



# Rapport projet : BodySynch



## Table des matières

Résumé.....	4
Mots-clés.....	4
Développement.....	4
Projet et gestion de projet.....	5
Vocabulaire spécifique au projet BodySynch .....	6
Table des illustrations .....	7
Contexte et analyse des besoins .....	8
Présentation du problème posé, des objectifs fixés.....	8
Présentation du problème posé .....	8
Objectifs fixés.....	9
État de l'art du domaine concerné, analyse concurrentielle.....	9
Etat de l'art des systèmes de suivi de la <b>glycémie</b> .....	9
Analyse concurrentielle .....	11
Analyse SWOT de BodySynch .....	12
Analyse de besoins fonctionnels, des exigences non-fonctionnelles .....	13
Besoins Fonctionnels.....	13
Exigences non-fonctionnelles.....	14
Résultats obtenus .....	14
La conception de l' <b>application</b> .....	15
Architecture fonctionnelle et technique.....	15
Privacy Impact Assessment.....	19
Le corps de l' <b>application</b> .....	21
B2C : Business to Client.....	21
B2B : Business to Business .....	23
Le backend FireBase .....	24
Le modèle de <b>machine learning</b> .....	25
Les différences par rapport aux choix originaux.....	29
Gestion de projet.....	30
Planification .....	30
Planification temporelle.....	30
Planification organisationnelle .....	31

Identification des ressources et des risques projet .....	33
Ressources .....	33
Risques.....	33
Problèmes rencontrés et leur accordance avec les risques .....	36
Bilan et perspectives .....	38
Bilan .....	38
État d'achèvement du projet .....	38
Difficultés rencontrées et solutions .....	38
Perspectives.....	39
Quels sont les prochains points à aborder ?.....	39
Quelles sont les perspectives d'amélioration à terme ? .....	39
Annexes.....	40
WBS et RACI originels.....	40
Extraits du CDCM.....	41
Détails de la modélisation des Use Cases .....	44
GANTT originel.....	46

## Résumé

Le présent rapport est un document relatant l'évolution du projet **BodySynch**, de ses débuts à son état actuel. Il décrit l'objectif du projet et son contexte, l'organisation de l'équipe, la conception de l'**application**, les différentes technologies utilisées pour l'**application** et comment se sont réparties les tâches dans le temps, de même que les possibilités d'amélioration du projet **BodySynch**.

## Mots-clés

## Développement

**Frontend** : Terme utilisé pour la portion de l'**application** permettant l'interface entre les données et l'utilisateur. Les éléments de frontend utilisés ici sont JavaScript (pour la programmation), HTML (pour la description des pages) et CSS (pour les propriétés graphiques). Le frontend englobe l'affichage des pages et la communication avec le backend.

**Backend** : Terme utilisé pour la portion de l'**application** s'occupant de gérer les données. C'est notamment l'API et les services métiers liés à cette API, qui permettent le traitement des données fournies à et par l'**application**. Il communique avec l'utilisateur via le frontend.

**Framework** : Outil utilisé pour la création de base d'un projet, soit une **application** ou un logiciel. Il s'agit du squelette de notre **application**. Son objectif est de simplifier notre projet, en nous créant une base que l'on peut ensuite améliorer. Chaque langage de programmation possède des framework différents.

**API** : Acronyme d'Application Programming Interface, que l'on traduit en français par interface de programmation applicative ou interface de programmation d'**application**. L'API est une solution informatique qui permet à des **applications** de communiquer entre elles et de s'échanger mutuellement des services ou des données. Une API permet d'accéder aux fonctions ou aux données d'une **application** à distance, généralement depuis une autre **application**, en passant par une interface applicative standard. Une requête est envoyée au logiciel cible dans un langage universel (JSON).

**IoT** : Internet of Things. L'Internet des objets est l'interconnexion entre l'Internet et des objets, des lieux et des environnements physiques.

**BaaS** : backend as a service. Le Backend-as-a-Service (BaaS) permet aux développeurs de se concentrer sur le frontend de leurs **applications** et d'exploiter les services backend sans avoir à les créer ou à les maintenir.

**UID** : Un user identifier ou UID permet d'identifier un utilisateur via une clé unique. Cette technique est utilisée principalement pour les droits d'accès à des ressources ou à des domaines et donc pour la sécurité du système.

**LSTM** : Long Short-Term Memory. Type de modèle de **machine learning**. Les LSTM ont été créés comme méthode permettant de gérer efficacement la mémoire à court et long terme grâce à leurs systèmes de portes.

## Projet et gestion de projet

**Spécifications** : Déterminées en phase de cadrage et de conception, elles déterminent les différentes **applications** du projet et ce qu'est censé faire chaque élément. On les sépare concrètement en deux types : Les spécifications fonctionnelles et techniques. Les spécifications fonctionnelles présentent ce que va faire le produit développé, tandis que les spécifications techniques détaillent comment il va le faire, avec quelles technologies, quelle **architecture** et quel matériel. Ici, nous avons essentiellement cité les spécifications fonctionnelles.

**WBS** : Work breakdown structure. Outil permettant la séparation en tâches du projet, afin d'en gérer l'attribution, les ressources et le temps et de faire la meilleure planification possible. Ici, le WBS a été allié à une planification temporelle à partir des échéances.

**Matrice RACI** : Mode de répartition des tâches, entre Responsable (celui responsable de l'exécution), Accountable (celui responsable de la supervision), Consulted (celui qui ne fait rien mais est considéré comme l'expert à consulter en cas de problème ou de question) et Informed (celui qui connaît l'existence de cette tâche sans y participer).

**PoC** : Proof of Concept. Est une méthode qui permet d'évaluer la faisabilité d'un projet. Il fait partie du cadrage du projet et est notamment très utilisé dans les méthodes Agile. Il peut être utilisé pour la création d'un nouveau produit ou d'un logiciel, ou encore à l'occasion d'une réorganisation ou du lancement d'un service.

**MVP** : Minimum Viable Product. Le Minimum Viable Product est une version du produit destinée à être commercialisée pour être testée par le client final. Elle est en état de fonctionner, même s'il ne s'agit pas encore de la version finale : il peut par exemple manquer des fonctionnalités qui ne sont pas encore prêtes.

**CDCM** : Cahier des charges marketing. Le cahier des charges (CDC) marketing est un document rédigé par le service marketing pour faire une demande de développement auprès des services R & D. Il contient les principaux éléments nécessaires pour comprendre le positionnement marketing du futur produit. Il mentionne également des informations relatives au délai du projet dans sa globalité depuis la demande jusqu'à la mise sur le marché.

**C4 Model** : C4 est l'acronyme de Context, Container, Component et Code. C'est un type de diagramme architectural pour une **application**. L'élément central du C4 model est l'approche en différents niveaux de zoom de la représentation du système à modéliser. Ces vues permettent d'avoir des détails précis ou au contraire une vue globale.

**RGPD** : Règlement général de la protection des données.

**SWOT** : Strengths, Weaknesses, Opportunities, Threats. L'analyse SWOT permet d'identifier les forces, faiblesses, opportunités et menaces d'un projet donné ou d'un business plan dans son ensemble.

**PIA** : Privacy Impact Assessment. Le Privacy Impact Assessment est une méthode conçue pour réaliser des analyses d'impact requises par le RGPD concernant la protection des données personnelles, appliqué depuis le 25 mai 2018 dans toute l'UE.

**CRUD** : Create, Read, Update, Delete. Désigne les quatre opérations de base pour la persistance des données, en particulier le stockage d'informations en base de données.

**B2C** : Business to Business. Désigne un produit destiné à des entreprises, ou des professionnels.

**B2B** : Business to client. Désigne un produit destiné à des particuliers.

**Stack technique** : Anglicisme très couramment utilisé dans le domaine du développement logiciel et qui représente l'ensemble des technologies et composants utilisés pour développer et faire tourner un produit logiciel. Aussi appelé environnement technique.

## Vocabulaire spécifique au projet **BodySynch**

**CGM** : Continuous Glucose Monitoring. Il s'agit d'un type de capteur capable de mesurer le glucose interstitiel des cellules en temps réel et de renvoyer une mesure à intervalles réguliers. Il se différencie du capteur ampérométrique de mesure ponctuelle par la passivité de la prise de mesure (le capteur est posé sur le bras pendant un temps donné) et la continuité des mesures.

**Métabolisme** : Ensemble des transformations qui s'accomplissent dans l'organisme vivant.

**Dispositif médical** : On entend par dispositif médical « tout instrument, appareil, équipement, matière, produit, à l'exception des produits d'origine humaine, ou autre article utilisé seul ou en association, y compris les accessoires et logiciels nécessaires au bon fonctionnement de celui-ci, destiné par le fabricant à être utilisé chez l'homme à des fins médicales et dont l'action principale voulue n'est pas obtenue par des moyens pharmacologiques ou immunologiques ni par métabolisme, mais dont la fonction peut être assistée par de tels

moyens. Constitue également un dispositif médical le logiciel destiné par le fabricant à être utilisé spécifiquement à des fins diagnostiques ou thérapeutiques. (Sante.gouv.fr)

***Glycémie post-prandiale*** : Glycémie d'un corps après consommation de nourriture.

## Table des illustrations

Figure 1 : Analyse SWOT de <b>BodySynch</b> .....	12
Figure 2 : Architecture des systèmes embarqués dans le produit .....	16
Figure 3 : C4 model de BodySync, niveau Container .....	17
Figure 4 : Diagramme Use Case résumé de <b>BodySynch</b> .....	18
Figure 5 : Schéma relationnel de la base de données en UML .....	19
Figure 6 : Storyboard du B2C.....	22
Figure 7 : Storyboard du B2B.....	24
Figure 8 : Format de la base de données FireStore .....	25
Figure 9 : Données nécessaires à l'authentification .....	25
Figure 10 : Conception d'un modèle de <b>machine learning</b> .....	26
Figure 11 : Schéma d'un réseau LSTM.....	27
Figure 12 : Positionnement des risques sur une matrice gravité-vraisemblance .....	35
Figure 13 : Diagramme risques/solutions .....	35

# Contexte et analyse des besoins

## Présentation du problème posé, des objectifs fixés

### Présentation du problème posé

Le projet **BodySynch** découle d'une prise de conscience croissante quant à l'importance cruciale de la **santé Métabolisme** dans le bien-être général des individus. Cette prise de conscience est alimentée par une augmentation alarmante des maladies chroniques telles que l'obésité, le diabète de type 2 et les maladies cardiovasculaires, attribuées en grande partie à des modes de vie sédentaires et à des habitudes alimentaires déséquilibrées dans notre société moderne.

Face à cette réalité, le défi majeur réside dans le manque de personnalisation des outils et des ressources accessibles pour un suivi efficace. Actuellement, la surveillance de la **santé métabolique** est souvent réactive, les individus ne prenant conscience des problèmes que lorsqu'ils présentent des symptômes ; De même, lorsque des ressources de surveillance existent, elles sont limitées à une seule constante (**glycémie**, sommeil...), rendant l'analyse des constantes de santé étreinte et inefficace. Quelques exemples sont les **applications** en lien avec le fitness, le suivi pur de la **glycémie** ou l'enregistrement manuel de ses constantes.

Il est à noter que même lorsque les individus sont motivés pour améliorer leur **santé métabolique** avant de présenter des symptômes, ils se heurtent souvent à des obstacles pour accéder à des professionnels de la santé spécialisés, ou à des ressources fiables pour les guider dans leur parcours de santé, comme le manque de temps ou de professionnels disponibles, ou le manque de pertinence des ressources. Ces dernières emploient des programmes plus génériques, reposant sur des données moins précises que ce que représente le suivi de la **glycémie** en temps réel : Cela entraîne des difficultés à trouver les bons conseils de manière rapide et empirique. On peut également citer le problème d'absence de personnalisation dans les programmes mis en place dans le monde du sport et de la nutrition, ce qui est un grand frein à la gestion de sa **santé métabolique** : Les variations entre solutions ne sont pas suffisantes pour couvrir les variations d'utilisateurs en recherche.

Cette situation crée un besoin urgent de solutions innovantes qui permettent un suivi en temps réel de la **santé métabolique**, ainsi que le fournissement de conseils personnalisés et d'un soutien professionnel, optionnel selon les préférences de l'utilisateur. Le suivi en temps réel de la **glycémie** représente une solution phare pour répondre à ce besoin : De la surveillance de la **glycémie** découle un suivi de la perte de poids, une explication possible à des fatigues irrégulières en lien avec une hypoglycémie chronique, une meilleure compréhension de sa capacité de concentration, d'endurance et des réserves d'énergie à sa disposition pour les activités sportives, mais également les activités stressantes, demandant une quantité d'énergie importante.



Dans ce contexte, **BodySynch** se positionne comme une **application** intelligente conçue pour répondre à ces besoins spécifiques. En permettant aux utilisateurs de suivre leur santé métabolique en temps réel, de recevoir des conseils personnalisés, si souhaité, d'un suivi professionnel, **BodySynch** vise à autonomiser les individus dans la gestion proactive de leur santé et de leur bien-être.

## Objectifs fixés

Dans le cadre du projet **BodySynch**, plusieurs objectifs ont été établis par le client (voir Extraits du CDCM) pour orienter nos efforts vers la réalisation d'une solution complète et innovante pour le suivi et l'amélioration de la santé métabolique. Ces objectifs ont été soigneusement définis afin de répondre aux besoins des utilisateurs tout en assurant la faisabilité technique et la viabilité du projet. Voici les principaux objectifs fixés :

- **Développement de la *PoC*** : L'objectif principal était de concevoir et de réaliser une preuve de concept avancée de l'**application BodySynch**. Cette *PoC* devait démontrer la faisabilité technique de l'**application** en intégrant des fonctionnalités clés telles que la connexion et l'agrégation de données à partir de différents biocapteurs et **application** du marché, en mettant particulièrement l'accent sur l'utilisation des capteurs *CGM* pour le suivi en temps réel de la **glycémie**.
- **Création du *MVP*** : une fois la *PoC* établie, l'objectif suivant était de développer un *MVP* fonctionnel de l'**application BodySynch**. Le *MVP* devait offrir une expérience utilisateur de qualité tout en mettant en œuvre les fonctionnalités essentielles permettant aux utilisateurs de suivre leur santé métabolique. L'intégration de fonctionnalités optionnelles telles que recevoir des conseils personnalisés et bénéficier d'un suivi professionnel a été envisagée selon les besoins et les préférences des utilisateurs.
- **Intégration d'un modèle d'apprentissage automatique** : Un autre objectif crucial était d'identifier et d'intégrer un modèle d'apprentissage automatique approprié pour permettre la prédiction de l'évolution de la **glycémie** à partir des données recueillies. Cette fonctionnalité vise à fournir aux utilisateurs des recommandations encore plus précises et personnalisées pour les aider à mieux gérer leur santé métabolique.

## État de l'art du domaine concerné, analyse concurrentielle...

### Etat de l'art des systèmes de suivi de la **glycémie**

L'étude de la **glycémie**, en quoi ça consiste ?

L'étude de la **glycémie** est essentielle pour une personne diabétique, mais il s'agit également d'une donnée intéressante pour une personne en bonne santé, afin d'améliorer son bien-être et

comprendre son organisme. Avant de s'attaquer à la conception de l'**application**, il a donc fallu comprendre quelles étaient les solutions disponibles et le format des valeurs à étudier.

