

MEMOIRE

Présenté à

**Institut Supérieur d'Informatique
Et de Multimédia de Sfax**

En vue de l'obtention du

DIPLOME D'INGENIEUR

Spécialité : Génie logiciel

Intitulé

**E_Bike : Application web Gestion tracker
Et de Configuration**

Par

Sirine TRABELSI

*Soutenu le 24 Juin 2024, devant le jury
composé de :*

Mme.

Président

Mme.

Membre

Mme. Faiza GHOZZI

Encadrante

M.

Invité Entreprise

Dédicace

À chaque début correspond une fin, et derrière chaque fin se cache un succès. Avec cette conviction en tête, je dédie ce travail :

À mes chers parents Slim et Najla, dont les prières incessantes, le soutien indéfectible et les encouragements constants ont été les piliers de mon parcours vers la réussite.

À ma chère sœur, pour son soutien inestimable tout au long de mes études, qui m'a toujours inspiré et motivé à donner le meilleur de moi-même.

À mes grands-parents, pour leur soutien inconditionnel et leurs encouragements chaleureux qui ont illuminé mon chemin à chaque étape de ma vie.

À mes amis, qui ont été présents à mes côtés, m'ont encouragé et soutenu avec leur amitié sincère tout au long de cette période cruciale.

À tous ceux que j'aime, qui ont contribué, chacun à leur manière, à ma croissance personnelle et à la réalisation de mes aspirations.

Remerciement

En préambule, je souhaite exprimer ma sincère reconnaissance envers Allah pour nous avoir octroyé la force et la détermination nécessaires afin de mener à bien ce projet.

De même, je saisirai cette opportunité pour exprimer ma gratitude envers l’Institut Supérieur d’Informatique et de Multimédia qui nous a accordé cette faveur et nous a ainsi permis d'enrichir nos connaissances dans le domaine de l'informatique.

Je tiens également à exprimer ma profonde gratitude envers **Mohamed Souheib SMIDA**, en qualité de leader technique, à monsieur **Aymen Khelifi** en tant que Product Owner ainsi que Madame **Améni Bellaj**, conceptrice senior en logiciel, et l'ensemble de l'équipe d'Actia Engineering Services pour leur assistance inestimable, leur suivi attentif, leur disponibilité sans faille, leur professionnalisme exemplaire et leurs conseils éclairés.

Je désire également adresser mes plus chaleureux remerciements à Madame **Jedidi Ghazzi Faiza**, notre encadrante dévouée, pour son dynamisme, son soutien constant et ses conseils judicieux. Sa compétence et sa pédagogie ont été d'une valeur inestimable dans la conduite de ce travail.

Pour finir, je souhaite exprimer ma gratitude et ma reconnaissance envers les membres du jury pour avoir pris le temps d'examiner attentivement ce modeste travail, pour leurs précieuses suggestions, leur présence lors de l'évaluation et pour l'honneur qu'ils m'ont fait en évaluant ce travail.

INTRODUCTION GENERALE	1
CHAPITRE 1 : CONTEXTE GENERAL DU PROJET	4
INTRODUCTION	5
1.1 L'ORGANISME D'ACCUEIL	5
1.1.1 Présentation	5
1.1.2 Domaines d'Activité et Savoir-faire	5
1.1.3 Engagement en Qualité et Récompenses	5
1.2 PRESENTATION DU PROJET.....	6
1.2.1 Mission à accomplir.....	6
1.2.2 Présentation de tracker SC350MG	6
1.3 METHODOLOGIE UTILISEE.....	6
1.3.1 Les méthodes agiles	6
1.3.2 La méthodologie Scrum.....	7
1.4 APPLICATION SIMILAIRE	8
1.4.1 VELCO FLEET MANAGEMENT :	8
1.4.2 GPSLive de Rewire Security :.....	8
1.4.3 Tableau comparative :.....	9
1.5 ETUDE DE L'EXISTANT	10
1.5.1 Présentation de E_Bike ACTIA	10
1.5.2 Critique de l'existant.....	12
1.5.3 Solutions proposées.....	13
1.6 CONCLUSION	13
CHAPITRE 2 : MISE EN PLACE DE L'ENVIRONNEMENT DE TRAVAIL	14
INTRODUCTION	15
2.1 SPECIFICATION ARCHITECTURALE.....	15
2.1.1 Architecture logicielle frontend	15
2.1.2 Architecture logicielle du backend.....	16
2.1.3 Architecture globale	17
2.2 SPECIFICATION TECHNOLOGIQUE.....	19
2.2.1 Environnement matériel	19
2.2.2 Environnements logiciels.....	20
2.3 PLANIFICATION ITERATIVE.....	25
2.3.1 Backlog de produit	25
2.3.2 Planification des Sprints	27
2.4 CONCLUSION	28
CHAPITRE 3 : SPRINT 1 : GESTION DES MAGASINS ET DES MODELES DE VELO.....	29
INTRODUCTION	30
3.1 BACKLOG DU SPRINT 1.....	30
3.2 ANALYSE DES BESOINS	31
3.2.1 Identification des acteurs	31
3.2.2 Diagramme de cas d'utilisation de sprint 1 :	32
3.2.3 Description textuelle des cas d'utilisation	32
3.3 MODELISATION CONCEPTUELLE	34
3.3.1 Diagramme de séquence.....	34
3.3.2 Diagramme de classe :	36
3.4 REALISATIONS :	37
3.5 TEST	39
3.6 REVUE DU SPRINT 1	41
3.7 CONCLUSION	41
CHAPITRE 4 : SPRINT 2 : COMMUNICATION AVEC LE TRACKER SC350MG	42
INTRODUCTION :	43
4.1 BACKLOG DU SPRINT 2 :	43
4.2 RECHERCHE SUR TRACKER SC350MG	44

4.2.1	<i>Architecture du Tracker</i> :	44
4.2.2	<i>Les types et format de messages</i> :	45
4.2.3	<i>Communication entre le serveur Tracker et le client web</i> :	46
4.2.4	<i>Etude comparatif</i>	47
4.2.5	<i>Simulation de serveur TCP</i> :	48
4.3	ANALYSE DES BESOINS	49
4.3.1	<i>Identification des acteurs</i>	49
4.3.2	<i>Diagramme de cas d'utilisation de sprint 2</i>	49
4.3.3	<i>Description textuelle des cas d'utilisation</i>	50
4.4	MODELISATION CONCEPTUELLE	51
4.4.1	<i>Diagramme de séquence</i>	51
4.4.2	<i>Diagramme de classe</i>	52
4.5	REALISATION	53
4.6	TEST	54
4.7	REVUE DU SPRINT 2	57
4.8	CONCLUSION	57
CHAPITRE 5 : SPRINT 3 : INTEGRATION DES STATISTIQUES		58
INTRODUCTION		59
5.1	BACKLOG DU SPRINT 3.....	59
5.2	ANALYSE DES BESOINS	61
5.2.1	<i>Identification des acteurs</i>	61
5.2.2	<i>Diagramme de cas d'utilisation</i>	61
5.2.3	<i>Description textuelle des cas d'utilisation</i>	62
5.3	MODELISATION CONCEPTUELLE	63
5.3.1	<i>Diagramme de séquence</i>	63
5.3.2	<i>Diagramme de classe</i>	65
5.4	REALISATION	65
5.5	TEST	70
5.6	REVUE DU SPRINT 3	73
5.7	CONCLUSION	73

Figure 1: Logo Actia ENGINEERING SERVICES	5
Figure 2: Cycle de vie de la méthodologie Scrum	8
Figure 3: Interface d'application web et mobile GPSLive	9
Figure 4: Etude de l'existant	11
Figure 5: Architecture Angular	15
Figure 6: Architecture microservice backend de E_Bike ACTIA	16
Figure 7: Architecture Spring Boot.....	17
Figure 8: Architecture globale.....	19
Figure 9: Logo draw.io.....	21
Figure 10: Logo Spring Boot	21
Figure 11: Logo Angular CLI	22
Figure 12: Logo JAVA.....	22
Figure 13: Logo TypeScript	22
Figure 14: Logo PostgreSQL	23
Figure 15: Logo Jira	23
Figure 16: Logo GitLab.....	24
Figure 17: Logo Microsoft Teams	24
Figure 18: Logo Outlook.....	24
Figure 19: Logo IntelliJ.....	25
Figure 20: Logo Postman	25
Figure 21:Planification des sprints	27
Figure 22:Diagramme de cas d'utilisation de sprint 1	32
Figure 23:Diagramme de séquence « Ajout modèle vélo »	35
Figure 24: Diagramme de séquence « ajout magasin ».....	36
Figure 25: Diagramme de classe module de configuration.....	36
Figure 26: Liste des Magasins.....	37
Figure 27: Formulaire d'ajout d'un magasin.....	38
Figure 28: Liste des modèles de vélo	38
Figure 29: Formulaire d'ajout d'un modèle de vélo.....	39
Figure 30: Ajout d'un store	40
Figure 31: Cas de test d'ajout d'un modèle de vélo pour un magasin	41
Figure 32: Types de messages envoyer par le tracker.....	45
Figure 33: Logiciel de simulation	49
Figure 34: Use case trajet et point de trajet	50
Figure 35 : Diagramme de séquence « consulter informations trajet ».....	51
Figure 36 : Diagramme de séquence « Envoie des informations sur trajet et point Trajet »	52
Figure 37: Diagramme de classe	52
Figure 38: Liste te information sur les trajets.....	53
Figure 39: Détaille d'un trajet.....	53
Figure 40: Représentation d'un trajet dans la carte géographique.....	54
Figure 41: Connexion et déconnexion de client TCP	55
Figure 42: Test de trip avec Postman	56
Figure 43: Liste des trips points pour le trajet.....	56
Figure 44: Use case sprint trois	61
Figure 45: Diagramme de séquence consulter statistiques.....	64
Figure 46: Diagramme de séquence statistique vélo	64
Figure 47: Diagramme de classe	65
Figure 48:Détaille d'un trajet.....	66
Figure 49: Liste des vélos.....	66
Figure 50: Liste des trajets pour un vélo	67

Figure 51: Statistique de la vitesse moyenne d'un vélo.....	67
Figure 52: Statistique pour la durée des trajets	68
Figure 53: Statistique pour distance des trajets.....	68
Figure 54: statistique de la distance totale	68
Figure 55: Statistiques de nombre de calories brûlées	69
Figure 56: Statistique de la vitesse moyenne pour totalité des vélos	70
Figure 57: Nombre de calories brûlée pour tous les vélos	70
Figure 58: Test unitaire filtre « date personnalisée »	71
Figure 59: Test unitaire pour le filtre année dernière	72
Figure 60: Suite test unitaire pour le filtre année dernière	73

Tableau 1: Tableau comparatif des applications similaire	10
Tableau 2: Caractéristiques du PC	19
Tableau 3: Caractéristiques du Tracker.....	20
Tableau 4: Backlog de produit	27
Tableau 5: Backlog sprint 1	31
Tableau 6: Les acteurs de sprint 1	31
Tableau 7: Description textuelle de cas d'utilisation « Ajout d'un modèle de vélo ».....	33
Tableau 8: Description textuelle de cas d'utilisation « Ajout d'un nouvelle magasin »	34
Tableau 9: Cas de test « ajout d'un nouveau magasin »	39
Tableau 10: Cas de test "ajout d'un modèle de vélo"	40
Tableau 11: Backlog Sprint 2	44
Tableau 12: Tableau de comparaison des données du trajet entre tracker et le mobile	47
Tableau 13: Tableau de comparaison des données point trajet entre tracker et le mobile	48
Tableau 14: les acteurs de sprint 2	49
Tableau 15 : Description textuelle « Consulter détaille trajet »	51
Tableau 16: Backlog Sprint 3	61
Tableau 17: Identification des acteurs.....	61
Tableau 18: Description textuelle « Consulter vitesse moyenne »	63

Introduction Générale

Ce stage a été réalisé dans le cadre de mon projet de fin d'études pour l'obtention du diplôme d'ingénieur en informatique, spécialité génie logiciel, à l'Institut Supérieur d'Informatique et de Multimédia de Sfax. Pendant ce stage de fin d'études de quatre mois chez Actia Engineering Services, j'ai eu l'opportunité de contribuer de manière significative à l'amélioration de la solution web existante, E_Bike ACTIA.

