



République Tunisienne
Ministère de l'enseignement supérieur
Et de la recherche scientifique
Direction Générale des Etudes Technologiques
Institut Supérieur des Etudes Technologiques de
Kasserine



Rapport de Projet de Fin d'Études

Présenté en vue de l'obtention du
Licence Appliquées en Technologies de L'informatique
Spécialité : Développement des Systèmes d'Information (DSI)

Par

Chayma Jbeli

Jihed Jebri

Lifestyle Manager

Encadrant académique :

Monsieur Saber MESSAOUDI,

Madame Layla NAKKAI

Dédicaces

À ma mère,

Tu es la lumière qui a guidé chacun de mes pas, la source infinie d'amour et de soutien. Ta force et ta tendresse ont forgé la personne que je suis aujourd'hui, et je t'en suis éternellement reconnaissante.

À mon père,

Ton amour inconditionnel, ta sagesse et ta patience sont pour moi des exemples à suivre. Merci pour tous les sacrifices que tu as faits pour moi, et pour m'avoir toujours encouragé à poursuivre mes rêves.

À mes frères,

Ceux qui ont toujours cru en moi et qui ont enduré mon tempérament. Je vous souhaite une abondance de bonheur et de réussite. Jusqu'à ce que nous nous revoyions.

À mes amis,

A mes ami(e)s, à qui je souhaite le succès, pour l'amitié qui nous a toujours unis.

Chayma Jbeli

Dédicaces

J'offre ce modeste travail,

A mes chers parents,

Mais aucune dédicace ne serait témoin de mon profond amour, mon immense gratitude et mon plus grand respect, car je ne pourrais jamais oublier la tendresse et l'amour dévoué par lesquels ils m'ont toujours entourer depuis mon enfance.

Je dédie aussi ce modeste travail,

A tous mes amis, et à tous ceux que j'aime et à toutes les personnes qui m'ont Prodigué des encouragements et se sont données la peine de me soutenir durant cette année de formation.

A mes chers formateurs et formatrices, sans aucune exception.

Sans oublier mon binôme pour son soutien moral, sa patience et sa compréhension tout au long de ce projet.

Jihed Jebri

Remerciement

Louange à dieu, qui nous a éclairés sur le chemin de la science et de la connaissance et nous a aidé à accomplir ce devoir de nous accorder le succès et l'achèvement de ce travail.

Tout d'abord, j'adresse mes meilleurs remerciements de tout mon cœur à ma famille qui m'a soutenu et n'oublier pas mes amis

Je tiens à exprimer ma profonde gratitude envers les membres du jury pour avoir pris le temps d'évaluer notre projet. Vos retours et vos conseils ont été d'une importance capitale pour son développement et son amélioration.

Un merci tout particulier à notre encadrant **Monsieur Saber Messaoudi**, dont l'accompagnement et les conseils éclairés ont été des éléments déterminants dans la réussite de ce projet. Sa disponibilité et son expertise ont été pour moi une source d'inspiration et d'apprentissage constant.

Mes plus sincères remerciements vont à **Mme Layla Nakaii** pour les précieux conseils et le temps qu'elle nous a consacré ainsi qu'à tous les professeurs qui y ont contribué directement ou indirectement.

Je souhaite également adresser mes remerciements à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet, ainsi qu'à mes proches pour leur soutien indéfectible.

Merci infiniment à chacun d'entre vous pour avoir rendu cette expérience unique et enrichissante pour moi.

Sommaire

Introduction générale.....	1
Chapitre I : Contexte générale.....	3
Introduction	4
I. Présentation de l’Institut Supérieur des Etudes Technologiques	4
II. Etude et critique de l’existant.....	5
III. Problématique.....	5
IV. Solution proposée	6
V. Méthodologie de travaille.....	6
1.1 Méthodologie Agile.....	6
1.2 Méthode Scrum	7
1.3 Equipe de travaille.....	8
Conclusion.....	9
Chapitre II : Analyse et spécification des besoins.....	10
Introduction	11
I. Capture des besoins	11
1. Indentification de l’acteur	11
1.4 Spécification des besoins fonctionnels.....	11
1.5 Spécification des besoins non fonctionnels.....	12
II. Planification de déroulement de projet.....	12
1. Backlog du produit	12
1.6 Plan de releases	13
1.7 Planification	14
1.8 Introduction UML	14

1.9	Diagramme de cas d'utilisation globale	15
	Conclusion.....	16
	Chapitre III : Conception.....	17
	Introduction	18
I.	Release1	18
1.	Sprint1 : Authentification, gérer profil et gérer calendrier.....	18
1.1	Diagramme de séquence du cas d'utilisation « s'authentifier »	18
1.2	Raffinement du cas d'utilisation « gérer profil »	19
1.3	Description textuelle du cas d'utilisation « modifier profil »	20
1.4	Diagramme de séquence du cas d'utilisation « modifier profil »	20
1.5	Raffinement du cas d'utilisation « gérer calendrier »	21
1.6	Description textuelle du cas d'utilisation « modifier profil »	21
1.7	Diagramme de séquence du cas d'utilisation « modifier profil »	22
1.8	Raffinement du cas d'utilisation « gérer calendrier »	23
1.9	Description textuelle du cas d'utilisation « Ajouter des taches ».....	23
1.10	Diagramme de séquence du cas d'utilisation « ajouter des taches ».....	24
1.11	Description textuelle du cas d'utilisation « Modifier des taches »	25
1.12	Diagramme de séquence du cas d'utilisation « modifier des taches »	26
II.	Release 2	26
1.	Sprint 2 : Gérer condition physique et gérer alimentation	26
1.1	Raffinement du cas d'utilisation « gérer condition physique ».....	26
1.2	Description textuelle du cas d'utilisation « ajouter des activités physiques » .	27
1.3	Diagramme de séquence du cas d'utilisation « ajouter des activités physiques »	28
1.4	Description textuelle du cas d'utilisation « supprimer des activités »	28
1.5	Diagramme de séquence du cas d'utilisation « supprimer des activités »	29

1.6	Raffinement du cas d'utilisation « gérer alimentation »	29
1.7	Description textuelle du cas d'utilisation « ajouter repas »	30
1.8	Diagramme de séquence du cas d'utilisation « ajouter repas ».....	31
III.	Release 3	31
1.	Sprint 3 : Gérer dépenses et consulter météo	31
1.1	Raffinement du cas d'utilisation « gestion de dépenses ».....	31
1.2	Description textuelle du cas d'utilisation « ajouter salaire »	32
1.3	Diagramme de séquence du cas d'utilisation « ajouter salaire ».....	32
1.4	Raffinement du cas d'utilisation « consulter météo »	33
1.5	Description textuelle du cas d'utilisation « ajouter payée ».....	34
1.6	Diagramme de séquence du cas d'utilisation « ajouter payée ».....	34
	Conclusion.....	37
	Chapitre IV : Réalisation.....	38
	Introduction	39
1.	Environnement de travail	39
1.1	Environnement matériel	39
1.2	Environnement logique	40
1.3	Technologies utilisées	41
2.	Jeu d'essai et scénarios de test	43
	Conclusion.....	48
	Conclusion générale	49
	Bibliographie	50

Liste des tableaux

Tableau 1 :équipe de travaille	8
Tableau 2 : Backlog du produit de l’application	13
Tableau 3 : plan de releases.....	14
Tableau 4 : tableau des priorités.....	16
Tableau 5 description textuelle du cas d’utilisation « s’authentifier ».....	18
Tableau 6 : description textuelle du cas d’utilisation « modifier profil »	20
Tableau 7 : description textuelle du cas d’utilisation « modifier profil »	22
Tableau 8 : Description textuelle du cas d’utilisation « Ajouter des taches »	24
Tableau 9 : description textuelle du cas d’utilisation « modifier taches »	25
Tableau 10 : Description textuelle du cas d’utilisation « ajouter des activités physiques »	27
Tableau 11 : Description textuelle du cas d’utilisation « supprimer des activités »	28
Tableau 12 : Description textuelle du cas d’utilisation « ajouter repas ».....	30
Tableau 15 : Description textuelle du cas d’utilisation « ajouter salaire ».....	32
Tableau 16 : Description textuelle du cas d’utilisation « ajouter payée ».....	34
Tableau 17 : descriptions des classes	35
Tableau 18 : Les caractéristiques de poste de travail.	39

Liste des figures

Figure 1:Architecture de Scrum.	7
Figure 2 : Diagramme de cas d'utilisation globale.....	15
Figure 3 : Diagramme de séquence relatif au cas d'utilisation « s'authentifier ».....	19
Figure 4 : Raffinement de cas d'utilisation « gérer profil ».....	19
Figure 5 : Diagramme de séquence du cas d'utilisation « modifier profil ».....	21
Figure 6 : Raffinement du cas d'utilisation « gérer calendrier ».....	21
Figure 7 : Diagramme de séquence du cas d'utilisation « modifier profil ».....	23
Figure 8 : Raffinement du cas d'utilisation « gérer calendrier ».....	23
Figure 9 : Diagramme de séquence du cas d'utilisation « ajouter des tâches ».....	25
Figure 10 : Diagramme de séquence du cas d'utilisation « modifier des tâches ».....	26
Figure 11 : Raffinement du cas d'utilisation « gérer condition physique ».....	27
Figure 12 : Diagramme de séquence du cas d'utilisation « ajouter des activités physiques ».....	28
Figure 13 : Diagramme de séquence du cas d'utilisation « supprimer des activités ».....	29
Figure 14 : Raffinement du cas d'utilisation « gérer alimentation ».....	30
Figure 15 : Diagramme de séquence du cas d'utilisation « ajouter repas ».....	31
Figure 19 : Raffinement du cas d'utilisation « gestion de dépenses ».....	32
Figure 20 : Diagramme de séquence du cas d'utilisation « ajouter salaire ».....	33
Figure 21 : Raffinement du cas d'utilisation « consulter météo ».....	33
Figure 22 : Diagramme de séquence du cas d'utilisation « ajouter payée ».....	34
Figure 23 : Diagramme de classe.....	35
Figure 24 : Visual Paradigm.....	40
Figure 25 : IntelliJ IDEA.....	40
Figure 26 : MySQL.....	41
Figure 27:API openWeather.....	41

Figure 28: API Nutritionx Track	41
Figure 29: java.....	42
Figure 30: Spring Boot.....	42
Figure 31: Angular	43
Figure 32:TypeScript.....	43
Figure 33: interface de connexion	44
Figure 34: interface d'inscription.....	44
Figure 35: interface d'accueil	45
Figure 36: interface météo.....	45
Figure 37: Dashboard	46
Figure 38: interface exercice	46
Figure 39: interface calendrier	47
Figure 40: interface d'alimentation.....	47
Figure 41: interface d'alimentation.....	48

Introduction générale

Dans le cadre de notre formation de licence en technologie informatique, à l'Institut Supérieur des Études Technologiques de Kasserine, nous amenés, à l'issue de notre cursus, à réaliser un projet de fin d'études.

Ce projet est une occasion importante pour enrichir nos acquis théoriques avec les aspects pratiques à l'aide d'une étude, d'une conception et d'une réalisation d'une application « Lifestyle Manager »

Dans le monde actuel, la gestion efficace du temps et des activités quotidiennes est un défi constant. Les obligations professionnelles, les engagements personnels, la gestion des finances et le maintien d'une bonne santé se chevauchent souvent, créant une désorganisation qui peut entraîner stress et inefficacité.

De nombreuses applications existent pour gérer ces aspects individuellement, mais cette fragmentation complique encore plus la vie des utilisateurs, qui doivent diviser entre plusieurs outils pour suivre leurs tâches, leur budget, leur condition physique, alimentation et les prévisions météorologiques.

Notre application est conçue pour résoudre ce problème en offrant une solution tout-en-un. Notre application regroupe les fonctionnalités essentielles pour suivre les calories, gérer un calendrier, contrôler les finances et obtenir des prévisions météorologiques, le tout dans une interface intuitive.

Avec notre application, les utilisateurs peuvent centraliser et organiser toutes les facettes de leur vie quotidienne. Les rappels et notifications personnalisées aident à mieux planifier et gérer les tâches, réduisant ainsi le stress et améliorant la productivité. En simplifiant la gestion du mode de vie, notre application permet aux utilisateurs de reprendre le contrôle de leur temps et de leurs activités.

En offrant une solution intégrée et complète, notre application vise à améliorer la qualité de vie de ses utilisateurs en rendant l'organisation quotidienne plus simple et plus efficace.

Notre rapport se compose de quatre chapitres qui abordent différents aspects de notre projet

- **Le premier chapitre** : présente l'étude de l'existant, l'analyse et la spécification des besoins.
- **Le deuxième chapitre** : comporte une étude détaillée des spécifications.
- **Le troisième chapitre** : décrit la partie conceptuelle de notre projet.
- **Le quatrième chapitre** : présent les détails de l'implémentation et les interfaces de notre application.

Nous concluons finalement avec une conclusion qui met en évidence les enseignements que nous avons acquis à la suite de la réalisation de notre application, ainsi que les améliorations souhaitées.

Chapitre I : Contexte générale

Introduction

Dans ce chapitre, nous débuterons par une présentation de l'Institut Supérieur des Études Technologiques. Ensuite, nous procéderons à une analyse critique de l'existant, en mettant en avant la solution que nous proposons. Nous aborderons également une analyse approfondie des besoins de notre application. Enfin, nous exposerons la méthodologie de travail que nous avons adoptée tout au long du développement de notre solution.

