

---

# APPLICATION DE SUIVI FITNESS

---

Document de Conception



## 1- Contents

2-	Identification .....	1
2.1	Identification du document.....	1
2.2	Historique des changements.....	1
3-	Contexte .....	1
3.1	Présentation du cadre du projet .....	1
3.2	Objectifs du document.....	1
3.3	Portée du document .....	2
3.4	Public cible .....	2
4-	Analyse et Spécification des besoins .....	2
4.1	Introduction .....	2
4.2	Objectifs du projet .....	2
4.3	Parties prenantes .....	2
4.4	Contraintes et exigences spécifiques .....	2
4.5	Analyse des besoins utilisateurs .....	3
4.6	Fonctionnalité de FitnessTrack .....	3
4.7	Besoins fonctionnels et non fonctionnels .....	3
4.7.1	Besoin fonctionnel .....	3
4.7.2	Besoin non fonctionnels .....	3
4.8	Contraintes et limitations.....	3
4.8.1	Limitations liées aux capacités des appareils .....	3
4.8.2	Restrictions budgétaires pour les ressources serveur .....	4
4.9	Utilisateurs et Rôles .....	4
4.9.1	Utilisateurs.....	4
4.9.2	Administrateurs .....	4
5-	Modélisation et conception.....	4
5.1	Modélisation conceptuelle.....	4
5.2	Modélisation interactive .....	5
5.3	Identification des cas d'utilisation .....	5
5.3.1	Description textuelle des cas d'utilisation .....	5
5.3.2	Diagramme de cas d'utilisation .....	8
6-	Conception du système .....	10
6.1	Introduction .....	10
6.2	Modélisation structurelle.....	10
6.3	Modélisation comportementale .....	11
6.4	Conception de la base de données .....	11
6.4.1	Introduction .....	11
6.4.2	Dictionnaire de données.....	11

6.4.3	Modèle conceptuel de données(MCD)	13
6.4.4	Modèle logique de données(MLD)	14
7-	Réalisation et mise en œuvre	14
7.1	Architecture du système	14
7.1.1	Principe d'architecture et conception	14
7.1.2	Phases de déploiement	14
7.1.3	Architecture globale du système	15
7.2	Mise en ligne	16
7.3	Validation	16
7.4	Langages utilisés	16
7.5	Outils et logiciels utilisés	17
7.6	Conclusion	17

# Application de suivi fitness

## 2- Identification

---

### 2.1 Identification du document

Projet :	FitnessTrack
Version du projet :	1.0
Version du document :	1.0
Sécurité du document :	Public
Date de création :	03/09/2024
Par :	NOEL Rockson

### 2.2 Historique des changements

Qui	Quand	Quoi
NOEL Rockson	03/09/2024	Création du document

## 3- Contexte

---

Avec l'augmentation de l'intérêt pour la santé et le bien-être, de nombreuses personnes se tournent vers des applications mobiles pour les aider à suivre leur condition physique. Cependant, la plupart des applications disponibles manquent de fonctionnalités sociales ou ne permettent pas une personnalisation suffisante des objectifs. FitnessTrack comble ces lacunes en offrant une plateforme conviviale qui combine le suivi de la condition physique avec des fonctionnalités sociales, permettant ainsi aux utilisateurs de rester motivés et engagés.

### 3.1 Présentation du cadre du projet

L'application FitnessTrack est conçue pour répondre au besoin croissant des utilisateurs de suivre leurs activités physiques, de définir et de suivre des objectifs de fitness, et de visualiser leurs progrès au fil du temps. L'application vise à offrir une expérience fluide et engageante, tout en restant accessible à un large public via des versions web et mobile.

### 3.2 Objectifs du document

Ce document a pour objectif de :

- Détailler les besoins fonctionnels et non fonctionnels de l'application FitnessTrack.
- Présenter la conception technique de l'application, y compris l'architecture logicielle, la base de données, et l'intégration des services tiers.
- Servir de référence pour les développeurs, les gestionnaires de projet, et les autres parties prenantes impliquées dans le projet.

### 3.3 Portée du document

Le document couvre l'ensemble du cycle de vie de développement de l'application FitnessTrack, depuis l'analyse des besoins jusqu'à la mise en œuvre technique, en passant par la conception de la base de données et l'architecture du système.

### 3.4 Public cible

Ce document est destiné à :

- **Développeurs** : Pour comprendre les spécifications techniques et les exigences de l'application.
- **Gestionnaires de projet** : Pour superviser le développement et s'assurer que les objectifs du projet sont atteints.
- **Testeurs** : Pour identifier les critères de validation et s'assurer que le produit final répond aux attentes.

## 4- Analyse et Spécification des besoins

---

### 4.1 Introduction

Cette section présente une analyse détaillée des besoins fonctionnels et non fonctionnels de l'application FitnessTrack, en se basant sur les attentes des utilisateurs finaux et les contraintes techniques identifiées.

### 4.2 Objectifs du projet

- **Suivi des activités** : Permettre aux utilisateurs de suivre leurs séances d'entraînement, incluant le type d'exercice, la durée, et l'intensité.
- **Personnalisation des objectifs** : Les utilisateurs peuvent définir des objectifs spécifiques (ex : perdre du poids, améliorer l'endurance) et suivre leur progression.
- **Analyse des performances** : Fournir des rapports et des visualisations des données pour aider les utilisateurs à comprendre leurs performances et à ajuster leur plan d'entraînement.
- **Fonctionnalités sociales** : Offrir des options pour partager les progrès avec des amis, participer à des défis, et recevoir des encouragements.

### 4.3 Parties prenantes

- **Utilisateurs finaux** : Amateurs de fitness cherchant à suivre leurs activités physiques.
- **Développeurs** : Responsables de la création et du maintien de l'application.
- **Gestionnaires de projet** : Chargés de superviser le développement et de s'assurer que le projet respecte les délais et les budgets.
- **Testeurs** : S'assurent que l'application fonctionne comme prévu et répond aux exigences des utilisateurs.

