscription d'un nouveau centre.

Autrement dit, le fichier excel est une copie de la table qui correspond au catalogue des centres de remise en forme. A chaque fois la table des centres est mise à jour, le fichier excel l'est aussi.

Ensuite, nous avons considéré que pour chaque centre seuls les attributs les plus pertinents sont pris en considération. Nous avons évalué les sept attributs nécessaires pour évaluer la similarité entre les centres à savoir :

- La localisation à grande échelle, exemple : Tunis, Sousse, Béja, etc.
- La localisation à petite échelle, exemple : El Manzah, El Manar, Cité olympique, etc.
- L'activité dominante du centre, exemple : musculation, fitness, yoga, etc.
- Le nombre d'activités proposées par le centre.
- Le nombre de services offerts par le centre.
- La moyenne des notes données au centre.
- Le nombre de personnes qui ont évalués le centre.

Le tableau 3.3 (page 50) illustre un exemple de catalogue de centres de remise en forme.

Index	Nom	Localisation Grande échelle	Localisation petite échelle	services	nbe activités	Sport Dominant	Note	Vote
0	Arena gym	Tunis	Manar 1	3	4	Fitness	2.7	0
1	California gym	Tunisa	Marsa	1	4	Musculation	2.5	0
2	Fitness Line	Tunis	Nasr 1	1	5	Musculation	4.33	3
3	Best Gym	Tunis	Mutuelle Ville	3	2	Relaxation	2.5	0
4	Laura SPA massage	Sousse	El Kantaoui	2	2	Relaxation	2.5	0
5	Aphordite SPA	Tunis	Lac1	1	2	Relaxation	4	1
6	Africa SimRacing	Tunis	Manouba	1	1	Collective sport	5	1
7	California gym platinum	Tunis	Lac 2	1	2	Musculation	5	1
8	Vitality SPA	Sousse	El Kantaoui	2	2	Relaxation	2.5	0
9	Tunisia first club	Tunis	Ariana	2	1	Divers	2.5	0
10	Gym box	Tunis	Manzah 6	2	2	Musculation	2.5	0
11	Green field	Tunis	Borj Touil	2	1	Collective sport	2.5	0
12	Gym one	Tunis	Ariana	1	3	Fitness	2.5	0

Tableau 3.3: Exemple d'un catalogue de centres de remise en forme.

3.5.3.2 Phase 2 : Construction de la matrice de similarité

La deuxième phase de notre modèle consiste à calculer la similarité cosiné entre les centres deux à deux, pour générer une matrice de forme carrée (n, n), symétrique et des 1 en diagonale, comme illustrée par la figure 3.4 (page 51).

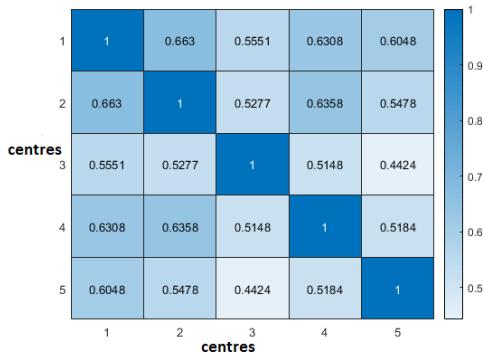


Figure 3.4: Exemple de matrice de similarité cosinus.

Cette matrice contient les probabilités de similarité entre chaque élément : elle sera de la forme carrée, symétrique et des 1 en diagonale.

Pour ce faire, notre modèle transforme chaque centre en un vecteur de chaînes de caractères et combine les sept attributs pertinents, cités précédemment. Ce vecteur est, ensuite, transformé en un vecteur numérique, en appliquant une fonction de vectorisation. En effet, rappelons que la mesure cosinus entre deux vecteurs ne peut être réalisée qu'avec des valeurs numériques.

Une fois tous les centres sont transformés en vecteurs numériques, notre modèle calcule la matrice de similarité cosiné pour obtenir une matrice carré (nn), tel que n correspond au nombre des centres et la valeur à la i^{me} ligne et j^{me} colonne correspond à la similarité entre le i^{me} et le j^{me} centre.

Si on considère l'exemple cité auparavant (tableau 3.3 page 50), le premier élément d'index 0 posséde les caractéristiques "Tunis, Manar, 1, 4, 3, Fitness, 2.5, 0". Ces paramètres correspondent respectivement à, la localisation à grande échelle, la localisation à petite échelle, l'activité dominante du centre, le nombre d'activités proposées par le centre, le nombre de services offerts par le centre, la moyenne des notes données au centre et le nombre de personnes qui ont évalués le centre. La vectorisation numérique de ce centre génère le vecteur suivant : [(0, 20), (0, 9), (0, 6)].

En appliquant le même processus sur tout le catalogue des centres, l'ensemble de vecteurs numériques est généré comme l'indique le tableau 3.4. En calculant la similarité entre les vecteurs deux à deux la matrice de similarité cosinus est obtenue est donnée par la figure 3.5.

[(0, 20), (0, 9), (0, 6)]
[(1, 20), (1, 12), (1, 13)]
[(2, 20), (2, 13), (2, 15), (2, 0)]
[(3, 20), (3, 14), (3, 21), (3, 16)]
[(4, 16), (4, 17), (4, 5), (4, 7)]
[(5, 20), (5, 16), (5, 8)]
[(6, 20), (6, 10), (6, 3), (6, 18)]
[(7, 20), (7, 13), (7, 8)]
[(8, 16), (8, 17), (8, 5), (8, 7)]
[(9, 20), (9, 1), (9, 4)]
[(10, 20), (10, 13), (10, 11)]
[(11, 20), (11, 3), (11, 18), (11, 2), (11, 19)]
[(12, 20), (12, 6), (12, 1)]

Tableau 3.4: Vectorisation du catalogue des centres de remise en forme, donné par le tableau 3.3.

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1	0.33	0.28	0.28	0	0.33	0.28	0.33	0	0.33	0.33	0.25	0.66
1	0.33	1	0.57	0.28	0	0.33	0.28	0.66	0	0.33	0.66	0.25	0.33
2	0.28	0.57	1	0.25	0	0.28	0.25	0.57	0	0.28	0.57	0.22	0.28
3	0.28	0.28	0.25	1	0.25	0.57	0.25	0.28	0.25	0.28	0.28	0.22	0.28
4	0	0	0	0.25	1	0.28	0	0	0.99	0	0	0	0
5	0.33	0.33	0.28	0.57	0.28	1	0.28	0.66	0.28	0.33	0.33	0.25	0.33
6	0.28	0.28	0.25	0.25	0	0.28	1	0.28	0	0.66	0.28	0.67	0.28
7	0.33	0.66	0.57	0.28	0	0.66	0.28	1	0	0	0.66	0.25	0.33
8	0	0	0	0.25	0.99	0.28	0	0	1	0	0	0	0
9	0.33	0.33	0.28	0.28	0	0.33	0.66	0	0.33	1	0.33	0.25	0.66
10	0.33	0.66	0.57	0.28	0	0.33	0.28	0.66	0	0.33	1	0.25	0.33
11	0.25	0.25	0.22	0.22	0	0.25	0.67	0.25	0	0.25	0.25	1	0.25
12	0.66	0.33	0.28	0.28	0	0.33	0.28	0.33	0	0.66	0.33	0.25	1

Figure 3.5: Matrice de similarité cosinus des centre du catalogue donné par le tableau 3.3.

3.5.3.3 Phase 3 : Filtrage des centres à recommander à chaque utilisateur

Afin de recommander les centres à un utilisateur, notre modèle compare la similarité de l'historique des visites de chaque utilisateur aux autres centres du catalogue. Les centres qui peuvent être recommandés sont ceux qui ont une certaine ressemblance. Cette phase consiste donc à construire un classifieur, qui a pour objectif d'extraire pour chaque centre, tous les centres ayant une similarité supérieure à un seuil donné. Cette similarité sera considérée comme le score du centre. Nous avons proposé de fixer la valeur du seuil à 50%, c.à.d les centres ayant une similarité supérieur à 0.5.

Rappelons que le processus d'extraction des centres à recommander est basé sur l'historique d'un utilisateur. C'est à dire que pour chaque centre déjà visité, une liste des centres qui lui sont similaires est proposée, ce qui peut provoquer une apparition de redondance dans le résultat final. Par exemple,

si un utilisateur a visité deux centres C_1 et C_2 , et que le centre C_3 est similaire à la fois aux deux centres visités. Le centre C_3 sera donc proposé à deux reprises pour l'utilisateur.

Pour remédier à ce problème, nous proposons d'appliquer un filtrage supplémentaire, afin de conserver une liste unique des centres ayant une similarité maximale par rapport aux centres visités par l'adhérent. Une fois la liste des centres à recommander est obtenue, elle sera affichée à l'utilisateur trié selon un ordre décroissant des scores des centres. Autrement dit, les centres ayant les plus grands scores seront affichés en premiers, puisqu'ils ont une grande similarité aux centres déjà visités.