En l'absence de diabète, la **glycémie** sanguine varie naturellement au cours de la journée autour de 1 g/L, en fonction des repas ou de l'activité physique. Elle est d'environ 0,8 g/L avant les repas ou après une activité physique. Elle augmente avec la prise de nourriture, puis diminue progressivement ensuite.

L'autosurveillance de la **glycémie** consiste à mesurer soi-même sa **glycémie** à un moment donné de la journée. Elle est réalisée :

- Le plus souvent à partir d'une goutte de sang prélevée à l'extrémité d'un doigt grâce à un auto-piqueur pour une mesure instantanée ;
- De façon plus récente, grâce à un appareil de mesure du glucose interstitiel. Cet appareil permet une mesure de glucose en continu.

Elle peut être réalisée à domicile, sur son lieu de travail, en voyage, dans un établissement sportif, etc.

Il est bon de noter sur les différences entre la mesure du glucose sanguin et du glucose interstitiel, que pour un niveau de sucre stable, le niveau de glucose interstitiel est égal au taux de **glycémie** sanguine ; pour un taux de sucre en augmentation, le niveau de glucose interstitiel est inférieur à celui de la **glycémie** sanguine ; enfin, pour un taux de sucre en baisse, le niveau de glucose interstitiel est supérieur à celui de la **glycémie** sanguine.

Cette différence est due à un décalage temporel car les glucides transformés en glucose passent du sang vers le liquide interstitiel. Il faut en être conscient pour comprendre les données renvoyées par les différents types de capteurs.

### Les différents types de lecture de la **glycémie**

De nos jours, la grande majorité des lecteurs de **glycémie** disponibles sur le marché sont des glucomètres à électrodes. Ils utilisent une méthode électrochimique, dite ampérométrie.

Dans un premier temps, la goutte de sang est aspirée par une bandelette de test. Le glucose sanguin va ensuite entrer en réaction avec une électrode enzymatique, ou capteur biologique. Ce capteur contient du glucose oxydase ainsi qu'un réactif. De multiples réactions chimiques vont générer un courant électrique : La force de ce courant étant proportionnelle au taux de glucose sanguin, le lecteur est capable de nous indiquer la **glycémie**. Cette réaction est instantanée, il ne faut donc que quelques secondes pour l'affichage du résultat.

Ce type de mesure a cependant le désavantage d'être uniquement instantanée, et ne permet donc pas de suivi en temps réel efficace.

Il existe également des capteurs de mesure du glucose interstitiel, ou **CGM** (mesure de glucose en continu). Ils fonctionnent également grâce à la technique reposant sur l'enzyme glucose oxydase ; Cependant, plutôt que d'avoir à se piquer le doigt, eux se posent sur le bras ou l'abdomen à partir d'un dispositif d'insertion automatique, et récupèrent des données en

temps réel (avec un décalage de quelques minutes cependant). Ces données sont ensuite stockées dans la base de données du capteur pour un temps donné (ce sont ces données qu'il faudra transmettre via réseau à l'**application**), puis envoyées au lecteur de données (souvent un lecteur peut être employé avec une appli fournie avec le capteur).

Obtenir une mesure en temps réel permet une avancée remarquable au niveau de suivi de la santé, tant des personnes diabétiques que des personnes en pleine santé : En effet, elle permet de passer outre les inconvénients des mesures ponctuelles, qui sont la possibilité non négligeable de manquer des alertes de glucose, l'incapacité de prédire la descente ou la remontée de sa **glycémie**, et l'obligation de devoir se piquer régulièrement. Le temps réel permet l'établissement de manière passive d'une courbe de suivi de la **glycémie**, et à partir de cette courbe de paramétrer des alertes en cas de changement important et de permettre à l'utilisateur de prédire de manière plus simple les variations de son taux de glucose ; cela permet un suivi extrêmement personnalisé, adapté aux variabilités de chaque individu et de son mode de vie.. Cela ouvre également la voie à l'analyse de données par un modèle **IA**, permettant de déléguer la prédiction de l'évolution de la **glycémie**, et, à terme, de se passer du capteur **CGM**. C'est un des objectifs, à terme, de **BodySynch**.

### Analyse concurrentielle

Dans le domaine de la santé et du bien-être, **BodySynch** fait face à une concurrence directe et indirecte. L'analyse suivante détaille ses concurrents, leurs forces et faiblesses, ainsi que les opportunités et menaces associées.

Un exemple de concurrent direct dans le domaine de la santé et du bien-être est **Levels Health**, qui offre un suivi en temps réel de la **glycémie** avec des recommandations pour optimiser la santé métabolique. Ce concurrent offre un suivi continu de la **glycémie** avec des conseils personnalisés pour améliorer les habitudes de vie. Ses forces résident dans la qualité de leur marque, qui se classe dans les haut-de-gamme étatsuniens. Cependant, ce type de concurrent peut être confronté à des défis de notoriété et de pénétration du marché, ainsi qu'à des contraintes réglementaires comme le **RGPD**, ou encore l'accès aux **CGM** qui peut être très coûteux ou (comme dans le cas des USA) se faire sous ordonnance. De même, comme ils se concentrent sur le marché américain, ils sont très peu implantés en France.

On peut également citer quelques concurrents indirects, comme **Lumens.me**, qui propose une **application** de suivi de la nutrition et du métabolisme accompagnée d'un capteur de CO<sub>2</sub>, offrant des conseils personnalisés pour améliorer la santé. On note également **Oura**, qui est une bague connectée surveille le sommeil, l'activité physique et le niveau de stress pour optimiser la santé globale, ou **Withings**, qui offre une gamme de dispositif de suivi de la santé, y compris des balances connectées et des tensiomètres, pour aider les utilisateurs à suivre leur santé générale. Ces concurrents proposent des solutions de suivi de la nutrition, du sommeil, de l'activité physique, et de la santé générale. Bien que leurs solutions ne se concentrent pas spécifiquement sur la **glycémie**, elles offrent des fonctionnalités complémentaires.

## Analyse SWOT de BodySynch

L'analyse **SWOT** (Figure 1) de **BodySynch** met en lumière les principales forces, faiblesses, opportunités et menaces auxquelles l'**application** est confrontée dans le contexte concurrentiel de l'industrie de la santé et du bien-être.

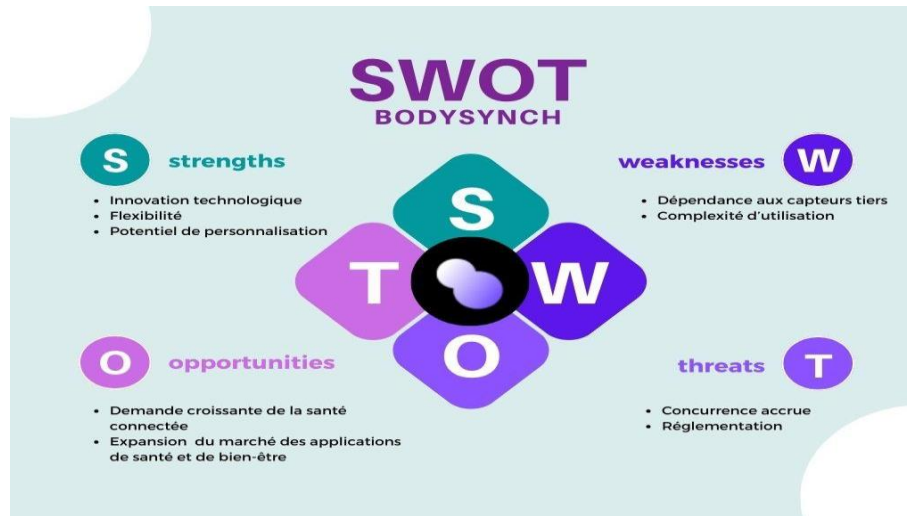


Figure 1 : Analyse SWOT de **BodySynch**

### Forces

**BodySynch** se distingue par son approche novatrice qui gère les données de différents capteurs et **applications** pour offrir une solution complète de suivi de la santé et du bien-être. Cette technologie de pointe lui confère un avantage concurrentiel en termes de valeur ajoutée pour les utilisateurs.

De plus, **BodySynch** apporte une flexibilité aux utilisateurs : En se concentrant sur la collecte de données à partir de capteurs **CGM**, elle offre une solution flexible et adaptable aux besoins des utilisateurs. Cette capacité à s'adapter aux préférences individuelles renforce son attrait sur le marché.

Il est également à noter que **BodySynch** représente un important potentiel de personnalisation : La capacité de l'**application** à intégrer une variété de sources de données lui permet d'offrir une expérience utilisateur hautement personnalisée. Cette approche centrée sur l'utilisateur renforce l'engagement et la fidélité des utilisateurs envers la plateforme.

### Faiblesses

A l'état actuel, **BodySynch** montre une dépendance importante aux capteurs et systèmes tiers : Elle a besoin de la disponibilité et de la fiabilité des capteurs **CGM** et d'autres dispositifs non intégrés à l'**application** pour collecter les données. Cette dépendance peut entraîner des risques liés à la disponibilité des données et à la qualité de la collecte.

Elle peut également se révéler complexe d'utilisation à cause de la quantité de données à traiter : cela peut constituer un obstacle pour certains utilisateurs, en particulier ceux moins familiers avec la technologie. Cette faiblesse nécessite une attention particulière pour garantir une expérience utilisateur intuitive et conviviale.

## Opportunités

**BodySynch** s'inscrit dans une réponse à une demande croissante pour la santé connectée : L'augmentation de la prévalence des maladies chroniques comme le diabète crée une demande croissante pour des solutions de ce type. **BodySynch** se veut bien positionnée pour répondre à cette demande en offrant une solution intégrée et personnalisée de suivi de la *santé métabolique*.

Elle permet également une expansion du marché des **applications** de santé et de bien-être : Le marché des **applications** de santé et de bien-être est en croissance constante (les applications de santé représentent 230 milliards de dollars dans le monde en 2023, ce qui correspond à une augmentation de 160 % par rapport à 2019) offrant des opportunités de développement pour les nouvelles innovations telles que **BodySynch**. L'entreprise peut capitaliser sur cette tendance en développant des partenaires stratégiques et en investissant dans la recherche et le développement.

## Menaces

Les menaces les plus notables sont celles d'une concurrence accrue : Le marché des **applications** de santé et de bien-être est hautement concurrentiel, avec de nombreux acteurs établis et de nouvelles startups qui entrent en compétition pour attirer les utilisateurs. **BodySynch** doit rester vigilant face à cette concurrence en continuant à innover et à différencier sa proposition de valeur.

On peut également citer l'importance de la réglementation : Les réglementations gouvernementales dans le domaine des dispositifs médicaux et de la protection des données peuvent imposer des contraintes supplémentaires à **BodySynch**. L'entreprise doit se conformer rigoureusement à ces réglementations et adopter des pratiques de gouvernance solides pour atténuer les risques associés.

## Analyse de besoins fonctionnels, des exigences non-fonctionnelles

### Besoins Fonctionnels

L'**application** se doit de remplir plusieurs besoins fonctionnels mis en lumière par le client et l'équipe projet : une surveillance et agrégation de données, un *MVP* mettant l'accent sur la connectivité et une importante adaptabilité et personnalisation.

**BodySynch** doit être capable de se connecter et d'agréger des données provenant de divers biocapteurs du marché, tels que les *CGM* et les montres connectées, ainsi que de différentes **applications** de fitness et de suivi du sommeil. Cette fonctionnalité permettra aux utilisateurs de visualiser en temps réel leur *santé métabolique*.

Un *MVP* de l'**application** mobile **BodySynch** doit pouvoir prouver la connectivité avec les biocapteurs et les **applications** tierces, la mesure et la prédiction de l'évolution de la **glycémie**, ainsi que la mise en relation avec des spécialistes de la santé métabolique.

L'**application** doit permettre aux utilisateurs de configurer leurs objectifs, leurs habitudes alimentaires, leurs préférences d'entraînement, etc. Elle doit également fournir des recommandations personnalisées en fonction des données de l'utilisateur et de ses besoins spécifiques.

Enfin, nous pouvons citer l'interaction avec les spécialistes comme une fonctionnalité aux ramifications plus optionnelles : **BodySynch** doit offrir la possibilité aux utilisateurs de se connecter avec des spécialistes de la *santé métabolique*, tels que des diététiciens, des nutritionnistes et des coachs sportifs, pour recevoir des conseils personnalisés et un suivi professionnel.

### Exigences non-fonctionnelles

Quelques exigences non-fonctionnelles qui sont remontées au cours de l'analyse du besoin sont notamment une nécessité de conformité réglementaire, en lien avec une importance mise sur la sécurité des données, ainsi qu'une importante performance et fiabilité et facilité d'utilisation

**BodySynch** doit respecter les normes et réglementations applicables dans le domaine de la santé et du bien-être, telles que la norme IEC 82304 pour les logiciels de santé. Il doit éviter toute revendication médicale pour rester dans le cadre hors-*DM* et réduire les coûts réglementaires. Également, étant donné la nature sensible des données de santé, **BodySynch** doit garantir la sécurité et la confidentialité des données des utilisateurs en conformité avec des normes telles que le *RGPD* et la norme IEC 81001-5-1 pour la sécurité des logiciels de santé.

L'**application** doit être performante et fiable, offrant une expérience fluide et réactive même lors de la manipulation de grandes quantités de données. Les fonctionnalités de suivi en temps réel et de prédiction de la *glycémie* doivent être précises. **BodySynch** doit être également convivial et facile à utiliser pour une large gamme d'utilisateurs, y compris les athlètes professionnels, les amateurs de fitness et les professionnels de la santé. L'interface utilisateur doit être intuitive et ergonomique pour une navigation aisée.

## Résultats obtenus

L'équipe a rejoint le projet à un stade d'idéation : Le client en avait une idée précise et un planning à respecter, et a également fourni un *CDCM* (Cahier des charges marketing) afin d'encadrer le travail demandé. Il contenait le contexte du projet, une présentation de l'**application**, les utilisateurs cibles de l'**application**, les objectifs attendus, les fonctionnalités requises ainsi qu'un planning macro avec des rendus à donner à des dates précises, et nous permettait de recevoir une première approche du besoin du client lié à l'**application**, ainsi que de ses exigences en termes de résultat.

A partir de ces données, l'équipe s'est organisée pour tout d'abord, arriver aux connaissances exigées pour une conception et un développement optimal du projet, notamment grâce à un compte-rendu de recherches à rendre pour le 8 novembre 2023, ensuite, à partir des



connaissances apportées par l'étude du marché et du besoin, faire des choix sur la priorisation des fonctionnalités, notamment déterminer ce qui était essentiel à rendre sur l'**application** et ce qui pouvait être implémenté plus tard, ou ce dont l'équipe était capable ou non. A partir de là, des choix ont été faits sur la *stack technique* du projet, choix qui ont évolué en fonction des demandes du client et de la mise en lumière de différents besoins : Des entretiens avec le client ont eu lieu à des intervalles réguliers, où il nous présentait les informations qu'il avait obtenu de ses analyses afin que nous mettions en commun nos études pour décider de la meilleure option possible. Cela a notamment mené à des changements de *stack technique* au cours de la conception et du développement, pour répondre au mieux aux attentes tout en se facilitant, nous-mêmes, le développement.