### 4.4 Contraintes et exigences spécifiques

- **Compatibilité multiplateforme** : L'application doit fonctionner sur Android, iOS, et les navigateurs web modernes.
- **Sécurité** : Les données des utilisateurs doivent être sécurisées, en particulier celles liées à l'authentification et aux informations personnelles.

- Performance : L'application doit charger rapidement et offrir une expérience utilisateur fluide, même sur des appareils à faible capacité.

#### 4.5 Analyse des besoins utilisateurs

- Inscription et connexion : Les utilisateurs doivent pouvoir s'inscrire via une adresse email, un numéro de téléphone, ou un compte social (Google, Facebook).
- Suivi des activités : L'utilisateur peut ajouter, modifier, et supprimer des séances d'entraînement. Chaque session doit être associée à des détails comme le type d'exercice, la durée, et les calories brûlées.
- Définition d'objectifs : L'utilisateur peut définir des objectifs à court et long terme (ex : courir 5 km en 30 minutes) et recevoir des notifications pour rester sur la bonne voie.
- Visualisation des statistiques : Offrir des graphiques et des tableaux pour que l'utilisateur puisse suivre ses progrès sur des périodes données.
- Notifications et rappels : Notifications push pour rappeler les séances d'entraînement ou signaler des objectifs atteints.

#### 4.6 Fonctionnalité de FitnessTrack

- Gestion des comptes utilisateurs : Créer, modifier, et supprimer des comptes.
- Suivi des activités : Ajouter, consulter, modifier, et supprimer des séances d'entraînement.
- Gestion des objectifs : Définir des objectifs et suivre leur progression.
- Notifications : Envoi de notifications push pour rappeler les séances d'entraînement et signaler les objectifs atteints.
- Fonctionnalités sociales : Partager des séances d'entraînement, participer à des défis, suivre les activités des amis.

#### 4.7 Besoins fonctionnels et non fonctionnels

##### 4.7.1 Besoin fonctionnel

- Enregistrement et connexion des utilisateurs : Authentification sécurisée avec Firebase.
- Gestion des séances d'entraînement : CRUD (Create, Read, Update, Delete) pour les séances.
- Suivi des objectifs : CRUD pour les objectifs de fitness.
- Affichage des statistiques : Génération de rapports et graphiques basés sur les données de l'utilisateur.
- Notifications : Gestion des notifications push pour rappel et motivation.

##### 4.7.2 Besoin non fonctionnels

- Performance : Le temps de réponse pour charger les données d'une séance d'entraînement ne doit pas dépasser 2 secondes.
- Sécurité : Les informations sensibles doivent être cryptées, et des mécanismes de sécurité tels que l'authentification à deux facteurs doivent être envisagés.
- Fiabilité : Le système doit avoir un taux de disponibilité supérieur à 99,5 %.
- Extensibilité : L'architecture doit permettre l'ajout de nouvelles fonctionnalités sans nécessiter de modifications majeures.

#### 4.8 Contraintes et limitations

##### 4.8.1 Limitations liées aux capacités des appareils

- Capteurs mobiles : L'application FitnessTrack repose en partie sur les capteurs des appareils mobiles (comme l'accéléromètre, le GPS, et le capteur de fréquence cardiaque). Cependant,

la précision de ces capteurs peut varier considérablement en fonction du modèle de l'appareil, ce qui peut affecter la précision des données enregistrées.

- **Stockage local** : La quantité de données pouvant être stockées localement sur les appareils mobiles est limitée, ce qui peut contraindre la capacité de l'utilisateur à accéder à l'historique complet des activités sans une connexion réseau active.
- **Compatibilité des appareils** : FitnessTrack doit être compatible avec une variété de dispositifs, notamment ceux fonctionnant sous Android et iOS. Cette diversité peut introduire des défis liés à la performance et à l'optimisation pour les appareils de faible capacité.

#### 4.8.2 Restrictions budgétaires pour les ressources serveur

- **Capacité des serveurs** : Les coûts associés au déploiement sur des infrastructures cloud peuvent limiter la capacité de traitement et de stockage initialement allouée, surtout pendant les phases de test bêta et les premières phases de déploiement. Il est possible que les performances du serveur soient affectées si le nombre d'utilisateurs augmente rapidement.
- **Scalabilité** : Bien que l'application soit conçue pour être scalable, les contraintes budgétaires pourraient limiter l'investissement dans des ressources de serveur supplémentaires ou dans des optimisations avancées, ce qui pourrait affecter la capacité à gérer une forte charge de trafic utilisateur.

### 4.9 Utilisateurs et Rôles

#### 4.9.1 Utilisateurs

- **Amateurs de fitness** : Les utilisateurs principaux de l'application FitnessTrack sont des individus cherchant à améliorer leur condition physique en suivant leurs activités, en fixant des objectifs personnels, et en surveillant leur progression. Ces utilisateurs peuvent être des sportifs occasionnels ou réguliers, et l'application doit s'adapter à leurs divers besoins et niveaux de compétence.

#### 4.9.2 Administrateurs

- **Gestion des utilisateurs** : Les administrateurs sont responsables de la gestion des comptes utilisateurs, incluant la création, la modification, la désactivation ou la suppression de comptes en cas de violation des conditions d'utilisation.
- **Gestion des contenus** : Ils veillent également à la qualité et à la pertinence des contenus partagés dans l'application, notamment les défis communautaires, les suggestions d'exercices, et les notifications globales. Les administrateurs peuvent également modérer les interactions sociales au sein de la plateforme pour assurer un environnement sûr et motivant pour tous les utilisateurs.

## 5- Modélisation et conception

---

### 5.1 Modélisation conceptuelle

Le contexte d'utilisation de FitnessTrack implique l'interaction de l'utilisateur avec l'application via une interface web ou mobile. Les principales interactions incluent la création de comptes, la gestion des séances d'entraînement, et l'affichage des progrès.