Le marché des vélos électriques connaît une croissance significative dans le monde entier, avec plus de 50 millions de vélos électriques vendus dans l'UE en 2021, soit une augmentation de 150 % par rapport à 16. La Chine, premier producteur de vélos électriques, a vendu 210 millions de vélos électriques par jour en 2016. En 2022, les vélos électriques ont continué de gagner des parts de marché dans l'UE, représentant 57 % des ventes de vélos aux Pays-Bas, 49 % en Autriche, 48 % en Allemagne et 47 % en Belgique. Cette tendance devrait se poursuivre à mesure que de plus en plus de personnes recherchent des options de transport durables et respectueuses de l'environnement.

La mission assignée consiste à concevoir une application web répondant à des besoins spécifiques liés à la gestion des traceurs installés sur les vélos électriques. Cette application doit permettre l'intégration d'un module de communication avec l'appareil Tracker SC350MG, permettant ainsi la collecte et le traitement en temps réel de données cruciales telles que le niveau le nombre de kilomètres parcourus, la durée, la vitesse moyenne et l'assistance moyenne pendant un trajet.

De surcroît, elle devra inclure des statistiques sur la durée, la vitesse moyenne, le nombre de kilomètres et le nombre de calories brûlées lors des trajets réalisés par chaque vélo du cycliste ou par la totalité de ses vélos.

En plus de ma contribution à l'implémentation du module de communication avec l'appareil Tracker SC350MG, j'ai été chargé de mettre en œuvre des fonctionnalités essentielles pour le projet. Cela comprenait notamment le module de configuration, qui contient la gestion des magasins et des modèles de vélos.

Un "tracker" est un dispositif électronique connecté à Internet, conçu pour suivre et surveiller un objet ou une personne en temps réel. Dans le contexte des vélos électriques, le tracker permet

E_Bike gestion tracker à travers TCP protocole

de collecter des données essentielles pour surveiller et gérer les performances des vélos, ainsi que pour assurer la sécurité des utilisateurs.

Les vélos électriques, également connus sous le nom de e-bikes, sont des bicyclettes équipées d'un moteur électrique qui assiste le cycliste pendant le pédalage. Ces vélos offrent une alternative écologique aux moyens de transport traditionnels et gagnent en popularité en raison de leur capacité à parcourir de longues distances tout en réduisant l'effort physique nécessaire.

Dans ce rapport, je présenterai en détail le processus de conception et de développement de l'application web de gestion de tracker, en mettant en évidence les différentes fonctionnalités implémentées et les technologies utilisées. Je partagerai également mon expérience au sein de Actia Engineering Services, ainsi que les défis rencontrés et les solutions apportées tout au long de ce projet.

Ce rapport comprendra Cinque chapitres, organisés comme suit :

Chapitre 1 : Contexte générale du projet

Ce chapitre introduit le contexte du projet, présente l'organisme d'accueil, Actia Engineering Services, décrit la mission assignée, et analyse l'existant pour identifier les points d'amélioration et les bonnes pratiques à adopter.

Chapitre 2 : Mise en place de l'environnement de travail

Ce chapitre comprendra la description de l'architecture de l'application, de l'environnement matériel et logiciel nécessaire, ainsi que du backlog de produit. Cette préparation stratégique est essentielle pour établir des bases solides, garantir une exécution efficace et améliorer notre capacité à gérer les défis futurs, assurant ainsi le succès du développement de l'application.

Chapitre 3 : Gestion des magasins et des modèles de vélo

Dans ce chapitre, nous amorçons notre premier sprint en nous attardant sur la gestion des magasins et des modèles de vélo. Il englobe une série d'étapes clés, notamment la définition du sprint backlog, l'analyse des besoins avec l'identification des acteurs, le diagramme de cas d'utilisation et la description textuelle des cas d'utilisation, ainsi que la modélisation conceptuelle à travers des diagrammes de séquence et de classe. Nous aborderons également la phase de réalisation, de test et de validation pour garantir la conformité et l'efficacité des fonctionnalités développées.

Chapitre 4 : Communication avec le tracker SC350MG

Ce chapitre s'attache à explorer notre deuxième sprint, centré sur le développement du module

E_Bike gestion tracker à travers TCP protocole

de communication avec le tracker SC350MG. Il offre un aperçu détaillé des différentes étapes impliquées dans ce processus, du sprint backlog initial à la conception, la réalisation et les tests.

Chapitre 5 : Intégration des statistiques

Dans ce cinquième chapitre, nous explorons notre troisième sprint dédié au module d'intégration des statistiques. En collaboration avec le Product Owner, nous avons défini les objectifs clés de ce sprint, qui seront abordés à travers différentes phases telles que la définition du backlog du produit, l'analyse des besoins, la conception, les tests et la validation. Ce chapitre offre un aperçu détaillé du processus de développement et des étapes essentielles pour assurer l'intégration réussie des statistiques dans notre application.

Chapitre 1 : Contexte général du projet

Introduction

Dans ce premier chapitre, nous nous plongeons dans le contexte général du projet en présentant l'organisme d'accueil, Actia Engineering Services. Nous décrivons ensuite la mission qui nous a été attribuée ainsi que la méthodologie choisie pour l'accomplir. Enfin, nous effectuons une analyse critique de l'existant, évaluant les forces et les faiblesses des systèmes en place, ainsi que l'étude des applications similaires.

1.1 L'organisme d'accueil

1.1.1 Présentation

Actia Engineering Services est un centre de recherche et développement (R&D) spécialisé dans les technologies de pointe appliquées aux secteurs de l'automobile et des télécommunications. Fondée en 2005, l'entreprise s'est distinguée par son expertise dans divers domaines, notamment les systèmes connectés, l'architecture électronique de véhicules, les solutions d'électromobilité et les services de diagnostic et de maintenance. Avec une équipe de plus de 800 collaborateurs, principalement des ingénieurs qualifiés, Actia Engineering Services se positionne comme un acteur majeur, offrant des solutions innovantes et adaptées.



Figure 1: Logo Actia ENGINEERING SERVICES

1.1.2 Domaines d'Activité et Savoir-faire

L'ingénierie dans les activités de Actia Engineering Services couvre un large éventail de compétences, allant du développement de logiciels embarqués à la réalisation des tests et des validations produits. L'entreprise excelle également dans le développement web et mobile, ainsi que dans le développement mécanique et électronique. Elle excelle également dans la conception et la commercialisation d'outillage de test et de production, renforçant ainsi sa position en tant que fournisseur de solutions complètes et intégrées.

1.1.3 Engagement en Qualité et Récompenses

La qualité et l'excellence sont au cœur des préoccupations d'Actia Engineering Services. Certifiée selon les normes ISO9001, ISO17025, ISO27001 et les normes CMMI dev v1.3, l'entreprise s'engage à fournir des produits et des services répondant aux plus hauts standards

E_Bike gestion tracker à travers TCP protocole

de qualité et de sécurité. Actia Engineering Services a été régulièrement récompensée pour son engagement envers l'innovation et l'excellence, avec des récompenses telles que le 1er prix du concours national tunisien de l'innovation en 2021 et le prix récompensant les avancées réalisées dans la gestion des ressources humaines en période de crise Covid-19 en 2020.

1.2 Présentation du projet

1.2.1 Mission à accomplir

La mission assignée consiste à concevoir une application web répondant à des besoins spécifiques liés à la gestion des trackers SC350MG installés sur les vélos électriques. Cette application doit permettre l'intégration d'un module de communication avec l'appareil Tracker SC350MG, permettant ainsi la collecte en temps réel de données cruciales telles que le niveau de charge de la batterie, le nombre de kilomètres parcourus, la durée, la vitesse moyenne et l'assistance moyenne pendant un trajet. De surcroît, elle devra inclure des statistiques sur la durée, la vitesse moyenne, le nombre de kilomètres et le nombre de calories brûlées lors des trajets réalisés par chaque vélo du cycliste ou par la totalité de ses vélos. Les exigences du projet englobent également la rétro-ingénierie des microservices existants pour la gestion des magasins et des modèles de vélo et l'ajout d'une authentification via les réseaux sociaux.

1.2.2 Présentation de tracker SC350MG

Le SC350MG est un tracker GNSS résistant à l'eau conçu pour les vélos électriques, les Cyclomoteurs électriques ainsi que les fauteuils roulants électriques. L'appareil exécute une variété de fonctions télématiques. SC350MG a également la capacité de collecter des données ou de contrôler à distance les vélos électriques via le port CAN ou UART, avec lequel il peut communiquer avec les contrôleurs des vélos, les instruments et les systèmes de gestion de batterie, permettant une gestion intelligente des vélos électriques.

1.3 Méthodologie utilisée

1.3.1 Les méthodes agiles

La méthodologie agile est une approche itérative et incrémentale qui met l'accent sur la collaboration interfonctionnelle, l'adaptation continue aux changements et la livraison rapide de produits de haute qualité. Elle favorise une approche flexible et réactive au développement logiciel, permettant aux équipes de s'adapter rapidement aux besoins changeants du projet. Parmi les méthodologies agiles les plus connues, on retrouve Scrum, Kanban, Extreme Programming (XP), Lean Software Development, et Crystal.

1.3.2 La méthodologie Scrum

La méthodologie Scrum est un cadre de développement agile qui repose sur plusieurs piliers essentiels. Les rôles clés incluent le Product Owner, le Scrum Master et l'équipe de développement.

Rôles	Description
Product Owner	Chargé de définir la vision du produit, de prioriser le backlog produit et de s'assurer que les besoins des parties prenantes sont pris en compte dans le développement du produit.
Scrum Master	Responsable de faciliter le processus Scrum, de s'assurer que l'équipe respecte les principes et les pratiques Scrum, et d'éliminer les obstacles qui entravent la progression de l'équipe.
Équipe de développement	Groupe auto-organisé et multidisciplinaire responsable de la réalisation des fonctionnalités du produit.

En ce qui concerne les artefacts, Scrum utilise le Product Backlog, une liste hiérarchisée des fonctionnalités à développer, le Sprint Backlog, une sélection de tâches à accomplir pendant le sprint, et l'Incrément de produit, une version opérationnelle du produit livrée à la fin de chaque sprint.

Artefacts	Description
Product Backlog	Liste hiérarchisée des fonctionnalités à développer, établie par le Product Owner en collaboration avec les parties prenantes et priorisée en fonction de la valeur commerciale.
Sprint Backlog	Sélection de tâches à accomplir pendant le sprint, déterminée par l'équipe de développement à partir du Product Backlog et basée sur les objectifs du sprint.
Incrément de produit	Version opérationnelle du produit, résultat du travail réalisé pendant le sprint et prête à être livrée à la fin de chaque itération.

La durée typique d'un sprint dans Scrum est de deux semaines à quatre semaines, ce qui permet à l'équipe de développement de se concentrer sur un ensemble spécifique de fonctionnalités et de livrer un résultat tangible à la fin de chaque itération. Les bénéfices de la méthode Scrum incluent une meilleure visibilité et transparence dans le processus de développement, une adaptation plus rapide aux changements des exigences du client, une amélioration continue grâce aux rétrospectives régulières et une augmentation de la satisfaction client grâce à une livraison fréquente de fonctionnalités.