## La conception de l'application

### Architecture fonctionnelle et technique

Afin d'approcher de manière la plus précise possible l'**architecture** de l'**application**, nous avons eu recours à plusieurs outils : Le premier, le plus simple, est un simple diagramme représentant les communications et l'interopérabilité entre les différents services utilisés, le deuxième est une **architecture** en conteneurs représentant les différents blocs de notre produit, le troisième un diagramme permettant de détailler les cas d'utilisation de notre produit et le dernier permet de représenter une vision de la base de données et des liens entre données.

Le diagramme Figure 2 est un schéma de l'**architecture** fonctionnelle de l'**application**, avec les grandes lignes du trajet des données et des différents accès à des tiers. Il nous permet de poser nos exigences *IoT* et de comprendre quel est le transit des données récupérées via l'**application** ou le capteur tiers déjà possédé par le client. Ce diagramme permet de constater que l'**application** est pensée pour récupérer des données à partir d'un compte déjà existant de l'utilisateur sur LibreLinkUp, L'**application** reliée aux capteurs Abbott Freestyle, via une *API\_REST*. Une récupération via fichiers locaux de l'**application** est envisagée en guise d'amélioration. Ces données se retrouveront sur le Cloud via un service *BaaS*, hébergeant également la base de données de l'**application**, qui communiquera avec le *frontend* de **BodySynch** pour renvoyer les données.

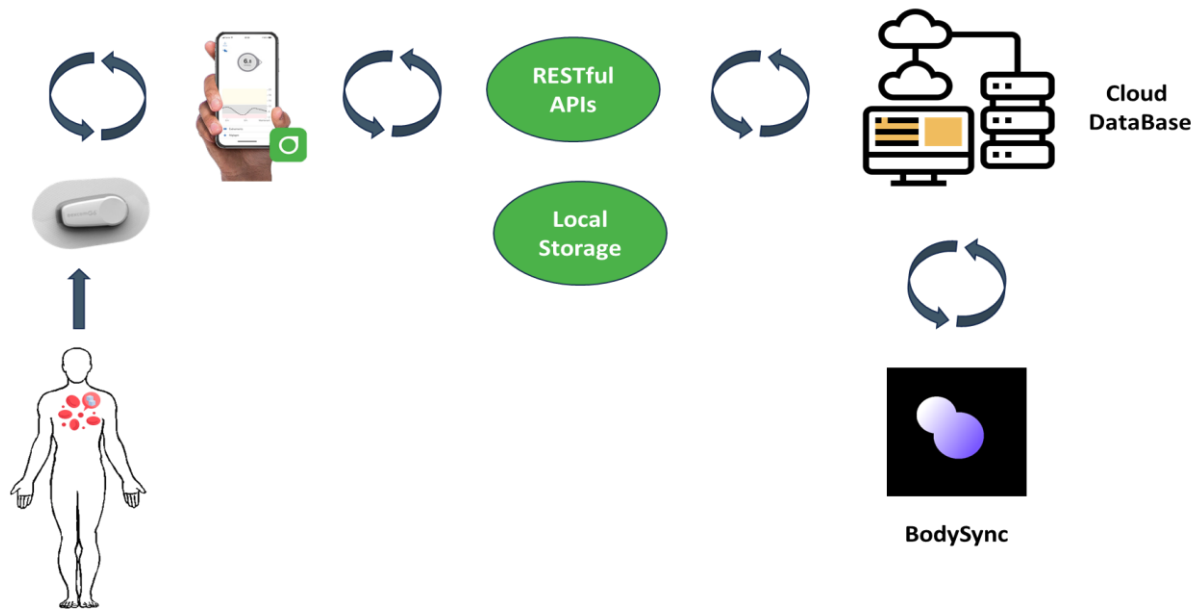


Figure 2 : Architecture des systèmes embarqués dans le produit

De même, un second diagramme fonctionnel (voir Figure 3) a été mis en place, se basant sur l'**architecture** du *C4 model*. L'élément central de ce modèle est l'approche en différents niveaux de zoom de la représentation du système à modéliser : ces vues permettent d'avoir des détails précis ou au contraire une vue globale, afin de pouvoir analyser à plusieurs niveaux l'**architecture** du code.

Ici, le niveau qu'il nous a paru le plus pertinent à utiliser était celui correspondant au niveau Container : Ce niveau permet de visualiser les différentes briques logicielles qui composent le système modélisé, comme les **applications** web et mobile, les APIs HTTP, les bases de données... Sur ce modèle, on voit notamment la connexion avec une *API* d'**application** tierce pour la récupération des données, une base de données temps réel reliée à l'**application** du capteur FreeStyle, et une autre intégrée dans l'**application** basée sur **FireBase**. Ces trois éléments communiquent avec le *frontend* de l'**application** mobile pour permettre à l'utilisateur de suivre ses constantes.

A noter que ce schéma, bien que la version finale soit rigoureusement identique en termes de communication et de types de conteneurs, a connu une petite évolution par rapport à la technique utilisée : **FlutterFlow** a remplacé Kotlin et Android Studio, comme ce sera étudié dans Le corps de l'**application**.



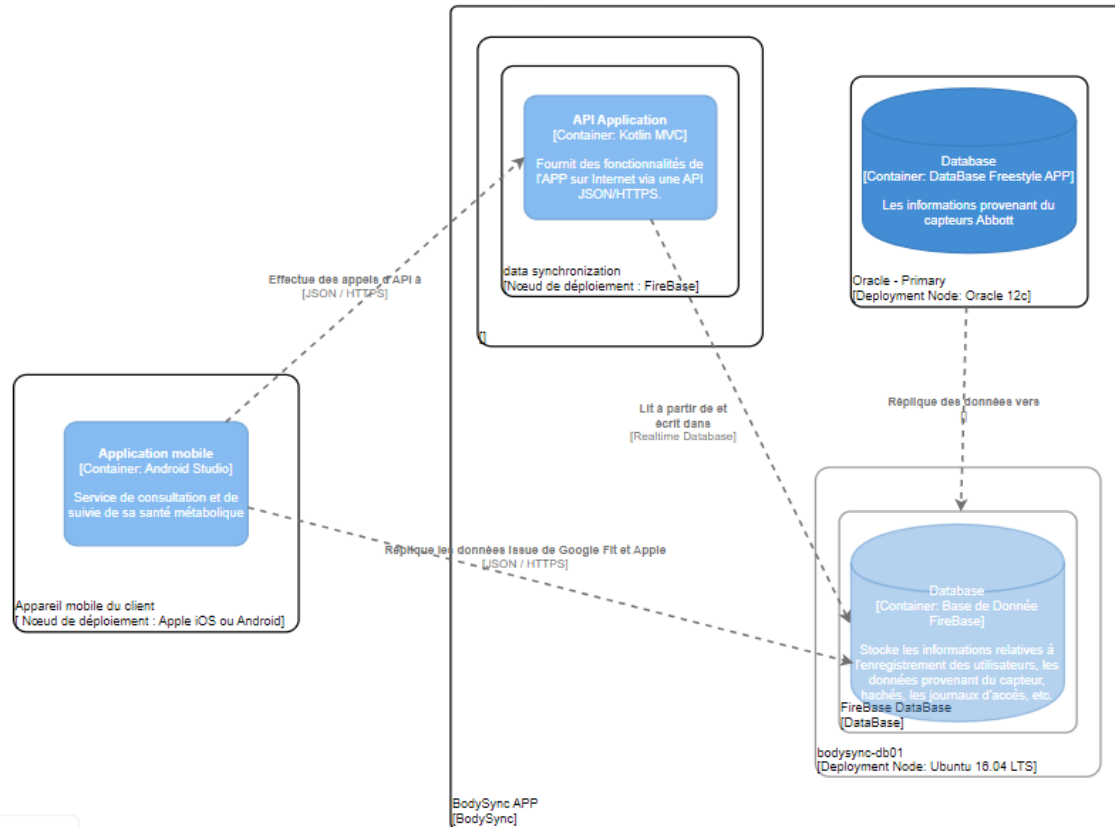


Figure 3 : C4 model de **BodySynchron**, niveau Container

Pour pouvoir répondre à ces contraintes d'**architecture**, il nous a fallu rechercher une **API** en lien avec l'**application** LibreLinkUp, permettant la récupération des données du client. L'idée de connexion directe (Bluetooth...) au capteur a également été envisagée, mais aucune ressource n'a été trouvée pour permettre une connexion sans avoir besoin de l'**application** tiers.

Les **API** retenues et testées au cours du développement sont les suivantes : [@diakem/libre-link-up-api-client - npm](#) : **API** basée sur Node.js pour intégration dans un projet, fournit également un dump HTTP afin de construire les appels à l'**API** de nous-même ; Les deux options ont été testées au cours du développement, via FlutterFlow et via le script Python fourni avec le projet afin de tester la validité de l'**API** ; [Script written in TypeScript that uploads CGM readings from LibreLink Up to Nightscout](#) : **L'API** fournie par NightScout, sous la forme d'un code TypeScript à intégrer dans son projet, pour ensuite récupérer les données sur l'**API** NightScout ; Le code TypeScript a été testé séparément, pour vérifier la validité de l'**API** avant de chercher à l'intégrer dans l'environnement de développement du frontend ; [Vital | Freestyle Libre API](#) : une **API** fournie par un tiers avec la garantie des mises à jour régulières, mais qui s'est révélée très vite derrière un paywall.

Pour ce qui est des cas d'utilisation (voir Figure 4), trois acteurs ont été identifiés dans ce projet : L'utilisateur performance/bien-être (abrégé en **UPE**), qui est la clientèle principale de l'**application**, l'utilisateur conseil (abrégé en **UC**), qui permettra de guider l'**UPE** dans l'optimisation de sa santé, et le capteur, ici **CGM**.

Selon ce diagramme de cas d'utilisation récapitulatif, UPE peut rentrer ses repas, rentrer son activité physique, consulter sa **glycémie**, consulter son profil ou demander conseil. UC pourra demander une mise en contact avec UPE, incluant une demande d'accès à ses données. Tous deux devront également gérer leur connexion, ainsi que leur inscription.

Le capteur, lui, n'a qu'un cas d'utilisation : Le traitement des données qu'il envoie. Ces données chemineront ensuite dans les packages interface et **IA** dépendant de leur utilisation.

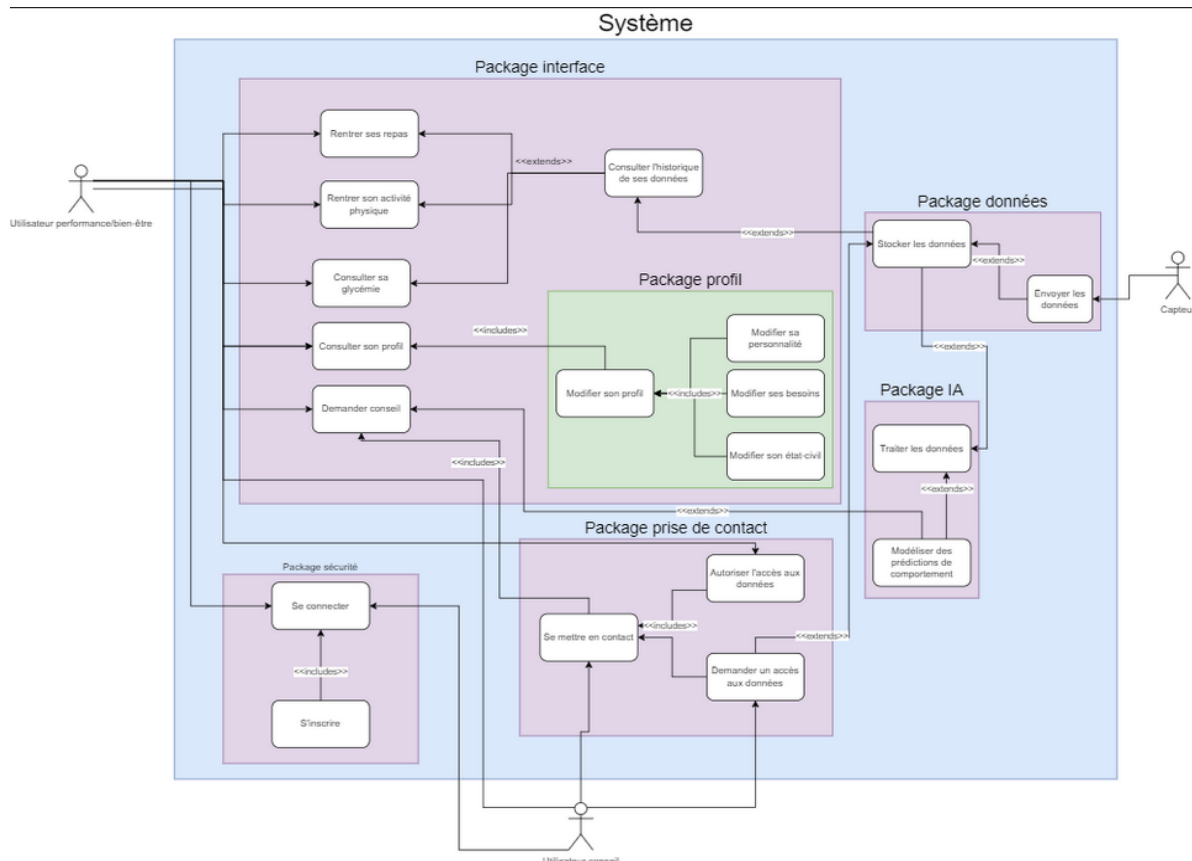


Figure 4 : Diagramme Use Case résumé de **BodySynch**

Enfin, la BDD (voir Figure 5) ici conçue avec l'aide des normes UML recoupe l'essentiel des données de l'**application**. Les classes **Message** et **TauxGlycémie** communiqueront avec l'**API** du capteur sélectionné afin de récupérer les données de l'**application** en temps réel. En ce qui concerne le mot de passe de l'utilisateur, ayant donné qu'il n'est pas stocké dans la base de données même, mais géré par l'authentification **FireBase**, il n'apparaît pas dans le schéma.

Dans les faits, **FireBase** est une base de données non-relationnelle, nous faisant travailler sur un modèle orienté clé-valeur plus qu'un modèle purement basé sur SQL : Cette **architecture** a cependant été suivie au maximum au cours de la conception de la base de données.