I. Présentation de l'Institut Supérieur des Etudes Technologiques

L'Institut Supérieur des Études Technologiques (ISET) de Kasserine est un établissement d'enseignement supérieur qui joue un rôle crucial dans le réseau des ISETs. Créé pour offrir une formation technologique de pointe, il permet à ses diplômés de s'intégrer efficacement dans le monde professionnel grâce à des mécanismes d'ouverture sur l'environnement.

Fonctionnant sous la tutelle du ministère de l'Enseignement Supérieur et de la Recherche Scientifique, et plus spécifiquement sous la supervision de la Direction Générale des Études Technologiques basée à Radès, l'ISET de Kasserine est un maillon essentiel dans le paysage éducatif tunisien.

Le Département des Technologies de l'Informatique (DTI) de l'ISET de Kasserine, créé au début de l'année universitaire 2004/2005, est un pilier de l'institut. Il offre une formation de licence en technologies de l'informatique, accessible aux nouveaux bacheliers via l'orientation universitaire ou les concours de réorientation.

La formation dispensée par le DTI vise à développer chez les étudiants des compétences techniques solides et à leur fournir les connaissances théoriques nécessaires dans le domaine des technologies de l'informatique. L'objectif est de préparer efficacement les étudiants à répondre aux défis du monde professionnel et de les doter des outils nécessaires pour exceller dans le domaine de l'informatique.

En somme, l'ISET de Kasserine, grâce à son Département des Technologies de l'Informatique, joue un rôle déterminant dans la formation des futurs professionnels de l'informatique en Tunisie.[1]

II. Etude et critique de l'existant

La gestion du style de vie est un défi complexe auquel de nombreux utilisateurs sont confrontés. De nos jours, il existe de nombreuses applications visant à améliorer la gestion du mode de vie. Ces applications se concentrent souvent sur des fonctionnalités spécifiques telles que le suivi des calories, la gestion des événements, la gestion des finances personnelles et les prévisions météorologiques.

Cependant, cette spécialisation crée une fragmentation, obligeant les utilisateurs à naviguer entre plusieurs outils pour gérer différents aspects de leur vie quotidienne. Cette dispersion augmente la complexité organisationnelle et peut réduire l'efficacité globale.

Passer fréquemment d'une application à l'autre pour suivre vos exercices, planifier des rendez-vous, gérer vos finances ou consulter les prévisions météorologiques peut s'avérer fastidieux et prendre beaucoup de temps. Étant donné que chaque application possède sa propre interface, ses paramètres et ses propres notifications, les utilisateurs peuvent rapidement se sentir submergés par ce déluge d'informations et d'outils. Cette fragmentation peut entraîner une diminution de la productivité et une augmentation du stress, rendant la gestion du mode de vie moins efficace et plus complexe.

Il existe donc un besoin croissant de solutions centralisant toutes ces fonctions sur une seule plateforme, simplifiant ainsi la gestion et améliorant l'efficacité de la vie quotidienne des utilisateurs.

III. Problématique

Comment l'application 'Lifestyle Manager' peut-elle évoluer pour intégrer des fonctionnalités avancées et personnalisées qui améliorent la gestion du style de vie des utilisateurs, tout en maintenant une interface utilisateur intuitive et facile à utiliser

Comment peut-elle s'adapter aux besoins spécifiques et changeants de chaque utilisateur pour offrir une expérience véritablement personnalisée ?

Et comment peut-elle le faire tout en restant à jour avec les dernières tendances et technologies dans le domaine de la gestion du style de vie, et en assurant la compatibilité avec une variété d'autres applications et appareils ?

IV. Solution proposée

Pour répondre à ce besoin croissant de simplification et d'efficacité, nous proposons une solution innovante :

Une application intégrée qui centralise toutes les fonctions essentielles de gestion du mode de vie en une seule plateforme. Cette application permettra aux utilisateurs de suivre leurs calories, gérer leurs événements, contrôler leurs finances personnelles et consulter les prévisions météorologiques, le tout depuis une interface unique et cohérente. En regroupant ces fonctionnalités disparates, l'application réduira le besoin de naviguer entre plusieurs outils, simplifiant ainsi la gestion quotidienne.

Grâce à cette centralisation, les utilisateurs bénéficieront d'une vue d'ensemble de leur style de vie, leur permettant de mieux visualiser et analyser leurs habitudes et performances. Les rappels et notifications provenant d'une seule source réduiront les distractions et aideront à maintenir un meilleur suivi des tâches et objectifs. En offrant des visualisations de données claires et des outils de planification intégrés, l'application aidera les utilisateurs à prendre des décisions plus informées et à optimiser leur temps et leurs ressources.

En somme, cette solution vise à réduire le stress lié à la gestion du mode de vie, à améliorer la productivité et à faciliter une organisation plus harmonieuse et efficace de la vie quotidienne. En intégrant toutes ces fonctionnalités essentielles, l'application propose une approche holistique et pratique pour aider les utilisateurs à atteindre leurs objectifs personnels et professionnels de manière plus cohérente et efficace.

V. Méthodologie de travail

1.1 Méthodologie Agile

La méthodologie Agile est une approche de gestion de projet qui met l'accent sur la flexibilité, l'amélioration continue et la collaboration avec les clients. Elle est née en réponse aux limites des méthodes traditionnelles en cascade, qui entraînaient souvent de longs cycles de développement et des processus rigides. [2]

1.2 Méthode Scrum

Scrum reprend la fluidité de la méthodologie Agile et la convertit en sprints stricts de deux semaines. Ces sprints sont analysés une fois terminés et constituent la base des améliorations à apporter lors du cycle suivant. Cette méthodologie de gestion de projet s'articule autour de sprints progressifs dont la vitesse et l'efficacité augmentent à chaque fois. Grâce à cette nature cyclique du développement, les équipes sont en mesure d'améliorer leur rapidité et leur flux de travail. [3]

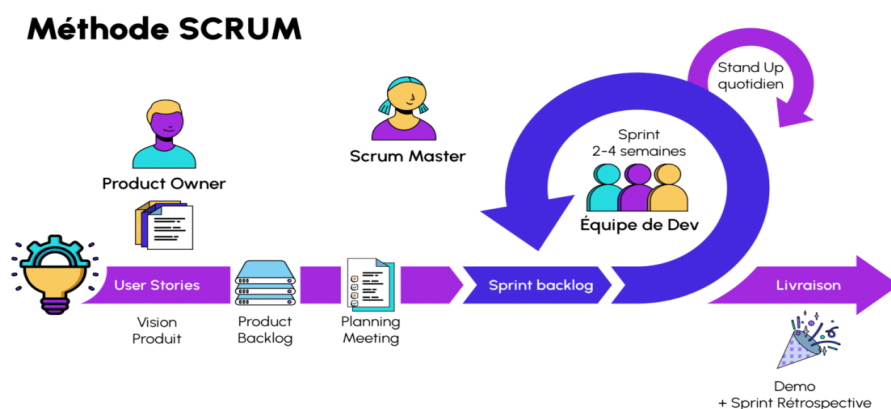


Figure 1: Architecture de Scrum.

Product Owner : est chargé de maximiser la valeur du produit issu du travail de l'équipe Scrum. Il définit clairement la vision et les objectifs du produit pour l'équipe, gère le Product Backlog et prend des décisions concernant les priorités et les fonctionnalités.

Scrum Master : est chargé de mettre en place Scrum au sein de l'équipe. Il facilite les événements Scrum, aide à résoudre les obstacles et soutient l'efficacité de l'équipe.

Équipe de développement : est composée de 3 à 9 membres et travaille sur les éléments du Product Backlog. Elle estime la charge des User Stories, développe le produit, effectue des tests et assure la livraison d'un incrément de haute qualité.

Sprint : est une période définie, généralement de 1 à 4 semaines, pendant laquelle l'équipe Scrum travaille sur un ensemble d'éléments du Product Backlog. À la fin du Sprint, un incrément de produit potentiellement livrable est produit.

Product Backlog : est une liste ordonnée des éléments nécessaires pour améliorer le produit. C'est la seule source de travail pour l'équipe Scrum. Les éléments du Product Backlog prêts à être réalisés lors d'un Sprint sont sélectionnés lors de la planification de ce dernier.

Sprint Backlog : est constitué des éléments du Product Backlog sélectionnés pour le Sprint en cours.

Daily Scrum : est une réunion quotidienne où l'équipe scrum se synchronise et ajuste son travail.

Sprint Review : est une réunion à la fin du Sprint pour inspecter l'incrément et adapter le Product Backlog.

Sprint Retrospective : est une réunion pour réfléchir sur le Sprint et améliorer les processus de l'équipe.

L'objectif de Scrum est de créer un environnement où une équipe peut travailler de manière collaborative, apprendre de ses erreurs, s'adapter aux changements et livrer le produit de la plus haute qualité possible. Il favorise la communication, la collaboration et l'amélioration continue.

1.3 Equipe de travail

Tableau 1 :équipe de travail

Rôle	Mission	Acteur
L'équipe (Team Scrum)	→Réalisation des user stories et élaboration des sprints.	Chayma Jbeli Jihed Jebri
Propriétaire du produit (Product Owner)	Présentation des caractéristiques et des fonctionnalités du produit à développer et approbation du produit à livrer. →Prioriser les fonctionnalités avant chaque planification du sprint. →Responsable test et qualité.	Chayma Jbeli
Scrum Master	→ Faciliter l'organisation de l'équipe. → S'assurer que la coopération entre les membres d'une équipe soit fluide.	Saber Messaoudi

	→ Avoir le rôle de faire en sorte que le déroulement des opérations se passe dans une ambiance la plus positive possible.	
--	---	--

Conclusion

Dans notre discussion, nous avons exploré les besoins fonctionnels et non fonctionnels d'une application de gestion de style de vie. Nous avons également examiné comment ces besoins pourraient être intégrés dans le développement logiciel Agile Scrum. En utilisant des méthodes itératives et collaboratives, l'équipe peut créer une application qui répond aux attentes des utilisateurs tout en garantissant la sécurité des données, la performance et l'accessibilité.

Chapitre II : Analyse et spécification des besoins

Introduction

Dans cette partie, nous allons préciser les besoins fonctionnels et non fonctionnels de l'application et les présenter en utilisant le diagramme de cas d'utilisation global et en fin nous allons décrire l'architecture de notre application et les environnements de développement utilisés.

I. Capture des besoins

1. Indentification de l'acteur

Dans cette partie, nous avons identifié un acteur principal pour notre application :

Utilisateur : ce dernier est chargé de créer un compte sur l'application, lui permettant ainsi d'accéder à diverses fonctionnalités telles que la consultation de son profil, l'ajout de tâches, le suivi de sa condition physique, la visualisation des dépenses, la consultation des informations d'alimentation ainsi que la vérification des prévisions météorologiques.

1.4 Spécification des besoins fonctionnels

Cette partie est réservée à la description des besoins fonctionnels que notre système doit effectuer. Ces besoins sont les suivants :

- **Gestion des tâches** : L'application offre un calendrier interactif où les utilisateurs peuvent enregistrer leurs tâches par date. Cela permet une visualisation claire de toutes les tâches à venir et aide les utilisateurs à planifier leur temps efficacement.
- **Suivi de la condition physique** : L'application devrait permettre aux utilisateurs de suivre divers aspects de leur condition physique, y compris les calories brûlées.
- **Gestion des dépenses** : L'application devrait offrir des outils permettant de suivre et de gérer les dépenses de manière efficace.
- **Gestion de l'alimentation** : L'application devrait permettre aux utilisateurs de suivre leur apport calorique et de planifier leurs repas en fonction de leurs objectifs nutritionnels. Elle devrait également calculer les calories totales de chaque repas.

- **Consultation de la météo :** L'application devrait intégrer des données météorologiques pour aider les utilisateurs à planifier leur journée en fonction des conditions météorologiques.

Notre application vise à être une solution complète pour la gestion du style de vie, offrant une gamme de fonctionnalités pour aider les utilisateurs à mener une vie plus organisée, saine et équilibrée.

1.5 Spécification des besoins non fonctionnels

- **Sécurité des Données :** La sécurité des données doit être une priorité, garantissant une protection rigoureuse des informations personnelles et financières des utilisateurs.
- **Performance :** Une réactivité rapide de l'application pour les conseils vestimentaires, le suivi des dépenses et les notifications du calendrier est cruciale pour une expérience utilisateur optimale.
- **Fiabilité :** Minimiser les temps d'indisponibilité de l'application afin de garantir une accessibilité constante pour les utilisateurs.
- **Adaptabilité :** L'application devrait fonctionner de manière fluide sur différentes plateformes, assurant une expérience utilisateur cohérente.
- **Accessibilité :** Concevoir l'application en tenant compte des besoins d'accessibilité, permettant à un large éventail d'utilisateurs de l'adopter.

II. Planification de déroulement de projet

1. Backlog du produit

L'objectif de cette approche consiste à établir une liste de user stories à développer par l'équipe. Ce dernier est établi dans sa forme initiale dès le démarrage du projet et avant de se lancer dans le premier sprint par le "Product Owner".

Ce tableau représente le Backlog du produit de l'application à développer :

Tableau 2 : Backlog du produit de l'application

Titre	Description (User story)	PRIORETE	SPRINT
Gestion profile	Modifier le profile	P1	1
Gestion des tâches	Création de tâches avec date, heure et détails. Priorisation des tâches.	P1	1
Suivi de la condition physique	Suivi des calories brûlées, des pas effectués, de la distance parcourue.	P2	2
Gestion de l'alimentation	Suivi de l'apport calorique et planification des repas.	P2	2
Consultation de la météo	Intégration des données météorologiques. Prévisions météorologiques en temps réel.	P3	3
Gestion des dépenses	Suivi des dépenses et création d'un budget.	P3	3

1.6 Plan de releases

Après avoir découpé notre travail en des sprints, nous a fallu regrouper ces sprints en des releases pour faciliter le travail. Le tableau représente la planification de nos releases en s'appuyant sur le Backlog de produit cité précédemment.