E_Bike gestion tracker à travers TCP protocole

La figure suivante illustre le cycle de vie de la méthodologie Scrum

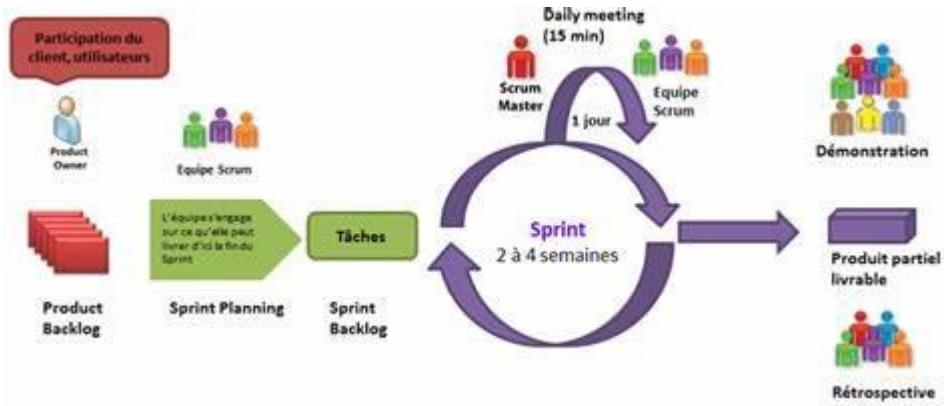


Figure 2: Cycle de vie de la méthodologie Scrum

1.4 Application similaire

1.4.1 VELCO FLEET MANAGEMENT :

Pour la communication avec le vélo électrique, la société VELCO utilise un tracker pour envoyer les données du vélo à une application web. Les fonctionnalités de Velco Fleet Management sont les suivantes :

- Suivi en temps réel de la position des vélos, du pourcentage de batterie, et de leur état (stationné ou en déplacement).
- Collecte et visualisation des données d'utilisation, telles que la distance moyenne parcourue, la vitesse, les kilomètres réalisés, ainsi que les économies en euros et en CO2 par rapport à une voiture.
- Possibilité de couper l'assistance électrique à distance et de bloquer les pédales.
- Surveillance en temps réel des données de maintenance préventive et curative.

1.4.2 GPSLive de Rewire Security :

L'application GPSLive de Rewire Security pour gérer le tracker GPS offre les fonctionnalités suivantes :

- Accès illimité : Permet un accès sans restriction aux fonctionnalités importantes du compte de suivi GPS Live.
- Tableau de bord : Affiche un récapitulatif de tous les véhicules, telles que des statistiques sur la durée des trajets, le nombre de kilomètres, et la vitesse moyenne.
- Suivi en temps réel : Affiche l'emplacement exact du véhicule, sa vitesse, son statut, et

E_Bike gestion tracker à travers TCP protocole

son adresse en temps réel.

- Historique : Permet de visualiser la rediffusion historique des voyages effectués sur une période allant jusqu'aux 3 derniers mois.
- Alertes en temps réel : Permet de configurer des alertes basées sur l'activité en temps réel, comme le mouvement et la vitesse du véhicule.
- Diagnostic : Affiche un résumé de toutes les informations de diagnostic relatives aux véhicules du compte.



Figure 3: Interface d'application web et mobile GPSLive

1.4.3 Tableau comparatif :

Ce tableau comparatif met en évidence les fonctionnalités offertes par deux solutions de suivi GPS, GPSLive de Rewire Security et Velco Fleet Management. Les deux systèmes sont conçus pour répondre aux besoins de suivi des positions, bien qu'il existe des différences significatives dans leur champ d'application et leurs fonctionnalités spécifiques. L'analyse de ces caractéristiques clés permet d'évaluer les avantages et les limites de chaque solution.

E_Bike gestion tracker à travers TCP protocole

Fonctionnalités	GPSLive	Velco Fleet Management
Accès illimité	Oui	Oui
Suivi en temps réel de l'emplacement, de vitesse	Oui	Oui
Historique des trajets et des détails du trajet	3 mois	Toute l'historique
Alertes en temps réel	Dépassement de vitesse	Vol
Informations de diagnostic	Affiche un résumé de toutes les informations de diagnostic relatives aux véhicules du compte.	Suivi en temps réel de la maintenance préventive et curative de la batterie et contrôleur
Données utilisateurs	Oui	Oui
Données telles que la distance, vitesse etc.	Oui	Oui
Sécurité contre les vols	Oui	Oui
Suivi GPS	Elle peut être utilisée pour les vélo électrique, les véhicules etc.	Spécifique au vélo électrique de Velco
Statistique sur les trajets, et les nombres de kilomètre	Oui (limitée à 3 mois)	Oui

Tableau 1: Tableau comparatif des applications similaires

Dans le cadre du développement de notre application web de gestion des trackers, cette analyse comparative nous permettra de mieux comprendre les fonctionnalités clés nécessaires pour répondre aux besoins de nos utilisateurs. En combinant les avantages de chaque solution et en tenant compte des exigences spécifiques de notre projet, nous serons en mesure de concevoir une application web efficace et appropriée, offrant des fonctionnalités robustes pour la gestion des trackers tout en répondant aux attentes de nos utilisateurs finaux.

1.5 Etude de l'existant

Nous entamerons cette section en explorant l'application mobile et web E_Bike ACTIA, un produit développé par la société ACTIA ES.

1.5.1 Présentation de E_Bike ACTIA

L'application E_Bike développée par ACTIA ES offre une solution complète et interconnectée

E_Bike gestion tracker à travers TCP protocole

pour les utilisateurs de vélos à assistance électrique (VAE), combinant une application mobile et une application web. La synchronisation entre ces deux interfaces permet une expérience utilisateur fluide et enrichissante. La partie mobile joue un rôle crucial en collectant les données du vélo via Bluetooth et en les transmettant au serveur Web via Internet.

Cette image vous illustre le fonctionnement de E_Bike ACTIA :

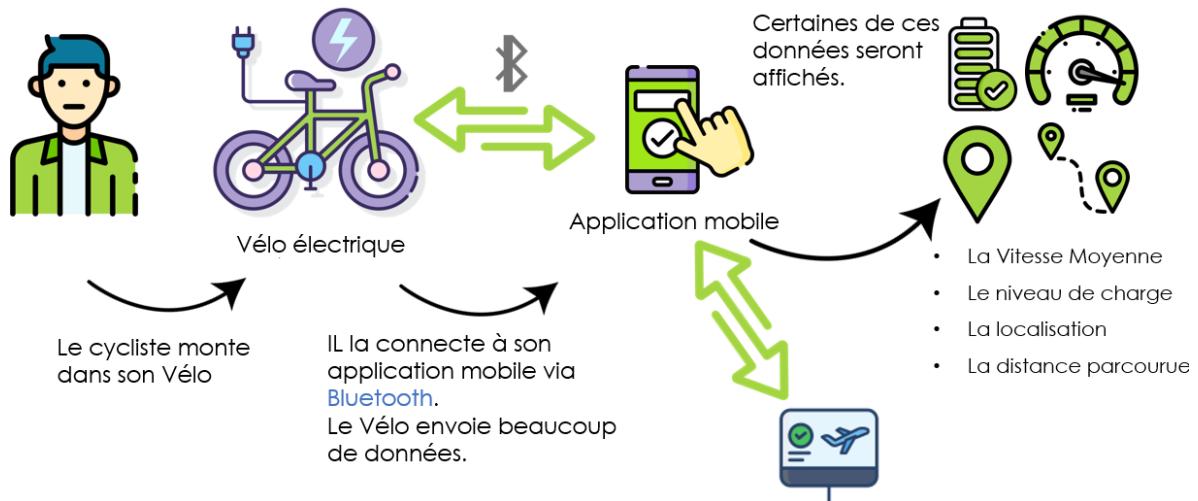


Figure 4: Etude de l'existant

Parmi les fonctionnalités, on trouve :

- Niveau de batterie
- Erreurs du système
- Détails des trajets (vitesse, distance parcourue, calories brûlées ...)
- Informations météorologiques
- Diagnostics du système électrique
- Alertes en cas d'événements survenant notamment liés au système moteur, au contrôleur, à la batterie, aux accessoires, ainsi qu'aux sous-systèmes.

Dans cette application, il existe 4 rôles :

Administrateur (Admin) qui assure :

- Contrôle global de la plateforme
- Gestion des utilisateurs et des données
- Surveillance du bon fonctionnement de l'application

Cycliste :

- Accès personnalisé à ses propres données et ses informations à propos les trajets
- Réception des données sur ses vélos (niveau de charge, localisation, etc.)
- Faire une réclamation en cas d'une non-conformité.

Administrateur SAV (Service Après-Vente) :

- Réception des réclamations des utilisateurs
- Gestion des défauts système et des erreurs
- Maintenance du parc de VAE

Utilisateur SAV (Service Après-Vente) :

- Traitement des problèmes signalés par l'Admin SAV
- Intervention sur les défauts et les erreurs du système

1.5.2 Critique de l'existant

Dans le cadre de l'évaluation de l'application E_Bike développée par ACTIA ES, plusieurs limitations et lacunes ont été mises en évidence, soulevant ainsi des points critiques à prendre en considération. Bien que E_Bike ACTIA offre une expérience interconnectée et interactive pour les utilisateurs de vélos à assistance électrique (VAE), plusieurs éléments méritent une attention particulière :

- **Contraintes liées à la disponibilité du réseau mobile :** Les utilisateurs peuvent rencontrer des difficultés dans les zones où la couverture réseau est faible ou inexistante, ce qui peut compromettre la collecte et la transmission des données de trajet.
- **Impact sur l'autonomie du smartphone :** L'utilisation intensive de l'application mobile peut entraîner une décharge rapide de la batterie du smartphone, ce qui peut être problématique pour les utilisateurs effectuant de longs trajets ou utilisant leur smartphone pour d'autres tâches essentielles.
- **Dépendance de l'application mobile :** Le passage par une application mobile pour la collecte des données avant de les envoyer à l'application web crée une dépendance à un intermédiaire. Cela peut entraîner des problèmes potentiels tels que des retards, des erreurs de transmission ou une perte de données. De plus, cette configuration peut empêcher la réception des données de trajet en temps réel, rendant les informations moins précises.
- **Dépendance des utilisateurs à leur smartphone lors des trajets :** Les utilisateurs doivent utiliser leur smartphone de manière constante pendant leurs trajets pour que l'application puisse collecter les données nécessaires. Cela peut les rendre dépendants de leur appareil, limitant ainsi leur capacité à utiliser leur smartphone pour d'autres activités ou à effectuer des trajets sans leur smartphone.

E_Bike gestion tracker à travers TCP protocole

La société a identifié ces problèmes et a proposé une solution consistant à ajouter un tracker dans le vélo électrique. Cela permettrait à l'utilisateur d'avoir le choix d'utiliser soit son smartphone, soit le tracker intégré pour collecter les données de manière indépendante, offrant ainsi une alternative aux utilisateurs qui ne souhaitent pas ou ne peuvent pas utiliser un smartphone pendant leurs trajets.

1.5.3 Solutions proposées

Pour remédier aux limitations identifiées dans l'application E_Bike et améliorer l'expérience utilisateur, une solution a été proposée : l'ajout d'un tracker intégré dans le vélo électrique.

- Permettre de réduire la dépendance au smartphone et d'assurer une collecte de données fiable même en zones à faible couverture réseau.
- Offrir aux utilisateurs la possibilité de choisir entre l'utilisation de leur smartphone ou du tracker intégré pour collecter les données, réduisant ainsi leur dépendance au smartphone.
- Mettre en place un mécanisme de synchronisation automatique et en temps réel pour réduire les erreurs de transmission et les pertes de données, garantissant que l'application web reçoit les données directement sans dépendre de l'application mobile.

Ces solutions visent à renforcer l'efficacité et la convivialité de l'application E_Bike développée par ACTIA ES, tout en offrant une alternative aux limitations actuelles identifiées. En intégrant un tracker autonome et en optimisant les performances de l'application, nous cherchons à améliorer l'expérience utilisateur et à fournir des données précises et en temps réel sur les trajets réalisés.

1.6 Conclusion

En conclusion, cette étude nous a permis d'avoir une vision claire de notre projet de gestion des trackers pour l'application web E_Bike ACTIA. En explorant l'organisation hôte, la mission assignée, la méthodologie adoptée et une analyse critique de l'existant, nous avons identifié plusieurs besoins et fonctionnalités clés. De plus, la comparaison entre GPSLive de Rewire Security et Velco Fleet Management a mis en lumière les différentes fonctionnalités disponibles dans le domaine de la gestion des trackers, fournissant ainsi un cadre précieux pour le développement de notre solution. Ces informations nous aideront à concevoir une application robuste et adaptée aux besoins de nos utilisateurs.