Reprenons l'exemple cité auparavant (tableau 3.3 page 50). Si on considère qu'un adhérent a visité 3 centres d'index 3, 5 et 7. D'après la matrice de similarité cosine, donnée par la figure 3.5, les centres à suggérer par notre système sont donnés par le tableau 3.5. On remarque que pour les centres d'index 3 et 7 le système de recommandation a suggéré un centre en commun, qui est le centre d'index 5, mais avec des similarités différentes. D'où la nécessité d'un filtrage avant affichage. Ensuite, en appliquant un tri décroissant des centres selon leurs similarités aux centres visités, la liste des index des centres à recommander est la suivante : 1, 7, 5, 10, 7, 3, et 2.

Index du centre visité	(Centres à recommander, Mesure de similarité)
3	(5, 0.57)
7	(1, 0.66), (2, 0.57), (5, 0.66), (10, 0.66)
5	(3, 0.57), (7, 0.66)

Tableau 3.5: Exemple de centres de remise à forme à recommander, selon la matrice de similarité donnée par la figure 3.5.

Conclusion

Dans cette partie nous avons introduit l'importance des systèmes de recommandations, leurs types ainsi que leurs avantages. Nous avons également dégagé une synthèse sur les différents types de systèmes de recommandation, afin de dégager le type le plus approprié à notre application. Par la suite, nous avons présenté la conception de notre modèle de système de recommandation. Notre système est basé sur une approche de type content-based et a pour objectif de comprendre le comportement d'un utilisateur et de lui proposer une liste personnalisée de recommandations.

IMPLEMENTATION ET REALISATION

Plan

1	Environnement de développement	56
4.1.1	Introduction NoSQL	56
4.1.2	Mongo DB	57
4.1.3	Mongo DB Atlas	58
4.1.4	Déploiement continu	59
4.1.5	Heroku : Déploiement sur un serveur	60
4.1.6	Développement des applications hybrides	60
4.1.7	Technologies Front-End / Back-end	61
4.1.7.1	Technologie Front-End : Ionic 4	61
4.1.7.2	Technologie Back-End : Node JS	61
4.1.7.3	Technologie Back-End (Module système de recommandation)	62
2	Architecture	62
4.2.1	Architecture matérielle	62
4.2.2	Architecture logicielle	63
3	Interfaces	65
4.3.1	Interfaces d'accueil	65
4.3.2	Scénario : Utilisation de l'application par l'adhérant	68
4.3.2.1	Interface d'authentification	68
4.3.2.2	Interface Carte	69
4.3.2.3	Interface Calendrier	71
4.3.2.4	Interface Page de la salle (En tant que client)	72
4.3.2.5	Interface réservation	72
4.3.2.6	Interface Découvrir	73
4.3.3	Scénario : utilisation de l'application par le gérant	74
4.3.3.1	Interface Page de la salle (En tant que gérant)	74

4.3.3.2	Interface Statistiques	75
4.3.3.3	Interface Scanner QR code	75
4.3.3.4	Interface Liste des adhérents	76
4.3.4	Scénario : utilisation de l'application par l'administrateur	77
4.3.4.1	Interface validations centres	77
4.3.4.2	Interface blocage des centres	77
4.3.4.3	Interface gestions des adhérents	78
4.3.4.4	Interface gestions des avis	78
4	Conclusion	79

Introduction

Dans ce chapitre, nous allons présenter des principes fondamentaux appliqués dans notre application ainsi que les outils techniques et l'architecture. Dans les chapitres précédents, nous avons détaillé la conception de notre application et présenté le modèle de système de recommandation sur lequel se base les recommandations des centres aux utilisateurs. Ce dernier chapitre est dédié à la présentation de la réalisation de la solution proposée. Nous présentons, dans un premier lieu, l'environnement logiciel utilisé pour le développement de notre application. Ensuite, nous exposerons les différentes technologies utilisées. Pour finir, nous illustrons le travail réalisé à travers un scénario de décrivant les principales fonctionnalités.

4.1 Environnement de développement

4.1.1 Introduction NoSQL

Les bases de données NoSQL (alias "Not Only SQL") ne sont pas tabulaires et stockent les données différemment des tables relationnelles [22]. Les bases de données NoSQL sont de différents types en fonction de leur modèle de données. Les principaux types sont le document, la valeur-clé, la colonne large et le graphique [23]. Ils fournissent des schémas flexibles et évoluent facilement avec de grandes quantités de données et des charges d'utilisateurs élevées.

Les bases de données NoSQL peuvent stocker des données de relation, cependant elles les stockent différemment des bases de données relationnelles.

De plus, la modélisation des données de relation dans les bases de données NoSQL peut sembler plus facile que dans les bases de données SQL, car les données associées n'ont pas besoin d'être réparties entre les tables.

Les bases de données NoSQL sont apparues à la fin des années 2000 alors que le coût du stockage diminuait considérablement, ceci a impacté directement la logique des modèles de données complexes et difficiles à gérer dans le seul but de réduire la duplication des données [24]. Les développeurs, plutôt que le stockage, devenaient le principal coût du développement logiciel, c'est ce qui explique le fait que les bases de données NoSQL sont considérées comme étant optimisées pour la productivité des développeurs.

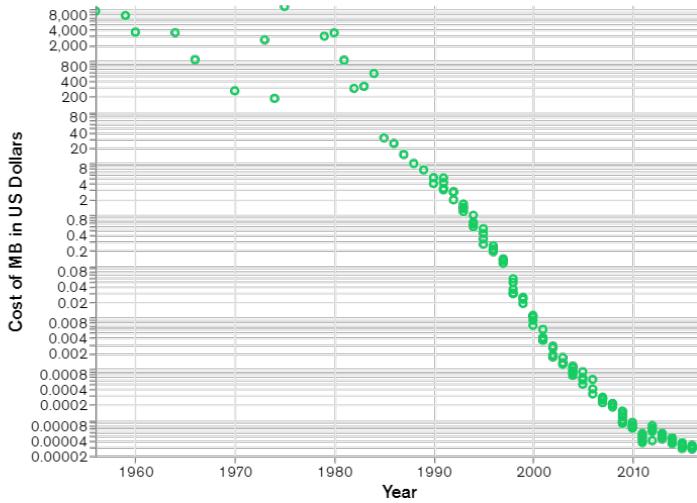


Figure 4.1: Coût de stockage des données par rapport au temps

4.1.2 MongoDB

MongoDB [25] est une base de données de documents construite sur une architecture évolutive qui est devenue populaire auprès des développeurs de tous types qui créent des applications évolutives à l'aide de méthodologies agiles [26].

MongoDB a été conçu pour les personnes qui créent des applications Internet et commerciales qui doivent évoluer rapidement. Ce type de base de données offre plusieurs avantages à savoir :

- Le modèle de données de document est un moyen puissant de stocker et de récupérer des données qui permet aux développeurs de se déplacer rapidement.
- Une base MongoDB peut prendre en charge d'énormes volumes de données et de trafic.
- Elle fonctionne sur tous les types de plates-formes informatiques, à la fois sur site et dans le cloud (à la fois les clouds privés et publics tels que AWS, Azure et Google Cloud).
- Elle est accessible à partir de tous les principaux systèmes ETL et de gestion de données.
- MongoDB dispose d'une communauté mondiale de développeurs et de consultants, il est donc facile d'obtenir de l'aide.

MongoDB reste une solution fortement utilisée par les développeurs. Ce qui en fait sa popularité est son langage puissant et le fait qu'il soit facilement intégrable dans toute application gérant des documents/objets. La figure 4.2 illustre les scores des utilisations des bases NoSQL, on remarque que les bases MongoDB sont largement utilisées au cours de ses dernières années.

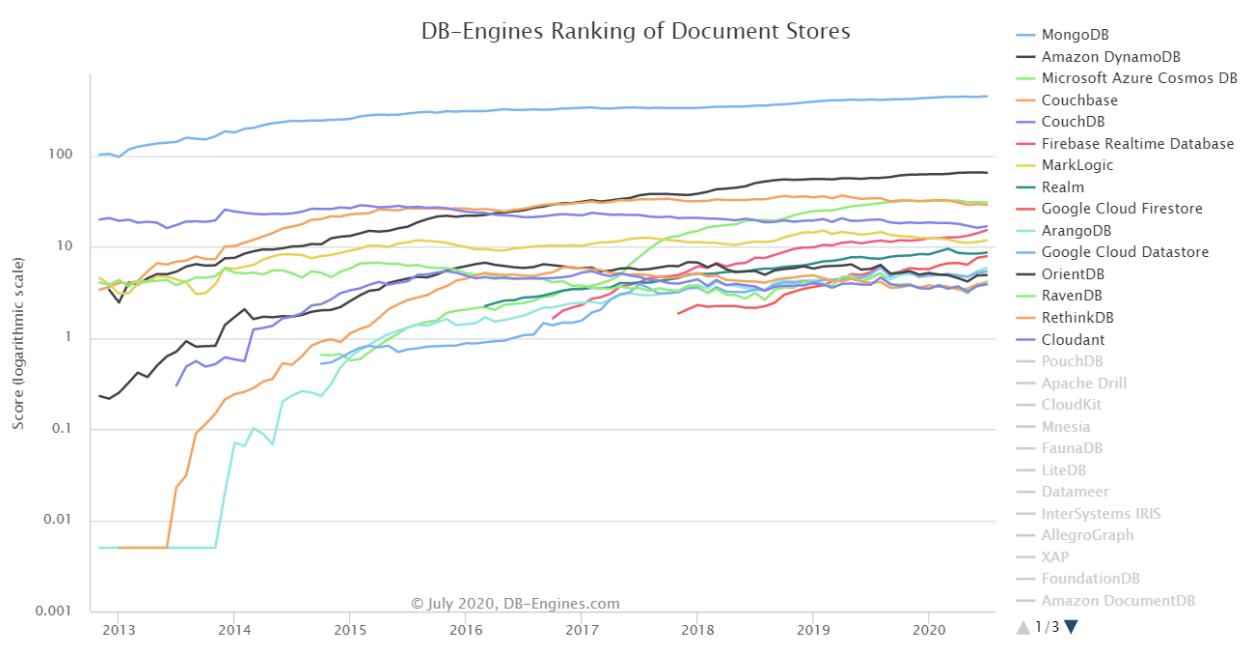


Figure 4.2: Graphe du score des bases de données NoSQL

4.1.3 Mongo DB Atlas

MongoDB Atlas est une base de données cloud entièrement gérée développée par les mêmes personnes qui ont construit MongoDB [27]. Atlas gère toute la complexité du déploiement, de la gestion et de la réparation des déploiements sur le fournisseur de services cloud de votre choix (AWS, Azure). Au total, MongoDB Atlas est présent sur 17 régions cloud à travers le monde.