Tableau 3 : plan de releases

Release 1		Release 2		Release 3	
Sprint 1		Sprint 2		Sprint 3	
Gestion de profil	Gestion des tâches	Suivi de la condition physique	Gestion de l'alimentation	Consultation de la Météo	Consultation de la Météo
01-03/13-03	14-03/04-04	15-04/25-04	10-05/19-05	21-05/27-05	28-05/03-06

1.7 Planification

Ce plan de release comprend trois releases et trois sprints soigneusement planifiés aux dates indiquées ci-dessous. La planification détaillée de l'application "LifeStyle Manager" permet une progression structurée et efficace du développement. Chaque sprint se concentre sur un aspect spécifique de l'application, garantissant une attention dédiée et une mise en œuvre complète des fonctionnalités. Cette approche agile offre flexibilité et adaptabilité aux changements, assurant que l'application répondra de manière optimale aux besoins des utilisateurs. En fin de compte, cette planification rigoureuse représente une étape clé vers la réalisation d'une application qui améliore la gestion du style de vie de ses utilisateurs.

1.8 Introduction UML

Unified Modeling Language (UML) est un langage de modélisation visuelle standardisé utilisé dans le domaine du génie logiciel pour fournir un moyen général, de développement et intuitif de visualiser la conception d'un système. UML aide à spécifier, visualiser, construire et documenter les artefacts des systèmes logiciels. [4]

1.9 Diagramme de cas d'utilisation globale

Un diagramme de cas d'utilisation UML représente les interactions entre les utilisateurs et un système en montrant les relations entre les acteurs et les différentes actions que les utilisateurs peuvent effectuer pour atteindre un objectif dans le système. En langage de modélisation unifié (UML), un diagramme de cas d'utilisation résume les informations des utilisateurs (également appelés acteurs) et leurs interactions avec le système.[5]

5.1 Diagramme de cas d'utilisation globale

La figure présente le diagramme de cas d'utilisation global

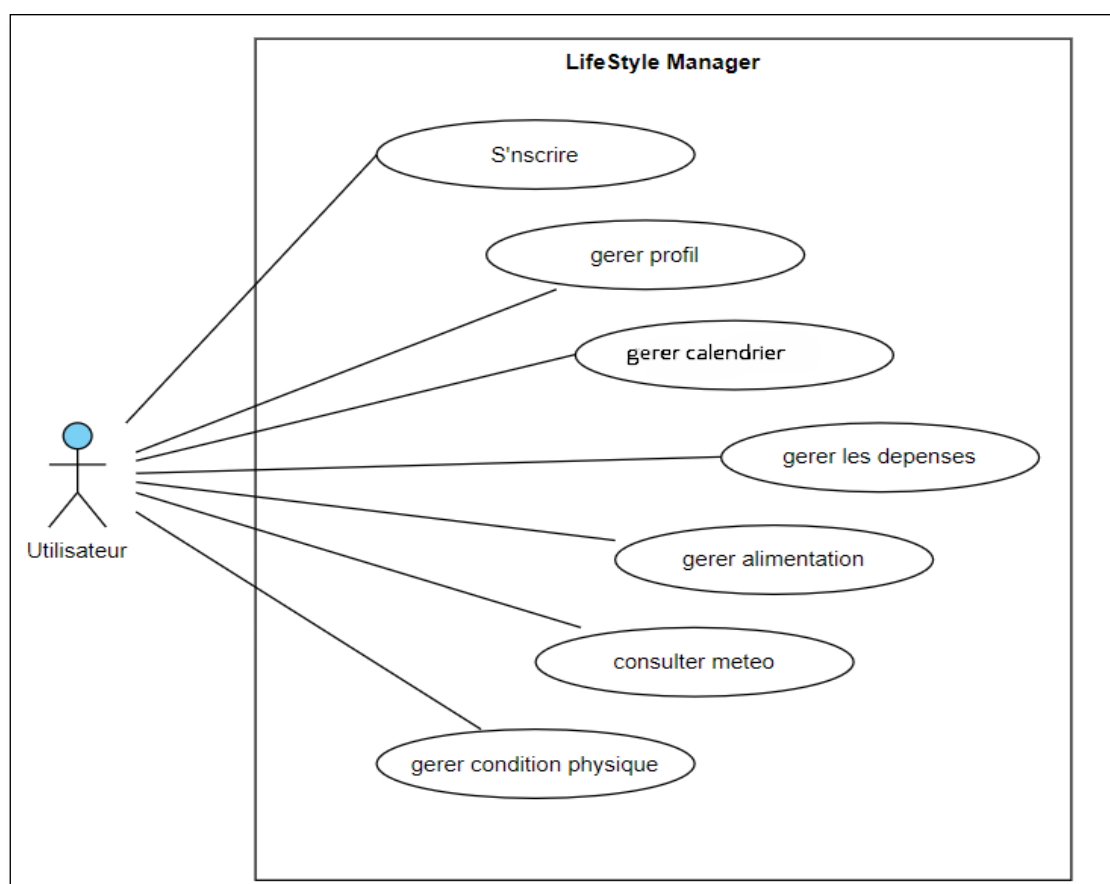


Figure 2 : Diagramme de cas d'utilisation globale.

5.2 Priorités des cas d'utilisation

Ce tableau présente des priorités des cas d'utilisation

Tableau 4 : tableau des priorités

Diagramme de cas d'utilisation	Description	Priorités
Gérer profil	L'utilisateur peut consulter, modifier et supprimer son profil	1
Gérer calendrier	L'utilisateur peut ajouter, modifier, supprimer, visualiser ses tâches	1
Gérer condition physique	L'utilisateur peut ajouter, supprimer, visualiser ses condition physique	2
Gérer alimentation	L'utilisateur peut ajouter ou supprimer des repas	2
Gérer dépenses	L'utilisateur peut ajouter salaire et visualiser ses dépenses	3
Consulter météo	L'utilisateur peut vérifier les prévisions météorologiques	3

Conclusion

Après avoir décrit les besoins fonctionnels et non fonctionnels attendus de notre application et présenté le fonctionnement général en s'appuyant sur le diagramme de cas d'utilisation globale, le chapitre suivant sera consacré à la spécification et la conception des sprints.

Chapitre III : Conception

Introduction

Ce chapitre est consacré à la phase de notre application. À cet effet, nous présenterons les cas d'utilisation à travers des descriptions textuelles et des diagrammes de séquence.

I. Release1

1. Sprint1 : Authentification, gérer profil et gérer calendrier

Dans cette partie, nous allons nous intéresser au premier Sprint.

Tableau 5 description textuelle du cas d'utilisation « s'authentifier »

Cas d'utilisation	S'authentifier
Acteur	Utilisateur
Objectif	Permet à chaque utilisateur d'inscrit sur notre application de s'authentifier avec ses coordonnées.
Pré condition	L'acteur n'est pas connecté.
Post condition	L'Acteur est connecté.
Scenario nominale	<ul style="list-style-type: none"> • L'acteur insère son identifiant et son mot de passe dans les champs prévus. • Le système vérifie la combinaison saisie avec les informations enregistrées. • Si les informations sont correctes l'acteur est dirigé vers sa page d'accueil.
Scenario alternative	<ul style="list-style-type: none"> • Si l'acteur n'a pas de compte il peut choisir de s'inscrire en cliquant sur un lien.

1.1 Diagramme de séquence du cas d'utilisation « s'authentifier »

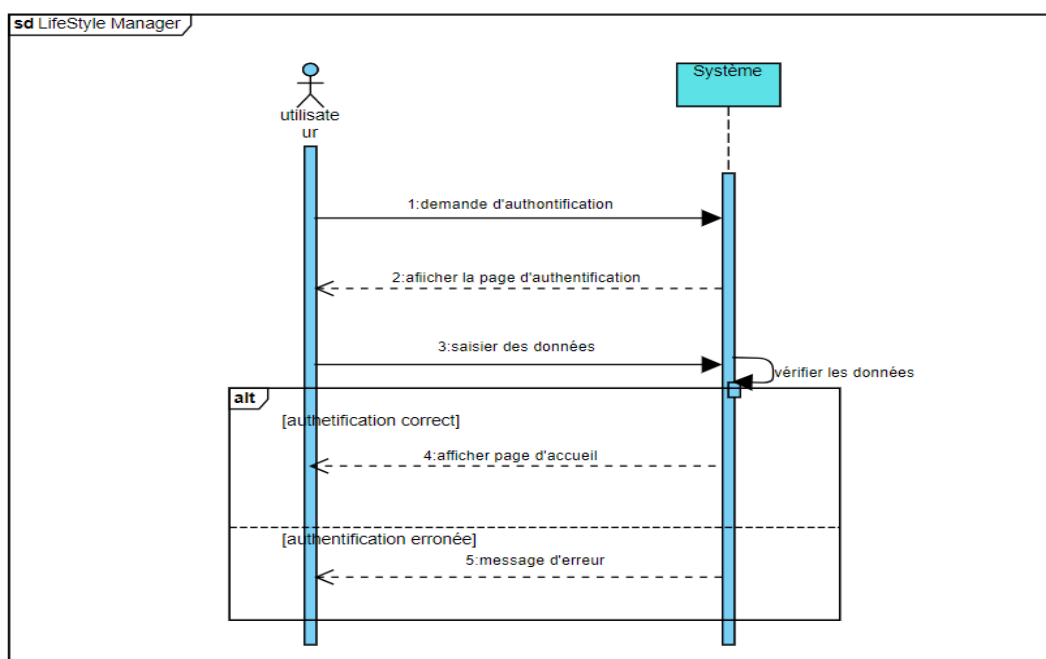


Figure 3 : Diagramme de séquence relatif au cas d'utilisation « s'authentifier ».

1.2 Raffinement du cas d'utilisation « gérer profil »

La figure représente le raffinement de cas d'utilisation « gérer profil ».

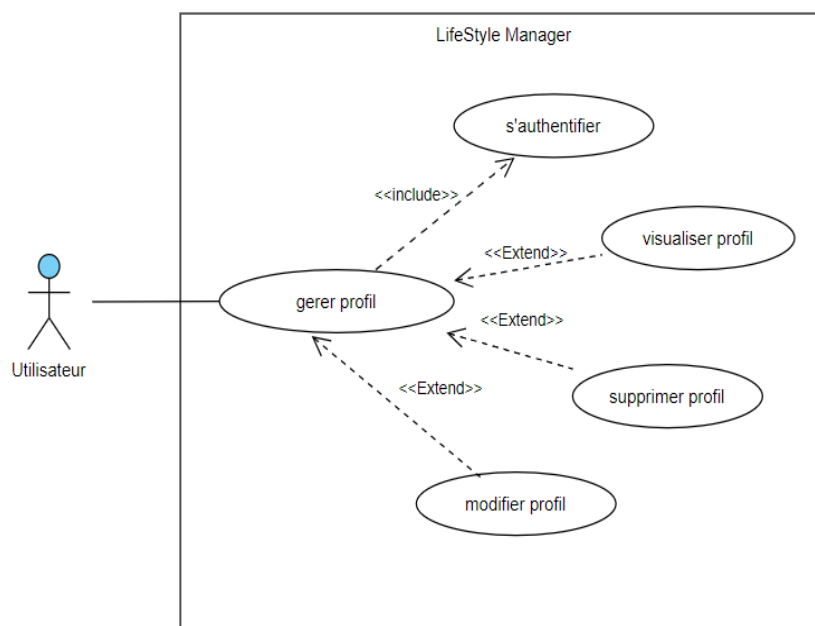


Figure 4 : Raffinement de cas d'utilisation « gérer profil ».