## 5.2 Modélisation interactive

Le modèle d'interaction du système de FitnessTrack est essentiel car il permet de comprendre comment les utilisateurs avec le Système de FitnessTrack, quels sont les besoins et les attentes des différents acteurs, et comment ces interactions se déroulent dans le système. Cela nous aide à définir les fonctionnalités de FitnessTrack de manière plus précise et à identifier les exigences spécifiques auxquelles il doit répondre.

## 5.3 Identification des cas d'utilisation

Chaque cas d'utilisation décrit une interaction spécifique que l'utilisateur peut avoir avec le système.

- **Cas d'utilisation 1 : Inscription**
  - **Description :** Un utilisateur crée un nouveau compte en fournissant une adresse email, un mot de passe, et d'autres informations pertinentes.
  - **Acteurs :** Utilisateur, Système d'authentification.
  - **Préconditions :** L'utilisateur ne doit pas déjà avoir un compte.
  - **Post-conditions :** Un nouveau compte utilisateur est créé et l'utilisateur est authentifié.
- **Cas d'utilisation 2 : Suivi d'une séance d'entraînement**
  - **Description :** L'utilisateur enregistre une nouvelle séance d'entraînement en spécifiant le type d'exercice, la durée, et d'autres détails.
  - **Acteurs :** Utilisateur, Base de données.
  - **Préconditions :** L'utilisateur doit être connecté.
  - **Post-conditions :** Les détails de la séance sont enregistrés dans la base de données et disponibles pour consultation ultérieure.
- **Cas d'utilisation 3 : Notification push pour rappeler les objectifs quotidiens ou hebdomadaires**
  - **Description :** L'utilisateur reçoit des notifications push pour lui rappeler ses objectifs quotidiens ou hebdomadaires, comme atteindre un certain nombre de pas, brûler un nombre spécifique de calories, ou compléter une activité programmée.
  - **Acteurs :** Utilisateur/Système de notification
  - **Préconditions :**
  - **Post-conditions :**
- **Cas d'utilisation 4 : Rappel pour les séances d'entraînement planifiées**
  - **Description :**
  - **Acteurs :**
  - **Préconditions :**
  - **Post-conditions :**
- **Cas d'utilisation 5 : Notifications pour les nouveaux badges ou récompenses obtenus**
  - **Description :**
  - **Acteurs :**
  - **Préconditions :**
  - **Post-conditions :**

### 5.3.1 Description textuelle des cas d'utilisation

CU-1 Inscription d'un utilisateur	
Acteur Principal	Utilisateur/Système d'authentification
Description	Un utilisateur souhaite créer un compte sur l'application FitnessTrack pour commencer à suivre ses activités de fitness. Il doit fournir des informations de base,



	telles que son nom, son adresse email, et un mot de passe. L'application lui offre également la possibilité de s'inscrire via un de ces comptes sociaux (Google ou Facebook).
Préconditions	L'utilisateur ne doit pas déjà avoir de compte sur l'application. L'application doit être connectée à internet pour vérifier l'email ou les informations de compte social
Scénario principal	<ul style="list-style-type: none"> <li>• L'utilisateur accède à la page d'inscription de l'application.</li> <li>• Il saisit son nom, son adresse email, et son mot de passe, ou choisit de s'inscrire via un compte social.</li> <li>• Le système vérifie que l'email ou les informations de compte social ne sont pas déjà associés à un compte existant.</li> <li>• Si l'email ou le compte social est valide et disponible, le système crée un nouveau compte utilisateur.</li> <li>• L'utilisateur reçoit un email de confirmation pour valider son compte.</li> <li>• Une fois le compte confirmé, l'utilisateur peut se connecter à l'application.</li> </ul>
Post-conditions	Une fois le nouveau compte utilisateur est créé et l'utilisateur est authentifié, l'utilisateur est redirigé vers le tableau de bord de l'application.

CU-2 Connexion d'un utilisateur	
Acteur Principal	Utilisateur/Système d'authentification
Description	Un utilisateur qui a déjà un compte souhaite se connecter à l'application pour accéder à ses données et fonctionnalités.
Préconditions	<ul style="list-style-type: none"> <li>• L'utilisateur doit avoir de compte sur l'application.</li> <li>• L'application doit être connectée à internet pour vérifier les informations d'identification.</li> </ul>
Scénario principal	<ul style="list-style-type: none"> <li>• L'utilisateur accède à la page de connexion de l'application.</li> <li>• Il saisit son adresse email et son mot de passe, ou choisit de se connecter via un compte social.</li> <li>• Le système vérifie les informations d'identification.</li> <li>• Si les informations sont correctes, l'utilisateur est authentifié.</li> <li>• L'utilisateur est redirigé vers le tableau de bord de l'application.</li> </ul>
Post-conditions	L'utilisateur est connecté à son compte et peut accéder à ses données et fonctionnalités.

CU-3 Suivi d'une séance d'entraînement	
Acteur Principal	Utilisateur/Base de données
Description	L'utilisateur souhaite enregistrer une nouvelle séance d'entraînement dans l'application, incluant le type d'exercice, la durée, et les calories brûlées.
Préconditions	<ul style="list-style-type: none"> <li>• L'utilisateur doit être connecté à l'application.</li> <li>• L'application doit être connectée à Internet pour sauvegarder les données dans la base de données (si le mode hors ligne n'est pas disponible).</li> </ul>
Scénario principal	<ul style="list-style-type: none"> <li>• L'utilisateur accède à la section "Ajouter une séance d'entraînement" dans l'application.</li> <li>• Il sélectionne le type d'exercice (ex : course, natation, musculation).</li> <li>• Il entre la durée de la séance et, si possible, les calories brûlées.</li> <li>• L'utilisateur valide les informations en appuyant sur le bouton "Enregistrer".</li> <li>• Le système enregistre la séance dans la base de données.</li> </ul>