Mongo DB Atlas offre :

- **Fonctionnalités de sécurité automatisées** : MongoDB Atlas garantit la confidentialité des données.
- **RéPLICATION intégrée** : La plate-forme fournit plusieurs serveurs pour une disponibilité permanente, pour assurer la stabilité du projet.
- **Sauvegardes et récupération à un moment donné** : Le système MongoDB Atlas protège contre la corruption des données.
- **Surveillance fine** : Mongo DB Atlas admet plusieurs outils, organisés de différentes manières, afin d'améliorer le matériel et stabiliser la performance.
- **Correctifs automatisés et mises à niveau en un clic** : MongoDB Atlas admet des correctifs automatisés.

4.1.4 Déploiement continu

Dans les applications web en cours de développement, il faut faire des livraisons effectives des fonctionnalités et des correctifs dès qu'ils sont prêts. Ce dynamisme au niveau de la plateforme déployée, nous oriente vers l'utilisation d'une méthode de déploiement continu [28].

Le déploiement continu est une stratégie de développement logiciel où toute validation de code qui réussit le cycle de test automatisé est automatiquement transférée dans l'environnement de production, propulsant ainsi les modifications vers les utilisateurs du logiciel [28].

Le déploiement continu fait partie de ce que l'on appelle le développement logiciel continu, et sont souvent associés aux méthodologies Agile et DevOps. Nous avons utilisé Git [29] et Github [30] comme outils de déploiement continu.

Git est un système de contrôle de version qui permet de gérer et de suivre l'historique du code source. Tandis que GitHub est un service d'hébergement cloud qui permet de gérer les référentiels Git.

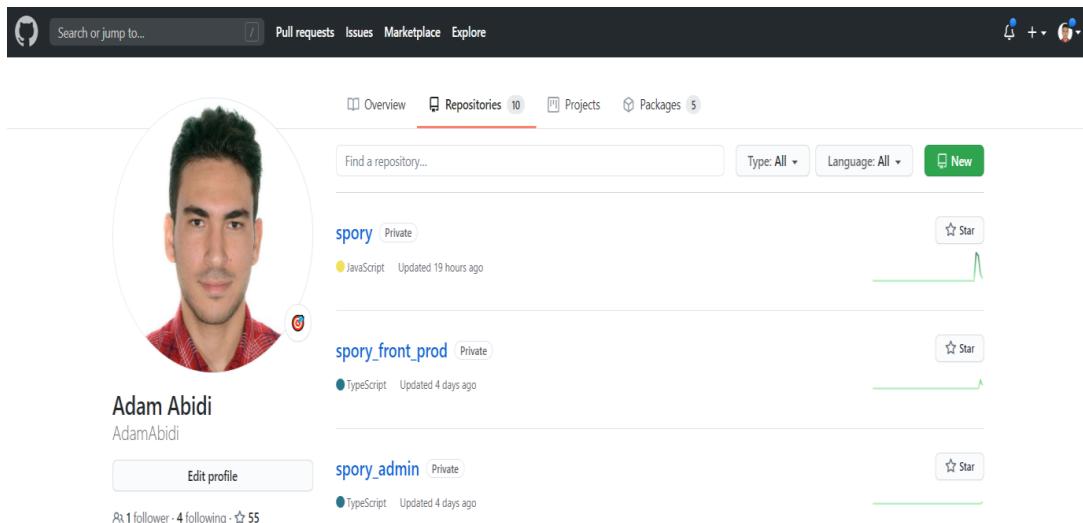


Figure 4.3: Interface de Github contenant les projets

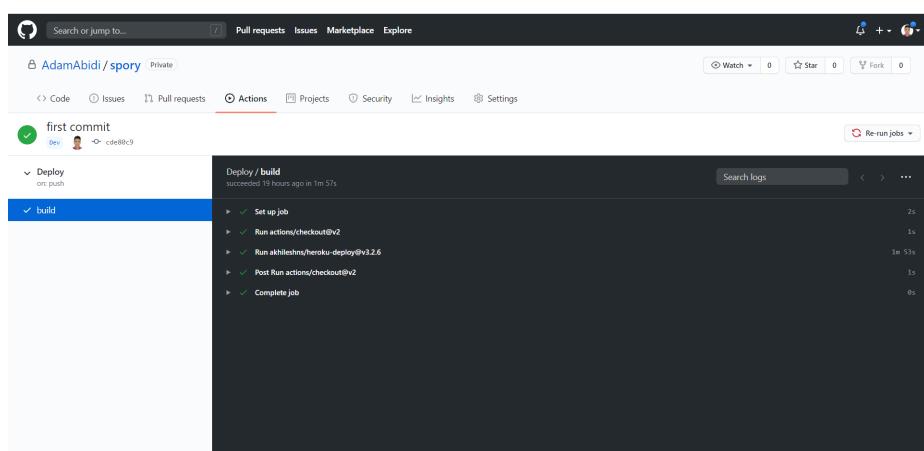


Figure 4.4: phase de test et automatisation de Github

4.1.5 Heroku : Déploiement sur un serveur

Heroku est une plateforme de services cloud dont la popularité s'est accrue ces dernières années. Heroku est simple à utiliser qu'il est le meilleur choix pour de nombreux projets de développement [31].

Avec un accent particulier sur la prise en charge des applications axées sur le client, cette plateforme permet le développement et le déploiement d'applications simples. En effet, la plateforme Heroku gère le matériel et les serveurs, ainsi les entreprises peuvent se concentrer sur le perfectionnement de leurs applications et non sur l'infrastructure qui les prend en charge.

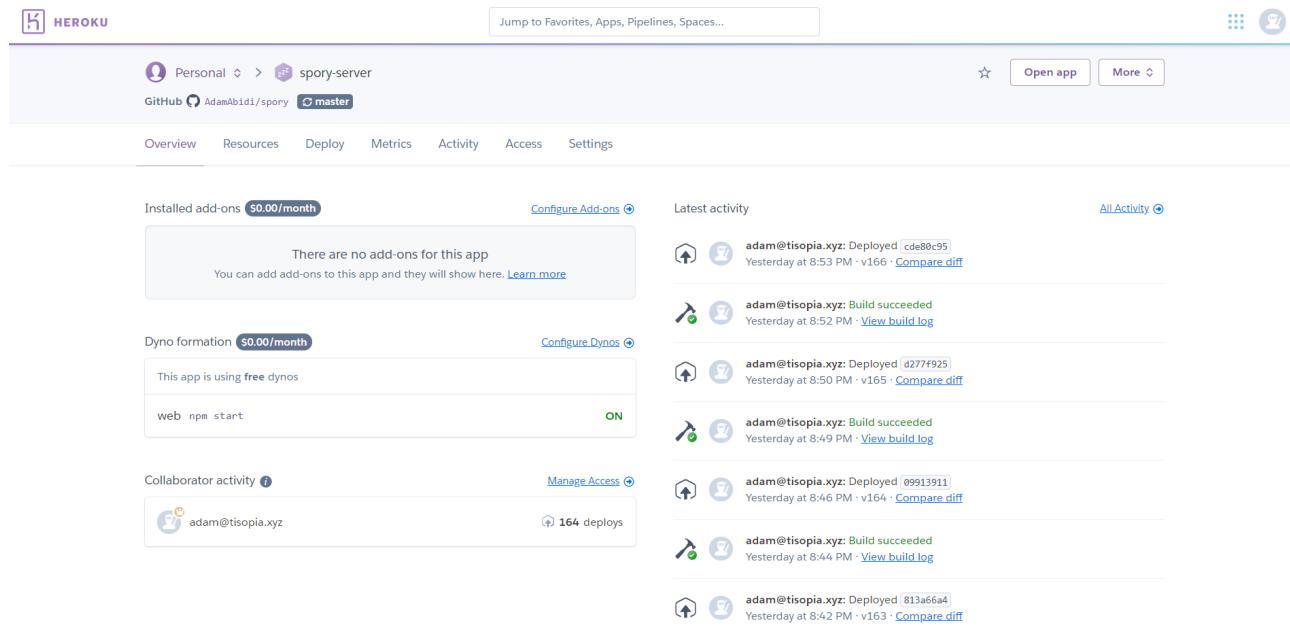


Figure 4.5: Interface de l'aperçu d'une application déployée sur Heroku

4.1.6 Développement des applications hybrides

Une application mobile hybride est développée à partir de langages web (HTML5, JavaScript, CSS...). Cependant, elle s'appuie également sur des technologies natives mobiles pour utiliser certaines fonctionnalités du smartphone. Bien que développée avec les technologies du web HTML5, JavaScript, CSS...), il s'agit tout de même bien d'une « application » dans le sens où elle sera téléchargée depuis les magasins d'applications mobiles et installée sur le mobile, contrairement à la application web qui n'est consultable que depuis un navigateur. Par exemple, LinkedIn est une application mobile hybride. A la différence des applications natives qui exigent de connaître des langages de programmation pointus et propres à chaque système d'exploitation, le développement hybride ne nécessite donc pas un apprentissage complexe. L'autre avantage des applications hybrides est qu'elles sont développées qu'une seule fois tout en étant compatibles avec les principaux systèmes d'exploitation

des smartphones : iOS, Android, Windows Phone. Les compétences web sont plus répandues que les compétences mobiles, les coûts de développement sont donc généralement moins importants.