1.3 Description textuelle du cas d'utilisation « modifier profil »

Ce tableau présente la description textuelle du cas d'utilisation « modifier profil »

Tableau 6 : description textuelle du cas d'utilisation « modifier profil »

Cas d'utilisation	Modifier profil
Acteur	Utilisateur
Objectif	Permet à chaque utilisateur de modifier ou visualiser son profil
Pré condition	Profil non modifier
Post condition	Le profil doit être affiché avec la modification
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur sélectionne « modifier profil » • Le système affiche le formulaire pré rempli • L'utilisateur modifie les informations • L'utilisateur enregistre les modifications
Scénario alternative	<ul style="list-style-type: none"> • Le système enregistre les données si valides, sinon affiche un message d'erreur

1.4 Diagramme de séquence du cas d'utilisation « modifier profil »

Le diagramme de séquence du cas d'utilisation « modifier profil » est illustré dans la figure :

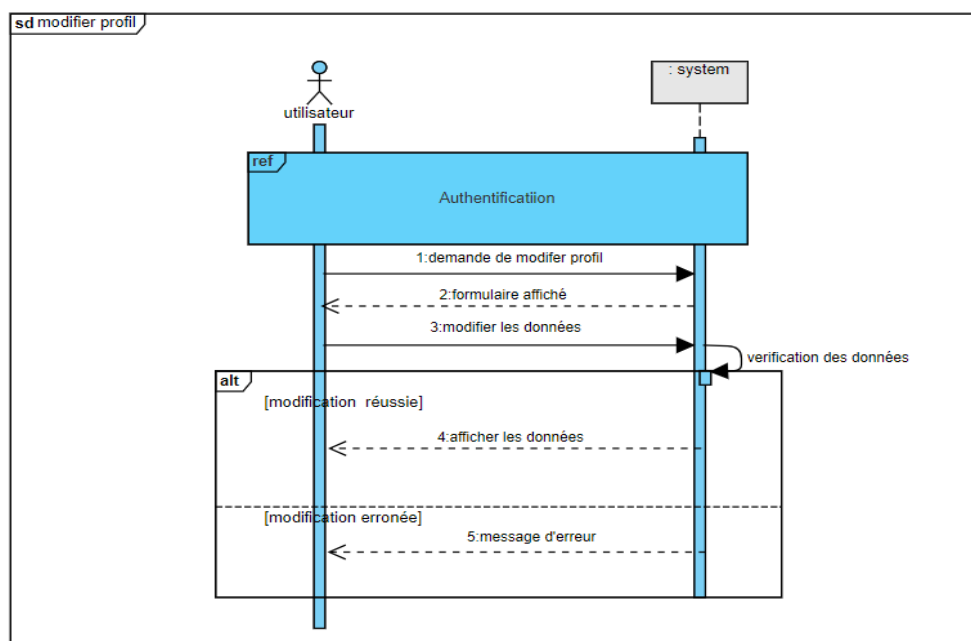


Figure 5 : Diagramme de séquence du cas d'utilisation « modifier profil »

1.5 Raffinement du cas d'utilisation « gérer calendrier »

La figure représente le raffinement de cas d'utilisation « gérer calendrier »

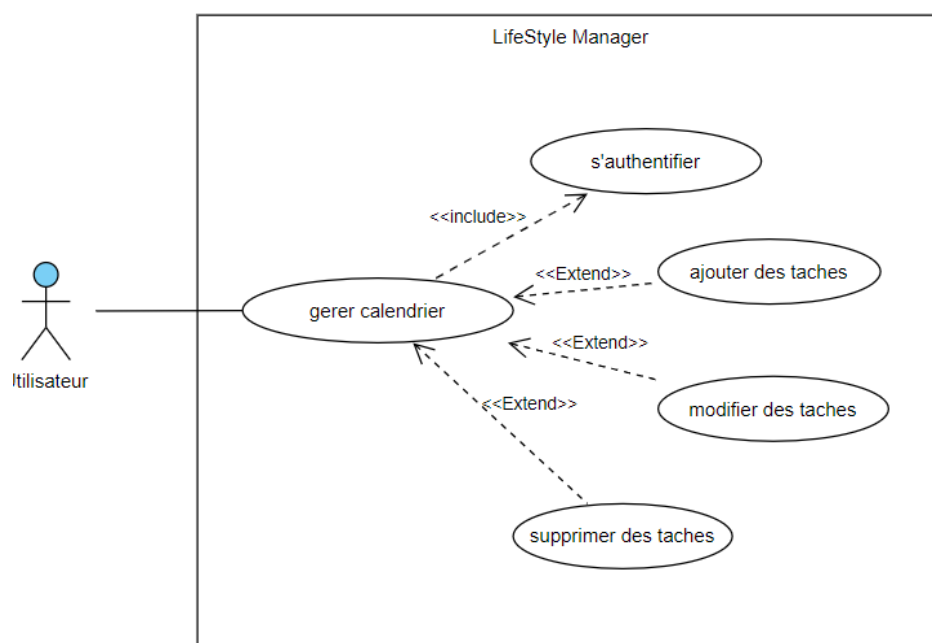


Figure 6 : Raffinement du cas d'utilisation « gérer calendrier ».

1.6 Description textuelle du cas d'utilisation « modifier profil »

Ce tableau présente la description textuelle du cas d'utilisation « modifier profil »

Tableau 7 : description textuelle du cas d'utilisation « modifier profil »

Cas d'utilisation	Modifier profil
Acteur	Utilisateur
Objectif	Permet à chaque utilisateur de modifier ou visualiser son profil
Pré condition	Profil non modifier
Post condition	Le profil doit être affiché avec la modification
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur sélectionne « modifier profil » • Le système affiche le formulaire pré rempli • L'utilisateur modifie les informations • L'utilisateur enregistre les modifications
Scénario alternative	<ul style="list-style-type: none"> • Le système enregistre les données si valides, sinon affiche un message d'erreur

1.7 Diagramme de séquence du cas d'utilisation « modifier profil »

Le diagramme de séquence du cas d'utilisation « modifier profil » est illustré dans la figure :

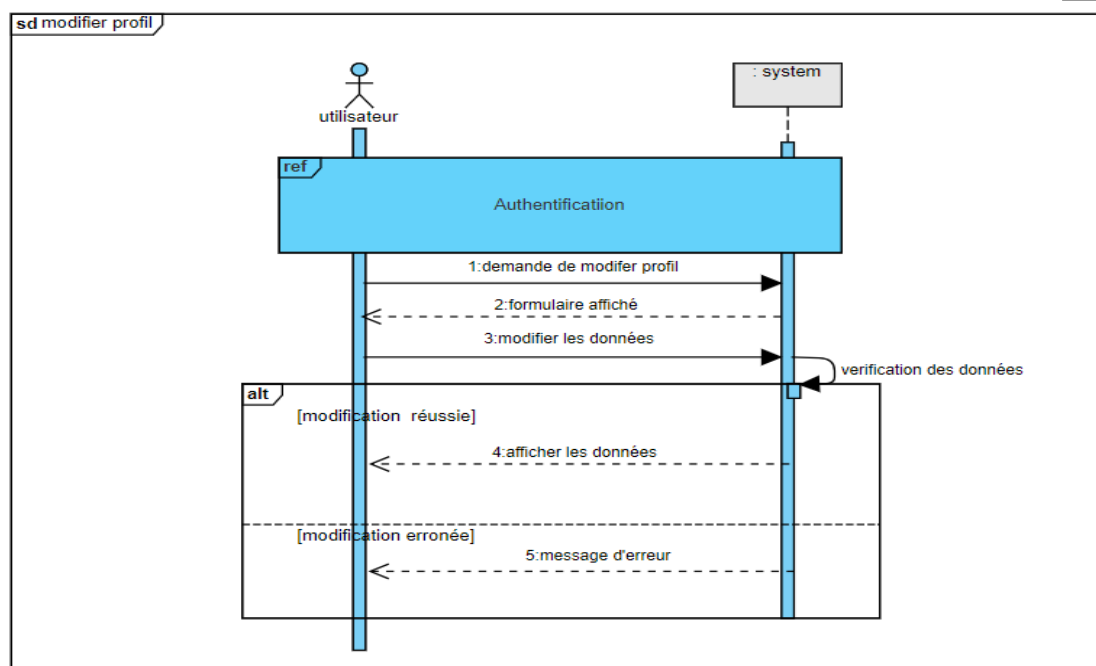


Figure 7 : Diagramme de séquence du cas d'utilisation « modifier profil »

1.8 Raffinement du cas d'utilisation « gérer calendrier »

La figure représente le raffinement de cas d'utilisation « gérer calendrier »

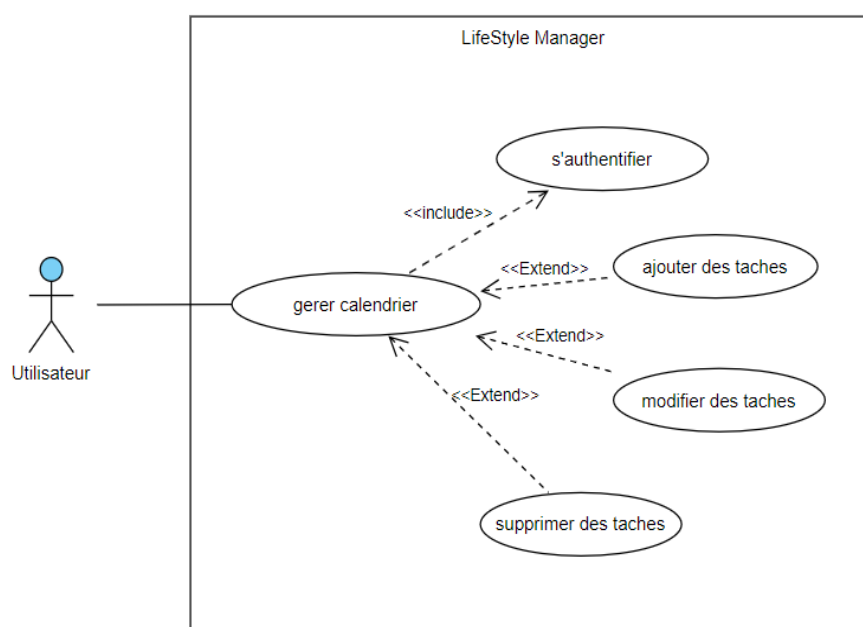


Figure 8 : Raffinement du cas d'utilisation « gérer calendrier ».

1.9 Description textuelle du cas d'utilisation « Ajouter des tâches »

Ce tableau présente la description textuelle du cas d'utilisation « Ajouter des tâches »

Tableau 8 : Description textuelle du cas d'utilisation « Ajouter des taches »

Cas d'utilisation	Ajouter des taches
Acteur	Utilisateur
Objectif	Permet de chaque utilisateur d'ajouter des taches
Pré condition	Tâche ne pas existe
Post condition	Liste des taches affichée avec tache ajouter
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur sélectionne « ajouter une tache » • Le système affiche le formulaire pour une tache • L'utilisateur saisit les détails de la tache • L'utilisateur confirme l'ajout de la tache
Scénario alternative	<ul style="list-style-type: none"> • Le système enregistre les données si valides, si-non affiche un message d'erreur

1.10 Diagramme de séquence du cas d'utilisation « ajouter des taches »

Le diagramme de séquence du cas d'utilisation « ajouter des taches » est illustré dans la figure :

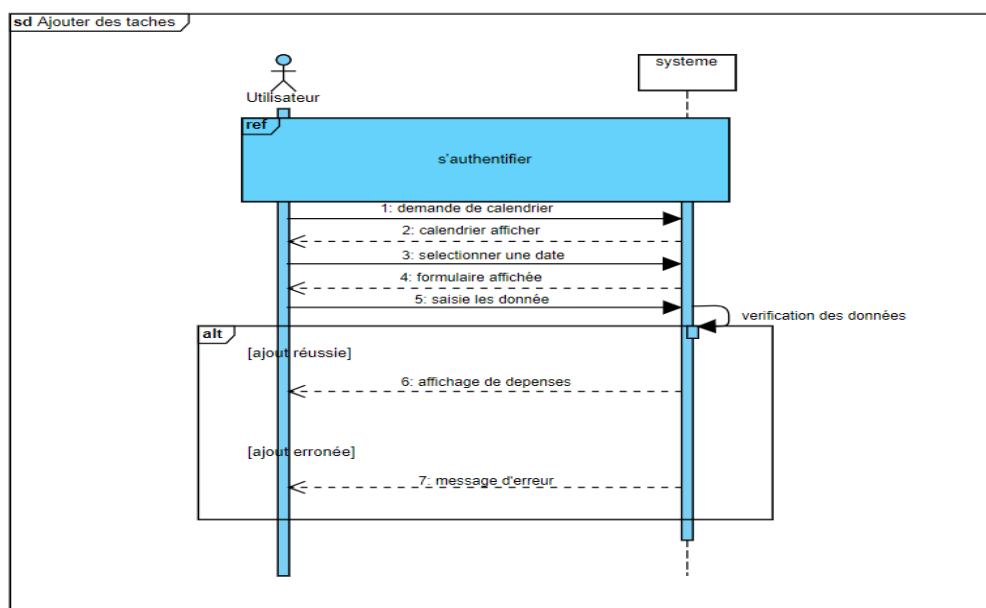


Figure 9 : Diagramme de séquence du cas d'utilisation « ajouter des taches »

1.11 Description textuelle du cas d'utilisation « Modifier des taches »

Ce tableau présente la description textuelle du cas d'utilisation « Modifier des taches ».