Chapitre 2 : Mise en place de l'environnement de travail

Introduction

Dans ce chapitre, nous mettrons en lumière la mise en place de l'environnement de travail. Nous décrirons les activités initiées lors de cette phase, telles que la planification du projet, la configuration de l'environnement de développement et la définition des objectifs. Nous soulignerons également l'importance stratégique dans l'établissement d'une base solide pour le succès du projet. Cette phase préliminaire revêt une importance capitale, car elle jette les fondements essentiels pour la réalisation efficace du projet. En mettant en place un cadre solide dès le départ, nous sommes mieux préparés à aborder les défis potentiels et à maximiser les chances de réussite tout au long du développement de l'application.

2.1 Spécification architecturale

2.1.1 Architecture logicielle frontend

Dans l'architecture Angular, les modules servent à organiser l'application en regroupant des fonctionnalités connexes. Chaque module peut contenir des composants, des services et d'autres artefacts.

Les composants sont les blocs de construction de l'interface utilisateur, représentant des éléments autonomes de l'application. Les templates fournissent la structure HTML des composants, définissant leur apparence et leur comportement.

Le système d'injection de dépendances (Injector) permet de fournir des instances de services et d'autres dépendances aux composants et autres classes.

Enfin, les métadonnées sont des informations fournies sous forme de décorateurs ou de décorations, utilisées pour annoter les classes Angular et fournir des informations supplémentaires sur leur fonctionnement et leur configuration.

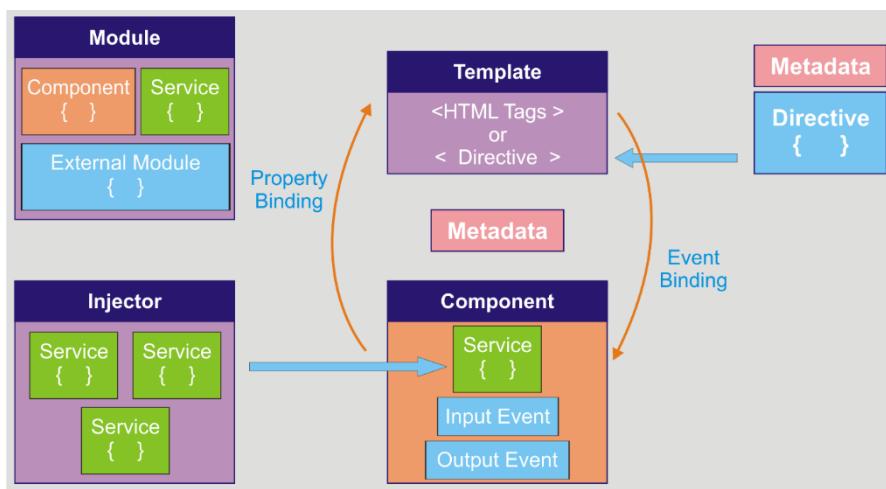


Figure 5: Architecture Angular

2.1.2 Architecture logicielle du backend

L'architecture microservices de notre application repose sur Spring Boot et comprend plusieurs composants clés, notamment la passerelle (gateway), le serveur d'authentification (UAA), le registre de services (registry) et d'autres microservices tels que diag, dashboard et ticketing.

Le serveur d'authentification (UAA) gère les identités et l'accès aux services, tandis que le registre (Registry) permet aux microservices de s'enregistrer et de découvrir les uns les autres. Passerelle (Gateway) joue également un rôle de chef d'orchestre il permet la redirection des microservices et la communication entre eux.

Les autres microservices comme diag, dashboard et ticketing sont responsables de fonctionnalités spécifiques de l'application.

Ces microservices communiquent entre eux via des API REST, et sont déployés indépendamment les uns des autres.

Cette figure représente les différant microservice de notre application

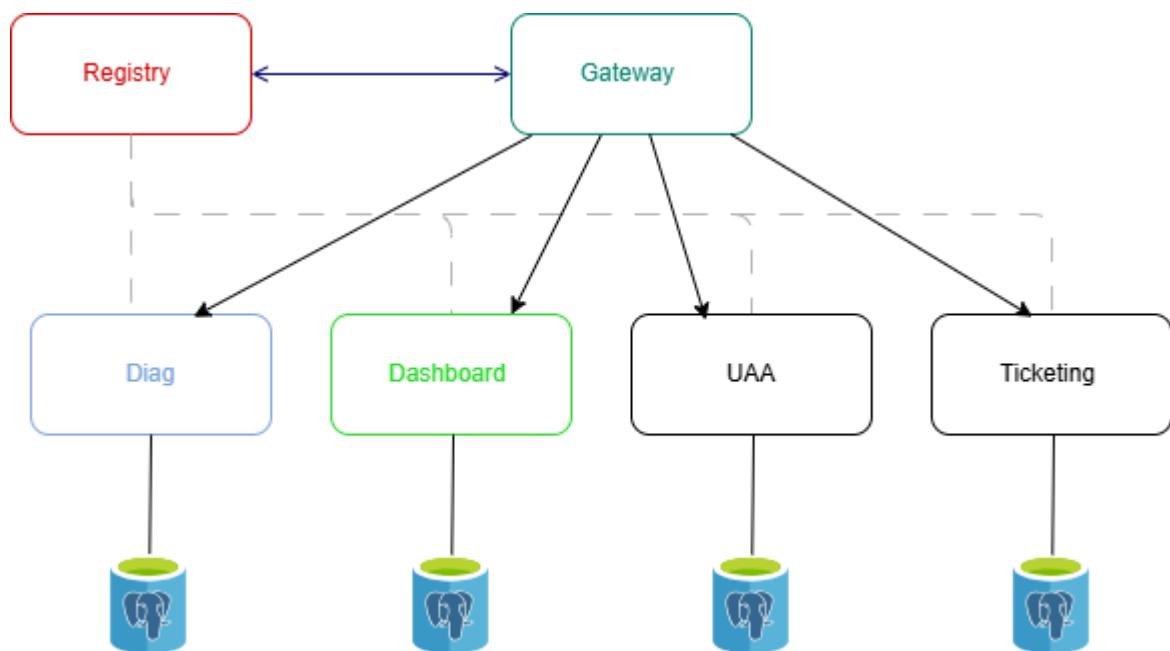


Figure 6: Architecture microservice backend de E_Bike ACTIA

Chaque microservice est généralement responsable d'une seule responsabilité métier et est construit autour d'un domaine spécifique de l'application. L'architecture d'un microservice Spring Boot se compose de quatre composants principaux :

- Les modèles (Models) pour représenter les entités métier.
- Les repositories pour accéder aux données.
- Les services pour encapsuler la logique métier.

E_Bike gestion tracker à travers TCP protocole

- Les contrôleurs (Controllers) pour gérer les requêtes HTTP.

Cette structure en couches offre une séparation claire des responsabilités, favorisant ainsi la réutilisation du code et la maintenance facile.

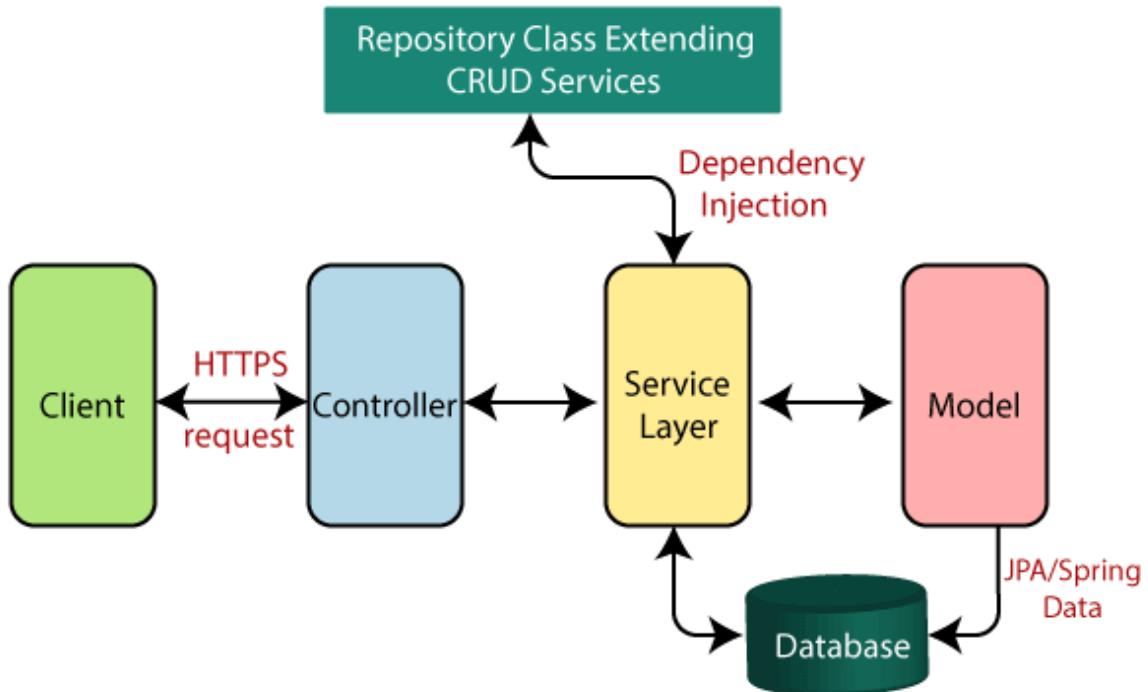


Figure 7: Architecture Spring Boot

2.1.3 Architecture globale

Dans l'architecture globale de notre système, il existe deux parties : la partie reliée au tracker et la partie reliée à la gestion de configuration.

- **Gestion de tracker :**

Chaque tracker est installé dans un vélo et contient une carte SIM pour assurer sa connexion à Internet. Le serveur Tracker envoie des requêtes d'informations, de configuration au tracker. Le tracker renvoie un message d'accusé de réception (ACK) pour confirmer la réception du message, puis il envoie des Trame de messages contenant plusieurs informations telles que le niveau de batterie, la longitude, la latitude et la vitesse.

Le serveur Tracker doit être capable de se connecter à Internet et d'écouter les connexions provenant des trackers à travers Internet. Cela permet aux trackers distants de se connecter au serveur Tracker depuis n'importe où dans le monde, tant qu'ils ont une connexion Internet. En utilisant des protocoles comme TCP, les trackers peuvent alors communiquer avec le serveur

E_Bike gestion tracker à travers TCP protocole

Tracker pour envoyer et recevoir des données.

Ensuite, un client TCP reste en écoute pour la réception d'une trame de message provenant du serveur Tracker. Le client TCP récupère cette trame, puis un service effectue le traitement de ces données et enregistre les données pertinentes dans la base de données PostgreSQL.

Dans la partie de gestion des trackers, lorsque l'utilisateur souhaite visualiser ses données, des requêtes HTTPS sont également utilisées. Angular envoie des demandes HTTPS au serveur backend Spring Boot pour récupérer les données traitées envoyées par le tracker.

- **Gestion de configuration**

Pour la gestion de configuration, L'architecture est basée sur une architecture trois tiers.

Front-end (Interface Utilisateur) : Angular

L'interface utilisateur est développée en Angular, permettant aux utilisateurs de créer, lire, mettre à jour et supprimer des données relatives aux magasins et modèles de vélos. Angular communique avec le serveur back-end via des requêtes HTTPS sécurisées.

Back-end (Logique Métier) : Spring Boot

Le serveur back-end est développé en utilisant Spring Boot, gérant la logique métier de l'application. Il reçoit les requêtes HTTPS du front-end, les traite et interagit avec la base de données pour récupérer ou mettre à jour les données nécessaires.

Persistance des données : PostgreSQL

La base de données PostgreSQL est utilisée pour stocker de manière persistante les informations sur les magasins et les modèles de vélos. Spring Boot se connecte à PostgreSQL pour lire et écrire des données dans la base de données.