4.1.7 Technologies Front-End / Back-end

4.1.7.1 Technologie Front-End : Ionic 4

Nous avons utilisés Ionic 4 comme outil de développement hybride [32]. En effet, Ionic [33] est une plateforme d'applications conviviale pour créer des applications multiplateformes avec une base de code, pour n'importe quel appareil. En 2018, Ionic figurait parmi les frameworks d'applications mobiles hybrides les plus préférés et les meilleurs pour les développeurs. Et, jusqu'à présent, il est en tête de liste pour les développeurs. Il possède des éléments et des mises en page de style natif. En effet, Ionic permet de :

- Fournir régulièrement les dernières mises à jour.
- Fournir une bibliothèque de composants HTML, CSS et JS.
- Fournir des fonctionnalités natives.
- Avoir des outils de personnalisation.



Figure 4.6: Logo d'Ionic

4.1.7.2 Technologie Back-End : Node JS

Node.js [34] est un environnement d'exécution open source multiplateforme pour le développement d'applications côté serveur et réseau. Les applications Node.js sont écrites en JavaScript et peuvent être exécutées dans l'environnement d'exécution Node.js sous OS X, Microsoft Windows et Linux. Node.js fournit également une riche bibliothèque de divers modules JavaScript qui simplifie dans une large mesure le développement d'applications Web utilisant Node.js. C'est une plateforme qui facilite la création des applications réseau rapides et évolutives. En effet, Node.js est adapté pour les applications en temps réel gourmandes en données qui s'exécutent sur des appareils distribués.



Figure 4.7: Logo de Node JS

4.1.7.3 Technologie Back-End (Module système de recommandation)

Nous avons utilisé Python 3.7 pour le développement du système des recommandations.

Python est un langage de programmation de haut niveau orienté objet avec une sémantique dynamique intégrée principalement pour le développement Web et d'applications [35]. Dans le domaine de l'intelligence artificielle, Python se distingue tout particulièrement en offrant un ensemble de librairies de très grande qualité, couvrant tous les types d'apprentissages disponibles sur le marché. Le tout accompagné d'une grande communauté [36].



Figure 4.8: Logo de Python

4.2 Architecture

Dans cette partie nous allons spécifier l'architecture matérielle ainsi que l'architecture logicielle de notre application.

4.2.1 Architecture matérielle

L'architecture la plus appropriée pour notre projet est une architecture 3-tiers. C'est une architecture de systèmes informatiques dans laquelle un ensemble de machines interconnectées partagent des ressources. L'architecture trois tiers, aussi appelée architecture à trois niveaux ou architecture à trois couches. L'architecture matérielle du système est divisée en trois niveaux ou couches :

- Couche de présentation : Affichage dédiée pour le client.
- Couche de traitement : Logique métier dédiée pour le serveur.
- Couche d'accès aux données : Serveur base des données.

C'est une architecture basée sur l'environnement client–serveur.

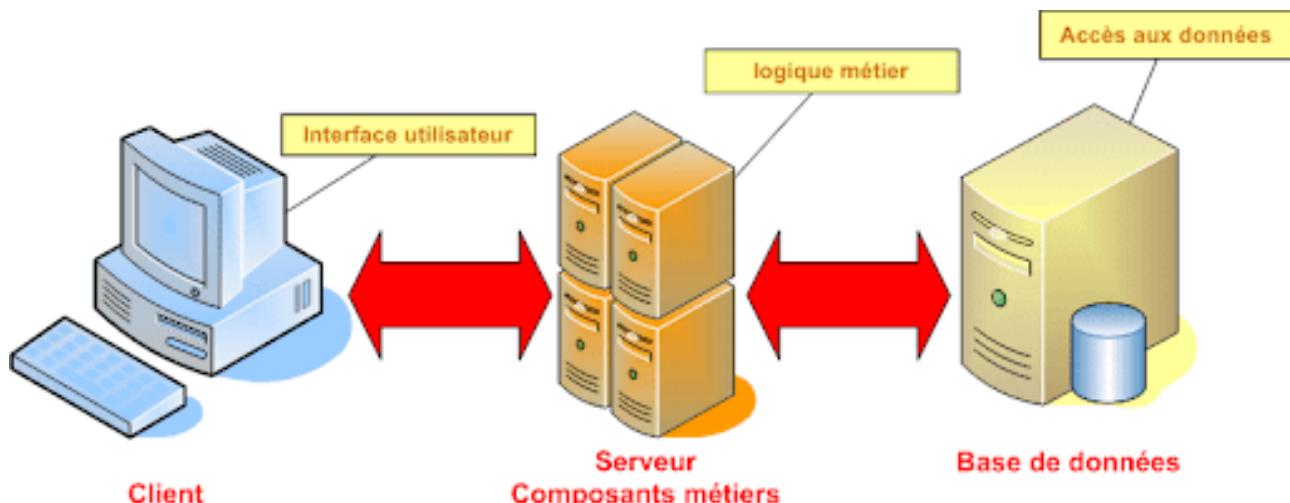


Figure 4.9: Architecture matérielle 3 tiers

4.2.2 Architecture logicielle

Notre projet adopte MVC comme architecture logicielle : c'est un est un motif d'architecture logicielle, très populaire pour les applications web. MVC est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

- Un modèle (Model) : contient les données à afficher. C'est l'élément qui contient les données ainsi que de la logique en rapport avec les données : validation, lecture et enregistrement. Il peut, dans sa forme la plus simple, contenir uniquement une simple valeur, ou une structure de données plus complexe. Le modèle représente l'univers dans lequel s'inscrit l'application.
- Une vue (View) : contient la présentation de l'interface graphique. C'est la partie visible d'une interface graphique. Une vue se peut être un diagramme, un formulaire, des boutons, etc. Elle contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle.
- Un contrôleur (Controller) : contient la logique concernant les actions effectuées par l'utilisateur. C'est le module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue.

C'est une architecture basée sur l'environnement client–serveur.