Tableau 9 : description textuelle du cas d'utilisation « modifier taches »

Cas d'utilisation	Modifier des taches
Acteur	Utilisateur
Objectif	Permet de chaque utilisateur de modifier des taches
Pré condition	Tache non modifier
Post condition	Liste des taches modifiée vas être affichée
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur sélectionne « modifier » • Le système affiche le formulaire pour une tache • L'utilisateur saisit les détails de la tache • L'utilisateur confirme la modification de la tache • Le système enregistre les données

1.12 Diagramme de séquence du cas d'utilisation « modifier des tâches »

Le diagramme de séquence du cas d'utilisation « modifier des tâches » est illustré dans la figure :

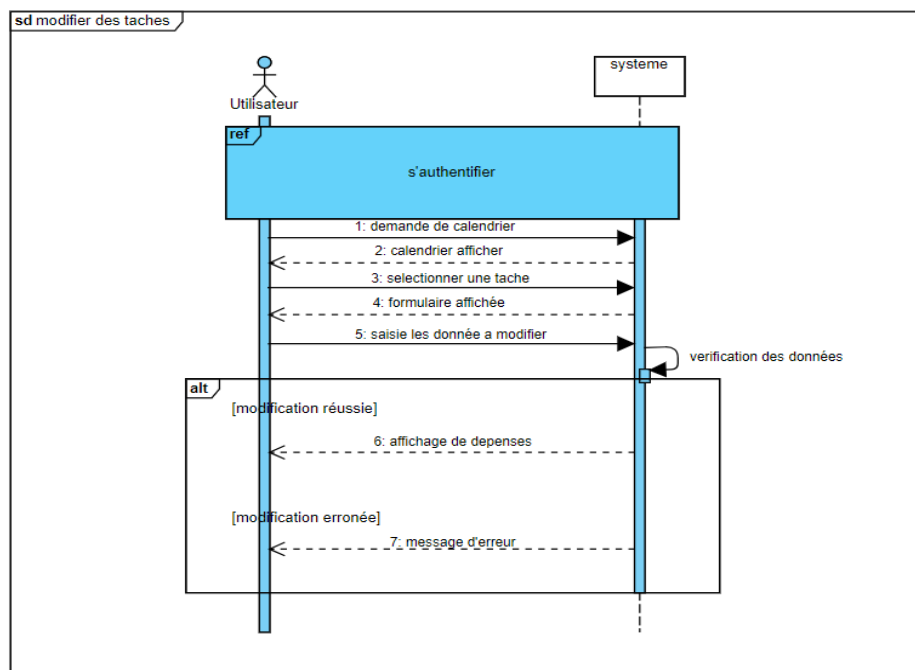


Figure 10 : Diagramme de séquence du cas d'utilisation « modifier des tâches »

II. Release 2

1. Sprint 2 : Gérer condition physique et gérer alimentation

1.1 Raffinement du cas d'utilisation « gérer condition physique »

La figure représente le raffinement de cas d'utilisation « gérer condition physique »

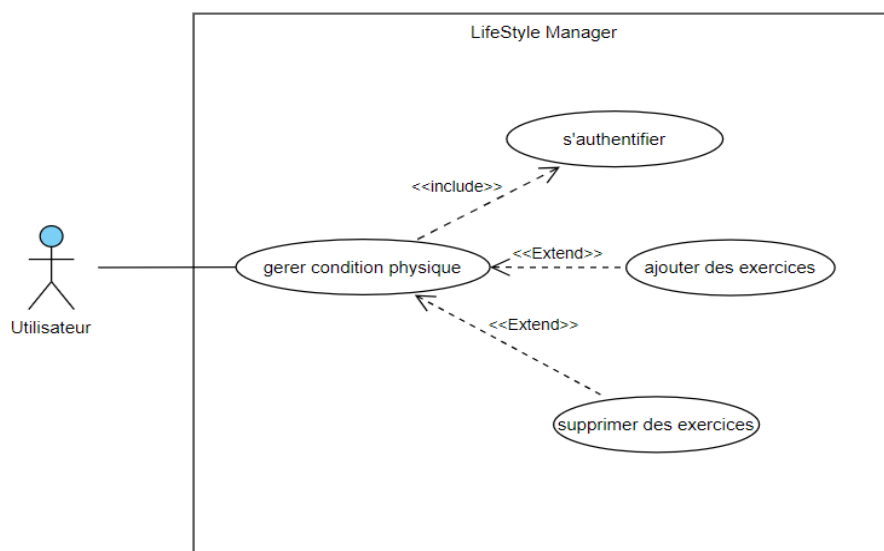


Figure 11 : Raffinement du cas d'utilisation « gérer condition physique ».

1.2 Description textuelle du cas d'utilisation « ajouter des activités physiques »

Ce tableau présente la description textuelle du cas d'utilisation « ajouter des activités physiques »

Tableau 10 : Description textuelle du cas d'utilisation « ajouter des activités physiques »

Cas d'utilisation	Ajouter des activités physiques
Acteur	Utilisateur
Objectif	Permet à l'utilisateur d'ajouter ses activités physiques
Pre condition	Activité physique n'existe pas
Post condition	Activité physique ajoutée
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur sélectionne l'option « ajouter » • Le système affiche le formulaire a rempli • L'utilisateur saisie les détails de l'activité physique • L'utilisateur confirme l'ajout
Scénario alternative	<ul style="list-style-type: none"> • Le système enregistre les données si valides, sinon affiche un message d'erreur

1.3 Diagramme de séquence du cas d'utilisation « ajouter des activités physiques »

Le diagramme de séquence du cas d'utilisation « ajouter des activités physiques » est illustré dans la figure :

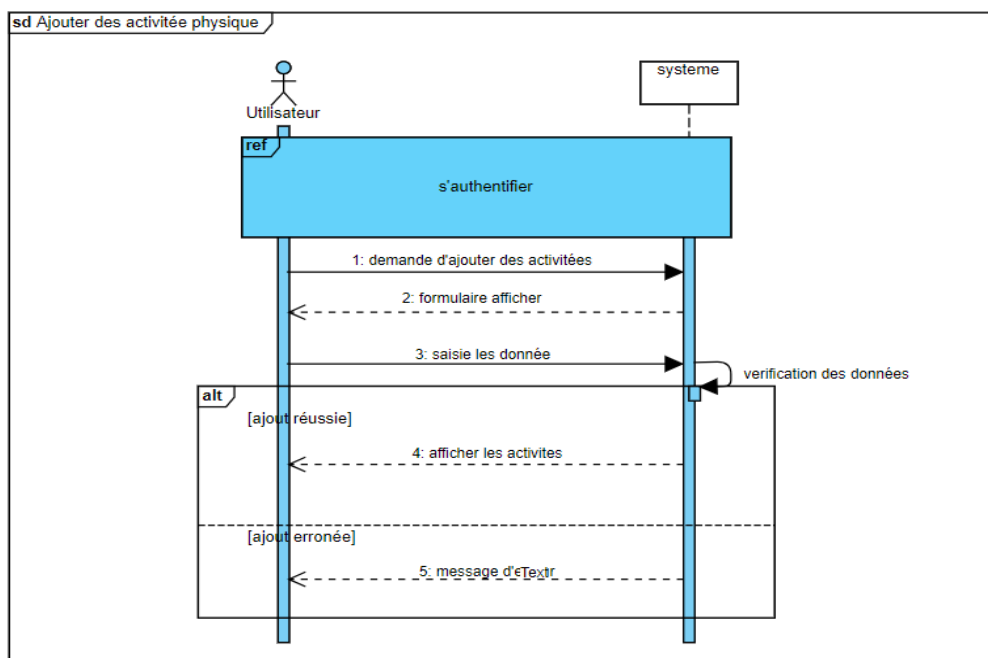


Figure 12 : Diagramme de séquence du cas d'utilisation « ajouter des activités physiques ».

1.4 Description textuelle du cas d'utilisation « supprimer des activités »

Ce tableau présente la description textuelle du cas d'utilisation « supprimer des activités »

Tableau 11 : Description textuelle du cas d'utilisation « supprimer des activités »

Cas d'utilisation	Supprimer des activités
Acteur	Utilisateur
Objectif	Permet à l'utilisateur de supprimer les activités
Pre condition	Activité existe
Post condition	Activité supprimée
Scénario nominale	<ul style="list-style-type: none"> Le système affiche la liste des activités L'utilisateur sélectionne l'activité qu'il souhaite supprimer L'utilisateur confirme la suppression

	<ul style="list-style-type: none"> • Le système supprime l'activité sélectionnée • Le système enregistre les données
Scénario alternative	<ul style="list-style-type: none"> • Si valides, sinon affiche un message d'erreur

1.5 Diagramme de séquence du cas d'utilisation « supprimer des activités »

Le diagramme de séquence du cas d'utilisation « supprimer des activités » est illustré dans la figure :

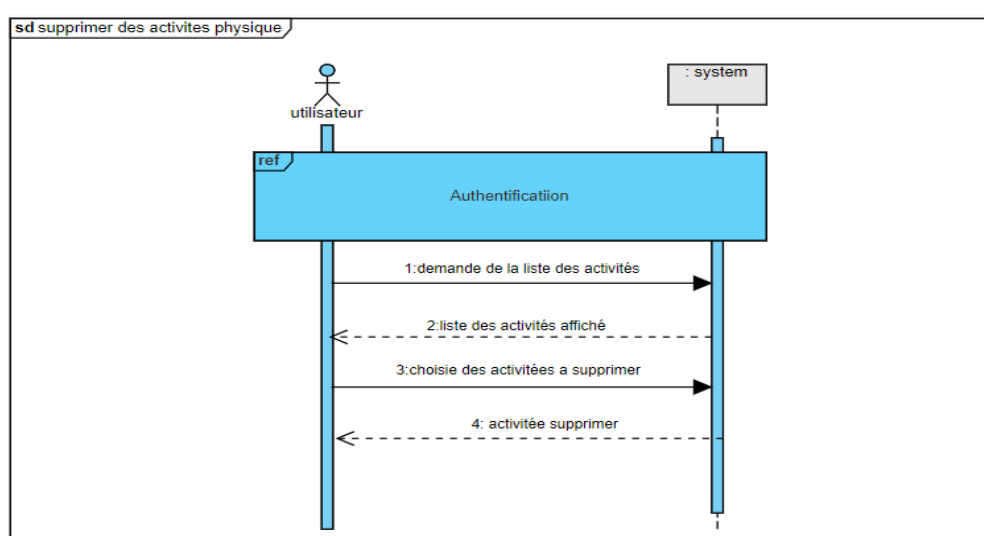


Figure 13 : Diagramme de séquence du cas d'utilisation « supprimer des activités ».

1.6 Raffinement du cas d'utilisation « gérer alimentation »

La figure représente le raffinement de cas d'utilisation « gérer alimentation »

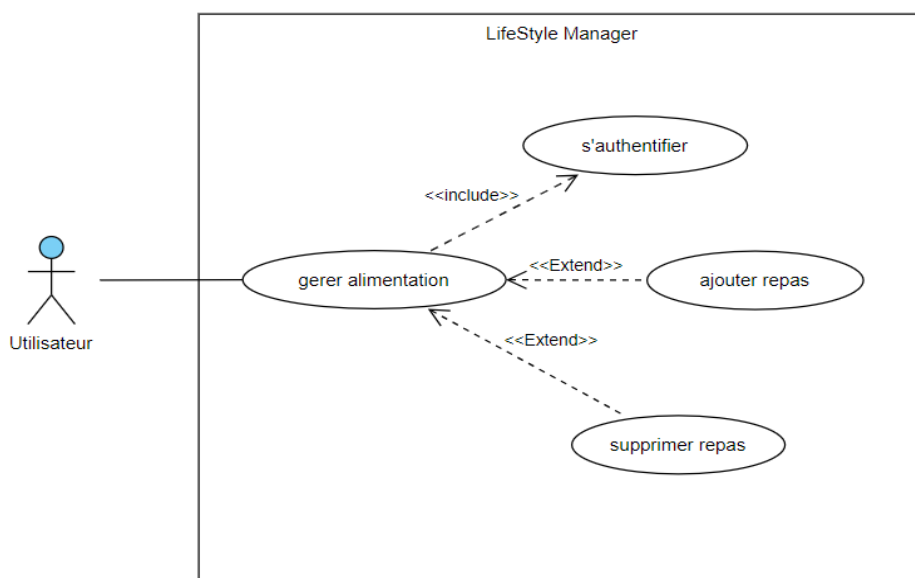


Figure 14 : Raffinement du cas d'utilisation « gérer alimentation ».

1.7 Description textuelle du cas d'utilisation « ajouter repas »

Ce tableau présente la description textuelle du cas d'utilisation « ajouter repas »

Tableau 12 : Description textuelle du cas d'utilisation « ajouter repas ».

Cas d'utilisation	Ajouter repas
Acteur	Utilisateur
Objectif	Ajouter des repas
Pre condition	Liste repas vide
Post condition	Liste des repas ajoutée sera affichée
Scénario nominale	L'utilisateur remplir le champ par le nom du repas L'utilisateur confirme l'ajout avec cliquer sur « ajouter »
Scénario alternative	Le système faire la vérification des donner : Si le repas existe une liste vas être affichée avec les données

1.8 Diagramme de séquence du cas d'utilisation « ajouter repas »

Le diagramme de séquence du cas d'utilisation « ajouter repas » est illustré dans la figure :

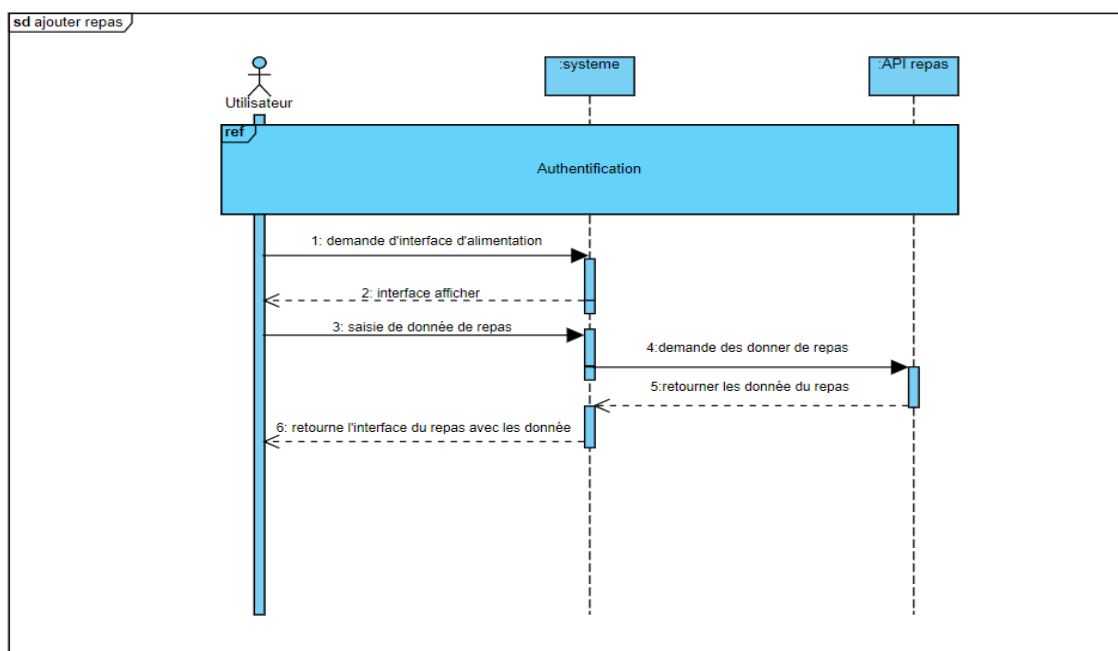


Figure 15 : Diagramme de séquence du cas d'utilisation « ajouter repas ».