	<ul style="list-style-type: none"> <li>• L'utilisateur reçoit une confirmation que la séance a été enregistrée avec succès.</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• La nouvelle séance d'entraînement est enregistrée et visible dans l'historique de l'utilisateur.</li> <li>• Les statistiques de l'utilisateur sont mises à jour en fonction de cette nouvelle séance.</li> </ul>

CU-4 Notifications push pour rappeler les objectifs quotidiens ou hebdomadaires	
Acteur Principal	Utilisateur/Système de notifications
Description	L'utilisateur reçoit des notifications push pour lui rappeler ses objectifs quotidiens ou hebdomadaires, comme atteindre un certain nombre de pas, brûler un nombre spécifique de calories, ou compléter une activité programmée.
Préconditions	L'utilisateur doit être connecté à l'application et avoir défini des objectifs. Les notifications push doivent être activées sur l'appareil de l'utilisateur.
Scénario principal	<ul style="list-style-type: none"> <li>• L'utilisateur définit des objectifs quotidiens ou hebdomadaires dans l'application (par exemple, marcher 10 000 pas par jour).</li> <li>• Le système planifie des notifications push en fonction des objectifs définis.</li> <li>• À des moments prédéfinis (ex : le matin pour les rappels quotidiens, le début de la semaine pour les rappels hebdomadaires), le système envoie une notification push à l'utilisateur.</li> <li>• La notification rappelle à l'utilisateur son objectif et peut inclure des messages motivants (ex : "Vous avez déjà fait 7 000 pas aujourd'hui, continuez !").</li> <li>• L'utilisateur voit la notification et peut cliquer dessus pour ouvrir l'application et consulter ses progrès.</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• L'utilisateur est rappelé de ses objectifs quotidiens ou hebdomadaires via une notification push.</li> <li>• La notification encourage l'utilisateur à atteindre ses objectifs.</li> </ul>

CU-5 Rappels pour les séances d'entraînement planifiées	
Acteur Principal	Utilisateur/Système de rappels
Description	L'utilisateur reçoit des rappels avant ses séances d'entraînement planifiées pour ne pas oublier de les effectuer.
Préconditions	<ul style="list-style-type: none"> <li>• L'utilisateur doit avoir planifié des séances d'entraînement dans l'application.</li> <li>• Les rappels doivent être activés sur l'appareil de l'utilisateur.</li> </ul>
Scénario principal	<ul style="list-style-type: none"> <li>• L'utilisateur planifie une séance d'entraînement pour une date et une heure spécifiques dans l'application.</li> <li>• Le système enregistre cette séance et planifie un rappel.</li> <li>• Un certain temps avant la séance (par exemple, 30 minutes avant), le système envoie un rappel à l'utilisateur via une notification push.</li> <li>• La notification informe l'utilisateur de la séance à venir et lui permet de l'ouvrir directement dans l'application.</li> <li>• L'utilisateur voit la notification et peut confirmer ou reporter la séance.</li> </ul>
Post-conditions	<ul style="list-style-type: none"> <li>• L'utilisateur reçoit un rappel pour la séance d'entraînement planifiée via une notification push.</li> </ul>