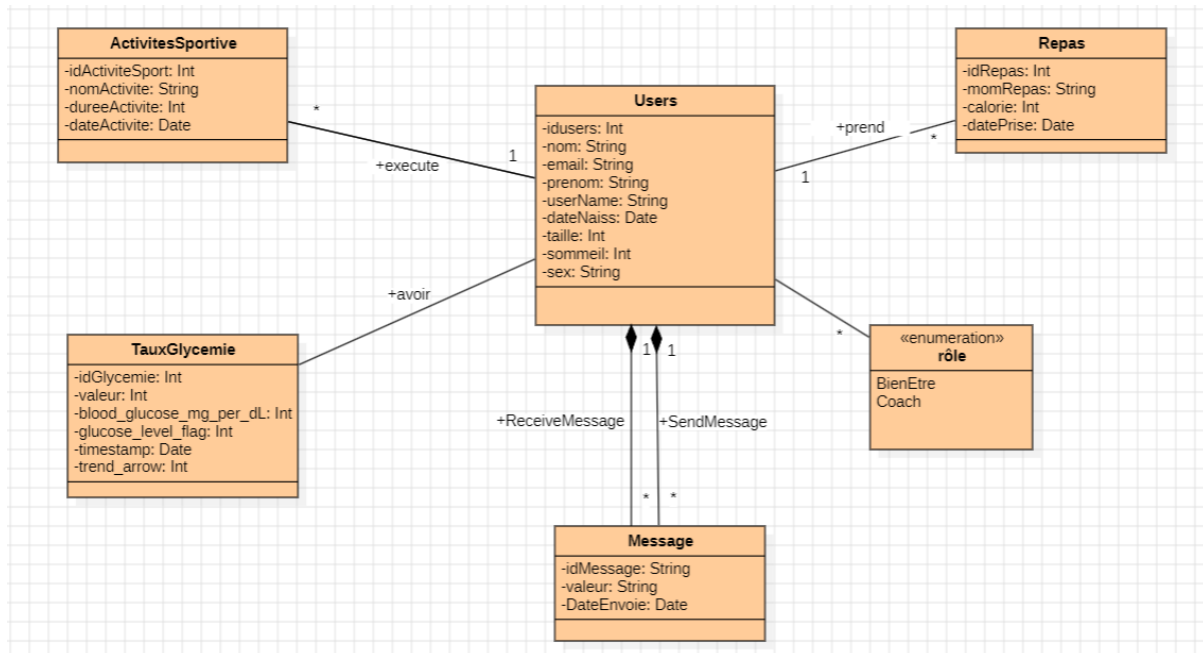


Figure 5 : Schéma relationnel de la base de données en UML

## Privacy Impact Assessment

Le **Privacy Impact Assessment**, abrégé en **PIA**, est une étude de l'impact de l'utilisation des données sur la vie privée des utilisateurs. Cette étude est essentielle pour chaque **application** centralisant des données privées, plus encore pour les données de santé, que nous traitons dans notre projet : Étudier leur sécurisation et leur impact est donc essentiel à la bonne marche de l'**application** et au respect du **RGPD** de la norme de développement 82304.

### Etude contexte

Le **contexte** nous a d'abord donné à étudier le traitement des données. Il s'agissait de définir clairement quelles données transitent : ici, il s'agit des données de santé issues de biocapteurs et d'**applications** reliées, relatives à la nutrition et à l'effort physique, avec pour enjeu de monitorer sa santé.

Les **responsabilités** liées au traitement qui se sont dégagées sont reliées à la société mère de l'**application**, notre commanditaire projet ; Les sous-traitants sont notre service **BaaS Firebase**, en charge de l'hébergement de l'**application**, et les professionnels de santé disposant du profil conseil.

Le **référentiel applicable** du **RGPD** pour le traitement des données est le RS-01 sur la gestion des vigilances sanitaires.

Le cycle de vie des données a également été étudié, et posé sous la forme suivante :

**Collection des données** : Récupération en temps réel des données via le capteur associé. Intervalles déterminés par le capteur (cinq minutes dans le cas d'Abbott FreeStyle 2, quinze secondes pour Dexcom G4).

**Stockage des données** : Les données sont ensuite conservées dans la base de données connectée au capteur concerné

### Accès aux données :

- Accès aux données par l'**IA** : L'**IA** a accès aux données de tous les utilisateurs pour nourrir son développement. Toutes les données de tous les utilisateurs consentants la nourrissent.
- Accès aux données par les utilisateurs client : Un client ne peut accéder qu'à ses propres données. Il peut également donner ou non son consentement pour qu'elles soient traitées par l'**IA** ou par les conseillers qui leur sont liés.
- Accès aux données par les utilisateurs conseil : Un conseiller ne peut accéder qu'aux données des patients ayant donné leur consentement (pendant la mise en relation)
- Accès aux données par le responsable de traitement : Le responsable de traitement doit pouvoir accéder à toutes les données récoltées pour la maintenance et le respect de la qualité. Elles seront cependant chiffrées pour éviter toute diffusion.

**Export des données hors de l'application** : Uniquement en cas de fin d'utilisation, en même temps que la suppression.

**Conservation des données** : Conservation en base active pendant la durée d'utilisation des données (10 ans), puis en archivage intermédiaire pour la durée d'activité du compte. Une fois le compte supprimé, ces données seront elles aussi supprimées.

Etude des principes fondamentaux

Les **principes fondamentaux** du **PIA** sont de déterminer la finalité et la légalité du traitement des données. Le consentement éclairé de l'utilisateur a été jugé nécessaire à l'utilisation de l'**application**. Il a également fallu déterminer de quelle manière les données collectées sont adéquates, permanentes et limitées à l'absolue nécessité : Deux cas ont été identifiés, à savoir, les données d'état-civil du patient, uniquement utilisées pour monter un dossier utilisateur, et les données glycémiques, traitées par l'**IA** à but informatif pour l'utilisateur.

Pour appliquer ces principes fondamentaux, il a fallu réfléchir aux mesures protectrices des droits de l'utilisateur. Notamment les droits à l'information sur le traitement des données, donner l'accès facile aux conditions d'utilisation et permettre d'obtenir de manière évidente le consentement des personnes concernées, avant le début de l'utilisation de l'**application**. Le droit d'accès et à la portabilité se devait également d'être respecté : Un export des données a été envisagé en cas de fin d'utilisation de l'**application**, sous un format lisible par toute **application**. De même, le support doit être prêt à procéder à l'effacement des données en cas de prise de contact à cet objectif de l'utilisateur.

Etude des risques

L'étude des risques se fait en trois étapes : Définition des mesures existantes ou prévues pour le développement de l'appli, relier ces mesures et étudier à partir leur impact sur trois risques : Accès illégitime, modification et suppression des données, et représenter ces risques et ces mesures sur une vue d'ensemble (voir Risques reliés à la sécurité des données). Elle est essentielle à un **PIA** car permet d'identifier le niveau de protection des données nécessité et toutes les actions à mettre en œuvre pour minimiser la fréquence ou l'impact du risque.

## Le corps de l'application

Pour construire l'**application**, nous avons utilisé l'**application FlutterFlow** pour le *frontend*. C'est une plateforme low-code, permettant une construction d'interfaces graphiques de manière rapide et efficace grâce à la disposition dans une page de widgets, auxquels sont reliés des actions le cas échéant. **FlutterFlow** se sépare en trois grandes fonctions : La construction de l'interface graphique, avec les différents widgets utilisables tels que des icônes, des boutons, des conteneurs de texte, des graphiques et tout ce qui était nécessaire pour l'affichage des données et la bonne navigation de l'**application** ; la mise en place d'actions sur les widgets concernés afin de paramétrer la navigation, les calculs, la recherche et l'affichage des données nécessaires ; enfin la mise en place d'appels au *backend* et aux **API** externes, pour récupérer les données soit de **FireBase**, soit des **API** employées par le projet. L'environnement permet également d'écrire ses propres fonctions, en langage Flutter, un *framework* développé par Google afin de pouvoir développer des **applications** multiplateformes.

Pour la base de données nous avons utilisé la base de données "Firestore" de **FireBase**. Il s'agit d'une base de données NoSQL : Firestore utilise un modèle de données basé sur les collections et les documents, où les données sont stockées sous forme de paires clé-valeur dans des documents organisés en collections. Il prend en charge des fonctionnalités telles que la recherche en temps réel, la synchronisation des données hors ligne et la gestion des autorisations, ce qui en fait un outil puissant pour le développement d'**applications** qui nécessitent un stockage de données flexible et évolutif. **FireBase** nous a permis de gérer le stockage des données, d'ouvrir une base de données temps réel pour récupérer les données du capteur **CGM**, et de permettre une gestion des authentifications par e-mail simple et efficace remplissant les contraintes de sécurité posées précédemment, notamment la vérification des e-mails.

De plus, **FireBase** et **FlutterFlow** sont facilement reliables l'un à l'autre via **FlutterFlow** pour récupérer, afficher, ajouter, supprimer et modifier des données de la base.

### B2C : Business to Client

Le **B2C** est destiné aux clients, et se présente en format mobile pour une vocation d'**application**. Pour le moment, un seul type de client est recensé dans la base de données, l'utilisateur Performance : C'est sur lui que nous nous sommes basés pour développer cette interface, étant donné qu'il s'agit de notre cible principale. Actuellement, l'**application** comprend treize pages, organisées suivant le storyboard ci-après (voir Figure 6 : Storyboard du B2C).

Elle compte une page de connexion, donnant sur la page d'accueil où le client pourra accéder aux différentes fonctionnalités ; Depuis la page de connexion, l'utilisateur est également en mesure de créer un profil qui sera enregistré dans la base de données **FireBase**, avec son rôle

Le **CRUD** est géré pour les utilisateurs, les ajouts de repas et d'activités sportives. Ces actions agissent sur la base de données, ainsi que, dans le cas du profil de l'utilisateur, pour l'authentification, et permettent à l'utilisateur de la base de données de gérer ses données comme il l'entend. Il n'est pour le moment pas géré sur la **glycémie**, car il a été jugé que seul l'affichage des données suffisait, tout particulièrement pour un suivi en temps réel ; Cependant, on peut considérer l'ajout de points dans le graphique dans le cas où un utilisateur voudrait utiliser l'**application** avec son lecteur de **glycémie** sanguine.



22

## B2B : Business to Business

Le **B2B** est destiné aux coachs, et se présente en format desktop pour une utilisation sur navigateur. Seuls les UC, intitulés “Coach” dans la base de données, doivent pouvoir se connecter à cette interface. Actuellement, l'**application** comprend neuf pages, organisées suivant le storyboard ci-après (voir Figure 7 : Storyboard du B2B).

Elle compte une page de connexion, donnant sur la page d'accueil où le client pourra accéder aux différentes fonctionnalités ; Depuis la page de connexion, l'utilisateur est également en mesure de créer un profil qui sera enregistré dans la base de données **FireBase** ; Le rôle s'ajoute automatiquement via une action de **FlutterFlow** implémentée dans la création de profil. L'authentification est gérée par e-mail via le **backend FireBase**.

La page d'accueil donne sur toutes les fonctionnalités de l'**application** : L'accès à la liste de tous les utilisateurs recensés, et celles des utilisateurs connectés avec l'utilisateur conseil afin de suivre leurs constantes. L'utilisateur peut également accéder à la gestion de son profil.

La liste d'utilisateurs globaux permet de parcourir tous les utilisateurs de l'**application**, et de les rechercher par nom d'utilisateur. Par mesure de confidentialité, tant qu'un utilisateur n'est pas connecté avec un coach, seul son nom d'utilisateur et sa photo de profil (si elle existe) sont visibles. Un bouton d'ajout est accolé à chaque utilisateur pour permettre la connexion : Une fois cela fait, l'**UID** de l'utilisateur Performance est ajouté dans la liste d'utilisateurs connectés de l'utilisateur Conseil sur la base de données.

La liste d'utilisateurs connectés est plus complète, et permet, elle, d'accéder au profil d'un utilisateur connecté : UC pourra notamment y lire la courbe de sa **glycémie** et les derniers repas pris. La même fonction de recherche y est implémentée, également basée sur les noms d'utilisateur.

Enfin, le profil de UC permet, outre de gérer ses données, d'accéder aux deux pages de recherche pour une navigation plus fluide. Depuis la page de gestion de profil, UC peut modifier ses informations personnelles, ou changer son mot de passe ; la fonction changement de mot de passe étant une fonction secondaire, implémentée via code customisé et non les actions par défaut de **FlutterFlow**, elle n'est pas complète et n'a pas été jugée prioritaire.

Le **CRUD** est géré pour les utilisateurs. Ces actions agissent sur la base de données ainsi que sur l'authentification, et permettent à l'utilisateur de la base de données de gérer ses données comme il l'entend. UC ne peut cependant modifier que ses propres données, selon les exigences établies par le **PIA** : Les données des utilisateurs connectés à son profil sont



uniquement

lisibles.

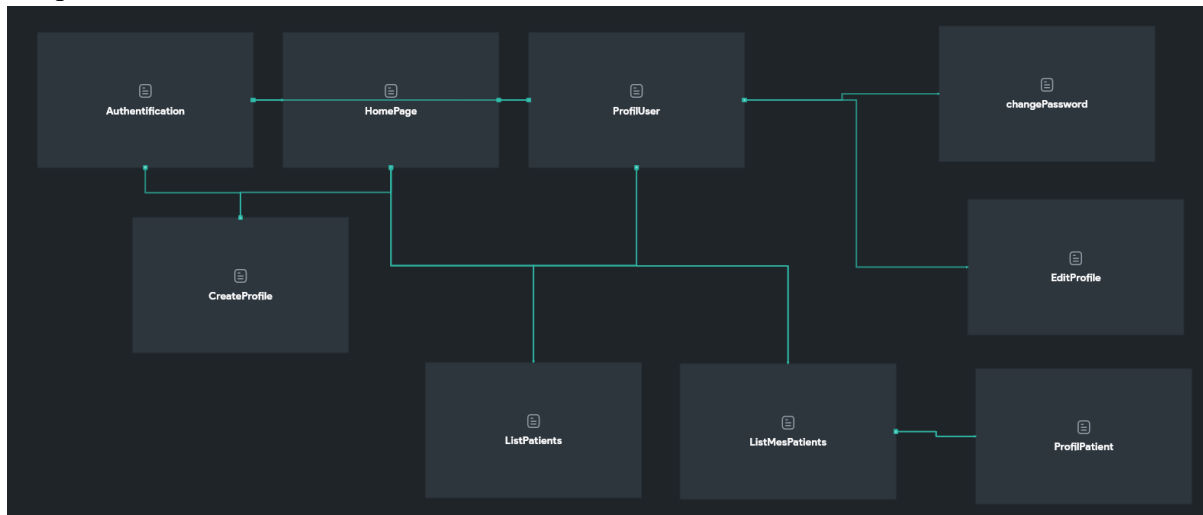


Figure 7 : Storyboard du B2B

## Le backend **FireBase**

**FireBase** est un outil d'aide au développement **BaaS** (Backend as a Service) hébergé par Google, permettant un hébergement dans le cloud des données de l'**application** développée et une gestion de bases de données noSQL, sous format clé-valeur. Il propose de nombreuses fonctionnalités utiles au développement d'une **application**, comme celles qui nous intéressent ici, soit la gestion du nom de domaine, celle de stockage des données, de suivi des données en temps réel, et d'authentification. De fait, comme il s'agit d'une base de données NoSQL, le schéma défini dans la conception n'est pas absolu ; cependant, il existe plusieurs moyens de le simuler, notamment avec des associations de clés dans les actions de l'**application**, comme celle qui permet d'enregistrer dans une liste reliée à un UC les **UID** d'un UPE.