III. Release 3

1. Sprint 3 : Gérer dépenses et consulter météo

1.1 Raffinement du cas d'utilisation « gestion de dépenses »

La figure représente le raffinement de cas d'utilisation « gestion de dépenses »

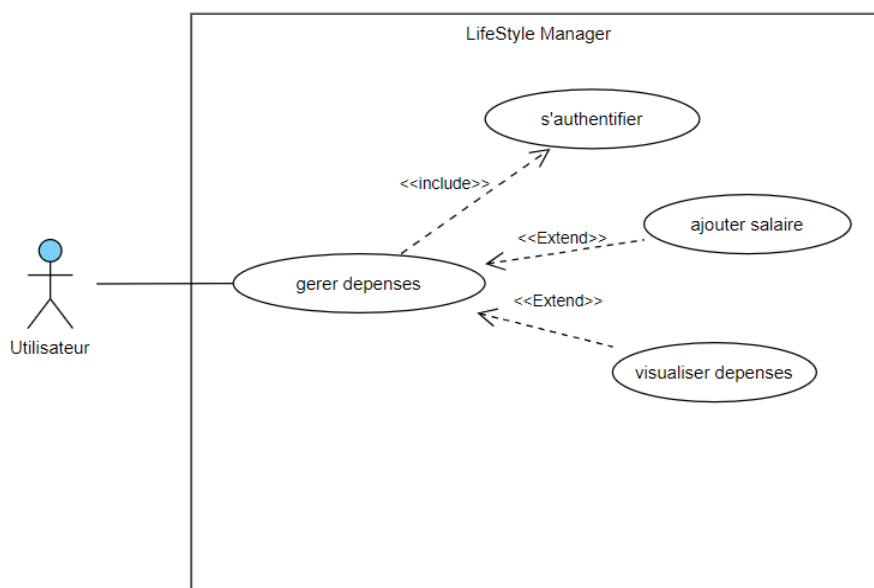


Figure 16 : Raffinement du cas d'utilisation « gestion de dépenses »

1.2 Description textuelle du cas d'utilisation « ajouter salaire »

Ce tableau présente la description textuelle du cas d'utilisation « ajouter salaire »

Tableau 13 : Description textuelle du cas d'utilisation « ajouter salaire »

Cas d'utilisation	Visualiser dépenses
Acteur	Utilisateur
Objectif	Visualiser les dépenses
Pre condition	Salaire inconnu
Post condition	Salaire ménager
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur remplir le champ par leur salaire • Le système vérifier les données et affiche le ménage-ment des dépenses

1.3 Diagramme de séquence du cas d'utilisation « ajouter salaire »

Le diagramme de séquence du cas d'utilisation « ajouter salaire » est illustré dans la figure :

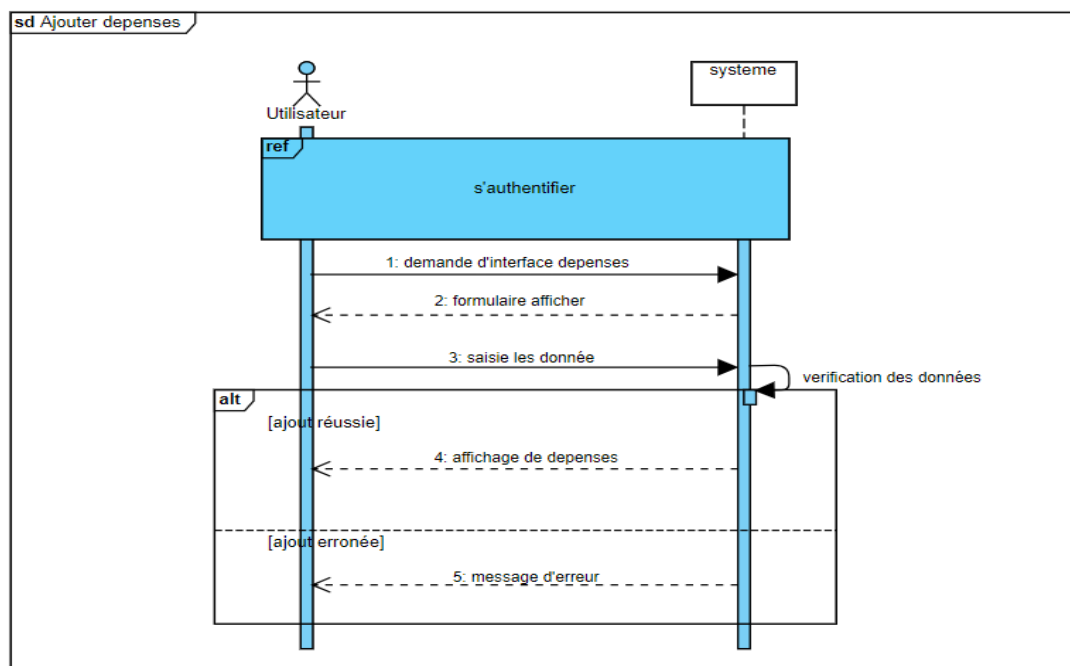


Figure 17 : Diagramme de séquence du cas d'utilisation « ajouter salaire ».

1.4 Raffinement du cas d'utilisation « consulter météo »

La figure représente le raffinement de cas d'utilisation « consulter météo ».

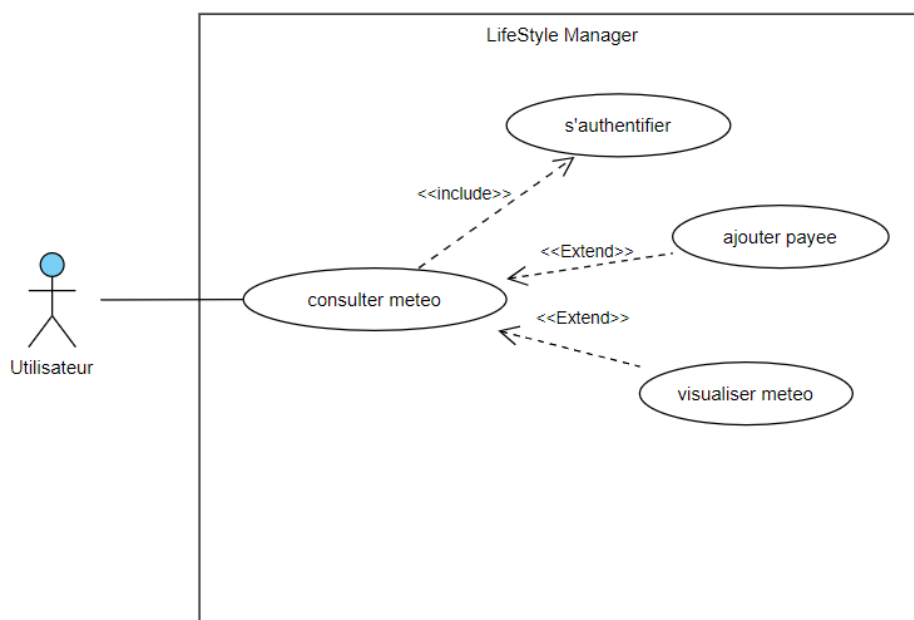


Figure 18 : Raffinement du cas d'utilisation « consulter météo ».

1.5 Description textuelle du cas d'utilisation « ajouter payée »

Ce tableau présente la description textuelle du cas d'utilisation « ajouter payée » :

Tableau 14 : Description textuelle du cas d'utilisation « ajouter payée ».

Cas d'utilisation	Ajouter payée
Acteur	Utilisateur.
Objectif	L'utilisateur peut consulter la météo.
Pré condition	S'authentifier.
Post condition	La page météo doit être affichée.
Scénario nominale	<ul style="list-style-type: none"> • L'utilisateur remplir le champ par le nom de payée. • Le système affiche les données de la météo.

1.6 Diagramme de séquence du cas d'utilisation « ajouter payée »

Le diagramme de séquence du cas d'utilisation « ajouter payée » est illustré dans la figure :

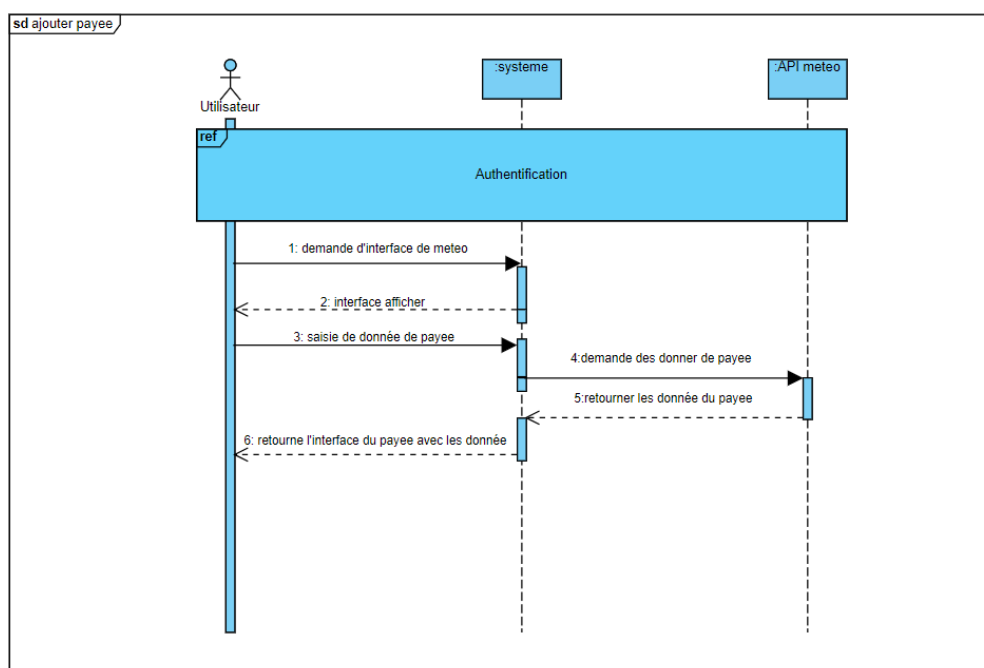


Figure 19 : Diagramme de séquence du cas d'utilisation « ajouter payée »

IV. Diagramme de classes :

Dans cette section, nous allons élaborer le diagramme de classes de notre application, qui représente la structure statique du système en utilisant le langage de modélisation unifié

(UML). Ce type de diagramme de structure décrit les classes, leurs attributs, leurs opérations ou méthodes, ainsi que les relations qui existent entre elles.

La figure représente le diagramme de classe :

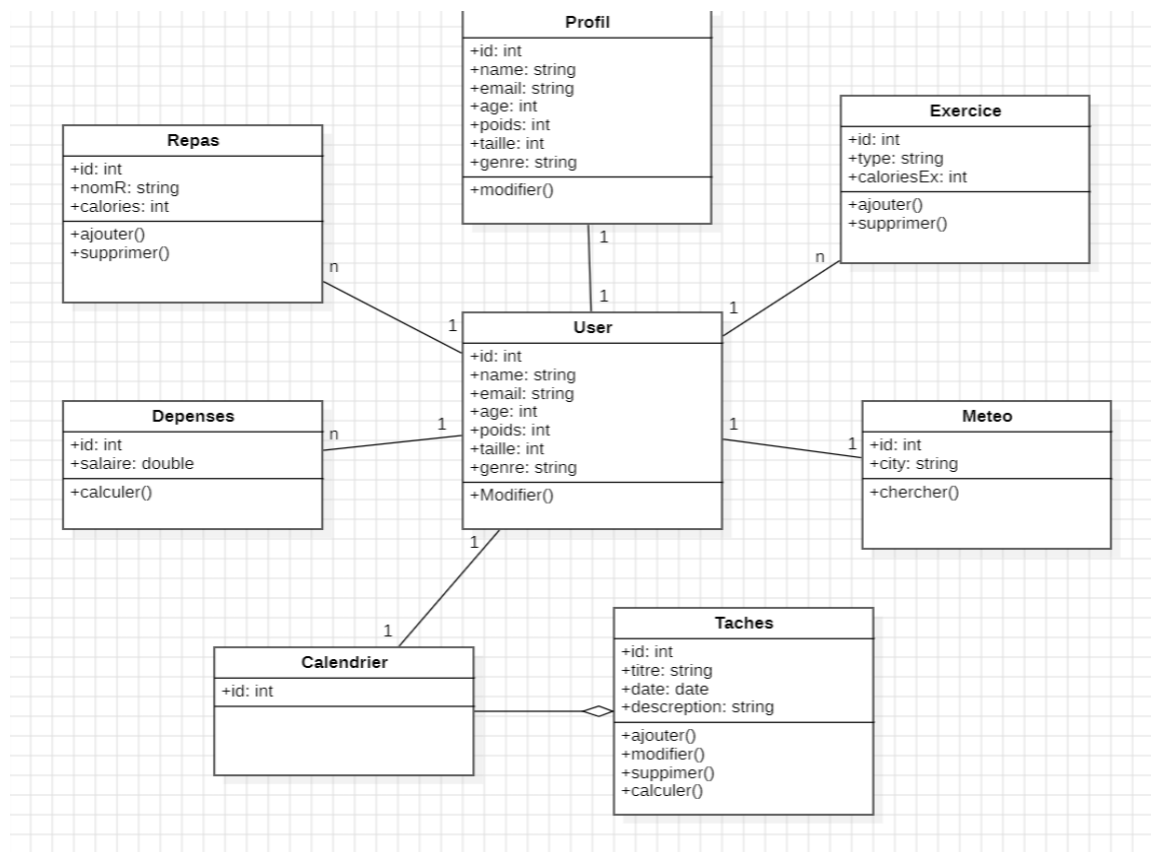


Figure 20 : Diagramme de classe.