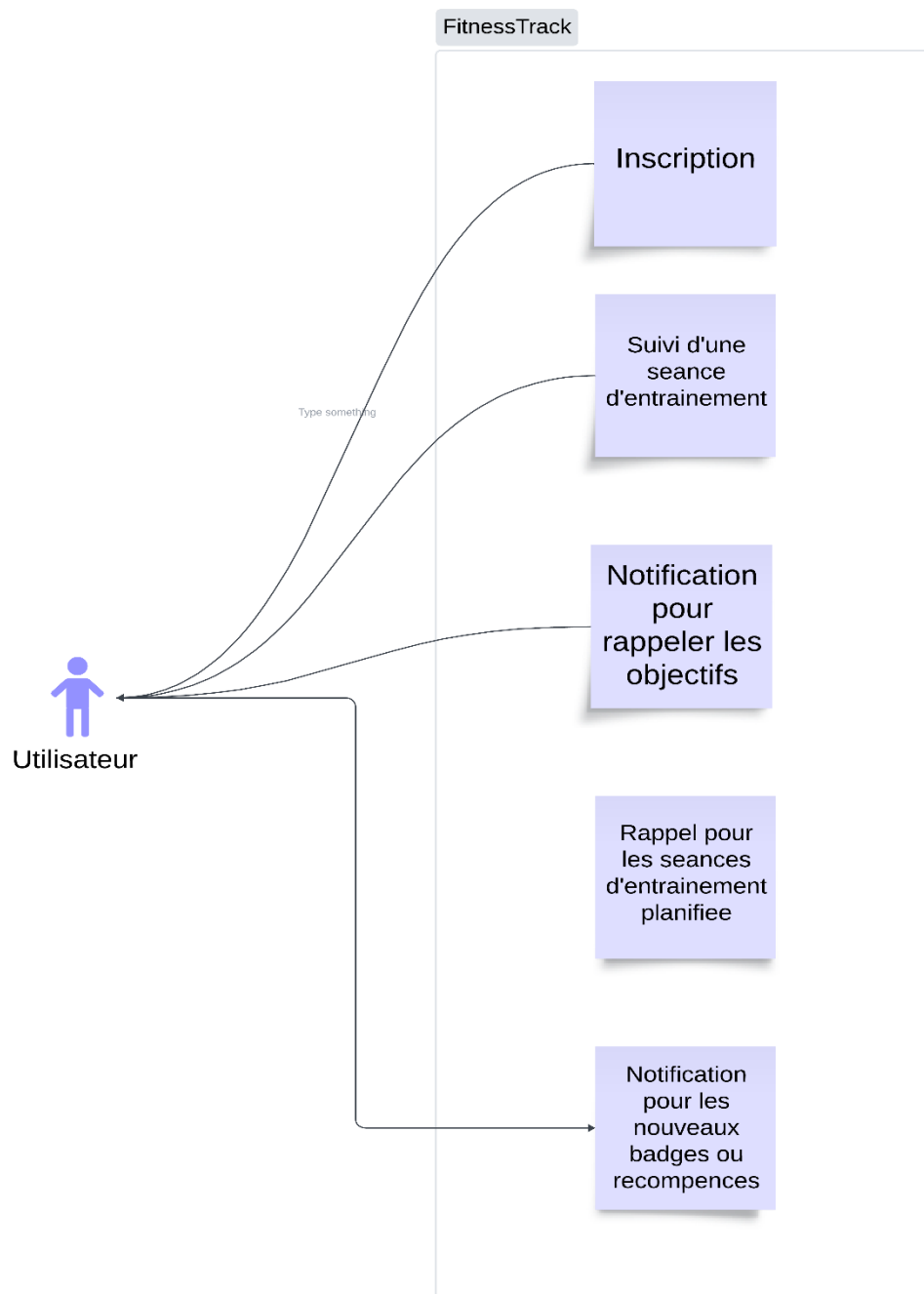


Figure 1 : Diagramme de cas d'utilisation de FitnessTrack

## Conclusion

L'analyse des besoins réalisée pour l'application FitnessTrack met en évidence les exigences critiques pour répondre aux attentes des utilisateurs, qui sont des amateurs de fitness désireux d'améliorer leur condition physique. En identifiant clairement les besoins fonctionnels et non fonctionnels, ainsi

que les contraintes et limitations techniques, nous avons défini les bases sur lesquelles l'application sera conçue et développée.

Les fonctionnalités clés, telles que le suivi des séances d'entraînement, la définition d'objectifs, la gestion des utilisateurs, et les notifications, ont été soigneusement étudiées pour garantir une expérience utilisateur fluide, engageante et motivante. Les parties prenantes et les rôles utilisateurs ont également été pris en compte pour assurer que chaque interaction avec l'application soit optimisée pour les différentes catégories d'utilisateurs.

Les contraintes techniques, telles que les limitations des appareils et les restrictions budgétaires, ont été intégrées dans notre approche pour concevoir une application à la fois performante et scalable, tout en respectant les réalités du marché et les attentes des utilisateurs. Cette analyse constitue donc un fondement solide pour la phase de conception du système, assurant que FitnessTrack répondra efficacement aux besoins de ses utilisateurs tout en respectant les contraintes définies.

Cette analyse exhaustive des besoins nous permet de nous engager dans la phase de conception avec une vision claire et détaillée des exigences à remplir, garantissant ainsi la réussite du projet FitnessTrack.

## 6- Conception du système

---

### 6.1 Introduction

La conception du système FitnessTrack repose sur l'architecture en couches, favorisant la séparation des responsabilités, la modularité, et la facilité de maintenance. Cette section détaille la structure globale du système, y compris la modélisation structurelle et comportementale, ainsi que la conception de la base de données. Chaque composant est conçu pour répondre aux besoins fonctionnels et non fonctionnels identifiés précédemment tout en respectant les contraintes techniques et budgétaires.

La modélisation joue un rôle essentiel dans le processus de développement de l'application de suivi fitness. Elle permet de représenter de manière abstraite et structurée les différentes facettes du système, ce qui facilite sa compréhension, sa conception et sa mise en œuvre.

### 6.2 Modélisation structurelle

La modélisation structurelle du système décrit l'organisation des composants logiciels en couches distinctes :

- **Couche de présentation (Frontend)** : Cette couche comprend l'interface utilisateur (UI) de l'application, disponible en versions web et mobile. Elle est responsable de la capture des interactions des utilisateurs et de la présentation des données sous forme visuelle. Les technologies utilisées incluent HTML5, CSS3, JavaScript, et des Framework comme React/Angular pour le web, et flutter pour le mobile.
- **Couche de logique métier (Backend)** : Cette couche implémente les règles métier de l'application, gère les interactions entre les utilisateurs et les données, et applique les logiques de traitement nécessaires. Elle est développée en Node.js avec Express.js pour

gérer les requêtes HTTP, et est connectée à la base de données pour stocker et récupérer les informations nécessaires.

- **Couche de données** : Cette couche est responsable de la gestion des données de l'application. Elle inclut la base de données NoSQL (Firestore), où sont stockées les informations relatives aux utilisateurs, aux séances d'entraînement, aux objectifs, et aux notifications. Firestore est également utilisé pour la gestion des authentifications et des notifications push.
- **Couche de services** : Elle comprend l'intégration de services externes comme Firebase pour l'authentification, les notifications, et le stockage de fichiers. Elle gère également les interactions avec des API tierces, si nécessaire.

## 6.3 Modélisation comportementale

La modélisation comportementale décrit le fonctionnement interne du système en termes d'interactions entre les composants et les utilisateurs :

- **Diagrammes de séquence** : Ces diagrammes illustrent les interactions chronologiques entre les différents composants du système pour accomplir des tâches spécifiques comme l'inscription, la connexion, le suivi des séances d'entraînement, et l'envoi de notifications push. Chaque séquence montre comment les messages sont échangés entre la couche de présentation, la couche de logique métier, et la couche de données pour réaliser une fonction.
- **Diagrammes d'activités** : Ils détaillent les processus métier, comme la gestion des objectifs de fitness ou la planification des séances d'entraînement, en représentant les différentes étapes du processus et les décisions prises à chaque point.

## 6.4 Conception de la base de données

### 6.4.1 Introduction

La base de données est conçue pour stocker et gérer les données de l'application de manière efficace et scalable. Elle doit supporter les opérations courantes telles que l'enregistrement des utilisateurs, le suivi des activités de fitness, et la gestion des objectifs et récompenses.