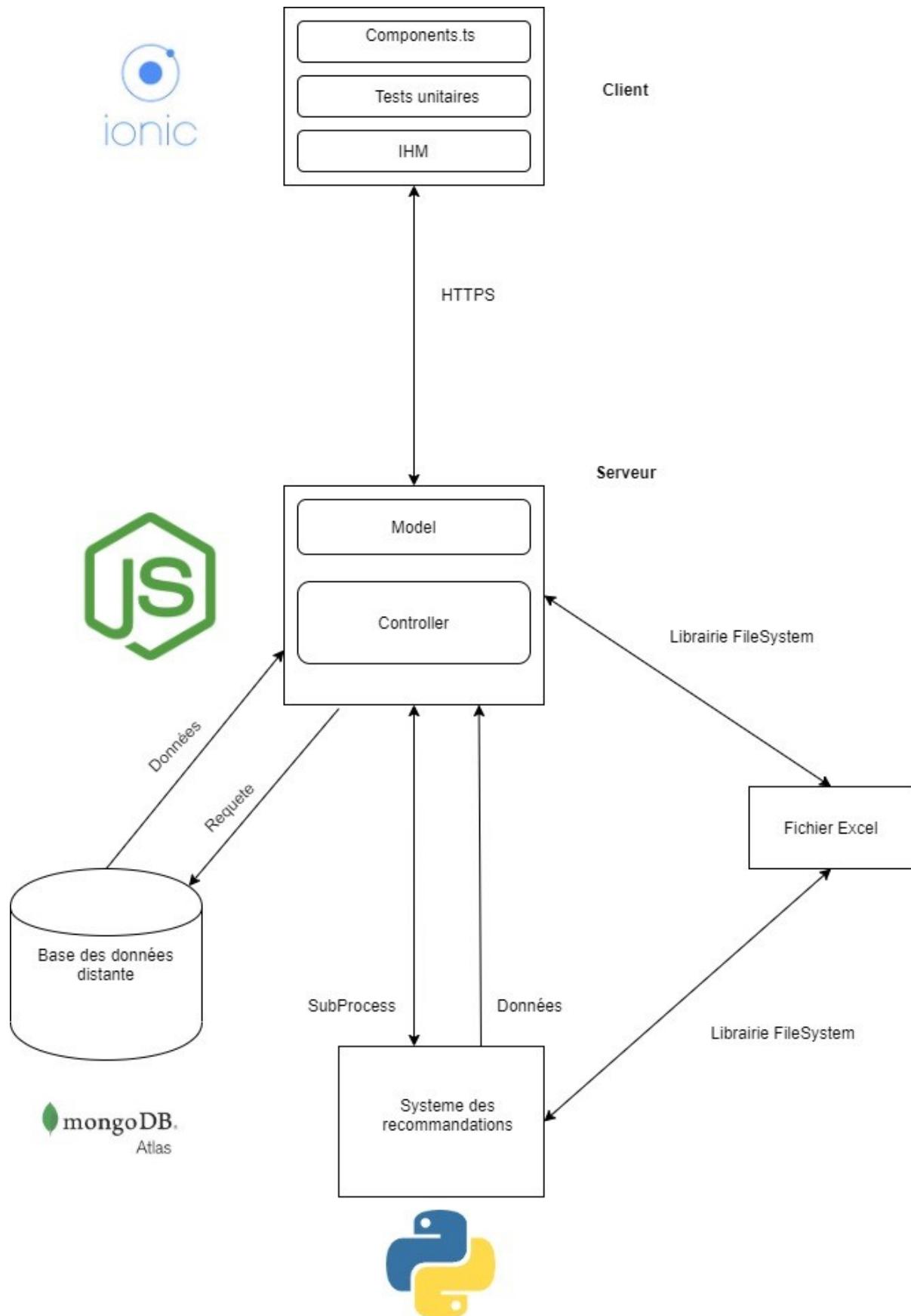


Figure 4.10: Architecture logicielle détaillée

4.3 Interfaces

4.3.1 Interfaces d'accueil

C'est la première interface qui apparaît à tout utilisateur. La plateforme présente des «slides» comme bienvenue contenant des images explicatives et permettant d'introduire le produit.



Figure 4.11: Page d'accueil

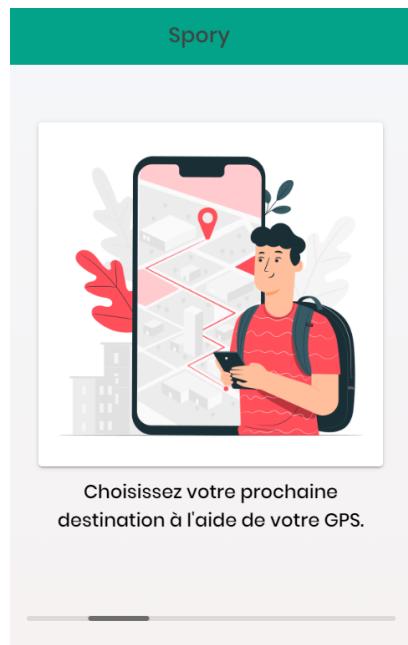


Figure 4.12: Slide d'accueil : présentation de géolocalisation

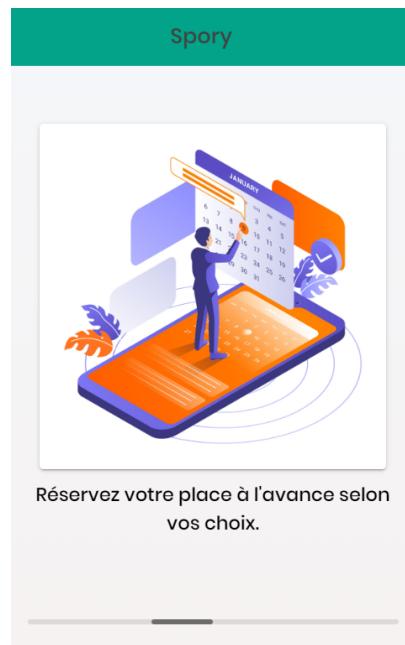


Figure 4.13: Slide d'accueil : présentation du calendrier de réservation

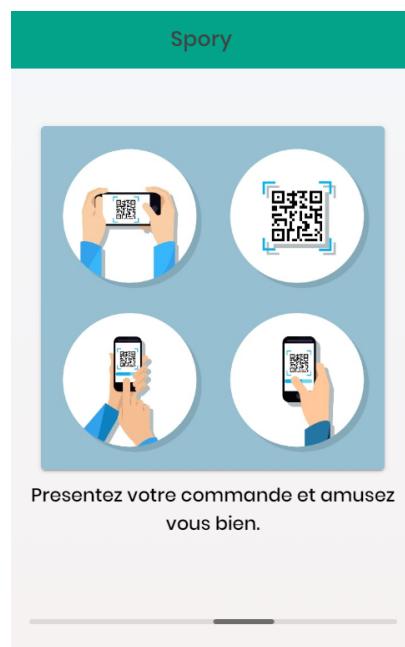


Figure 4.14: Slide d'accueil : présentation du QR code



Figure 4.15: Slide d'accueil : présentation d'autres services



Figure 4.16: Slide d'accueil : Invitation de passer à l'authentification

4.3.2 Scénario : Utilisation de l'application par l'adhérant

4.3.2.1 Interface d'authentification

Le client se retrouve dans cette interface afin de créer un compte en tant que adhérent ou gérant, une validation de mail est exigée par la saisie d'un code envoyé dans la boite de réception de l'adresse tapée. Tout client peut se connecter en tant que adhérent ou gérant et récupérer son mot de passe via un mail. Lorsque le client acquiert un compte d'adhérent ou gérant, une auto-connexion s'établit dès la consultation suivante de l'application. La plateforme offre le choix d'utiliser l'application comme un invité.

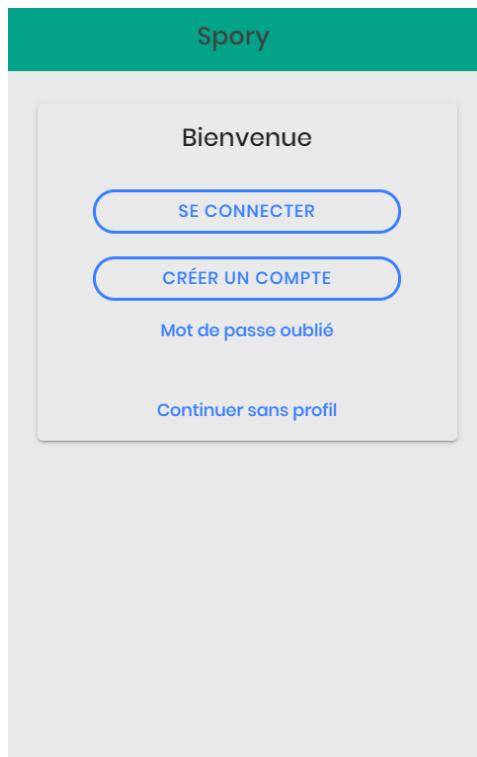


Figure 4.17: Interface d'authentification

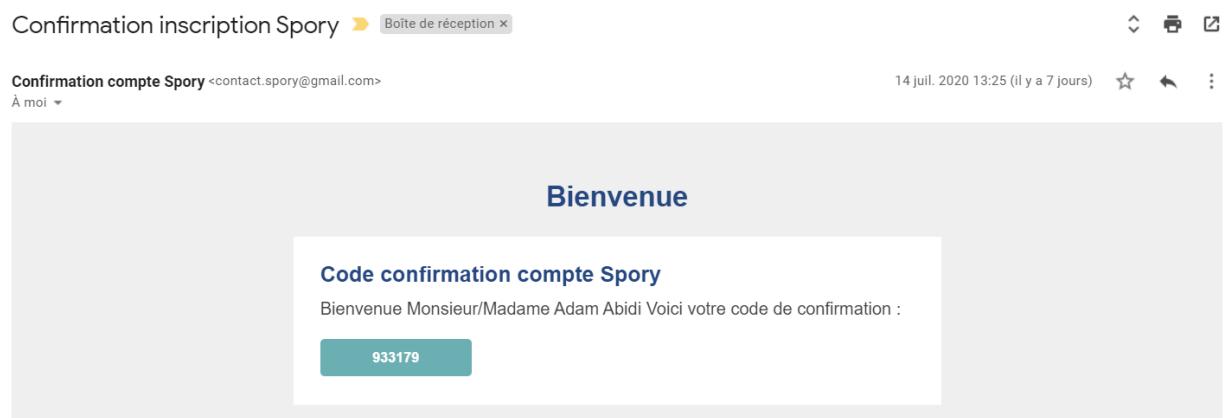


Figure 4.18: le mail de vérification de l'adresse électronique

4.3.2.2 Interface Carte

Dans cette interface, la carte géographique s'étend sur la majorité de l'écran, elle positionne le client au centre et lui remonte les différents centres de remise en forme tout autour, ceci lui permet d'avoir une visibilité globale sur les distances et le nombre des centres disponibles. Le client a le choix entre consulter la carte, choisir un centre, tracer la route ou bien consulter la liste des clubs, tout ceci reste paramétrable via un filtre selon le type de divertissement. Le client peut ensuite visiter la page officielle du centre ou consulter les autres onglets relatifs aux autres centres. Seul un adhérent peut bénéficier de l'option « profil » et « calendrier » afin d'encourager les invités de créer leurs propres profils.