Description des classes :

Ce tableau présente les descriptions des classes :

Tableau 15 : descriptions des classes

Nom de la classe	Noms des attributs	Description
User	Id	Clé primaire de la table qui represente l'identification de chaque utilisateur
	Name	Indique le nom de l'utilisateur
	Email	Indique l'email de l'utilisateur
	Age	Indique l'age de l'utilisateur
	Poids	Indique le poids de l'utilisateur
	Taille	Indique la taille de l'utilisateur
	Genre	Indique le genre de l'utilisateur
Taches	IdTache	Cle primaire de la table qui represente l'identification de chaque tache
	Titre	Indique le titre de la tache
	Date	Indique la date du la tache
	Descreption	Indique la descreption du la tache
Exercices	IdExercice	Cle primaire de la table qui represente l'identification de chaque exercice
	Type	Indique le type de l'exercice
	Durée	Indique la durée de l'exercice
	CaloriesEx	Indique la calorie du l'exercice
	IdRepas	Cle primaire de la table qui represente l'identification de chaque repas

Repas	NomR	Indique le nom de chaque repas
	Calories	Indique la calorie du chaque repas
Météo	IdMeteo	Cle primaire de la table qui represente l'identification de le meteo
	City	Indique le nom de chaque city

Conclusion

Dans ce chapitre, nous avons étudié les divers besoins et présenté les différents cas d'utilisation, accompagnés de leur description détaillée, leur diagramme de séquence et leur diagramme de classe. Cette analyse nous a permis d'avoir une vue d'ensemble des fonctionnalités nécessaires au système. Le prochain chapitre portera sur la réalisation de l'application.

Chapitre IV : Réalisation

Introduction

Ce dernier chapitre du rapport présente le travail accompli. Tout d'abord, nous détaillerons l'environnement de travail, qui se compose de deux parties : le matériel et le logiciel qui soutiennent notre application. Ensuite, nous exposerons les choix technologiques que nous effectués. Nous passerons ensuite en revue les différentes tâches que nous avons réalisées en présentant quelques interfaces, et nous fournissons un chronogramme récapitulatif décrivant tous les étapes de mise en œuvre de notre système.

1. Environnement de travail

Tout au long de réalisation de notre application, nous avons utilisé des matériels et des logiciels bien particuliers.

1.1 Environnement matériel

Nous avons utilisé deux postes de travail comme environnement matériel pour assurer la réussite de la réalisation, ces derniers présentent les caractéristiques suivantes :

Tableau 16 : Les caractéristiques de poste de travail.

Poste de travail		
Pc	Lenovo	HP
Système d'exploitation	Windows 11 Famille Unilingue	Windows 11 Famille
RAM	8,00 Go	8,00 Go
Type de système	Système d'exploitation 64 bits	Système d'exploitation 64 bits
Processeur	11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz 3.19 GHz	I5 11 Gen mx350/intel irise

1.2 Environment logical

- **Outlie de conception**

Visual Paradigm Online C'est un outil gratuit de modélisation et de conception de Logiciels, qui permet aux utilisateurs de créer des diagrammes UML, des cartes conceptuelles, des organigrammes et d'autres types de diagrammes.



Figure 21 : Visual Paradigm.

- **Outil de développement**

IntelliJ IDEA est un environnement de développement intégré (IDE) extrêmement puissant et largement adopté pour la programmation en java. Concu et maintenu par JetBrains, il se décline en deux éditions : Community et Ultimate. Cet IDE offre une multitude de fonctionnalités qui accélèrent le développement, tout en favorisant la qualité du code produit.

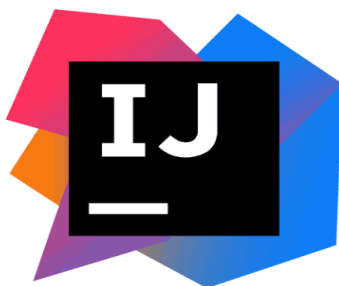


Figure 22 : IntelliJ IDEA.

MySQL est un système de gestion de base de données relationnelle open-source qui stocke des données dans un format structuré à l'aide de lignes et de colonnes C'est un logiciel qui permet aux utilisateurs de créer, gérer et manipuler des bases de données Développé par MySQL AB, qui appartient maintenant à Oracle Corporation, MySQL est réputé pour sa fiabilité, sa scalabilité et sa facilité d'utilisation.



Figure 23 : MySQL

1.3 Technologies utilisées

- **API et bibliothèques utilisées**

L'API OpenWeather est une plateforme élégante et largement reconnue qui fournit toutes les informations météorologiques nécessaires à la prise de décision pour n'importe quelle localité dans le monde. Elle est alimentée par des solutions de machine learning convolutionnelles et capable de fournir des données météorologiques actuelles et une variété de produits de prévisions pour n'importe quel emplacement, à l'échelle mondiale.



Figure 24: API openWeather

L'API Nutritionx Track est conçue pour faciliter au maximum le suivi des calories des consommées lors des repas et des calories dépensées pendant l'exercice. Cette API permet aux utilisateurs de suivre leur apport calorique et leurs activités physiques, offrant ainsi une solution complète pour la gestion de la santé.



Figure 25: API Nutritionx Track

- **Langages de programmation**

Java: C'est un langage de programmation orienté objet de haut niveau, développé par Sun Microsystems et désormais maintenu par Oracle. Il a été conçu pour être portable, ce qui signifie que le code Java peut être exécuté sur n'importe quel système d'exploitation sans nécessiter de modifications supplémentaires. Java est largement utilisée dans le développement d'applications Web, d'applications mobiles, de jeux et autres applications logicielles.



Figure 26: java

Spring Boot est un framework Java qui facilite la création et l'exécution d'applications Java. Il simplifie le processus de configuration et d'installation, permettant aux développeurs de se concentrer davantage sur l'écriture de code pour leurs applications.



Figure 27: Spring Boot

Angular est un framework open-source très utilisé pour le développement d'applications Web, conçu en TypeScript par Google. Les développeurs frontend utilisent des Framework tels qu'Angular ou React pour rendre la présentation et la manipulation des données plus efficaces et dynamiques. Au fil du temps, Angular a bénéficié de mises à jour régulières qui l'ont rendu de plus en plus performant, notamment avec le découpage des fonctionnalités de base en différents modules.



Figure 28: Angular

TypeScript est un sur-ensemble de JavaScript qui ajoute un typage statique optionnel. Développé et maintenu par Microsoft Corporation, TypeScript combine la syntaxe familière de JavaScript avec des fonctionnalités supplémentaires et un système de types robuste. Il améliore la maintenabilité du code en détectant les erreurs tôt pendant le développement, rendant ainsi le code plus facile à comprendre et à modifier.



Figure 29: TypeScript

2. Jeu d'essai et scénarios de test

- **Interface de connexion :**

Lorsque l'utilisateur accède à l'application, il est nécessaire de saisir l'e-mail et le mot de passe correctement, puis cliquer sur le bouton "Valider" pour accéder à l'interface suivante.

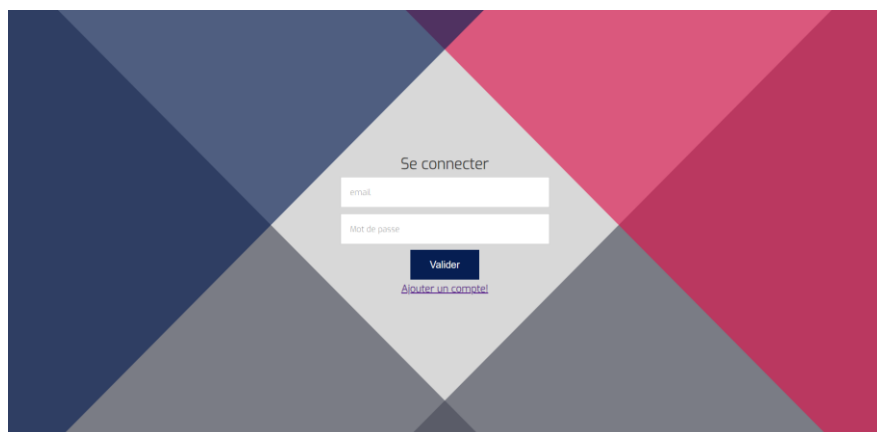


Figure 30: interface de connexion

Lorsqu'un utilisateur accède au site pour la première fois, il doit cliquer sur le lien "Ajouter un compte". Cela l'amènera à l'interface d'inscription.

- **Interface d'inscription**

Dans cette interface, l'utilisateur doit remplir tous les champs avec précision en fournissant les informations requises. Une fois le compte créé avec succès, l'utilisateur peut cliquer sur le bouton "Se connecter" pour accéder à l'interface suivante.

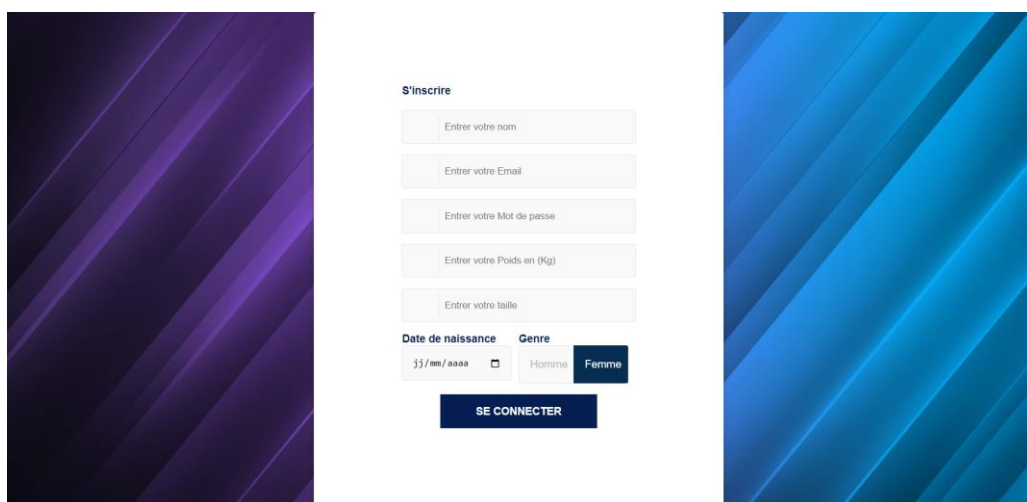


Figure 31: interface d'inscription

- **Interface d'accueil**

Cette interface d'accueil est conçue pour offrir une expérience conviviale et intuitive. Elle présente une série de cartes, chacune représentant une fonctionnalité et un service spécifique. Chaque carte permet à l'utilisateur d'accéder à la section correspondante. L'utilisateur peut simplement cliquer sur la carte qui l'intéresse pour accéder à son interface.

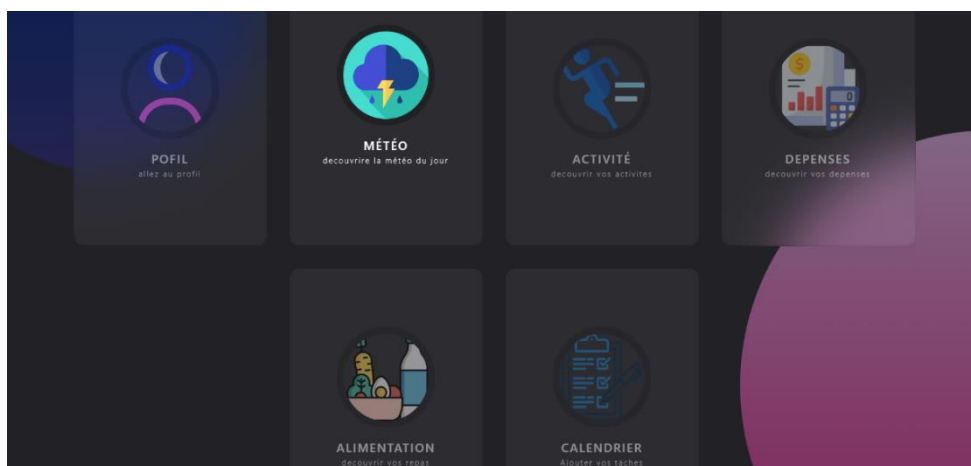


Figure 32: interface d'accueil

- **Interface météo**

Cette interface meteo propose une simple et efficace interface pour connaître les conditions météorologiques actuelles. L'utilisateur doit saisir le nom de sa ville, le clicke sur la bouton « chercher » la page affiche la température, la description , l' humidité , nom de sa ville et la date actuelle.