### 6.4.2 Dictionnaire de données

Voici les différentes collections utilisés pour la conception de notre base de données (MongoDB) :

- Collection 'Users'

Attribut	Type	Description
id	UUID	Identifiant unique de l'utilisateur (PK)
username	String	Nom du user
email	String	Adresse email du user
password_hash	String	Hachage du mot de passe pour l'authentification
created_at	DateTime	Date de création du compte utilisateur

Workouts	Array of Objects	Listes des séances d'entraînement associés a l'utilisateur
Goals	Array of Objects	Liste des objectifs de fitness définis par l'utilisateur
Notifications	Array of Objects	Liste des notifications envoyées a l'utilisateur
badges	Array of Objects	Liste des badges obtenus par l'utilisateur

- Collection 'Workouts' (Embarquée dans 'Users')

Attribut	Type	Description
Workout_id	UUID	Identification unique de la séance d'entraînement
Workout_type	String	Type d'entraînement (ex. natation, course)
Duration	String	Durée de la séance (minutes)
Calories_burned	Integer	Nombre de calories brûlées pendant la séance
Workout_date	Date	Date à laquelle la séance a lieu

- Collection 'Goals' (Embarquée dans 'Users')

Attribut	Type	Description
Goal_id	UUID	Identification unique de l'objectif
Goal_type	String	Type d'objectif (ex. poids à perdre ou distance à courir)
Target_value	Integer	Valeur cible de l'objectif (e. 15 kg, 3 KM)
Deadline	Date	Date limite pour atteindre l'objectif
achieved	Boolean	Statut indiquant si l'objectif a été atteint

- Collection 'Notifications' (Embarquée dans 'Users')

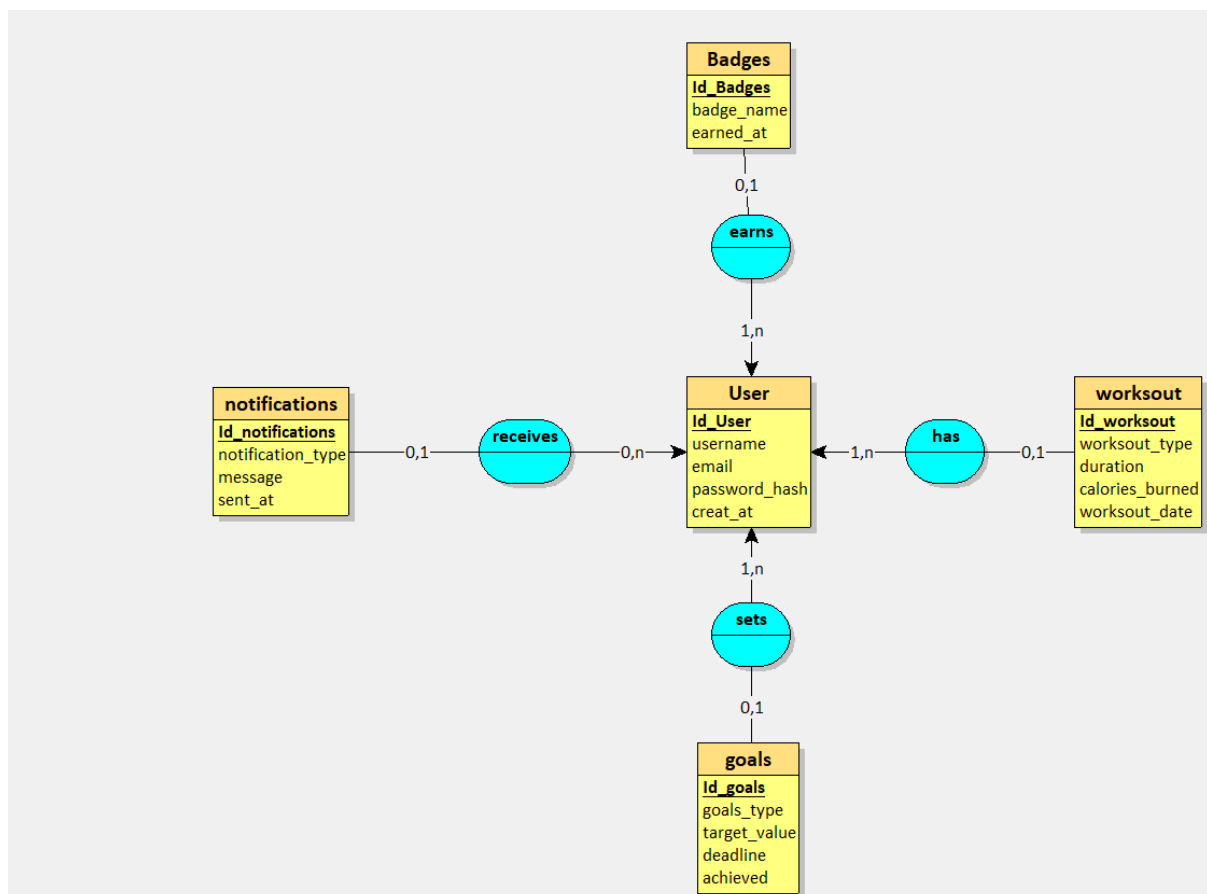
Attribut	Type	Description
Notification_id	UUID	Identification pour les notifications
Notification_type	String	Type de notification (ex. rappel, récompense)
Message	Long String	Message de la notification
Sent_at	DateTime	Date et heure d'envoi de la notification

- Collection 'Badges' (Embarquée dans 'Users')

Attribut	Type	Description
Badge_id	UUID	Identification unique pour le badge
Badge_name	String	Nom du badge
Earned_at	Date	Date à laquelle le badge a été obtenu