E_Bike gestion tracker à travers TCP protocole

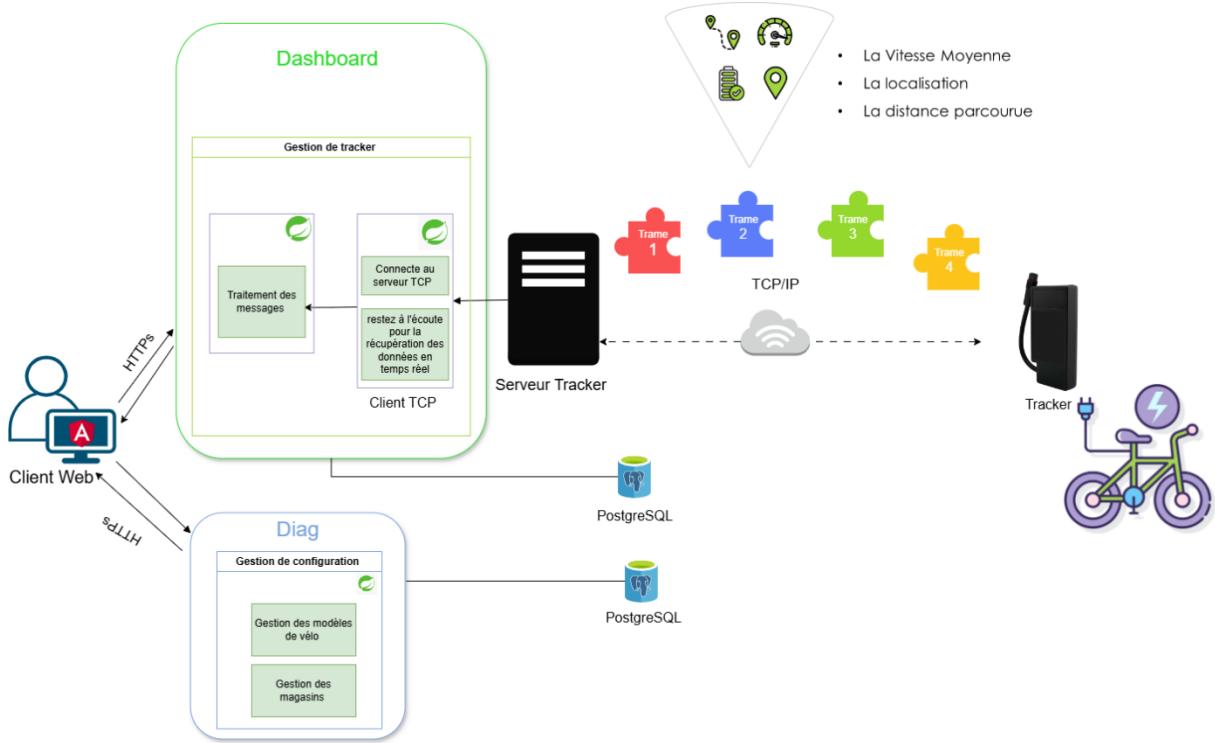


Figure 8: Architecture globale

2.2 Spécification technologique

2.2.1 Environnement matériel

Dans le cadre de ce projet, nous avons utilisé un PC doté des spécifications suivantes :

Marque	Lenovo
Processeur	Intel® core™ i7 13ème génération
Mémoire vive	16 Go
Système d'exploitation	Windows 10

Tableau 2: Caractéristiques du PC

Un tracker dotée des fonctionnalités suivantes :

Dimensions	95*45*16.2mm
Poids	88.5g
Tension de fonctionnement	8~60V DC
Batterie de secours	500mAh
Température de fonctionnement	-20°C ~ +70°C
Capteur	Accéléromètre triaxial, pour la détection de mouvement
Résistant à l'eau	IPX5

E_Bike gestion tracker à travers TCP protocole

CAN	1 x CAN haut & CAN bas
Types ECU pris en charge	Bosch, Bafang
Ports série	1 x UART
Types ECU pris en charge	Rhino2
Sorties numériques	2 x sorties numériques
Entrée numérique	1 entrée de déclenchement positive
USB BLE	BLE 5.2, pour le déverrouillage des vélos électriques
Type GNSS	Récepteur GNSS tout-en-un u-blox
Sensibilité	Autonome : -147 dBm
Démarrage à chaud	-156 dBm
Reprise	-160 dBm
Suivi	-162 dBm
Précision de position (CEP)	Autonome : < 2.5m
TTFF (Open Sky)	Démarrage à froid : 30 secondes en moyenne
Démarrage à chaud	27 secondes en moyenne
Démarrage rapide	1 seconde en moyenne

Tableau 3: Caractéristiques du Tracker

2.2.2 Environnements logiciels

Pour la mise en œuvre de notre projet, nous avons exploité plusieurs Framework, logiciels et technologies, lesquels seront examinés en détail dans la section suivante :

- **Outil de conception :**

Draw.io :

draw.io est un outil de diagramme en ligne qui offre une interface conviviale pour créer une variété de diagrammes, tels que des organigrammes, des diagrammes de flux, des diagrammes UML, et bien plus encore. Il permet aux utilisateurs de collaborer en temps réel, d'importer et d'exporter des fichiers dans différents formats, et offre une intégration avec diverses plates-formes de stockage en ligne.



Figure 9: Logo draw.io

- **Framework de développement :**

Spring BOOT :

Spring Boot est un Framework Java qui simplifie le développement d'applications web en fournissant une configuration par défaut et en intégrant plusieurs bibliothèques et outils. Grâce à sa conception orientée convention plutôt que configuration, Spring Boot permet aux développeurs de se concentrer sur la logique métier sans se soucier de la configuration fastidieuse. Il offre une prise en charge native de l'application basée sur REST, une gestion simplifiée des dépendances et une intégration transparente avec d'autres projets Spring. En outre, Spring Boot facilite le déploiement et la mise à l'échelle des applications grâce à son intégration étroite avec les serveurs d'application et les conteneurs de servlet.



Figure 10: Logo Spring Boot

Angular 10 :

Angular est un Framework open source développée par Google, principalement utilisé pour construire des applications web monopages (SPA) dynamiques et évolutives. Il utilise Type Script comme langage de programmation et suit une architecture basée sur les composants, où chaque élément de l'interface utilisateur est encapsulé dans un composant réutilisable. Angular offre une large gamme de fonctionnalités, y compris la liaison de données bidirectionnelle, la gestion de la navigation, le routage, la validation de formulaire, la communication avec des serveurs distants via des requêtes HTTP, et bien plus encore. Avec sa robustesse, sa modularité et sa communauté active, Angular est largement adopté pour le développement d'applications web modernes.

E_Bike gestion tracker à travers TCP protocole



Figure 11: Logo Angular CLI

- **Les langages de programmation :**

Java :

Java est un langage de programmation polyvalent, apprécié pour sa portabilité, sa fiabilité et sa sécurité. Utilisé dans une variété de domaines, notamment le développement d'applications web, mobiles et d'entreprise, il est soutenu par une vaste communauté de développeurs et une riche collection de bibliothèques et de Framework.



Figure 12: Logo JAVA

TypeScript :

TypeScript est un langage de programmation open source développé par Microsoft. Il s'agit d'un sur-ensemble de JavaScript qui ajoute des fonctionnalités de typage statique optionnel, permettant aux développeurs de détecter et de corriger les erreurs de typage à la phase de développement plutôt qu'à l'exécution. TypeScript est largement utilisé pour le développement d'applications web côté client, notamment avec les Framework Angular et React, ainsi que pour le développement côté serveur avec Node.js.



Figure 13: Logo TypeScript

- **Système de gestion de base de données :**

PostgreSQL :

PostgreSQL est un système de gestion de base de données relationnelle (SGBDR) open source,

E_Bike gestion tracker à travers TCP protocole

conçu pour être robuste, extensible et conforme aux normes ANSI SQL. Connue pour sa fiabilité et sa performance, PostgreSQL prend en charge un large éventail de fonctionnalités avancées, notamment les transactions ACID, les vues matérialisées, les index multicolonnes, la réPLICATION et la montée en charge horizontale. Il est utilisé dans de nombreux environnements de production, allant des petites applications aux grandes applications.



Figure 14: Logo PostgreSQL

- **Outils communication et de collaboration :**

Jira :

Jira est un outil de gestion de projet puissant et flexible utilisé par les équipes de développement logiciel et d'autres secteurs pour planifier, suivre et gérer les tâches et les problèmes. Il offre des fonctionnalités avancées telles que la gestion des tickets, le suivi du temps, la planification agile, les tableaux Kanban et les rapports personnalisés. Grâce à son interface conviviale et à sa personnalisation étendue, Jira facilite la collaboration et l'organisation des équipes pour optimiser l'efficacité et la productivité.



Figure 15: Logo Jira

GitLab :

GitLab est une plateforme de gestion du cycle de vie des applications DevOps, offrant un ensemble complet d'outils pour le développement de logiciels, la collaboration, l'intégration continue (CI), le déploiement continu (CD) et la gestion de projet. Il permet aux équipes de coder, de tester, de déployer et de surveiller leurs applications dans un seul endroit, favorisant ainsi la collaboration et l'efficacité tout au long du processus de développement. GitLab prend en charge la gestion de code source avec Git, le suivi des problèmes, la planification agile, la gestion de pipelines CI/CD, la surveillance des performances et bien plus encore.

E_Bike gestion tracker à travers TCP protocole



Figure 16: Logo GitLab

Microsoft Teams :

Microsoft Teams est une plateforme de communication et de collaboration qui permet aux équipes de travailler ensemble de manière efficace, même à distance. Intégré à la suite Microsoft 365, Teams offre des fonctionnalités telles que les conversations en ligne, les réunions vidéo, le partage de fichiers, les calendriers partagés, les intégrations d'applications tierces et bien plus encore. Il fournit un espace de travail numérique centralisé où les membres d'une équipe peuvent communiquer, collaborer et coordonner leurs activités, améliorant ainsi la productivité et la cohésion de l'équipe.



Figure 17: Logo Microsoft Teams

Outlook :

Outlook est une application de messagerie électronique et de gestion de calendrier développée par Microsoft. Elle offre des fonctionnalités telles que l'envoi et la réception d'e-mails, la gestion de contacts, le calendrier pour planifier des rendez-vous et des réunions, ainsi que des tâches et des notes pour organiser les activités quotidiennes. Outlook est largement utilisé dans les environnements professionnels et personnels pour gérer la communication et l'organisation de manière efficace et centralisée.



Figure 18: Logo Outlook

• **Les outils de développement :**

IntelliJ IDEA :

IntelliJ IDEA est un environnement de développement intégré (IDE) développé par JetBrains, principalement utilisé pour le développement d'applications Java, Kotlin, Groovy, Scala et

E_Bike gestion tracker à travers TCP protocole

d'autres langages de programmation. Il offre des fonctionnalités avancées telles que l'achèvement automatique du code, la refactorisation intelligente, le débogage, les outils de test et l'intégration avec des systèmes de gestion de versions tels que Git. IntelliJ IDEA est apprécié pour sa puissance, sa facilité d'utilisation et son large éventail de plugins.

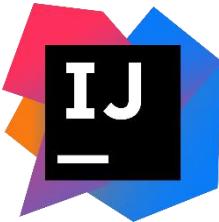


Figure 19: Logo IntelliJ

Postman :

Postman est une plateforme de développement d'API qui permet aux développeurs de créer, tester, documenter et partager facilement des API. Il propose une interface conviviale pour envoyer des requêtes HTTP vers des API, examiner les réponses, organiser les collections d'API, automatiser les tests, créer des scénarios de test complexes, générer des rapports et bien plus encore. Postman est largement utilisé dans le processus de développement d'API pour faciliter le développement, la collaboration et le débogage des API.



Figure 20: Logo Postman

2.3 Planification itérative

2.3.1 Backlog de produit

Le tableau suivant montre le backlog de produit pour la période de mon stage, qui a commencé le 13 février 2024 et prendra fin le 30 juin.