**FireBase** possède également de nombreuses fonctions accessoires, comme le monitoring de l'**application** qui y est reliée, ou la construction de statistiques en lien avec les visites, les bugs et les fonctionnalités de l'**application**, qui n'ont pas été jugées utiles à la conception du **MVP** mais peuvent prouver leur utilité dans le développement du produit final, notamment pour coller aux exigences du **PIA**.

Le **backend** ne comprend pas de limites aux **applications** qui y sont reliées : Le **B2C** comme le **B2B** utilisent le même **backend**, la même base de données et le même système d'authentification. Si cela est essentiel au bon fonctionnement du produit global, cela pose néanmoins des problèmes sécuritaires comme la gestion de connexions conditionnelles, c'est-à-dire empêcher un UPE de se connecter sur l'**application B2B** avec ses identifiants **B2C** ; L'équipe a donc dû réfléchir à ces failles de sécurité avec les options données par **FlutterFlow**.



La base de données Firestore se présente sous le modèle suivant (voir *Figure 8*). Le mot de passe est géré via l'authentification et n'est donc pas visible dans les collections utilisateur, mais chaque authentification est bien reliée à une des colonnes de la base de données, et on ne peut se connecter avec une adresse mail déjà existante.

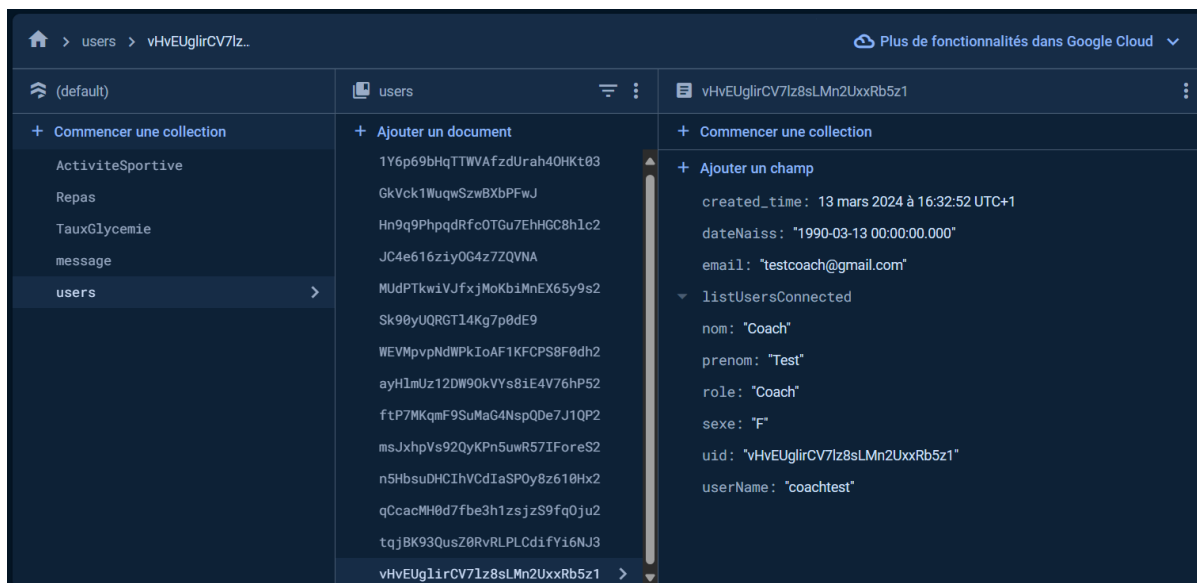


Figure 8 : Format de la base de données Firestore

L'authentification se présente sous le modèle suivant (voir Figure 9). Les adresses mail utilisées sont ici des placeholder, sauf celle anonymisée, utilisée pour tester la fonction de vérification de son adresse mail implémentée par **FlutterFlow** via le backend **FireBase** : Elle permet de gérer à partir de l'**UID** et de l'adresse mail de l'utilisateur ses données de connexion. Le mot de passe est également invisibilisé, et ce même aux administrateurs, afin de préserver la confidentialité des données.

Recherchez par adresse e-mail, numéro de téléphone ou ID utilisateur					Ajouter un utilisateur		
Identifiant	Fournisseurs	Date de création	Dernière connexion	UID utilisateur			
[REDACTED]	[REDACTED]	3 avr. 2024	3 avr. 2024	msJxhpVs92QyKPn5uwr57IF...			
testcoach3@gmail.com	[REDACTED]	21 mars 2024	21 mars 2024	n5HbsuDHCihVCdlaSP0y8z61...			
test122@gmail.com	[REDACTED]	21 mars 2024	21 mars 2024	ftP7MKqmF9SuMaG4NspQDe...			
testcoach2@gmail.com	[REDACTED]	13 mars 2024	13 mars 2024	ayHlmUz12DW90kVYs8iE4V7...			
testcoach@gmail.com	[REDACTED]	13 mars 2024	22 mars 2024	vHvEUglirCV7lz8sLMn2UxxRb...			

Figure 9 : Données nécessaires à l'authentification

## Le modèle de machine learning

Le modèle de **machine learning** doit répondre à un besoin de prédiction basée sur l'apprentissage automatique des niveaux de glucose chez des personnes non diabétiques. Sa

conception se base sur les étapes décrites sur le diagramme suivant (voir Figure 10 : Conception d'un modèle de **machine learning**) :



Figure 10 : Conception d'un modèle de **machine learning**

### Acquisition des données & Préparation des données

Ce travail se veut un travail préliminaire, approximatif et estimatif, et non être un travail dans des conditions réelles. Pour répondre à ce besoin, nous avons généré un jeu de données avec des données factices par **IA (OpenIA)**. Le jeu de données est généré de manière aléatoire en utilisant des distributions probabilistes pour chaque caractéristique. Voici les étapes générales et les prérequis utilisés pour générer le jeu de données :

#### **Participant ID, Âge et Sexe :**

- Les identifiants de participants sont attribués de manière séquentielle.
- Les âges des participants sont générés aléatoirement selon une distribution uniforme entre 18 et 80 ans.
- Le sexe de chaque participant est attribué de manière aléatoire, avec une répartition équitable entre masculin et féminin.

#### **Repas :**

- Les repas consommés par les participants sont choisis aléatoirement parmi une liste prédéfinie de repas tels que 'Petit déjeuner', 'Déjeuner', 'Dîner' ou 'Collation'.

#### **Aliment, Portion (g) et Glucides (g) :**

- Pour chaque participant, une liste aléatoire d'aliments consommés est générée.
- Chaque aliment est choisi aléatoirement parmi une liste prédéfinie d'aliments disponibles.
- Les portions de chaque aliment sont générées aléatoirement à partir d'un ensemble de poids prédéfinis.
- Les quantités de glucides dans chaque portion d'aliment sont également générées aléatoirement en fonction de l'aliment choisi. A noter que les protéines et lipides entrent aussi en ligne de compte, mais pour simplifier l'établissement du modèle, elles n'ont pas été considérées ici.

#### **Glycémie postprandiale (mg/dL) :**

- La **glycémie** postprandiale (après le repas) de chaque participant est calculée en fonction des aliments consommés, de leurs portions et de leurs quantités de glucides.
- Une valeur de **glycémie** postprandiale est attribuée en fonction de l'influence de chaque aliment sur la **glycémie**, ajustée en fonction de la portion consommée.
- En combinant ces étapes, un ensemble de données simulé est créé pour représenter les habitudes alimentaires et la **glycémie** postprandiale de plusieurs participants.

- Cette étude est à vocation expérimentale. Il est important de prendre en compte le fait que le modèle d'apprentissage est centré sur la prédiction de la **glycémie** et certains aliments pris par l'utilisateur : Par mesure de facilité, certaines données n'ont pas été prises en compte lors de l'entraînement du modèle.

### Choix du modèle

Introduits par Hochreiter et Schmidhuber en 1997, les réseaux de mémoire à long terme (voir Figure 11 : Schéma d'un réseau LSTM) sont d'une grande précision dans les tâches de prédiction de séries temporelles complexes, car ils tirent parti de leur capacité à établir des liens entre les dépendances à long terme sur plus de 1 000 pas de temps. Les **LSTM** sont bien adaptés aux données temporelles et ont montré des avantages dans la traduction automatique, la reconnaissance vocale et la prédiction de la mortalité dans les unités de soins intensifs, sur la base des dossiers médicaux électroniques.

Pour approcher une résolution de l'étude de la **glycémie**, nous avons donc décidé d'utiliser une approche basée sur les séries temporelles, étant donné que nos données sont issues de capteurs **CGM**. L'architecture flexible des **LSTM** offre un large éventail de configurations possibles.

La nature récurrente de la **LSTM** permet de **transmettre l'état de la cellule** non seulement à la cellule suivante, mais aussi à une couche **LSTM** profonde. Ces couches prennent l'état des cellules des couches précédentes comme entrée, par opposition aux données d'entrée, ce qui permet de former des niveaux d'abstraction de plus en plus élevés. Grâce à ces capacités accrues, plusieurs séries parallèles peuvent être analysées efficacement, formant une **prédiction unique**. Cette méthode peut être utilisée pour introduire des variables statiques en créant une série temporelle constante pour chacune d'entre elles. Toutefois, la complexité sera accrue, ce qui nécessitera davantage de couches et de données pour obtenir **une précision suffisante**.

### Entraînement du modèle

Une fois le modèle **LSTM** construit et compilé, l'étape suivante consiste à entraîner le modèle sur les données d'entraînement. Cette étape est cruciale car elle permet au modèle d'apprendre à partir des données et d'ajuster ses paramètres internes pour minimiser la perte (loss) lors de la prédiction.

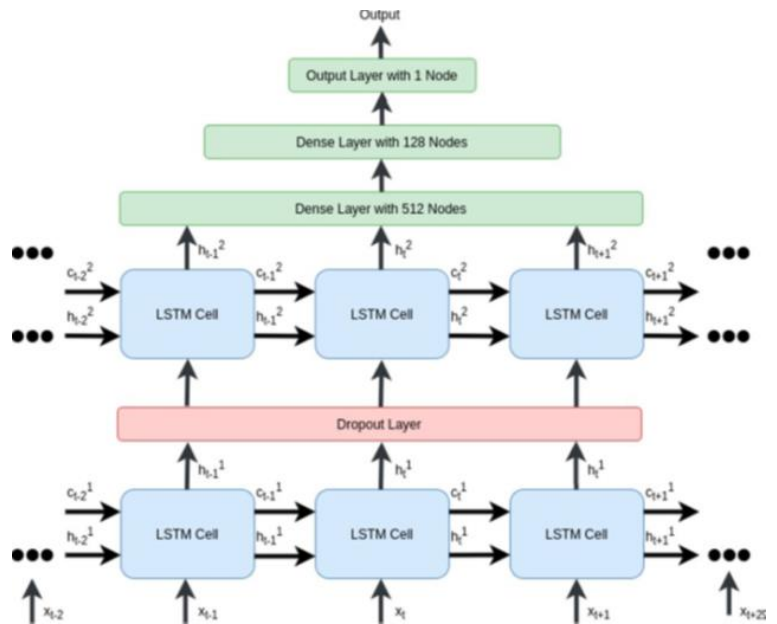


Figure 11 : Schéma d'un réseau LSTM

L'entraînement du modèle *LSTM* implique de fournir les données d'entraînement ( $X_{train}$ ) ainsi que les étiquettes correspondantes ( $y_{train}$ ). Les données d'entraînement sont généralement divisées en lots (batches) pour faciliter le processus d'optimisation. L'optimiseur, dans ce cas "Adam", est utilisé pour ajuster les poids du modèle afin de minimiser la perte calculée à l'aide de la fonction de perte spécifiée, dans ce cas "mse" (Mean Squared Error).

Pendant l'entraînement, le modèle parcourt plusieurs époques (epochs), c'est-à-dire des itérations sur l'ensemble des données d'entraînement. À chaque époque, le modèle ajuste ses poids en utilisant l'optimiseur pour améliorer sa performance sur les données d'entraînement. L'objectif de l'entraînement est d'optimiser les poids du modèle de telle sorte qu'il puisse généraliser correctement aux données qu'il n'a pas encore vues, c'est-à-dire les données de test. Il est important de surveiller la performance du modèle sur un ensemble de données de validation distinct pour éviter le surapprentissage (overfitting), où le modèle s'adapte trop précisément aux données d'entraînement et ne parvient pas à généraliser correctement.

Une fois l'entraînement terminé, le modèle est prêt à être évalué sur des données de test pour évaluer sa performance et sa capacité à généraliser à de nouvelles données.

### Prédiction & Evaluation

Après avoir entraîné le modèle *LSTM*, il est temps de l'utiliser pour effectuer des prédictions sur un ensemble de données de test. Ce processus permet d'évaluer la capacité du modèle à généraliser à de nouvelles données qu'il n'a pas encore vues.

Tout d'abord, nous utilisons la méthode `predict` du modèle pour obtenir les prédictions sur les données de test. Ces prédictions sont initialement normalisées, donc nous utilisons la méthode `inverse_transform` de l'objet `scaler` pour les ramener à l'échelle d'origine des valeurs de glucose.

Ensuite, nous identifions les valeurs prédites qui correspondent à chaque catégorie d'Indice de Pondération Glycémique (IPG). Les catégories comprennent les valeurs normales, prédiabétiques et diabétiques, déterminées en fonction des seuils spécifiés pour chaque catégorie.

Les valeurs prédites sont ensuite affichées dans chaque catégorie d'IPG pour une analyse plus détaillée.

Après avoir effectué les prédictions, nous évaluons la performance du modèle en utilisant plusieurs métriques d'évaluation couramment utilisées dans les tâches de régression :

- Erreur moyenne absolue (MAE) : Cette métrique mesure la moyenne des écarts absolus entre les valeurs prédites et les valeurs réelles. Elle fournit une indication de l'erreur moyenne de prédiction du modèle.
- Erreur quadratique moyenne (MSE) : Cette métrique calcule la moyenne des carrés des écarts entre les valeurs prédites et les valeurs réelles. Elle donne plus de poids aux grandes erreurs et est utile pour évaluer la précision du modèle.
- Coefficient de détermination ( $R^2$ ) : Aussi connu sous le nom de coefficient de détermination, cette métrique mesure la proportion de la variance dans la variable cible qui est expliquée par le modèle. Une valeur proche de 1 indique

Les résultats de l'évaluation du modèle de prédiction du glucose sont remarquablement prometteurs. L'erreur moyenne absolue (MAE) est extrêmement faible, à seulement 0.00065, ce qui indique que les prédictions du modèle diffèrent en moyenne de seulement cette petite fraction par rapport aux valeurs réelles de glucose. De même, l'erreur quadratique moyenne (MSE) est très basse, à 6.22e-07, soulignant une précision exceptionnelle dans les prédictions. Enfin, le coefficient de détermination ( $R^2$ ) est pratiquement égal à 1, s'élevant à 0.999996, ce qui suggère que le modèle explique près de 100% de la variance des valeurs de glucose. Ces résultats combinés témoignent d'une performance exceptionnelle du modèle dans la prédiction du glucose, offrant une base solide pour sa fiabilité et son utilisation future. Force est de cependant noter qu'en raison de quantité insuffisante de données notre modèle est exposé à un sur-apprentissage. Dans un futur proche, des méthodes seront utilisées dans le but de réduire le sur-apprentissage et le sous-apprentissage de notre modèle.