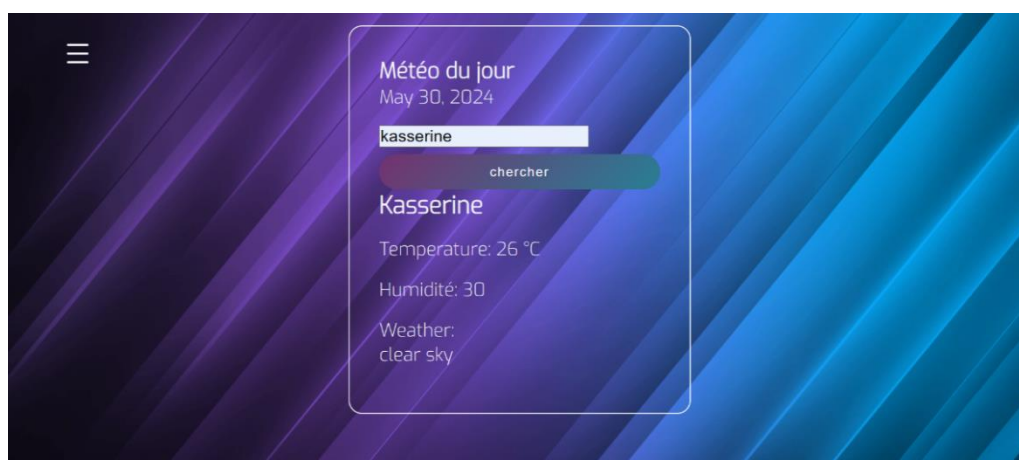


Figure 33: interface météo

- **Dashboard**

A été conçu pour faciliter la navigation entre les différentes interfaces. Il comprend des boutons et des cartes qui permettent à l'utilisateur d'accéder directement à des pages spécifiques en fonction de ses choix.

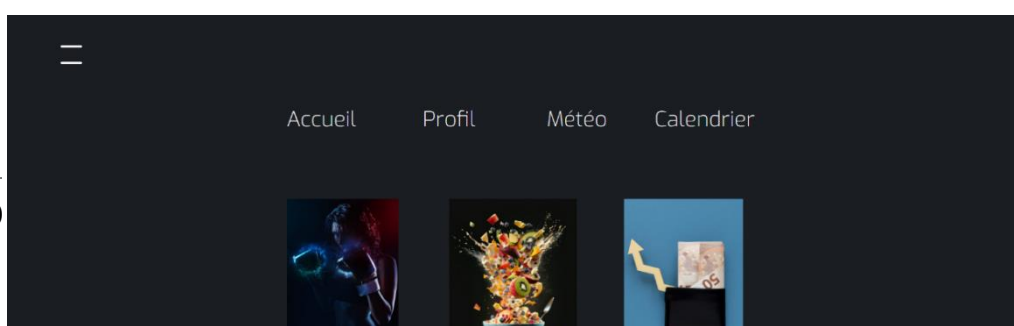


Figure 34: Dashboard

- **Interface exercice**

Dans cette interface, l'utilisateur doit remplir le formulaire, puis cliquer sur le bouton «Ajouter». Ensuite, le système calcule les calories et affiche le résultat.

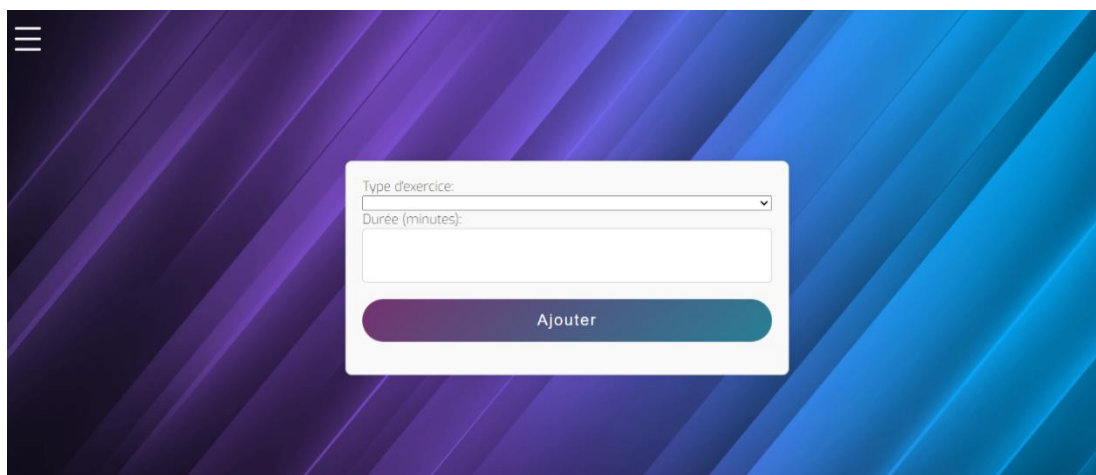
A screenshot of a web application interface for adding an exercise. The background is a dark purple and blue gradient with diagonal lines. In the top left corner, there is a white hamburger menu icon. Centered on the screen is a white rectangular form. Inside the form, there are two input fields: the first is labeled "Type d'exercice:" and has a dropdown arrow; the second is labeled "Durée (minutes):" and is a text input field. Below these fields is a large, rounded rectangular button with a purple-to-blue gradient, labeled "Ajouter" in white text.

Figure 35: interface exercice

- **Interface calendrier**

Pour ajouter une tâche, l'utilisateur peut simplement cliquer sur la date souhaitée dans le calendrier. Une fois la tâche ajoutée, elle apparaîtra sur la date correspondante dans le calendrier. Les utilisateurs peuvent cliquer sur une tâche existante pour voir ses détails ou pour la modifier.

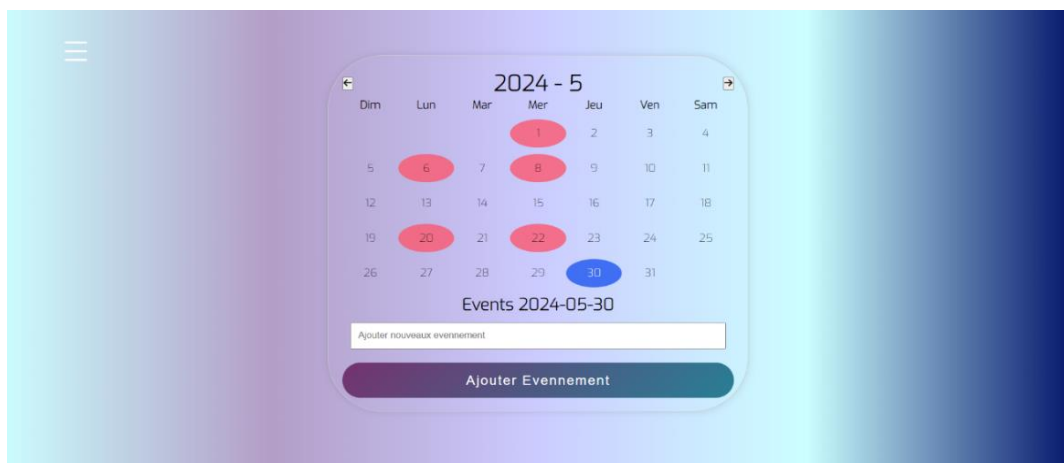


Figure 36: interface calendrier

- **Interface d'alimentation**

Lorsqu'un utilisateur ajoute un repas, il peut entrer les détails de chaque aliment L'application calcule alors les calories pour chaque aliment en se basant sur une base de données d'un API d'aliments et de valeurs caloriques. Le total des calories pour le repas entier est ensuite calculé et affiché.

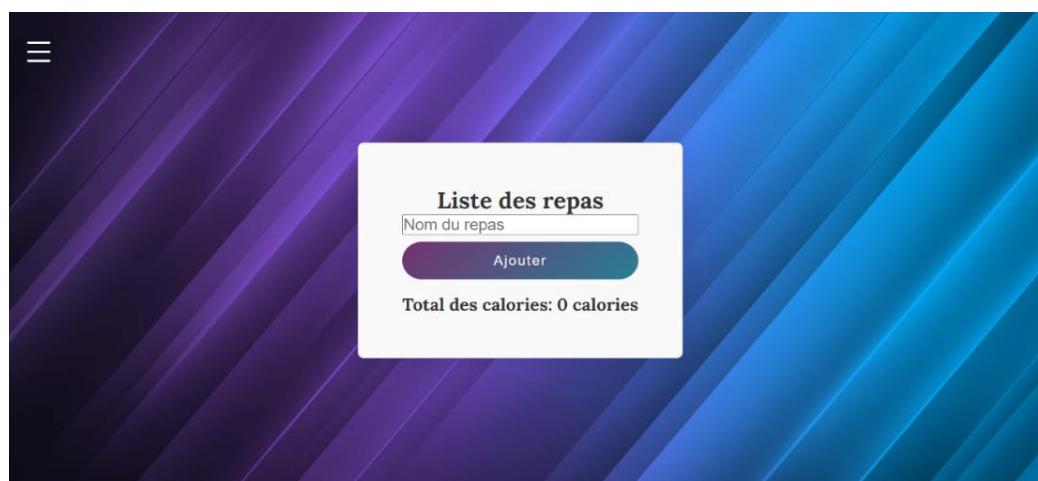


Figure 37: interface d'alimentation

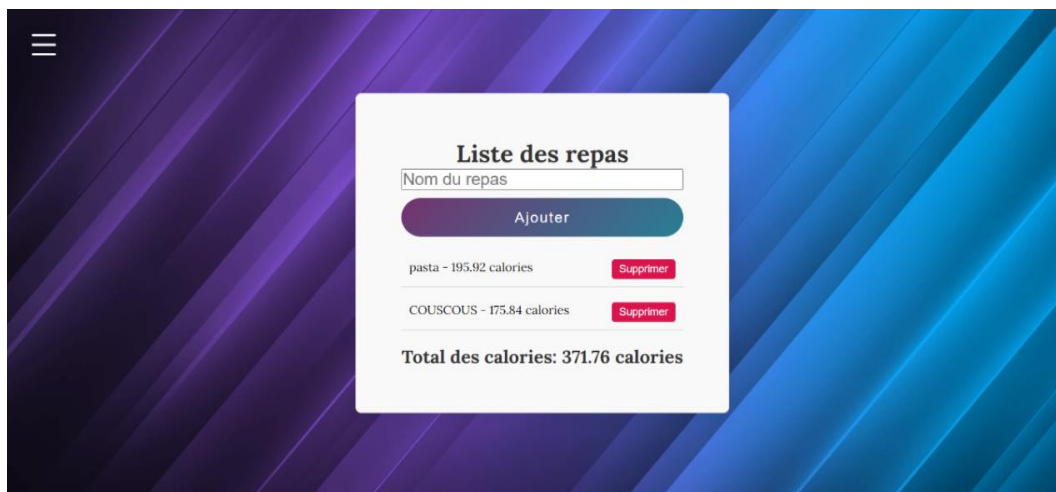


Figure 38: interface d'alimentation

Conclusion

Ce chapitre constitue une synthèse complète du travail réalisé tout au long du projet de conception et de développement, en mettant en lumière les résultats obtenus. Il détaille les environnements matériels et logiciels utilisés pour la construction de l'application. De plus, il présente certaines interfaces clés de l'application. Enfin, le chapitre se clôture avec la présentation du chronogramme des tâches, offrant ainsi une vue d'ensemble du déroulement du projet.

Conclusion générale

Ce projet de fin d'étude, réalisé au sein de L'Institut Supérieure des Études Technologique de Kasserine a pour objectif de faire une étude, une conception, et une Réalisation d'une application de « LifeStyle Manager ». En effet c'est un projet très rentable qui permet de simplifier le style de vie des utilisateurs.

Notre projet "LifeStyle Manager" représente une solution globale et novatrice pour répondre aux défis complexes de gestion du temps et des activités dans le monde contemporain. En combinant diverses fonctionnalités telles que la surveillance des calories, la gestion du calendrier, le contrôle des finances et l'accès aux prévisions météorologiques au sein d'une seule plateforme, notre application vise à simplifier la vie de ses utilisateurs et à améliorer leur qualité de vie.

L'une des principales forces de notre application réside dans sa capacité à intégrer de manière transparente ces différents aspects de la vie quotidienne. Contrairement à de nombreuses applications spécialisées qui fragmentent davantage les tâches, notre application offre une approche holistique qui permet aux utilisateurs de centraliser et d'organiser toutes les facettes de leur vie en un seul endroit.

Enfin, ce projet nous a permis d'appliquer de manière concrète nos connaissances théoriques acquises au cours de notre formation en technologie informatique. La réalisation de cette application nous a également permis de développer des compétences pratiques essentielles pour notre future carrière, notamment en matière de conception logicielle, de développement d'interfaces utilisateur et de gestion de projet.

En somme, notre projet "LifeStyle Manager" représente bien plus qu'une simple application. Il incarne notre vision d'une vie simplifiée et plus efficace, où la technologie peut être utilisée de manière intelligente pour améliorer notre quotidien.

Bibliographie

- [1] : <http://www.isetks.rnu.tn/fr/linstitut/présentation/>
- [2] : <https://ideascale.com/fr/blogues/definition-de-la-methodologie-agile>
- [3] : <https://ideascale.com/fr/blogues/definition-de-la-methodologie-agile/>
- [4] : <https://geeksforgeeks.org/unified-modeling-language-uml-introduction>
- [5] : <https://www.lucidchart.com/pages/fr/diagramme-de-cas-dutilisation-uml>
- [6] : <https://online.visual-paradigm.com/>