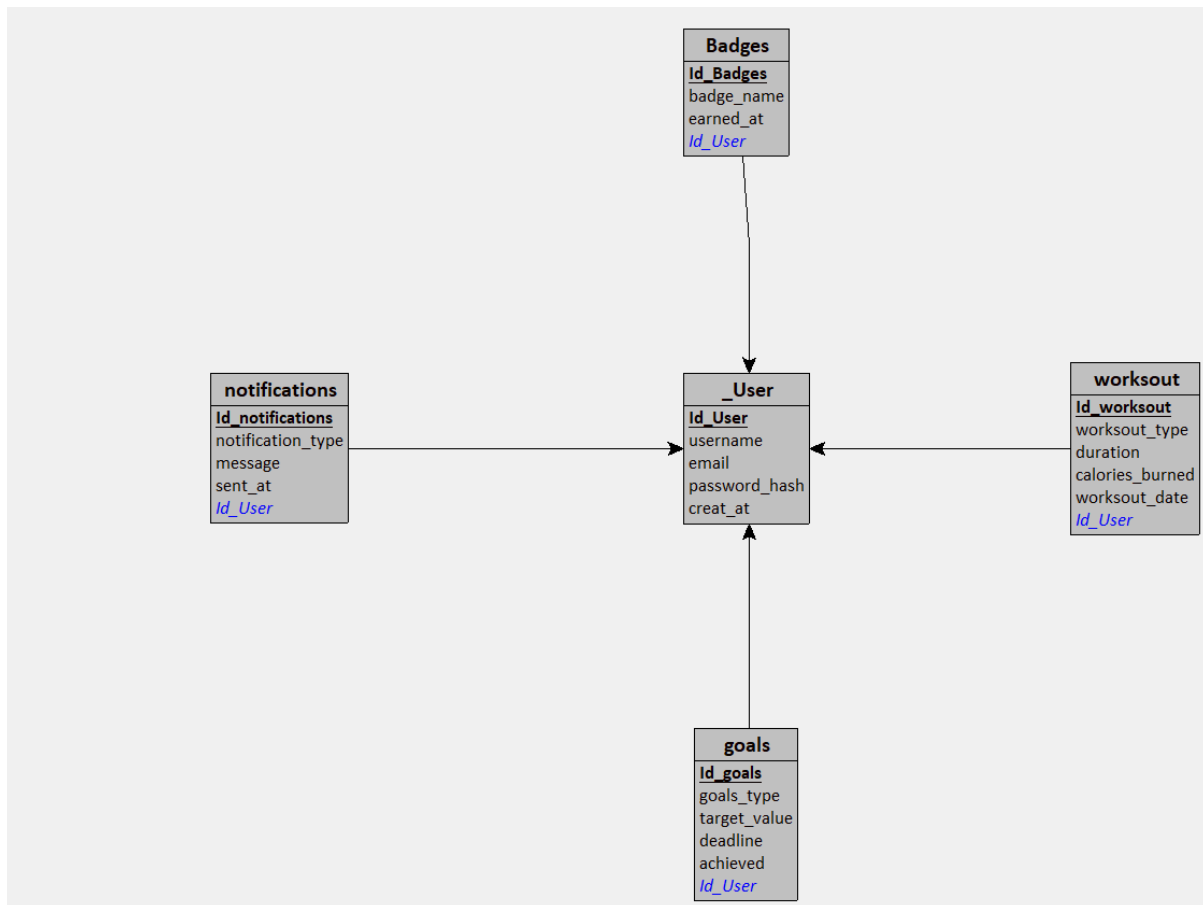
- Les 'id' seront automatiquement générés par MongoDB et agiront comme clé primaire pour chaque document.
- Les collections 'Workouts', 'Goals', 'Notifications' et 'badges' sont des documents embarqués dans la collection 'Users', ce qui permet de récupérer toutes les informations liées à un utilisateur en une seule requête.
- Les **type de données** comme 'String', 'Integer', 'Boolean' etc. sont utilisés pour définir la nature des informations stockées

#### 6.4.3 Modèle conceptuel de données(MCD)





#### 6.4.4 Modèle logique de données(MLD)



## 7- Réalisation et mise en œuvre

### 7.1 Architecture du système

#### 7.1.1 Principe d'architecture et conception

L'architecture de **FitnessTrack** est conçue pour offrir une solution cohérente et performante à la fois pour les plateformes web et mobiles. Elle repose sur une structure en couches pour séparer les responsabilités et faciliter la maintenance et l'évolution du système. Les principes de conception incluent l'utilisation de technologies modernes, la garantie de la performance, et la scalabilité.

#### 7.1.2 Phases de déploiement

Le déploiement du système est prévu en plusieurs phases :

- **Phase 1** : Déploiement initial des fonctionnalités de base (authentification, suivi des séances d'entraînement, objectifs).
- **Phase 2** : Ajout des notifications et rappels, ainsi que la gestion des récompenses.
- **Phase 3** : Intégration des fonctionnalités sociales et de partage.

### 7.1.3 Architecture globale du système

L'architecture de **FitnessTrack** est divisée en plusieurs composants clés :

#### 1) Frontend Mobile

- **Technologie** : Développé avec **Flutter** pour une expérience utilisateur cohérente sur iOS et Android.
- **Communication** : Le front-end mobile communique avec le back-end via des **API REST** pour récupérer et envoyer des données.
- **Responsabilités** : Affichage des données utilisateur, suivi des activités, gestion des objectifs, et interaction avec les services back-end.

#### 2) Frontend Web

- **Technologie** : Développé avec **React.js** ou **Angular** pour créer une interface web dynamique et réactive.
- **Communication** : Utilisation des **API REST** pour l'échange de données avec le back-end.
- **Responsabilités** : Fournir une interface utilisateur pour le suivi des activités, la gestion de la nutrition, l'analyse des progrès, et les fonctionnalités sociales.