## Les différences par rapport aux choix originaux

La *stack technique* prévue à l'origine se basait sur une association de Kotlin avec **FireBase**, pour rendre une **application** purement Android. Malgré les avantages évidents que présentait cette stack au vu des compétences de l'équipe projet en la matière, divers problèmes ont été identifiés : L'impossibilité ou la difficulté d'itérer l'interface à grande vitesse, ce qui était essentiel pour le développement du projet à partir d'une évaluation de la concurrence et des besoins client, le portage Android unique qui limitait sur la mise en forme de l'**application** (notamment l'impossibilité de construire un *B2B* orienté desktop) et l'importance des tâches liées au développement du frontend, qui auraient causé un développement très lent, bien que régulier car le besoin de formation serait minime.

Pour pallier ce problème, le choix de **FlutterFlow** répondait à toutes nos attentes : L'interface low-code permettait une itération bien plus rapide, avec des changements de thème efficaces et des positionnements de fonctions mis en place de manière efficace et limitant les bugs, le service proposait le format orienté Android pour les utilisateurs clients et le format desktop pour les utilisateurs business, ce qui lui permettait de remplir un objectif que Kotlin ne pouvait pas combler et la capacité d'implémenter les tâches nécessaires au bon fonctionnement de l'**application** avec l'aide des actions. Le seul inconvénient majeur étant la protection du service derrière un paywall, limitant les fonctions plus poussées comme, notamment, les appels à *l'API* via **FlutterFlow** ; Pour surpasser ce paywall, il a fallu demander une licence éducative au service. De même, le langage de programmation de **FlutterFlow** se basant sur un *framework* non vu en cours, Flutter, la compréhension de chaque action et la correction des bugs nécessitent un temps de formation plus important, ajouté à celui dédié à la prise en main de l'interface de l'environnement de développement.

**FlutterFlow** permet également une connexion efficace à **FireBase**, qui était notre base de données choisie dans la première *stack technique*. De nombreuses fonctions telles que

l'authentification utilisateur, le stockage de données et la gestion de nom de domaine étaient gérées par **FireBase** et facilement implémentables dans **FlutterFlow**, ce qui nous a permis un gain de temps immense compensant celui pris par les formations à la technologie. Le choix de **FlutterFlow** a donc été jugé plus avantageux par l'équipe projet pour compenser les risques de développement et répondre au mieux à la problématique.

## Gestion de projet

### Planification

#### Planification temporelle

Le projet a connu de nombreux changements en termes de gestion au cours des six mois de son déroulement, notamment à cause d'un changement de technologie proposé par le client à la fin du mois de décembre, afin de pouvoir mieux répondre à ses exigences, répondre aux difficultés éprouvées par l'équipe technique et ouvrir la voie à la prochaine équipe qui s'y engagera. De fait, l'organisation a également énormément évolué, et le planning strict auquel nous devions nous tenir pour chaque étape de l'**application** est devenu beaucoup plus agile, basé sur un macro-planning imposé basé sur le précédent, avec des deadlines larges.

Notre organisation prévue pour les six mois était la suivante :

En termes de recherche, l'équipe s'est répartie chaque point sur lequel il a fallu approfondir nos connaissances. La tâche finale est la présentation des recherches, qui a eu lieu le 8/11.

Pendant la conception, la même idée de séparation a été envisagée : Cependant, là, beaucoup plus de tâches dépendantes ont été identifiées. Ainsi, les recherches sur le **BaaS** utilisables ont dû prendre fin plus tôt, le 25/11, puisque cette tâche était essentielle à la conception de la stack technique fonctionnelle. De même, les études sur le **RGPD** et l'élaboration du **PIA** se sont révélées bloquantes pour la conception de la **stack technique** : Il fallait être en mesure de déterminer les mesures de sécurité essentielles par rapport à la protection des données avant de choisir les différents éléments de la **stack technique**.

La **PoC** se découpe selon les tâches suivantes : Un temps d'un mois à partir du 18/12 alloué à la formation aux technologies employées, qui une fois achevé devait permettre de débiter chaque partie de la **PoC** : La collection de données, le modèle prédictif, le frontend et l'interface. Ces tâches devaient pleinement débiter au retour de vacances de l'équipe projet, le 8/01/2024. Le développement de l'interopérabilité et des liens entre ces différentes couches débiterait ensuite le 28/01/2024. Toutes ces tâches devaient être achevées le 15/03/2024, date à laquelle l'équipe aurait mis en œuvre les tests globaux de chaque couche et les tests d'intégration jusqu'au 15/04/2024.

Le développement du **MVP** devait débiter en parallèle de la **PoC**, mais en seconde priorité pour l'équipe projet et pour un rendu le 15/04/2024.



Le macro-planning originel n'a pas énormément bougé : Nos objectifs restaient d'allouer un mois à compter du 18/12/2023 afin de prendre en main la technologie, qui s'est réduite à trois semaines puisque nous avons reçu la proposition de changement de stack technique le 28/12/2023 ; La **PoC** avait pour vocation d'être rendue le 15/03/2024, et le **MVP** à la date prévue de la soutenance projet, le 12/04/2024.

Pour suivre ces objectifs, l'équipe a choisi une organisation en sprints de deux semaines à chaque fois : Une liste d'objectifs était définie avec le client au cours d'une réunion précédant chaque sprint, qu'il fallait avoir rempli deux semaines plus tard. Cette organisation permettait un planning fluide et a donc donné un Gantt en constante variation, puisque suite à de nombreux problèmes de développement plusieurs objectifs ont été reportés de semaine en semaine, tandis que d'autres, plus facilement mis en place, étaient réalisés en un temps très court (parfois dans les deux jours suivant le début du sprint) : Il fallait cependant pour le coordinateur projet être constamment au courant de l'avancée de chacun pour s'assurer de la répartition des tâches et du bon respect du planning.

Chaque sprint se déroulait comme suit : Une réunion d'équipe le lundi, en heure de projet tutoré ou à distance, pour répartir les tâches identifiées à la réunion de clôture du sprint précédent, des retours réguliers sur les objectifs de sprint au cours des deux semaines, puis, en fin de sprint (entre jeudi et samedi précédant la clôture), une réunion de clôture avec le client pour présenter notre avancée, discuter des difficultés et identifier les prochains objectifs de sprint. L'équipe terminait ensuite les objectifs manquants jusqu'à la date prévue de fin de sprint, soit le dimanche.

L'agilité mise en place sur le projet a permis à l'équipe de s'adapter à des imprévus de type problèmes techniques, difficultés d'utilisation des ressources et emploi du temps complexe à gérer, malgré tout, la gestion n'était pas idéale puisqu'un problème technique au cœur de la **PoC** a entraîné un retard important et une nécessité de remanier la priorisation des tâches, suite à des discussions entre l'équipe et avec le client. De même, le travail sur le modèle **IA** s'est révélé bien plus rapide que prévu, permettant à l'équipe de rendre une première approche en février et réorganiser l'équipe pour pallier les difficultés rencontrées.

### Planification organisationnelle

L'organisation originelle était la suivante : à partir de la **stack technique** choisie à l'origine, placer un membre de l'équipe sur le développement **IA**, deux sur le développement de l'IoT et de la connectivité du projet, et un sur le frontend pour obtenir l'interface de l'**application** (voir WBS et RACI originels). Cette organisation a changé au cours des mois de développement pour faire passer deux développeurs sur l'interface de l'**application** (**B2C** et **B2B**) et une personne en charge de la gestion de projet technique, qui grâce à une analyse poussée des concurrents et des interfaces proposées par ces derniers, permettrait à l'équipe de développement d'orienter son travail et d'itérer sur l'interface de l'**application**. La dernière personne de l'équipe serait en charge de l'**IA**.

Devant la rapidité d'avancée du travail sur le modèle de **machine learning** et les problèmes rencontrés par l'équipe de développement de l'**application**, le développeur **IA** est venu en renfort une fois une première approche du modèle terminée.

Le **RACI** une fois modifié montre une répartition des tâches telle que :

Tâche récapitulative	Tâche	Rokaya	Noémie	Marc	Louna
Emploi du temps	Planif réunions	I	I	I	R, A
	Planif tâches	I	I	I	R, A
	Planif deadlines	I	I	I	R, A
Recherches	Recherches métabolisme	I	R	I	A
	Recherches glycémie	I	R	I	R A
	Recherches normes	R	I	R	A
	Familiarisation vocabulaire projet	R	R	R	R A
	Recherches capteurs CGM	I	I	I	R A
	Recherches RGPD	R	I	R	A
	Présentation des recherches	R	R	R	R A
Livrables documentés	Compte-rendu recherches	C	C	C	R A
	Note de cadrage	C	C	C	R A
	Dossier de conception	C	C	C	R A
	Rapport d'avancement	C	C	C	R A
	Rapport projet final	C	C	C	R A
	Soutenance	R	R	R	R A
	Privacy Impact Assessment	C	I	C	R A
Conception	Diagramme use cases	A	I	R	R
	Recherche et identification des technos	R A	R	R	R
	Recherche et identification des protocoles d'accès aux capteurs	I	R	C	R A
	Architecture	A	C	R	C
	Recherches projets avec stack technique choisie	R A	R	R	R
	Diagrammes de classe	R A	C	R	C
	BDD	C A	R	R	C
Analyse du marché	SWOT de BodySynch	RA	I	I	C
	Première approche des scénarios de démonstration	RA	I	I	C
	Préparation de sondages qualité client	RA	I	I	C
	Analyse concurrentielles	RA	I	I	C
	Analyse des interfaces des concurrents	RA	I	I	C
B2B	Navigation	C	C	C	R A
	Ergonomie	C	C	C	RA
	CRUD users	C	C	C	RA
	Connexion entre utilisateurs	C	C	C	RA
	Recherche utilisateurs	C	C	C	RA
B2C	Affichage du profil d'un utilisateur du B2C	C	C	C	RA
	Navigation	C	RA	R	I
	Ergonomie	C	RA	R	I
	CRUD users	C	RA	R	I
	Affichage de la glycémie	C	RA	R	I
Modèle IA	Page pour les repas	C	RA	R	I
	Page pour les activités sportives	C	RA	R	I
	Mise en forme des données	I	I	RA	I
	Choix du modèle	I	I	RA	I
	Entraînement du modèle	I	I	RA	I
Tâches adjointes	Prédiction et évaluation	I	I	RA	I
	Etablissement des objectifs de sprint	C	C	C	RA
	Prise de contact avec les partenaires	R	I	I	A
	Répartition des tâches selon les objectifs de sprint	C	C	C	RA
	Contrôle des livrables	I	I	I	RA
	Formations techniques	R	R	R	RA

Au cours des sprints, les tâches ont beaucoup évolué en fonction des besoins, de l'avancée et des nouvelles données de développement projet, mais la répartition des tâches à chaque début de sprint se faisait dans le respect de cette répartition initiale. Le seul changement de taille à considérer est la venue en renfort du développeur **IA** sur les interfaces de l'**application**.



## Identification des ressources et des risques projet

### Ressources

Un capteur **CGM** est mis à disposition de l'équipe projet, son achat ayant été demandé et validé par ISIS. L'équipe projet dispose également des ressources allouées par l'école au développement : matériel informatique, salles de travail, et heures dédiées à la réalisation du projet tutoré.

Les ressources humaines se constituent de l'équipe, du tuteur projet Adrien Defossez (contact : [adrien.defossez@univ-jfc.fr](mailto:adrien.defossez@univ-jfc.fr)), et du client et superviseur projet Rayan Oudbib (contact : [rayan.oudbib@etud.univ-jfc.fr](mailto:rayan.oudbib@etud.univ-jfc.fr)), assisté par Olivier Gibaud (contact : [oliviergibaud31@gmail.com](mailto:oliviergibaud31@gmail.com)).

L'équipe projet se constitue de Louna Bashti, responsable projet, Rokaya Rezouki, responsable technique, Mark Ezéchiél Bakayoko, développeur fullstack (spécialisation **IA** et **ML**), et Noémie Lipovac, développeuse fullstack (spécialisation IoT et connectivité).

Les réunions sont planifiées via Discord, et tout document utile est envoyé par mail à l'adresse donnée.

Les ressources logicielles se constituent des IDE VSCode et IntelliJIDEA, de l'environnement de développement low/no-code **FlutterFlow**, du service de **BaaS Firebase** se constituant d'une gestion de l'authentification, une gestion du domaine, une base de données temporelles et une base de données modèle clé-valeur pour le stockage des utilisateurs, Zoom en cas de communication urgente et SCOUT/Discord pour communiquer avec le client.

### Risques

#### Risques projet

Tableau de recensement des risques

Type de risque	Risque	Code risque
Matériel	Impossibilité de travailler sur sa machine à cause d'un bug d'IDE	RM1
	Ressources insuffisantes pour la conception d'une <b>application</b>	RM2
	Compétences insuffisantes	RM3
	Capteur non fonctionnel	RM4
Temporel	Mauvaise estimation de la tâche	RT1
	Imprévu humain (accident, maladie)	RT2
	Impossibilité de concilier les plannings	RT3
Humains	Discordance de l'équipe	RH1

	Indisponibilité d'un membre de l'équipe pour une longue période	RH2
--	---	-----

### Matrice des risques

Risques négligeables/Risques importants

Impact/Probabilité	1	2	3	4	5	6	7	8	9	10	Fort impact
1									RH2		
2					RM2		RH1	RT3			
3						RT2					
4				RM4				RM3			
5		RM1					RT1				
6											
7											
8											
9											
10											
Forte probabilité											

### Solutions proposées

**Risques temporels** : Prévoyance des plannings à l'avance avec répartition des tâches la moins bloquée possible afin de pouvoir assurer des remplacements efficaces ou une marge totale importante

**RM1** : Étudier l'emploi du temps d'ISIS pour avoir toujours accès à une machine, utiliser un système de partage de données (GitHub/GitLab) pour se faciliter les transferts de code

**RM2 et RM3** : Communication avec les professeurs en charge de l'enseignement des matières/les élèves mieux qualifiés pour recevoir leurs conseils sur les sujets difficiles ou demandant des ressources extérieures (validation du programme, des diagrammes, aide à la résolution des problèmes, etc.)

**RM4** : Emploi du système de simulation de données POSTMAN ou du jeu de données de secours fourni par le client

### Risques reliés à la sécurité des données

La gestion des données de santé est un enjeu majeur du projet. De ce fait, un **PIA** a été mis en œuvre au cours de la phase de conception du projet pour comprendre réellement quelles sont les données employées, quels sont les risques associés et comment en gérer les conséquences.

L'évaluation de la gestion des données nous a permis de constater le nombre de mesures qu'il faudrait implémenter dans les trois risques principaux concernant la gestion des données : L'accès illégitime à des données, les modifications non désirées de données et la perte de données. Les données étant de santé, elles sont d'autant plus critiques, d'où le fort impact du risque d'accès illégitime.

Le diagramme Figure 12 permet de situer les risques liés aux données sur une matrice vraisemblance/gravité. Elle a été faite en considérant le cas le plus grave, ce qui explique les fortes sévérités représentées ici : La gravité de l'accès aux données, notamment, est considérée comme maximale car perte du secret médical : Cela reste des données de santé, soit des données parmi les plus convoitées. Il faudra donc, au cours du développement, porter une attention toute particulière à la protection des données.