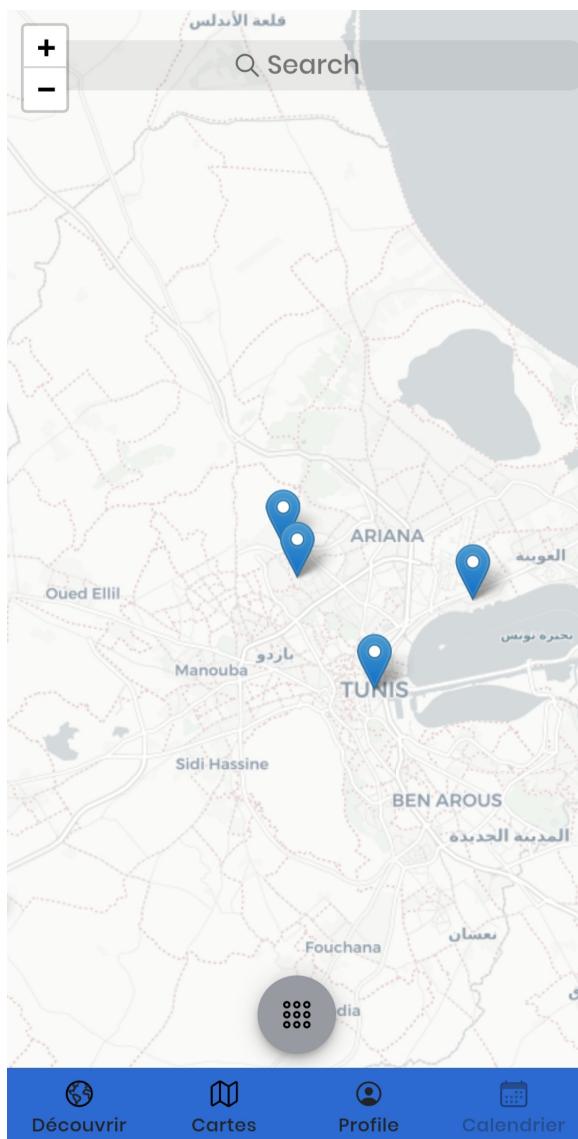


Figure 4.19: Carte initiée

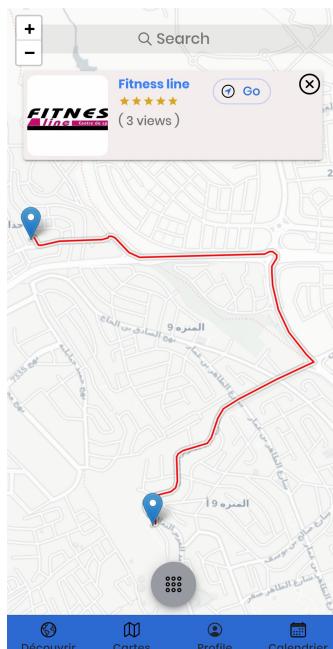


Figure 4.20: GPS tracé

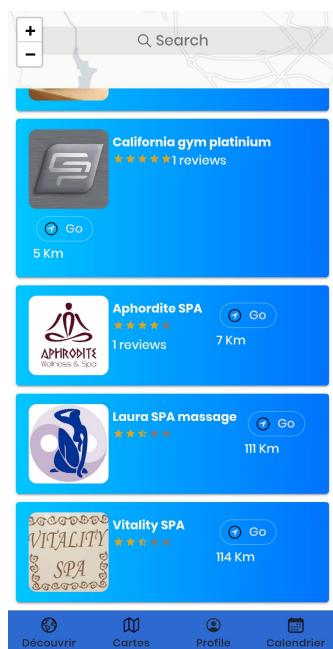


Figure 4.21: Interface Carte contenant la liste des salles

4.3.2.3 Interface Calendrier

Tout adhérent peut consulter son historiques des offres ainsi que les réservations d'avance. Tout adhérent peut voir le QR code de toute réservation générée par l'application ainsi que modifier la date de la réservation tout en vérifiant les disponibilités du centre et du client.

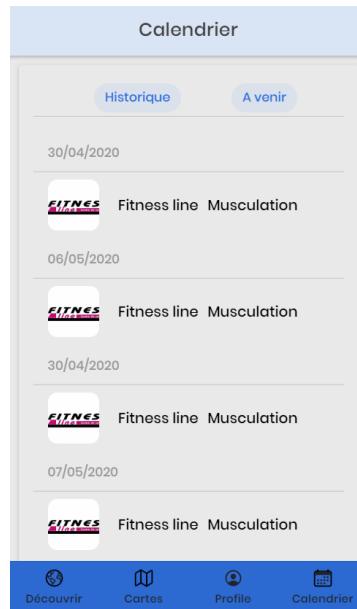


Figure 4.22: Historique des achats

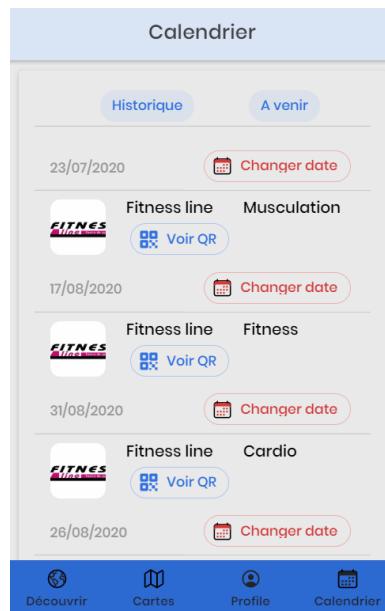


Figure 4.23: Les réservations à venir

4.3.2.4 Interface Page de la salle (En tant que client)

Chaque utilisateur peut visualiser le profil du centre, voir les images, les services et les avis des adhérents. Tout adhérent peut passer à la réservation en suivant les règles de l'application.



Figure 4.24: Page officielle du centre vu par l'adhérent

4.3.2.5 Interface réservation

Cette interface comporte les différents types de loisirs adoptés par le centre, ainsi que les offres de la semaine courante du type sélectionné. Tout adhérent ne peut réserver sa place que si la date est respectée, son crédit est suffisant et qu'au moins une place est vacante.

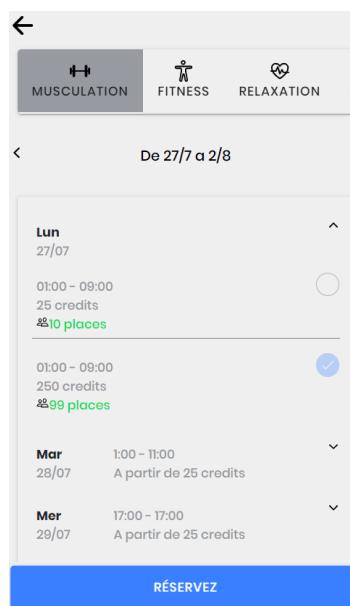


Figure 4.25: Interface de réservation

4.3.2.6 Interface Découvrir

Dans cette interface, la plateforme affiche les clubs les plus visités ainsi que les séances les plus populaires. L'application visualise aussi les nouveautés du mois. Seul un adhérent peut avoir accès à la fonctionnalité de la recommandation selon son profil.

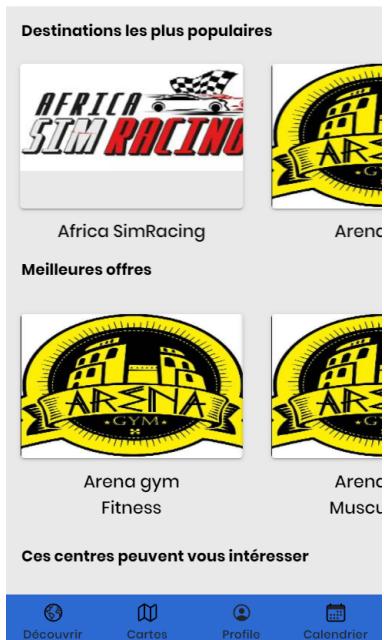


Figure 4.26: Interface Découvrir

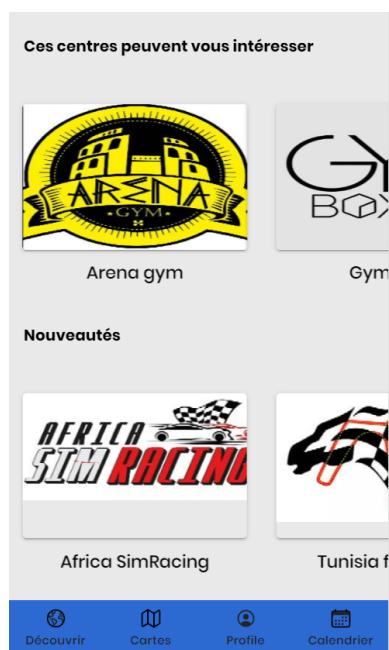


Figure 4.27: Interface Découvrir

4.3.3 Scénario : utilisation de l'application par le gérant

4.3.3.1 Interface Page de la salle (En tant que gérant)

Après l'authentification, le gérant est redirigé dans cette interface contenant le profil de la salle où il peut modifier ses images, la description, consulter les avis, ses séances ainsi que les modifier ou les supprimer. L'ajout d'une séance se fait par un remplissage de formulaire en suivant les champs à remplir : nom, description, heures de services, nombre maximal des adhérents par session, assistance et prix.