#### 3) Backend

- **Technologie** : Serveur basé sur **Node.js** avec **Express.js** pour gérer les requêtes HTTP et la logique métier.
- **API** : Fourniture d'**API REST** pour permettre la communication entre les front-end mobile et web et le back-end.
- **Responsabilités** : Traitement des demandes des utilisateurs, gestion des données, authentification, et interactions avec les bases de données.

#### 4) Base de Données

- **Firestore** : Utilisé pour les fonctionnalités nécessitant une synchronisation en temps réel et des mises à jour instantanées.
- **MongoDB** : Employé pour stocker des données flexibles, offrant une solution évolutive et adaptable aux différents types de données.
- **Responsabilités** : Stockage et gestion des données utilisateur, des activités, des objectifs, et des historiques.

#### 5) Sécurité

- Les communications entre les composants sont sécurisées via HTTPS, et l'authentification des utilisateurs est renforcée avec des tokens JWT.

#### 6) Services Supplémentaires

- **Firebase** :
  - **Authentification** : Gestion des utilisateurs, intégration avec divers fournisseurs d'identité (email, réseaux sociaux).

- **Notifications** : Envoi de notifications push pour tenir les utilisateurs informés des mises à jour importantes et des rappels.
- **Stockage** : Gestion des fichiers comme les photos et les vidéos, avec une intégration transparente pour les utilisateurs.
- 

## 7.2 Mise en ligne

La mise en ligne de FitnessTrack nécessite de relever plusieurs défis et contraintes. Il faut assurer la sécurité des données des utilisateurs grâce à des protocoles de cryptage et d'authentification solides, garantir la performance et l'évolutivité de l'application pour répondre à une utilisation croissante, et maintenir une disponibilité continue grâce à une planification minutieuse de la maintenance. La compatibilité avec différents appareils et navigateurs, la gestion efficace de la base de données, les mises à jour régulières pour ajouter des fonctionnalités et corriger les bugs, ainsi que la prise en compte des coûts associés à l'hébergement et à la conformité légale sont autant d'éléments cruciaux. La mise en ligne du système sera réalisée par étapes, en commençant par un déploiement bêta pour un groupe d'utilisateurs restreint, suivi d'une mise en production complète une fois les tests finaux réussis.

## 7.3 Validation

La validation de FitnessTrack revêt une importance capitale dans le processus global de développement et de déploiement du système. Elle vise à garantir que l'application FitnessTrack répond aux exigences fonctionnelles et techniques préalablement définies, tout en assurant sa fiabilité, sa performance et sa convivialité. La validation du SNGIA comprend plusieurs étapes clés, telles que les tests unitaires pour évaluer les fonctionnalités individuelles, les tests d'intégration pour vérifier l'interaction entre les différents modules, Réalisation et mise en œuvre et les tests de système pour évaluer l'ensemble du système dans un environnement similaire à celui de production.

Les tests de performance et de charge permettent d'évaluer la réactivité du système face à une utilisation intensive, tandis que les tests de sécurité s'assurent de la protection des données contre les menaces potentielles. La validation implique également la participation des utilisateurs finaux dans des scénarios réels, permettant de recueillir des retours d'expérience et d'identifier d'éventuelles améliorations. Une fois que l'application a passé avec succès toutes les étapes de validation, il est prêt à être déployé et mis à la disposition des utilisateurs pour une utilisation effective et bénéfique. Chaque fonctionnalité sera testée individuellement avant l'intégration, et un ensemble de tests finaux (tests fonctionnels, tests de charge, tests de sécurité) sera effectué avant le déploiement final.

## 7.4 Langages utilisés

L'application FitnessTrack utilise une combinaison de langages de programmation pour mettre en œuvre son architecture et ses fonctionnalités. Les langages couramment utilisés dans le développement de FitnessTrack comprennent :

- Frontend: HTML5, CSS3, JavaScript, TypeScript, React, React Native.
- Backend: Node.js, Express.js, Firebase Functions.
- Base de Données : Firebase Firestore, SQL pour certaines opérations analytiques.

Ces langages de programmation sont combinés de manière à mettre en œuvre l'architecture MVC (Modèle-Vue-Contrôleur) dans un contexte client-serveur, permettant ainsi une séparation claire des responsabilités et une gestion efficace des données, de la logique métier et de l'interface utilisateur au sein de FitnessTrack.

## 7.5 Outils et logiciels utilisés

Les outils et logiciel utilisés incluent :

Environnement de développement intégré (IDE) :

- Visual Studio Code pour le développement en HTML5, CSS3 et React.js
- Android Studio pour le développement mobile avec flutter

Système de gestion de base de données :

- Firestore pour une base de données en temps réel
- MongoDB pour les données flexibles

Outils de modélisation :

- Astah pour la création de diagramme de classe, de séquences et d'états

Serveur d'application :

- Serveur Node.js avec Express.js pour déployer et exécuter les composants back-end
- API REST pour la communication avec les front-end mobile et Web

Contrôle de version :

- Git
- GitHub

Outils de tests :

- Jest
- Cypress

## 7.6 Conclusion

En somme, Ce document de conception de FitnessTrack a été élaboré avec pour objectif de fournir une version complète et détaillée de la conception de cette application novatrice. A travers les différents chapitres, nous avons exploré les besoins et les fonctionnalités essentielles de FitnessTrack, défini une architecture globale solide et détaillée les aspects clés de la modélisation fonctionnelle, statique et dynamique. Nous avons aussi examiné en profondeur la conception de la base de données, les choix technologiques pour la réalisation et la mise en œuvre, ainsi que les stratégies de test et d'assurance qualité, donc la conception du système FitnessTrack a été élaborée pour offrir une application robuste, évolutive, et performante, capable de répondre aux besoins des utilisateurs tout en respectant les contraintes identifiées. Chaque composant du système a été soigneusement structuré pour faciliter l'interaction avec l'utilisateur, la gestion des données, et la fourniture de services fiables, contribuant ainsi à une expérience utilisateur fluide et motivante.