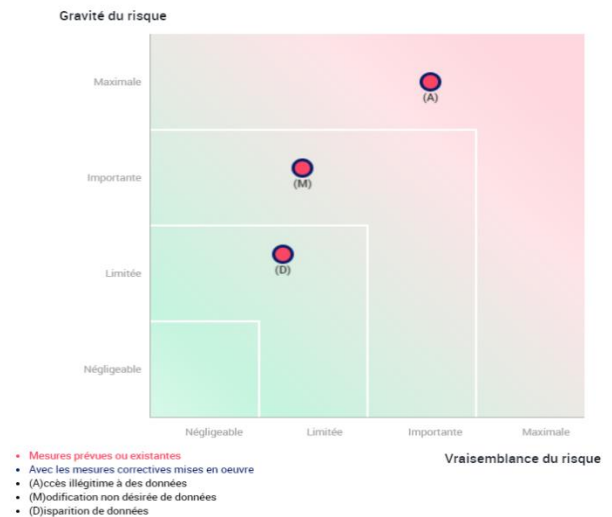


Figure 12 : Positionnement des risques sur une matrice gravité-vraisemblance

Le même risque se pose pour la modification non-désirée de données, qui s'avère cependant plus limitée en termes de vraisemblance puisque, pour modifier les données, il faut pouvoir y accéder. Les utilisateurs n'auront accès qu'à la modification de leurs données personnelles : celles récupérées par les capteurs seront verrouillées.

Le graphique Figure 13 quant à lui représente la gestion du risque, avec les impacts potentiels de chaque risque, les menaces menant à chaque risque, les différents acteurs soumis à ces risques et enfin, les mesures existantes et celles qu'il faudra immédiatement mettre en place au cours du développement pour être conforme à la norme 82304.

On peut constater, notamment, que les acteurs les plus à risque pour les données sont soit l'administrateur informatique, soit un attaquant externe : Il faudra donc protéger ces données de toute agression extérieure et prévoir au maximum de la sauvegarde de données, ainsi qu'un encadrement le plus strict possible du mode administrateur.

De même, les risques impactent surtout l'utilisateur

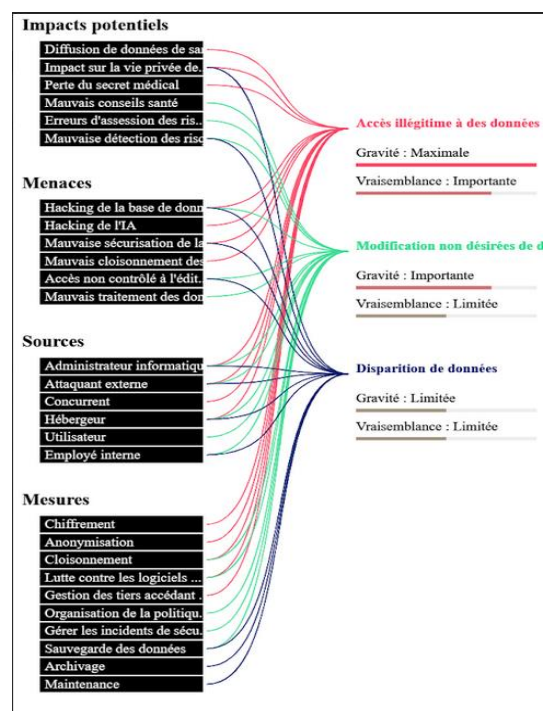


Figure 13 : Diagramme risques/solutions

: Outre une perte de secret médical, on doit envisager un impact sur la vie privée de l'utilisateur, voire des erreurs de détection des risques sur sa santé. Gérer ces conséquences devient donc capital à notre projet.

### Problèmes rencontrés et leur accordance avec les risques

Les problèmes rencontrés ont surtout été de l'ordre technique, bien que de petites erreurs de communication au sein de l'équipe se soient multipliées, aggravant le poids des ennuis.

En termes de soucis techniques, nous avons notamment eu des problèmes de communication graves avec **l'API** du capteur choisi, entraînant un retard d'un mois : Le retard étant dû à la découverte successive de différents obstacles.

Le premier obstacle était lié à la communication avec **l'API** : Ayant changé de stack technique après le choix de **l'API**, nous nous sommes rendus compte que les ressources fournies en complément de cette **API** pour permettre la connexion n'étaient pas adaptées à **FlutterFlow**, nous obligeant à manipuler ces ressources pour correspondre à l'interface où chercher un autre moyen d'accès (autre **API** ou autre script de connexion) : Une fois un moyen trouvé, le mettre en œuvre sur **FlutterFlow** s'est révélé impossible, le service étant barré par un paywall que nous n'avons pu franchir qu'en milieu de mars. Date avant laquelle nous nous sommes rendus compte en creusant l'accès pur à l'API grâce aux scripts fournis en ressource que **l'API** choisie étant développée par un tiers, sa mise à jour était très aléatoire et pouvait être bloquée par une mise à jour du système d'Abbott même. Il est à noter que cette constatation entraînait le risque très important que même dans le cas où ce problème se résolvait, les headers des appels **API**, notamment la version de l'appli LibreLinkUp, pouvaient facilement devenir obsolètes et non entretenus correctement.

Plusieurs **API** ont été testées pour répondre à ce problème : **l'API CGM** basée sur NodeJs @diakem, **l'API CGM** fournie par NightScout et **l'API GGM** fournie par TryVital, toutes identifiées pendant la phase de recherche et de conception. Chacune de ces **API** nous a fait rencontrer un problème différent : **l'API** @diakem était fournie par un tiers et les ressources fournies n'étaient plus fonctionnelles ni à jour, tout comme **l'API** NightScout, et TryVital s'est révélée être payante, demandant des ressources que nous n'avions pas.

Enfin, devant l'échec de ces trois pistes, le support d'Abbott a été contacté le 29/02/2024 : à la date du présent rapport, aucune réponse n'a été obtenue.

Ce problème est relié aux risques RM2, RT1 et RM4 : ressources insuffisantes et capteur non fonctionnel. Pour pallier ce risque et permettre un test des différentes fonctions de l'**application** sans avoir cette communication avec le capteur, un jeu de données glycémiques fourni par le client a été utilisé, et nous avons monté un jeu de données utilisateur test. Cette organisation était faite pour pouvoir développer un maximum des fonctions, afin que, le jour où une solution serait trouvée par la prochaine équipe, ce soit le seul gros point à gérer.

Plusieurs problèmes ont été rencontrés en ce qui concerne la gestion des données, mineurs cependant : Il s'agissait notamment de gérer les connexions conditionnelles et l'accès aux données de chaque utilisateur, selon le **PIA** et les contraintes projet. Actuellement, un utilisateur enregistré dans **FireBase** peut se connecter aux deux **applications** : C'est très

probablement un souci lié à la fonction d'authentification de **FireBase** même. **FlutterFlow** ne permet pas de gérer facilement les connexions conditionnelles, ce qui conduit à envisager plusieurs solutions : L'une, mise en œuvre dans le **B2B** mais encore imparfaite, consiste en une attribution et une vérification de rôles dans la base de données. Les rôles, donnés par les *spécifications* projet, seraient vérifiés à chaque tentative de connexion : Une absence de correspondance entraînerait un échec de la connexion. La gestion de cette solution via **FlutterFlow** est cependant compliquée à implémenter, et si actuellement la fonction existe, elle n'est pas utilisée, les tests ayant révélé des problèmes dans la vérification du rôle.

Un autre problème lié à la confidentialité des données est celui de connexion entre **applications**, ainsi qu'entre utilisateurs. Il est moins technique que légal, et consiste en la gestion des données utilisateurs visibles par un coach qui n'est pas connecté avec un particulier : L'objectif est de limiter ces données au maximum à un nom d'utilisateur (et potentiellement une photo de profil, si elle existe). La fonction de recherche simple implémentée par **FlutterFlow** permet de filtrer par nom d'utilisateur et donc de limiter au maximum l'affichage des données d'utilisateur lambda pour un coach, mais l'objectif, à terme, est de suivre un système similaire à celui de Discord, par demande d'ami pouvant être refusée sur la seule base d'un nom d'utilisateur. Cet objectif n'a pas été atteint pendant le temps de développement, mais a été noté en perspective d'évolution future.

Enfin, la gestion de beaucoup des risques a été liée à la communication entre membres de l'équipe projet : Il fallait notamment prendre un temps de planifications de réunion, de vérification des livrables et de l'avancement du projet et de nouvelles des différents acteurs de manière très régulière pour pouvoir tenir les sprints. Cette communication n'a pas été optimale, entraînant des absences à des réunions de sprint ou de présentation d'avancement, ou des retards de livrables entraînés par une mauvaise communication sur les problèmes rencontrés ou des réponses tardives. Ces problèmes, liés au risque RT3, ont pu être gérés lorsque l'accumulation a commencé à atteindre des seuils importants, mais entre le manque de temps et des plannings très aléatoires ou mal conciliés, cela a contribué à une prise de retard.

Cette prise de retard a pu être rattrapée, sauf pour ce qui est du problème technique décrit précédemment ; dans ce cas précis, il s'agissait de raisons hors du contrôle de l'équipe.

#### Communication avec les différents acteurs

L'équipe était en communication constante entre elle en distanciel, et via des réunions régulières toutes les semaines au moins pendant les heures dédiées au projet tutoré. La communication avec les clients projet se faisait de manière fréquente, entre les réunions d'information au cours du premier semestre, à une fréquence d'une toutes les deux à trois semaines, et celles de clôture de sprint, à une fréquence de toutes les deux semaines. De même, des informations étaient fréquemment échangées via Discord entre les réunions.

La communication avec le tuteur projet s'est limitée à une réunion pendant la phase de conception, puis un envoi régulier des comptes-rendus de réunion client pour donner une idée de notre avancée et de la fréquence de notre prise de contact. Cette organisation avait pour but de rendre compte à ISIS de manière efficace tout en consacrant un maximum de temps à la technique et à la communication avec le client, pour le bien du bon déroulement du projet.

Les livrables ont également été contrôlés par le client avant chaque envoi à ISIS, incluant le rapport d'avancement, le compte-rendu de recherches et le présent rapport projet.

## Bilan et perspectives

### Bilan

#### État d'achèvement du projet

Les deux **applications** comprennent actuellement toutes leurs fonctions essentielles en lien avec le [backend](#), bien que des ajustements soient nécessités pour pouvoir affirmer que le **MVP** est prêt à être mis sur le marché ; On ne peut cependant pas parler d'un objectif atteint à cause du problème rencontré lors de la recherche de connexion au capteur **CGM**, qui n'a pu être résolu pendant le développement. Le modèle **IA** a été pensé, développé et est prêt à être implémenté, l'interopérabilité avec l'**application** est cependant inexistante : Les données de l'**application** ne sont pas traitées par l'**IA** pour le moment.

#### Difficultés rencontrées et solutions

Les principales difficultés rencontrées étaient de l'ordre de gestion des ressources : Le problème en lien avec la connexion au capteur était la principale, et comme elle se révélait capitale au développement de la **PoC** interopérabilité, elle a pris beaucoup de temps à l'équipe projet. Il fallait également gérer sa planification, entre les différentes obligations de l'équipe et celles apportées par l'emploi du temps projet. La méthode agile s'est révélée être la solution la plus efficace pour pallier les difficultés de planification et les engagements de l'équipe, mais son **application** était expérimentale et autant basée sur l'apprentissage d'une nouvelle méthode que sur le bon déroulement du projet, ce qui a causé de nombreuses erreurs de jugement. De même, la communication entre différents membres de l'équipe se révélait parfois compliquée, et pas toujours au niveau de celle exigée pour réussir une planification en agile.

L'équipe a dû développer une grande capacité d'adaptation et de résolution rapide des problèmes afin de pouvoir se maintenir en course, et les problèmes rencontrés pendant le projet ont été une source d'apprentissage sur ce qu'était la gestion d'une équipe, d'un projet d'innovation avec des limites serrées et comment optimiser la priorisation des tâches pour arriver au produit le plus avancé possible.

## Perspectives

### Quels sont les prochains points à aborder ?

Le principal point à aborder est de pouvoir s'assurer de la connexion entre le **CGM** et l'**application**. Reprendre le contact initié avec le support d'Abbott pour pouvoir trouver une solution, ou réinitier les recherches pour trouver une **API** viable et utilisable avec **FlutterFlow**, sont deux options possibles. De même, réfléchir aux différentes fonctionnalités optionnelles de l'**application**, comme la possibilité d'instaurer un chat entre utilisateurs, est aussi une perspective d'évolution. On peut également inclure le début de la réflexion autour de l'implémentation de l'**IA** développée dans l'**application**, afin de faire en sorte qu'elle traite les données, et que les résultats de ses prédictions puissent être affichés dans l'**application**. Il faudra également achever de développer le design de l'**application** à partir des maquettes fournies par le client, afin d'obtenir une charte graphique uniforme, tout comme affiner les différentes fonctionnalités en fonction des retours clients afin de se rapprocher au maximum de la version finale.

Une autre piste à réfléchir serait d'étudier la précision du temps réel, afin d'éviter au maximum le décalage temporel entre la lecture des constantes via le capteur et la réception des données sur l'application : Analyser le système de recalage du capteur **CGM** en cours d'étude est une première étape vers un écart minimum.

### Quelles sont les perspectives d'amélioration à terme ?

A terme, l'**application** doit pouvoir offrir une sécurité maximale et une expérience utilisateur confortable. Plusieurs perspectives envisagées sont l'établissement d'un système de gamification pour UPE, qui pourra disposer de mini-objectifs et de succès atteignables via différents paramètres, comme la stabilisation de sa **glycémie**, la régularité de ses activités sportives ; On peut également songer à une connexion à l'**application** via compte Google ou Apple. De même, l'objectif à terme est de pouvoir utiliser un capteur **CGM** et les données qu'il recueille sans passer par l'**application** reliée à l'outil : Cela permettrait d'étendre les perspectives d'utilisation aux clients en possession d'un modèle qui n'a pas été traité au cours du développement, comme les capteurs DEXCOM. Au-delà de ça, on peut également considérer une connectivité avec d'autres types de capteurs de surveillance de constantes de santé, comme une montre connectée.