ID	Sprint	Fonctionnalité	ID	User story	Estimation	Priorité
1	Gestion de configuration et d'authentification	Gestion des magasin	1.1	En tant que Admin SAV, je souhaite gérer les magasins.	4 Semaine	Haute
		Gestion des modèles de vélo	1.2	En tant que Admin SAV, je souhaite gérer le modèle de vélo de chaque magasin.		

E_Bike gestion tracker à travers TCP protocole

		Authentification	1.3	En tant qu'utilisateur de l'application web E_Bike ACTIA, je souhaite utiliser mon compte Facebook ou Google pour m'authentifier.		
2	Communication avec le tracker	Etude Tracker	2.1	En tant que développeur je dois effectuer une étude sur le tracker.	4 Semaine	Haute
			2.2	En tant que développeur je dois faire une communication avec le tracker.		
			2.3	En tant que cycliste, je souhaite consulter la liste de mes trajets.		
			2.4	En tant que cycliste je souhaite consulter le détail de mon trajet.		
3	Intégration des statistiques	Gestion statistique pour un seul vélo	3.1	En tant que cycliste, je souhaite accéder aux statistiques de mes trajets pour un vélo spécifique, y compris la durée, la vitesse, la distance parcourue et les calories brûlées. Je souhaite pouvoir filtrer ces données par semaine, mois, année	4 Semaine	Moyenne

E_Bike gestion tracker à travers TCP protocole

				ou date personnalisée pour obtenir des informations pertinentes pour ce vélo particulier.	
	Gestion des statistiques pour totalité des vélo	3.2		En tant que cycliste je souhaite accéder à d'autres statistiques générales qui englobent l'ensemble des vélos que je possède, telles que le nombre total de kilomètres parcourus, le temps total passé à vélo et le nombre total de calories brûlées sur l'ensemble de mes trajets.	

Tableau 4: Backlog de produit

2.3.2 Planification des Sprints

La planification des sprints est une étape cruciale dans la méthodologie agile, permettant à l'équipe de développement de définir les objectifs et les livrables pour une période donnée. Dans notre processus, nous utilisons Jira, un outil de gestion de projet agile, pour faciliter cette planification. La capture d'écran de Jira ci-dessous illustre notre tableau de planification des sprints.

The screenshot shows the Jira backlog with four items:

- EBIKE-TR Sprint Configuration**: 6 tickets, ACTIF, 11/03/24 14:43 - 08/04/24 14:43, Afficher les pages associées
- EBIKE-TR Sprint Stats**: 1 ticket, ACTIF, 29/04/24 14:45 - 27/05/24 14:45, Afficher les pages associées
- EBIKE-TR Sprint COMM Tracker**: 4 tickets, 08/04/24 14:44 - 06/05/24 14:44, Afficher les pages associées, Démarrer le sprint
- EBIKE-TR Sprint kickoff**: 2 tickets, 13/02/24 14:51 - 12/03/24 14:51, Afficher les pages associées, Démarrer le sprint

Figure 21:Planification des sprints

2.4 Conclusion

Dans ce chapitre, nous avons présenté l'architecture du projet et les outils et équipements utilisés. Nous avons ensuite développé le premier artefact Scrum, le backlog de produit. Dans le chapitre suivant, nous présentons le premier sprint.

Chapitre 3 : sprint 1 : Gestion des magasins et des modèles de vélo

Introduction

Dans ce premier sprint, nous avons décidé, en collaboration avec le Product Owner, de débuter par la partie de gestion des magasins et des modèles de vélos. Cette décision a été prise afin de nous adapter à l'architecture de l'application web et de mieux comprendre le projet dans son ensemble. Pour ce faire, nous avons entrepris la rétro-ingénierie des microservices existants, ce qui nous permettra d'appréhender plus efficacement leur fonctionnement et leur intégration dans notre système. En concentrant nos efforts sur cette première partie, nous visons à établir une base solide pour la suite du développement, en assurant une compréhension approfondie des composants essentiels du système et en alignant nos priorités avec les besoins fonctionnels définis dans le backlog de produit.

3.1 Backlog du sprint 1

Le tableau ci-dessus présente les différentes user stories du sprint 1, chacune accompagnée de ses tâches associées. Ces user stories sont axées sur les besoins de l'administrateur du service après-vente (SAV) concernant la gestion des magasins et des modèles de vélo. Chaque tâche est détaillée pour indiquer les actions spécifiques nécessaires pour implémenter ces fonctionnalités, telles que la rétro-ingénierie des micro-services, les tests d'API existantes, la correction et l'ajout de nouvelles API, ainsi que le développement des interfaces utilisateur nécessaires pour gérer les stores et les modèles de vélo.

ID	User story	ID	Taches
3.1	En tant que Admin SAV, je veux gérer les Magasins	3.1.1	Rétroingénierie de micro service de diag et de la partie frontend de l'application E_Bike ACTIA
		3.1.2	Ajouter le menu de configuration de store et modèle de vélo
		3.1.3	Crée la partie de l'affichage de la liste des magasins avec pagination tout en respectant le style de l'application E_Bike ACTIA.
		3.1.4	Crée le formulaire d'ajout d'un magasin

		3.1.5	Crée le formulaire de modification d'un magasin
		3.1.6	Crée le formulaire de suppression d'un magasin
		3.1.7	Implémenté les tests unitaire
3.2	En tant que Admin SAV, je veux gérer les modèles de vélo de chaque magasin	3.2.1	Crée la partie de l'affichage de la liste des modèles de vélo de chaque magasin avec pagination tout en respectant le style de l'application E_Bike ACTIA.
		3.2.2	Crée le formulaire d'ajout d'un modèle de vélo pour un magasin.
		3.2.3	Crée le formulaire de modification d'un modèle de vélo.
		3.2.4	Crée le formulaire de suppression d'un modèle de vélo pour un magasin.
		3.2.5	Implémentation des tests unitaire.

Tableau 5: Backlog sprint 1

3.2 Analyse des besoins

3.2.1 Identification des acteurs

Dans ce sprint, l'acteur principal est l'admin SAV, qui a déjà des responsabilités établies dans l'application telle que la maintenance des parcs de VLE, la gestion et l'intervention en cas de défauts système et d'erreurs, ainsi que la réception, l'analyse et l'affectation des réclamations. En plus de ces tâches existantes, ce sprint se concentrera sur une nouvelle fonctionnalité : la gestion des modèles de configuration.

Le tableau suivant représente les différents rôles de l'acteur Admin SAV :

Acteur	Rôle
Admin SAV	<ul style="list-style-type: none"> Maintenance des parcs de VLE Gestion et intervention des défaut système et des erreurs Réception, Analyse et affectation des réclamation Gestion de modèle de configuration

Tableau 6: Les acteurs de sprint 1

3.2.2 Diagramme de cas d'utilisation de sprint 1 :

La figure ci-dessous représente le diagramme de cas d'utilisation de ce sprint. Les cercles en bleu représentent les fonctionnalités ajoutées dans ce sprint.

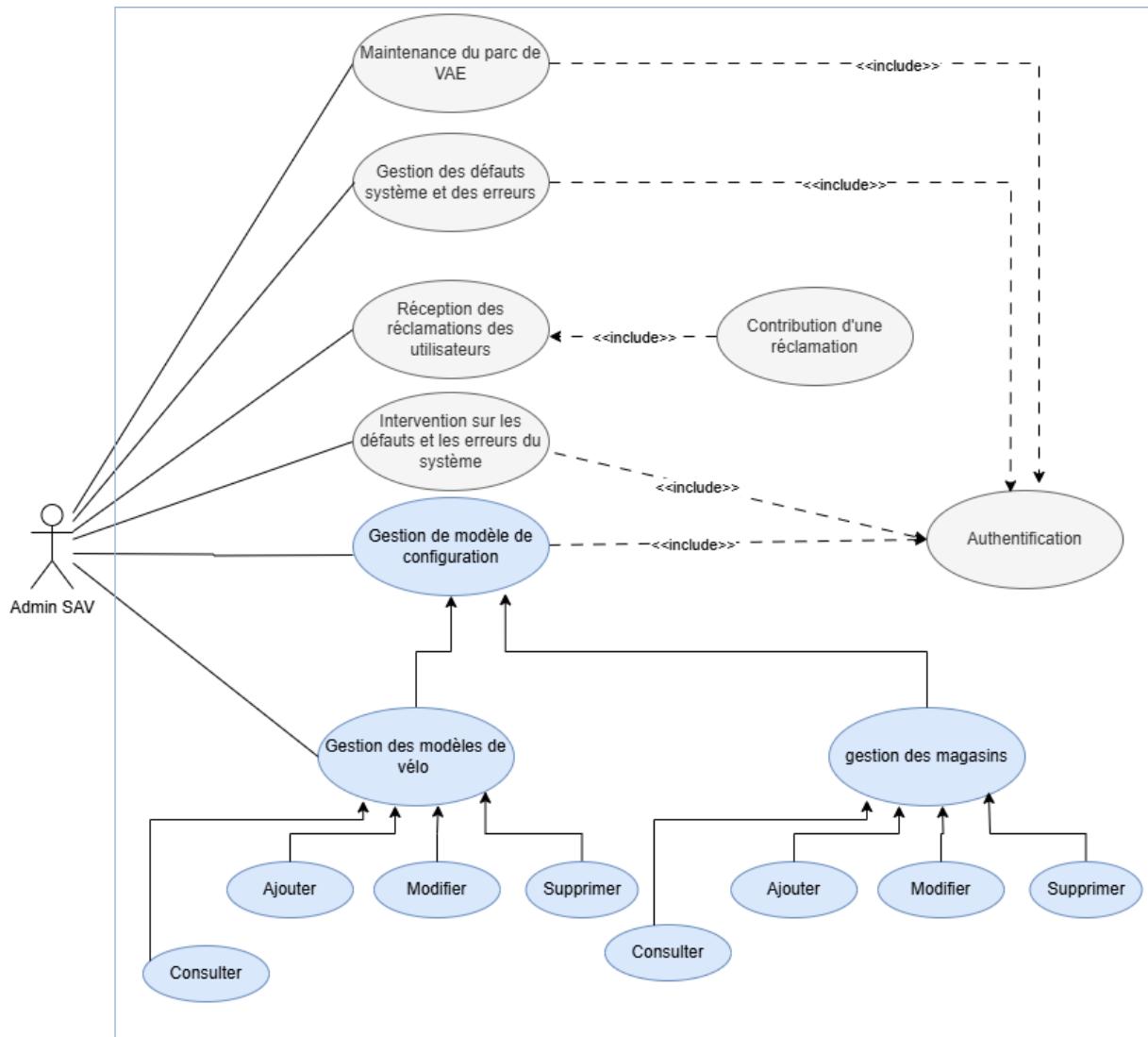


Figure 22:Diagramme de cas d'utilisation de sprint 1

3.2.3 Description textuelle des cas d'utilisation

Les deux descriptions textuelles suivantes détaillent les cas d'utilisation "Ajout d'un modèle de vélo" et "Gestion des magasins", impliquant l'acteur Admin SAV, dans le cadre du sprint en cours.

Description textuelle de cas d'utilisation « Ajout d'un modèle de vélo »

Titre	Ajout d'un modèle de vélo
Acteur	Admin SAV
Précondition	- Disponibilité d'une connexion internet.

E_Bike gestion tracker à travers TCP protocole

	<ul style="list-style-type: none"> - L'admin SAV est déjà Authentifié. - Le magasin dont les quel on va ajouter un modèle de vélo existe.
Postcondition	Après l'exécution des actions de l'administrateur du SAV sur la gestion des modèles de vélo, la liste des modèles de vélo est mise à jour dans la base de données du système
Scénario nominal	<ol style="list-style-type: none"> 1.L'admin SAV accède au formulaire d'ajout de modèle de vélo et saisit les informations du modèle de vélo telles que le nom, le diamètre de roue, le code, etc. 2.Le système affiche la liste des magasins existants. 3.L'admin SAV choisit le magasin auquel il souhaite affecter le nouveau modèle de vélo. 4.L'admin SAV enregistre le nouveau modèle de vélo. 5.Le système enregistre le nouveau modèle de vélo dans le magasin sélectionné et affiche une alerte de succès.
Scénario Alternatif	<p>A1 : le code de ce modèle de vélo existe déjà</p> <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur qui indique que le modèle de vélo existe déjà 2. Retour au point 2.