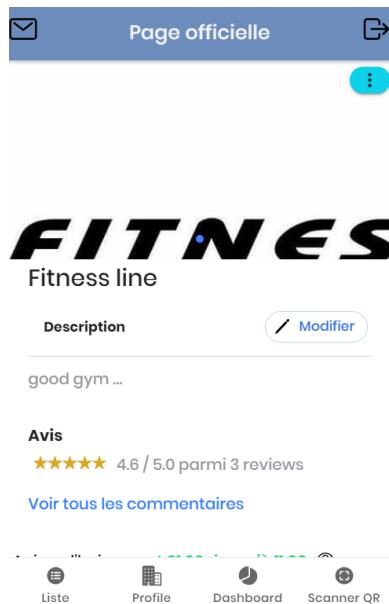


Figure 4.28: Page officielle vue par le gérant

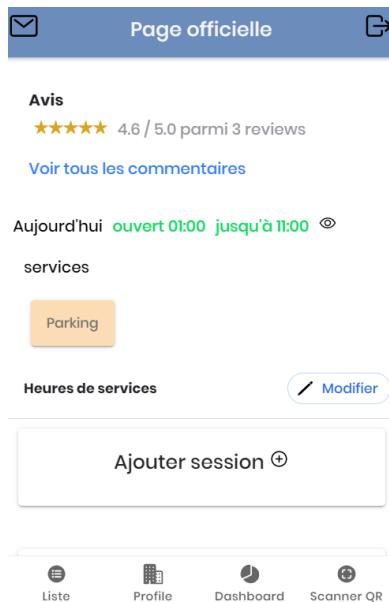


Figure 4.29: Page officielle contentant l'option d'ajout des sessions

4.3.3.2 Interface Statistiques

Le gérant peut consulter ses statiques dynamiques, cette interface contient les revenus générés par l'application, le pourcentage d'activité horaire du centre, le pourcentage des âges et le pourcentage du genre des clients. Le gérant peut choisir le mois et l'année correspondants et afficher les dashboards comme il est possible d'afficher le total.



Figure 4.30: Interface des statistiques dynamiques

4.3.3.3 Interface Scanner QR code

Le gérant scanne le QR code du client à l'aide de cette interface qui génère des tests unitaires afin de vérifier son état. L'interface affiche alors un message spécifique pour toute erreur : date invalide, identifiant invalide, Inexistence du code ou bien code déjà utilisé.

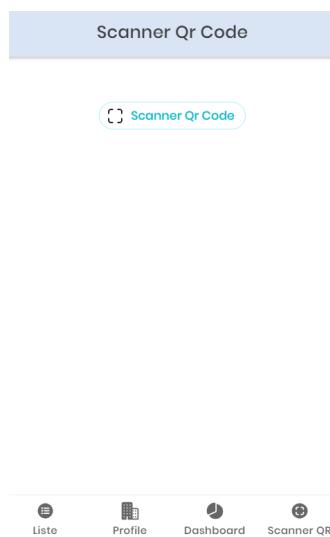


Figure 4.31: Interface de scan des codes QR

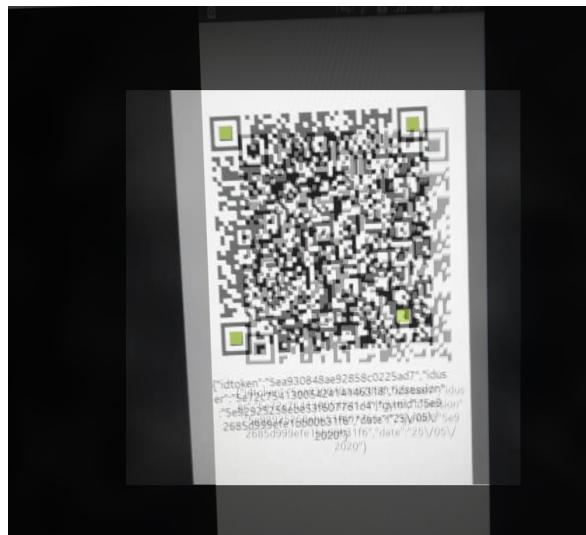


Figure 4.32: Process réel de scan exécuté sur mobile

4.3.3.4 Interface Liste des adhérents

Le gérant peut consulter la liste des adhérents en choisissant la date. La liste contient la photo de profil de l'utilisateur ainsi que son nom, la séance visitée et son horaire.

Liste des adhérents		
Adhérent info	Session	Début & Fin
Adam Abidi	Musculation	06:00 18:00
Fathi el haddaoui	Cardio	06:00 20:00
Saif bouchnek	Fitness	06:00 22:00
Kamilia miaoui	Yoga	06:00 20:00
Sinda chikhaoui	Fitness	06:00 19:00

Changer date
17/07/2020

Liste
 Profile
 Dashboard
 Scanner QR

Figure 4.33: Liste des adhérents pour une date fixée

4.3.4 Scénario : utilisation de l'application par l'administrateur

4.3.4.1 Interface validations centres

Dans cette interface, l'administrateur valide les centres en attente suite à leur inscription ou bien bloqués auparavant, suite à leur validation, les centres peuvent utiliser l'application et seront visibles aux différents adhérents.

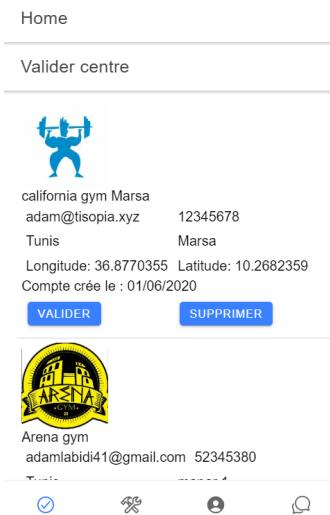


Figure 4.34: Interface de validation des centres

4.3.4.2 Interface blocage des centres

A travers cette interface, l'administrateur a la possibilité de bloquer les centres validés. Suite à cette action, les centres ne peuvent plus interagir avec l'application et ne sont plus visibles.

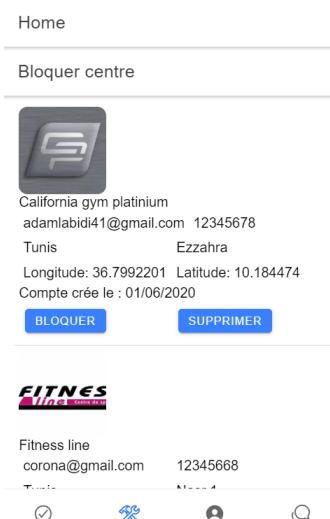


Figure 4.35: Interface de blocage des centres

4.3.4.3 Interface gestions des adhérents

Dans cette interface, l'administrateur gère les adhérents en les bloquant ou validant leurs profils ou même en les supprimant définitivement. Dans le cas d'une suppression définitive, l'utilisateur sera toujours présent dans la base de données et sera donc en quelque sorte Blacklisté, il ne pourra plus recréer son profil avec les mêmes identifiant ni en utilisant la même adresse mail

Home

Gestion clients

Saif bouchnek
saif@gmail.com tel : 45678934
CIN : 737373737

BLOQUER **SUPPRIMER**

Fathi el haddaoui
fathi@gmail.com tel : 24567898
CIN : 12345677

BLOQUER **SUPPRIMER**

🔍 🔁 🙁 💬

Figure 4.36: Interface de gestion des adhérents

4.3.4.4 Interface gestions des avis

Dans cette interface, l'administrateur consulte les avis des adhérents, il a la possibilité de supprimer l'avis, bloquer l'utilisateur ou le supprimer.

Home

gestion commentaires

Adam Abidi
Bonne expérience

SUPPRIMER COMMENTAIRE

BLOQUER **SUPPRIMER CLIENT**

Adam Abidi
Je recommande ce studio

SUPPRIMER COMMENTAIRE

BLOQUER **SUPPRIMER CLIENT**

🔍 🔁 🙁 💬

Figure 4.37: Interface de gestion des commentaires

4.4 Conclusion

Dans ce chapitre, nous avons présenté les technologies et les outils utilisés dans notre système. Nous avons par la suite détaillé la mise en œuvre de la solution en illustrant des captures d'écran des différentes interfaces réalisées en expliquant les différents scénarios possibles.

Conclusion générale

L'avènement des plateformes digitales et leur ascension continue en raison des divers avancées technologiques ont fait d'elles des acteurs économiques incontournables.

En effet, ces plateformes ont su s'adapter aux besoins et aux exigences des consommateurs en leur proposant des services adaptés à leurs préférences et en leur conférant le pouvoir d'interagir afin d'aider à mieux synthétiser la masse de données en expansion continue.