## Annexes

### WBS et RACI originels

Tâche récapitulative	Tâche	Nomenclature	Dépendances
Emploi du temps	Planif réunions	A1	/
	Planif tâches	A2	/
	Planif deadlines	A3	/
Recherches	Recherches métabolisme	B1	
	Recherches glycémie	B2	
	Recherches normes	B3	
	Familiarisation vocabulaire pr	B4	
	Recherches capteurs CGM	B5	
	Recherches RGPD	B6	
	Présentation des recherches	B7	B1, B2, B3, B5
Livrables documentés	Compte-rendu recherches	C1	B7
	Note de cadrage	C2	B7
	Dossier de conception	C3	D1, D4, D6, D7
	Rapport d'avancement	C4	C3
	Rapport projet final	C5	G7
	Soutenance	C6	G7
	Privacy Impact Assessment	C7	B6, B3
Conception	Diagramme use cases	D1	C1
	Recherche et identification de	D2	C1
	Recherche et identification de	D3	C1
	Architecture	D4	D2, D3
	Recherches projets avec stack	D5	D4
	Diagrammes de classe	D6	D1, D2, D3
	BDD	D7	D1, D2, D3
Tests POC	Couche collection de données	F1	E2
	Couche modèle prédictif	F2	E3
	Couche frontend	F3	E4
	Interopérabilité	F4	E5
	Développement sécurité	F5	E6
	Intégration	F6	E2, E3, E4, E5, E6
MVP	Navigation	G1	E2
	Ergonomie	G2	E2
	"Home"	G3	E2
	"MyForm"	G4	E2
	"Blog"	G5	E2
	"Profile"	G6	E2
	"More"	G7	E2
Tests MVP	Navigation	H1	G1
	Ergonomie	H2	G2
	"Home"	H3	G3
	"MyForm"	H4	G4
	"Blog"	H5	G5
	"Profile"	H6	G6
	"More"	H7	G7
	Intégration	H8	H1, H2, H3, H4, H5, H6, H7



Tâches	Rokaya	Noémie	Marc	Louna
Mise en forme des livrables	R	R	R	R;A
Rendu des livrables	R	I	I	A
Communication (tuteur et client)	R; A	I	I	I
Expression du besoin	R; A	R	R	R
Analyse du besoin	R; A	R	R	R
Organisation des réunions	R; A	I	I	R
Compte-rendu réunions	A	R	R	R
Planning	I	I	I	R;A

Tâche récapitulative	Tâche	Rokaya	Noémie	Marc	Louna
Emploi du temps	Planif réunions	I	I	I	R, A
	Planif tâches	I	I	I	R,A
	Planif deadlines	I	I	I	R,A
Recherches	Recherches métabolisme	I	R	I	A
	Recherches glycémie	I	R	I	R A
	Recherches normes	R	I	R	A
	Familiarisation vocabulaire projet	R	R	R	R A
	Recherches capteurs CGM	I	I	I	R A
	Recherches RGPD	R	I	R	A
	Présentation des recherches	R	R	R	R A
Livrables documentés	Compte-rendu recherches	C	C	C	R A
	Note de cadrage	C	C	C	R A
	Dossier de conception	C	C	C	R A
	Rapport d'avancement	C	C	C	R A
	Rapport projet final	C	C	C	R A
	Soutenance	R	R	R	R A
	Privacy Impact Assessment	C	I	C	R A
Conception	Diagramme use cases	A	I	R	R
	Recherche et identification des technos	R A	R	R	R
	Recherche et identification des protocoles d'accès aux capteurs	I	R	C	R A
	Architecture	A	C	R	C
	Recherches projets avec stack technique choisie	R A	R	R	R
	Diagrammes de classe	R A	C	R	C
	BDD	C A	R	R	C
	Formation technos	R	R	R	R A
POC	Couche collection de données	I	R	C	R A
	Couche modèle prédictif	R A	I	R	I
	Couche frontend	R A	R	R	R
	Interopérabilité	R A	R	R	R
	Développement sécurité	RA	R	R	R
Tests POC	Couche collection de données	I	R	C	R A
	Couche modèle prédictif	R A	I	R	I
	Couche frontend	R	I	R	A
	Interopérabilité	I	R	C	RA
	Développement sécurité	R	I	R	A
	Intégration	R	R	R	RA
MVP	Navigation	RA	I	R	I
	Ergonomie	RA	I	R	I
	"Home"	RA	I	R	I
	"MyForm"	RA	I	R	I
	"Blog"	RA	I	R	I
	"Profile"	RA	I	R	I
	"More"	RA	I	R	I
Tests MVP	Navigation	RA	I	R	I
	Ergonomie	RA	I	R	I
	"Home"	RA	I	R	I
	"MyForm"	RA	I	R	I
	"Blog"	RA	I	R	I
	"Profile"	RA	I	R	I
	"More"	RA	I	R	I
	Intégration	RA	I	R	I

## Extraits du CDCM

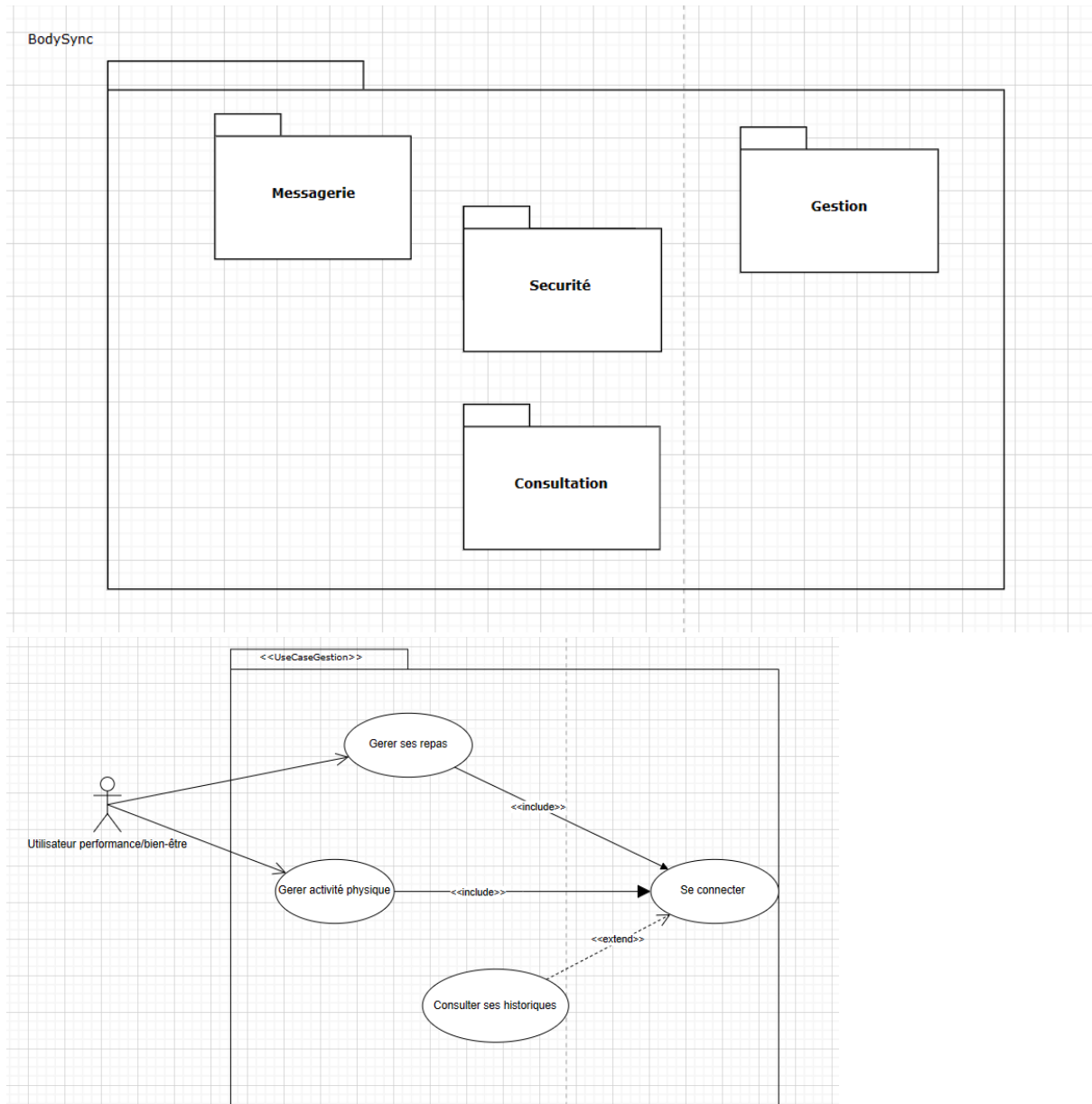
Abbreviation	Priority	Description
--------------	----------	-------------

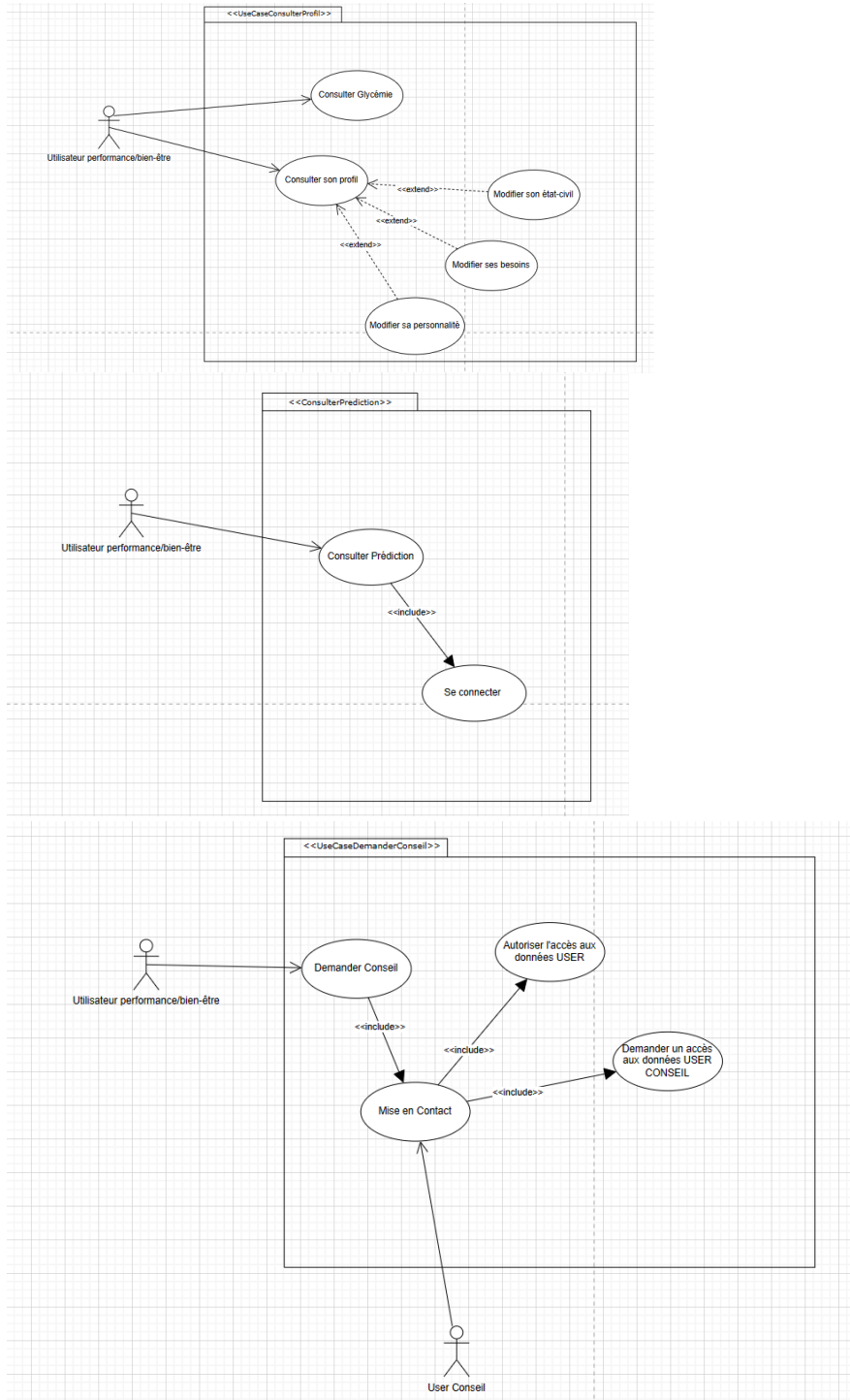
<b>M</b>	<b>Must have</b>	These provide the <b>minimum usable subset of requirements</b> which the project guarantees to deliver, otherwise the project is a failure. These may be defined using some of the following: - No point in delivering on target date without this; if it were not delivered, there would be no point deploying the solution on the intended date - Not legal without it - Unsafe without it - Cannot deliver a viable solution without it
<b>S</b>	<b>Should have</b>	Should Have requirements are defined as: - <b>Important but not vital</b> - May be painful to leave out, but the solution is still viable - May need some kind of workaround, e.g., management of expectations, some inefficiency, an existing solution, paperwork etc. The workaround may be just a temporary one
<b>C</b>	<b>Could have</b>	Could Have requirements are defined as: - <b>Wanted or desirable but less important</b> - Less impact if left out (compared with a Should Have)
<b>W</b>	<b>Will not have</b>	These are requirements which the project team has agreed <b>will not be delivered</b> (as part of this timeframe), at least not this time around. They are nonetheless recorded because they help clarify the scope of the project and because they could be delivered in a future version.

ID	Designation / feature	Priority			
General requirements					
	Expérience utilisateur révolutionnaire, intuitive et prestigieuse.	M			
	Interopérabilité avec différents biocapteurs, montres connectées et CGM.	M			
	Synchronisation efficace des données entre l'application mobile, les serveurs et les biocapteurs.	M			
	Sécurité des données personnelles et de santé, conformément aux RGPD.		S		
BodySync : Page Home					

	<p>Afficher en temps réel l'évolution journalière de la <b>glycémie</b> provenant d'un CGM</p> <p>Présenter des indicateurs personnalisés pour afficher l'historique et les actions à venir (repas, entraînements)</p> <p>Présenter et accéder aux actualités de l'entreprise</p> <p>Présenter l'historique des repas et activité physique avec les détails essentiels. Chaque élément doit permettre d'accéder à la sous-page "More" pour plus d'informations.</p> <p>Permettre d'ajouter un repas en utilisant l'appareil photo avec reconnaissance par ML et visualisation des macro en RA. Alternativement, les utilisateurs peuvent scanner un code-barre ou ajouter manuellement un repas.</p> <p>Permettre d'ajouter une séance d'entraînement, soit automatiquement en se connectant à Apple Health, soit manuellement.</p> <p>Permettre d'ajouter la connexion à un biocapteur via bluetooth ou NFC.</p>	M M M M M	S S		
<b>BodySync : Sous-Page More</b>					
	Fournir des informations détaillées sur une activité physique ou un repas sélectionné.	M			
<b>BodySync : Page MyForm</b>					
	Accéder à une liste de profils de spécialistes en santé métabolique disponibles pour un suivi professionnel.	M			
	Spécifier ses habitudes sportives et nutritionnelles, ses objectifs et ses spécificités.	M			
	Accéder à un bilan de son métabolisme avec l'évolution des performances et des métriques biologiques au fil du temps, un récapitulatif des habitudes nutritionnelles et un profil glycémique ambulatoire (PGA) transférable à son docteur.		S		
	Conseiller sur les prochaines actions à mener (choix d'un spécialiste, quantité de sport, nutrition...) de manière personnalisée (historique, nutrition, sport).	M			
<b>BodySync : Page Blog</b>					
	Donner accès à un blog éducatif en ligne contenant des cours, des articles, des interviews et des études sur la santé métabolique.	M			
<b>BodySync : Page Profile</b>					
	Configurer les paramètres de son profil, notamment les connexions aux biocapteurs, les objectifs physiques et les habitudes alimentaires en cours, les préférences d'entraînements etc.	M			

## Détails de la modélisation des Use Cases





## GANTT originel