Tableau 7: Description textuelle de cas d'utilisation « Ajout d'un modèle de vélo »

Description textuelle de cas d'utilisation « Ajout d'un magasin »

Titre	Ajout d'un magasin
Acteur	Admin SAV
Précondition	<ul style="list-style-type: none"> - Disponibilité d'une connexion internet. - L'admin SAV est déjà Authentifié.
Postcondition	Après l'exécution des actions de l'administrateur du SAV sur la gestion des magasins, la liste des magasins est mise à jour dans la base de données du système
Scénario nominal	<ol style="list-style-type: none"> 1.L'admin SAV accède au formulaire d'ajout de magasin. 2.L'admin SAV remplit le formulaire d'ajout d'un magasin en saisissant le code et le nom. 3.L'admin SAV enregistre le nouveau magasin.

E_Bike gestion tracker à travers TCP protocole

	4.Le système enregistre le nouveau magasin et affiche une alerte de succès.
Scénario Alternatif	A1 : le code d'un magasin est codé exactement de trois chiffres. 1. Le système affiche un message indique que le code doit avoir une longueur égale à trois.
Scénario d'échec	E1 : le code ou le nom de ce magasin existe déjà. L'enchâinement E1 démarre au point 1 du scénario nominal. 1 : le système affiche une alerte qui indique que ce code ou se nom de magasin existe déjà.

Tableau 8: Description textuelle de cas d'utilisation « Ajout d'un nouvelle magasin »

3.3 Modélisation conceptuelle

3.3.1 Diagramme de séquence

Le diagramme de séquence suivant représente l'ajout d'un modèle de vélo pour un magasin

E_Bike gestion tracker à travers TCP protocole

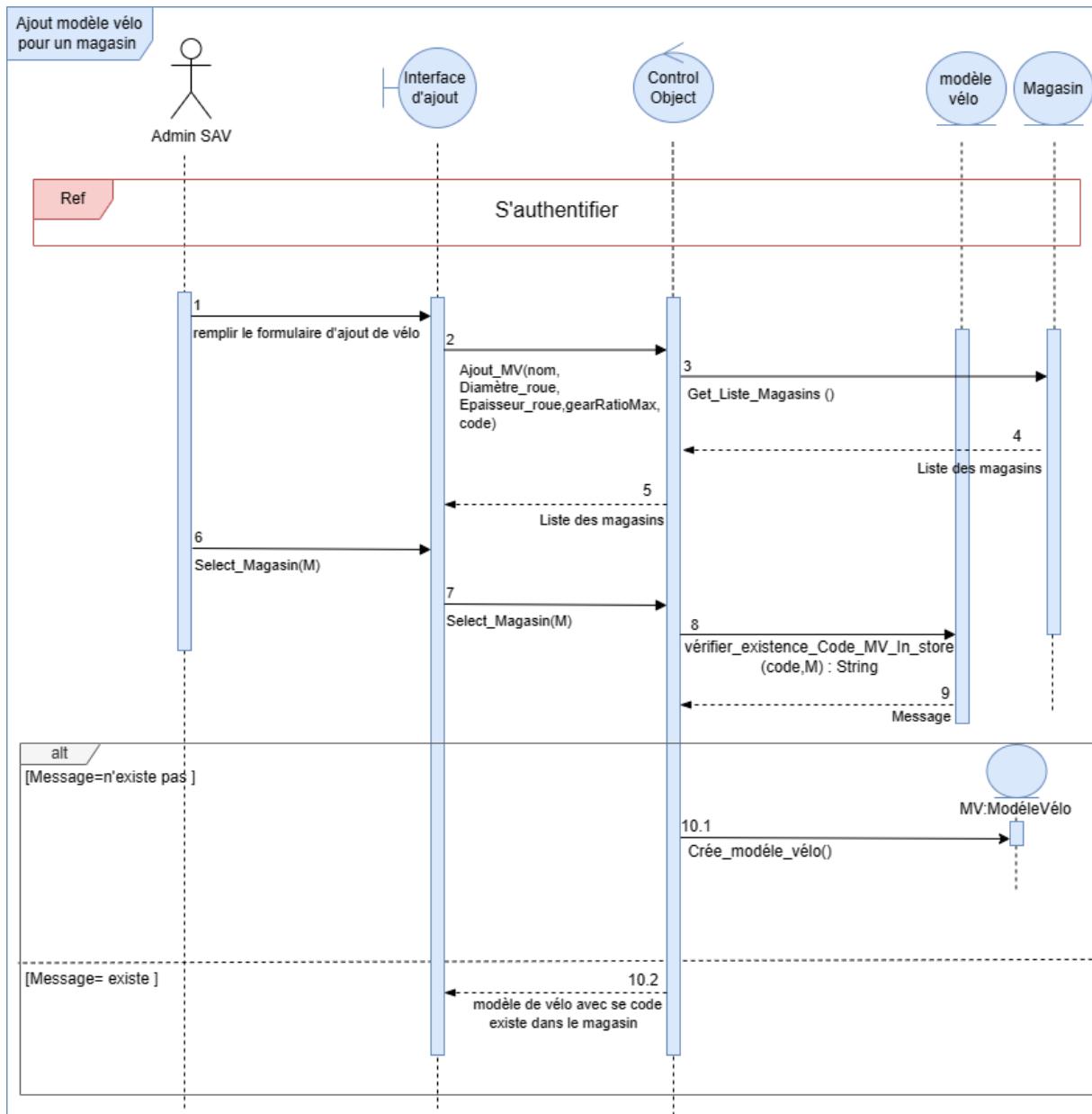


Figure 23: Diagramme de séquence « Ajout modèle vélo »

E_Bike gestion tracker à travers TCP protocole

La figure suivante représente le diagramme de séquence d'ajout d'un nouvel magasin

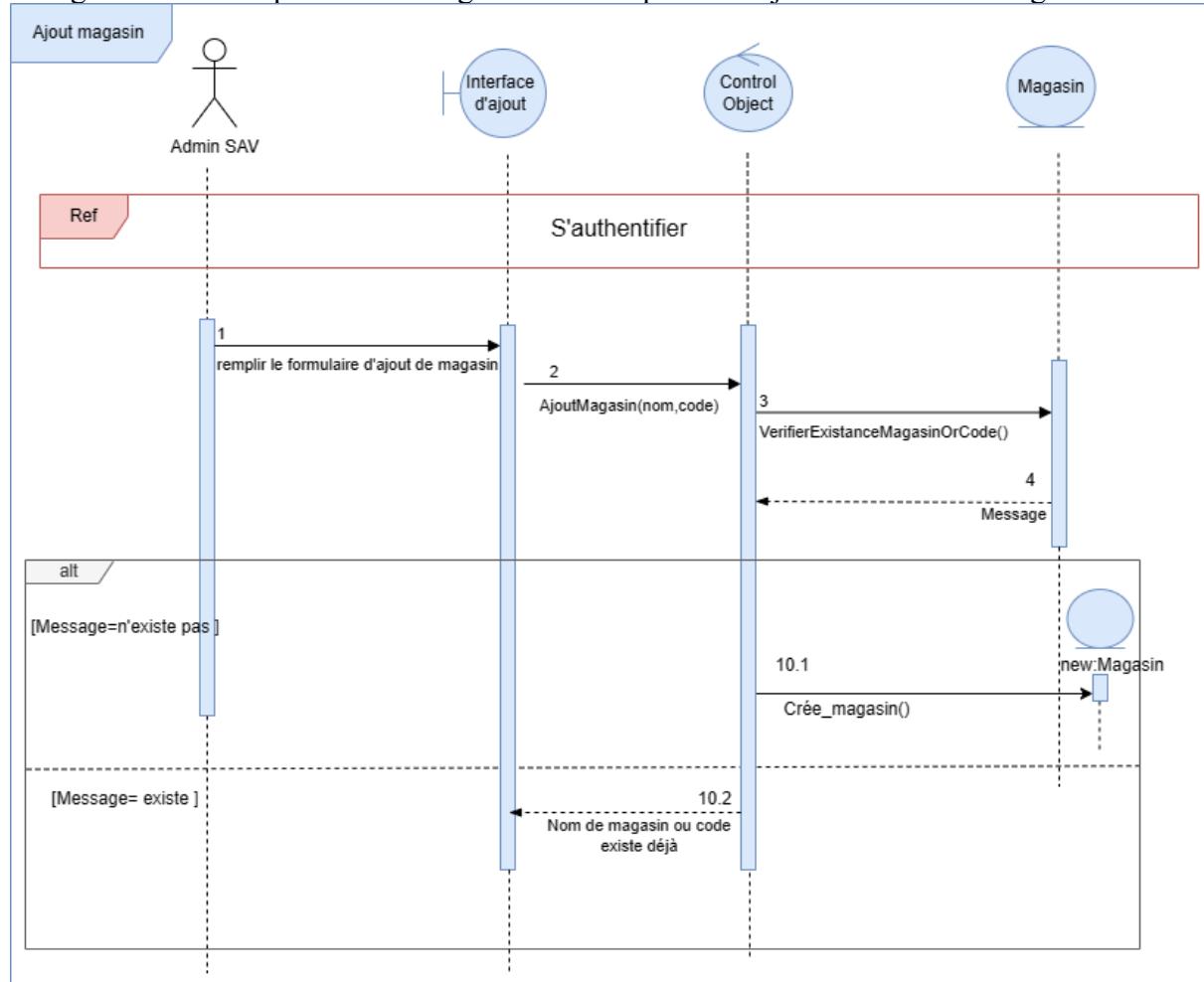


Figure 24: Diagramme de séquence « ajout magasin »

3.3.2 Diagramme de classe :

La figure ci-dessous représente le diagramme de classe de se sprint :

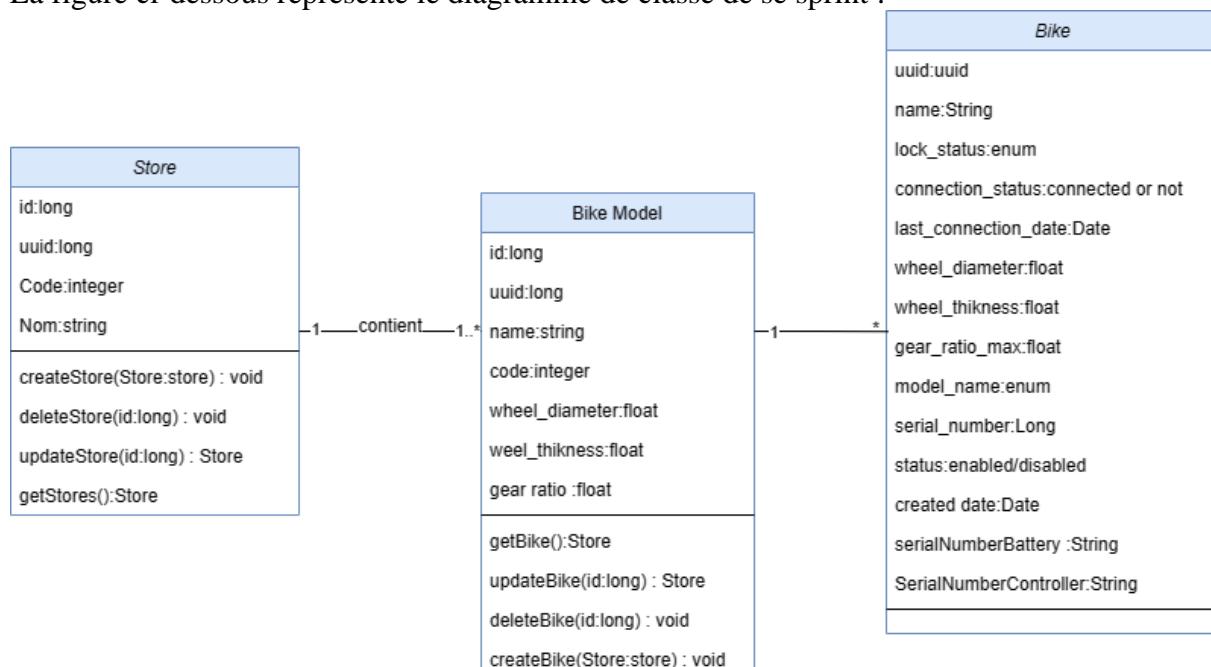


Figure 25: Diagramme de classe module de configuration

3.4 Réalisations :

Dans cette section, nous présentons plusieurs captures d'écran pour illustrer l'accomplissement des différentes tâches du sprint.