C'est dans ce cadre que s'oriente le travail réalisé et dont le but a été de mettre en œuvre une plateforme digitale mobile orientée vers le sport, régissant des technologies récentes et destinée à remplacer la démarche classique avec un système de réservation dynamique et offrant à la fois le principe « Pay As You Go » mais aussi un système de recommandation personnalisé des centres de remise en forme.

Cette application a eu pour objectif d'aider les utilisateurs à choisir le centre le plus adéquats à leurs contraintes en intégrant d'une part un système de géolocalisation, en leur permettant d'interagir avec l'application via un système de notation mais aussi en intégrant un principe de dématérialisation des abonnements via un module QR-CODE.

Nous avons suivi une méthode Agile afin de résoudre le problème de découverte tardive des anomalies de conception et satisfaire au mieux le client en assurant une gestion de projet plus facile. Le projet a ainsi été découpé en plusieurs sous tâches afin d'avoir le plus de transparence possible d'une part mais surtout un meilleur contrôle de la qualité des développements effectués. Nous avons ainsi débuté par la présentation du contexte du projet et des différentes problématiques constatées, nous avons étudié l'existant et les objectifs de notre solution, nous avons par la suite identifié les différents acteurs et surtout les différents besoins fonctionnels et non fonctionnels pour aboutir à une étude conceptuelle.

Nous avons ensuite fait l'étude des différents systèmes de recommandations, leurs principes de fonctionnement, avantages et inconvénients ainsi que leurs évaluations et mis en exergue la démarche adoptée.

Enfin, nous avons clôturé le présent travail par la phase de réalisation et l'implémentation ainsi que la présentation des outils techniques, l'architecture matérielle et logicielle de l'application et la présentation des différents scenarios décrit par des captures d'écran.

Pour conclure, il est important de considérer que l'application Spory est toujours en cours de développement, notre travail reste donc ouvert à plusieurs améliorations et extensions.

Conclusion générale

Autre point qui n'est pas des moindres et qui concerne le paiement en ligne en Tunisie, ce dernier demeure une tache nécessitant d'autres ressources extérieures d'où l'intérêt de mettre en œuvre un algorithme qui consommera une API spécifique, disponible dans notre pays et qui ne nuirait pas à la stabilité de notre application.

Autres pistes d'améliorations sont prévues comme la mise en place d'une messagerie qui permettrait aux divers adhérents d'interagir entre eux mais qui leur permettrait aussi de passer des réclamations directement aux divers gérants ou même à l'administrateur de la plateforme.

Il serait tout aussi important de mettre au point un système de fidélisation qui encouragerait les amateurs de rester actifs tout en utilisant notre application.

Enfin, nous avons prévu de préparer un suivi digital de nutrition couplé avec un second système de recommandations de produits alimentaires et la possibilité d'une intervention personnelle d'un nutritionniste.

Nous aurons ainsi une plateforme Digitale qui couvrirait réellement tous les besoins afin de garantir à tous une remise en forme optimisée, simple et pratique.

Bibliographie & Netographie

- [1] F. SCHEID, *Le marketing digital*. EYROLLES, 2019.
- [2] S. KEMP. (Feb, 2020). Digital 2020 : April Global Statshot, adresse : <https://datareportal.com/reports/digital-2020-april-global-statshot>.
- [3] ——, (Feb, 2020). Digital 2020 : Tunisia, adresse : <https://datareportal.com/reports/digital-2020-tunisia>.
- [4] WIKIPEDIA. (). Digital marketing, adresse : https://en.wikipedia.org/wiki/Digital_marketing.
- [5] ——, (). Webmarketing, adresse : <https://fr.wikipedia.org/wiki/Webmarketing>.
- [6] D. CHAFFEY. (Apr, 2020). What is Digital Marketing ? A visual summary., adresse : <https://www.smartinsights.com/digital-marketing-strategy/what-is-digital-marketing/>.
- [7] L. BLOCH. (). Méthodes de conception, de spécification et de réalisation, adresse : <https://www.laurentbloch.org/Data/SI-Projets-extraits/livre008.html>.
- [8] T. THIRY, *Les pratiques de l'équipe agile*. Collection Informatique, 2019.
- [9] (). La conduite de projet, ou gestion de projet, adresse : <http://www.gestion-projet-informatique.vivre-aujourd'hui.fr/conduite-de-projet.html>.
- [10] C. DRUMOND. (). Scrum, adresse : <https://www.atlassian.com/agile/scrum>.
- [11] K. BECK, *Extreme Programming Explained : Embrace Change*. Addison Wesley, 2004.
- [12] K. SUTTON. (Jun. 2017). Extreme Programming Explained, adresse : <https://kellysutton.com/2017/06/20/extreme-programming-explained.html>.
- [13] P. BERGOUGNOUX, *Modélisation Conceptuelle de Données : Une Démarche Pragmatique*. Bergougnoux Pat, 2019.
- [14] H. BERSINI, *La programmation orientée objet - cours et exercices en UML 2 avec Java 6, C4, C++, Python, PHP 5 et LinQ*. Eyrolles, 2011.
- [15] (). StarUML documentation, adresse : <https://docs.staruml.io/>.
- [16] H. DENG. (Feb, 2019). Recommender Systems in Practice : How companies make product recommendations, adresse : <https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a>.

- [17] J. LE. (Oct, 2019). Recommendation System Series Part 2 : The 10 Categories of Deep Recommendation Systems That Academic Researchers Should Pay, adresse : <https://towardsdatascience.com/recommendation-system-series-part-2-the-10-categories-of-deep-recommendation-systems-that-189d60287b58>.
- [18] E. NEGRE. (Nov, 2018). Les systèmes de recommandation : une catégorisation, adresse : <https://interstices.info/les-systemes-de-recommandation-categorisation/>.
- [19] ——, *Systèmes de recommandation : Introduction*. iSTE Editions, 2015.
- [20] WIKIPEDIA. (). Cosine similarity, adresse : https://en.wikipedia.org/wiki/Cosine_similarity.
- [21] S. PRABHAKARAN. (Oct, 2018). Cosine Similarity – Understanding the math and how it works, adresse : <https://www.machinelearningplus.com/nlp/cosine-similarity/>.
- [22] WIKIPEDIA. (). NoSQL, adresse : <https://en.wikipedia.org/wiki/NoSQL>.
- [23] M. ROUSE. (Oct, 2011). NoSQL (Not Only SQL database), adresse : <https://searchdatamanagement.techtarget.com/definition/NoSQL-Not-Only-SQL>.
- [24] S. YEGULALP. (Dec, 2017). What is NoSQL? Databases for a cloud-scale future, adresse : <https://www.infoworld.com/article/3240644/what-is-nosql-databases-for-a-cloud-scale-future.html>.
- [25] MONGODB. (). The database for modern applications, adresse : <https://www.mongodb.com/>.
- [26] TUTORIALSPPOINT. (). MongoDB Tutorial, adresse : <https://www.tutorialspoint.com/mongodb/index.htm>.
- [27] MANGODB. (). MongoDB Cloud, adresse : <https://www.mongodb.com/cloud>.
- [28] M. H. CHAHINE. (Feb, 2020). Intégration, Livraison et Déploiement continu, adresse : <https://latavernedutesteur.fr/2020/02/24/livre-cftl-integration-livraison-et-deploiement-continu/>.
- [29] GIT. (), adresse : <https://git-scm.com/>.
- [30] GITHUB. (), adresse : <https://github.com>.
- [31] HEROKU. (). Heroku : Cloud Application Platform, adresse : <https://www.heroku.com/>.
- [32] I. ACADEMY. (Mar, 2019). How to Pass Data with Angular Router in Ionic [v4], adresse : <https://ionicframework.com/>.

Bibliographie & Netographie

- [33] IONIC. (). Ionic - Cross-Platform Mobile App Development, adresse : <https://ionicframework.com/>.
- [34] NODE.JS. (Jun. 2017), adresse : <https://nodejs.org/en/>.
- [35] PYTHON. (). Python For Beginners, adresse : <https://www.python.org/>.
- [36] D. MITCHINSON. (). Python Machine Learning Tutorial, adresse : https://www.pythontutorial.eu/machine_learning.php.

Résumé

Ce projet intitulé "Mise en place d'une application mobile intégrant le principe «Pay As You Go» et d'un module de recommandation interactif" rentre dans le cadre du projet de fin d'études qui vient de conclure notre formation d'ingénieur à la faculté des sciences de Tunis. Cette idée fait partie des besoins quotidiens, l'entreprise WeCode Land a pris la décision de digitaliser les abonnements des centres de manière dynamique et mobile. L'application concerne une migration numérique des réservations statiques aux réservations dynamiques certifiées par un QR code et accompagnée par un système de recommandations basé sur l'intelligence artificielle.

Mots clés : Système de recommandations, Mobile, Dynamique, Intelligence artificielle, Code QR

Abstract

This project entitled "Implementation of a mobile application integrating the «Pay As You Go» principle and a interactive recommendation" is part of the end of studies project which has just concluded our engineering training at the Faculty of Sciences of Tunis. This idea is part of the society daily needs, the company WeCode Land has made the decision to digitize the subscriptions of the centers in a dynamic and mobile way. The application concerns a digital migration from static reservations to dynamic reservations certified by a QR code and accompanied by a recommender system based on artificial intelligence.

Keywords : Recommendation system, Mobile, Dynamics, Artificial intelligence, QR Code