- **La page d'affichage de liste des magasins :**

Cette figure représente la liste des magasins, comprenant une pagination, un bouton permettant d'ajouter un nouveau magasin, ainsi que des boutons pour modifier ou supprimer un magasin.

The screenshot shows a web-based application interface for managing stores. On the left, there is a vertical sidebar with icons and labels: Tickets, Diagnostic, FAQs, Configuration (with 'Stores' selected), and Bike Models. The main content area has a header 'Stores' and a sub-header 'Stores'. At the top right is a green button '+ Create a new Store'. Below is a table with the following data:

Code	Name	Actions
504	FLYING_C	
502	ACTIA_AUTO	
505	YMAGINE	
178	SODIUM_CYCLE	
012	EURO_CYCLE	
111	MOBUS	
517	TEST	
011	LIPPERT	
147	BOSH	
753	test4	

At the bottom of the table, there are pagination controls: 'Items per page 10' and '1 - 10 of 20'.

Figure 26: Liste des Magasins

- **Page d'ajout d'un magasin :**

Cette page représente le formulaire d'ajout d'un nouveau magasin. Le code du magasin doit comporter trois chiffres. De plus, le nom du magasin sera enregistré en majuscules, et tout espace sera remplacé par un tiret bas.

E_Bike gestion tracker à travers TCP protocole

The screenshot shows a web-based application interface for managing bike stores. On the left, a sidebar menu includes 'Tickets', 'Diagnostic', 'FAQs', 'Configuration' (with 'Stores' selected), and 'Bike Models'. The main content area is titled 'Stores' and displays a list of existing stores with columns for 'Code' (e.g., 504, 502, 505, 178, 012, 111, 517, 011, 147, 753) and 'Name' (e.g., LIPPERT, BOSH, test4). A modal dialog box titled 'Add New Store' is open in the center, prompting for 'Code' (set to 12) and 'Name' (left empty). Below the modal, a message says 'Code must be 3 digits.' A 'Save' button is visible in the bottom right of the dialog. At the bottom of the page, there are pagination controls ('Items per page: 10', '1 - 10 Of 20') and a note 'Created by ACTIA ES 2024'.

Figure 27: Formulaire d'ajout d'un magasin

- **La page d'affichage de liste des modèles de vélo :**

Cette figure illustre la liste des modèles de vélos, accompagnés des magasins auxquels ils appartiennent. Cette liste comprend une pagination, un bouton d'ajout d'un nouveau modèle de vélo pour un magasin spécifique, ainsi que des options de modification ou de suppression pour chaque modèle de vélo.

The screenshot shows a list of bike models with columns for 'Name' (e.g., City Bike, MTB, test23, Folding), 'wheelDiameter' (e.g., 70, 40, 10, 50), 'gearRatioMax' (e.g., 8, 15, 10, 8), 'wheelThickness' (e.g., 0, 10, 10, 0), 'code' (e.g., 107, 74, 947, 30), 'Store' (e.g., ACTIA_AUTO, LIPPERT, XUBAKA, XUBAKA), and 'Actions' (represented by icons for edit and delete). A 'Create a new Bike Models' button is located at the top right of the list. At the bottom, there are pagination controls ('Items per page: 10', '1 - 10 Of 40') and a note 'Created by ACTIA ES 2024'.

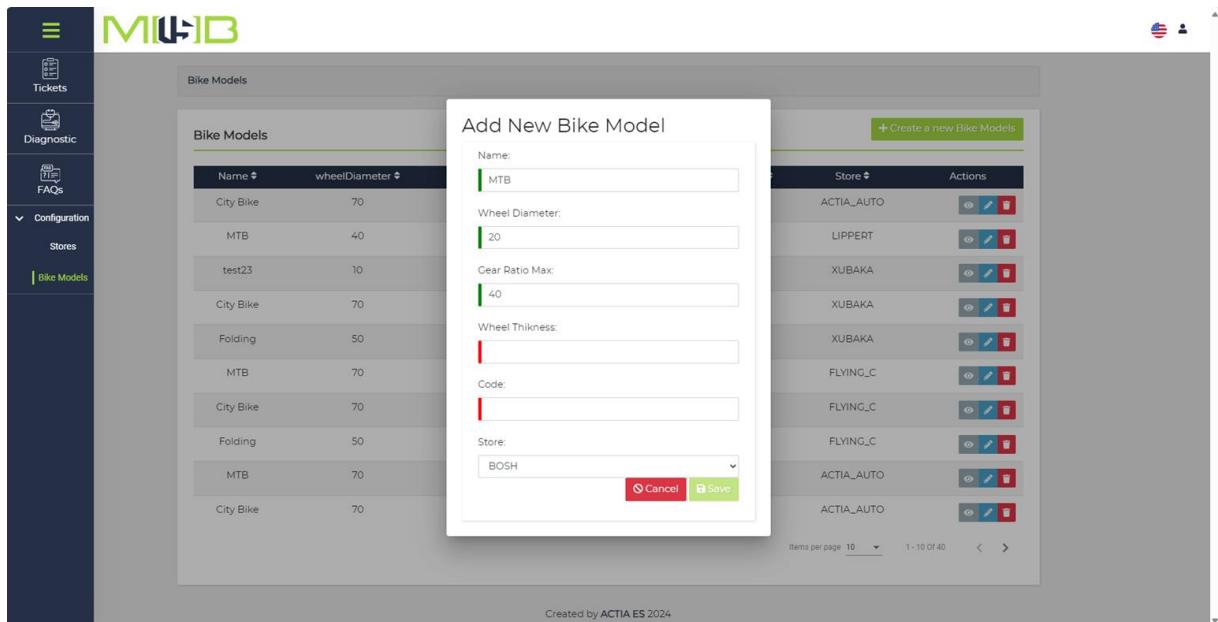
Figure 28: Liste des modèles de vélo

- **Formulaire d'ajout d'un nouveau modèle de vélo pour un magasin :**

La figure suivante présente le formulaire d'ajout d'un nouveau modèle de vélo pour un

E_Bike gestion tracker à travers TCP protocole

magasin spécifique.



The screenshot shows a web-based management system for bicycle models. On the left, a sidebar menu includes options like Tickets, Diagnostic, Configuration, Stores, and Bike Models. The main area displays a table of 'Bike Models' with columns for Name and wheelDiameter. A modal window titled 'Add New Bike Model' is open, prompting for details: Name (MTB), Wheel Diameter (20), Gear Ratio Max (40), Wheel Thickness (empty), Code (empty), and Store (BOSH). Below the table, a list of stores is shown with actions (Edit, Delete) for each: ACTIA_AUTO, LIPPERT, XUBAKA, FLYING_C, and ACTIA_AUTO. The bottom right corner indicates 'Created by ACTIA ES 2024'.

Figure 29: Formulaire d'ajout d'un modèle de vélo

3.5 Test

Dans cette section, nous avons entrepris de tester l'ajout d'un nouveau magasin. Il est crucial que les champs code et nom soient uniques pour chaque magasin ajouté.

Cas de test	Ajout d'un nouveau magasin
Acteur	Admin SAV
Prérequis	Authentifier
Scénario	<ol style="list-style-type: none">Le système affiche le formulaire d'ajout d'un magasin.Saisir le code et le nom de magasin.Enregistrer le nouveau magasin.
Résultat	Un message d'erreur s'affiche indiquant que ce code ou ce nom existe déjà

Tableau 9: Cas de test « ajout d'un nouveau magasin »

Pour faire le test de ce cas de test nous avons utilisée l'outil Postman, cette capture montre l'ajout d'un magasin dont le code ou le nom existe déjà

E_Bike gestion tracker à travers TCP protocole

The screenshot shows a Postman interface with the following details:

- Request URL:** `localhost:9996/api/stores/create`
- Method:** POST
- Body:** JSON (selected)


```

1 {
2 ...
3     "uuid": "e5cf14ec-fb9a-11ed-be56-0242ac120002",
4     "code": "111",
5     "name": "MOBUS"
6 }
```
- Response Status:** 400 Bad Request
- Response Body:**

```

1 {
2     "entityName": "diagStore",
3     "errorKey": "idexists",
4     "type": "https://www.jhipster.tech/problem/problem-with-message",
5     "title": "A new store cannot have the same name and code",
6     "status": 400,
7     "message": "error.idexists",
8     "params": "diagStore"
9 }
```

Figure 30: Ajout d'un store

Nous faisons un test d'ajout d'un modèle de vélo dans un magasin. Il est important de noter que l'ajout d'un nouveau modèle de vélo dans le même magasin nécessite que ce modèle possède un code et un nom uniques.

Cas de test	Ajout d'un modèle de vélo dans un magasin
Acteur	Admin SAV
Prérequis	Authentifier
Scénario	1.Le système affiche le formulaire d'ajout d'un modèle de vélo pour un magasin. 2.L'admin SAV saisit le nom, le code et d'autres informations relatives au modèle de vélo, ainsi que la sélection d'un magasin. 3.Enregistrer le nouveau modèle de vélo pour le magasin sélectionné.
Résultat	Un message d'erreur s'affiche indiquant que ce nom de modèle de vélo avec ce code existe déjà dans ce magasin.

Tableau 10: Cas de test "ajout d'un modèle de vélo"

E_Bike gestion tracker à travers TCP protocole

Cette figure représente le résultat avec Postman

The screenshot shows a POST request to `localhost:9996/api/bike-models`. The request body is a JSON object:

```
1 {  
2   ...  
3   "uuid": "68f8560c-8553-11ee-b9d1-0242ac120002",  
4   "name": "City Bike",  
5   "wheelDiameter": 70.0,  
6   "gearRatioMax": 8.0,  
7   "wheelThickness": 0.0,  
8   "code": "10",  
9   "storeId": 9  
10 }
```

The response status is 400 Bad Request, with the following JSON error message:

```
1 {  
2   "entityName": "diagBikeModel",  
3   "errorKey": "idexists",  
4   "type": "https://www.jhipster.tech/problem/problem-with-message",  
5   "title": "a bike model with the same name and code exist in the store",  
6   "status": 400,  
7   "message": "error.idexists",  
8   "params": "diagBikeModel"  
9 }
```

Figure 31: Cas de test d'ajout d'un modèle de vélo pour un magasin

3.6 Revue du sprint 1

La revue de sprint est une réunion organisée à la fin de chaque sprint, au cours de laquelle l'équipe de développement et les parties prenantes, telles que le Product Owner et le Scrum Master, se réunissent pour passer en revue le travail accompli au cours du sprint. Cette réunion offre l'occasion de présenter les fonctionnalités développées, de recueillir des commentaires et de s'assurer de l'alignement avec la vision et les exigences du produit.

Dans notre réunion de restitution, après avoir testé l'application, le Product Owner a confirmé les fonctionnalités de ce sprint.

3.7 Conclusion

Dans ce chapitre, nous avons mené à bien notre sprint 1, qui met en avant le modèle de configuration. Dans le prochain chapitre, notre attention sera portée sur le module de communication avec le tracker.

Chapitre 4 : Sprint 2 : Communication avec le Tracker SC350MG

Introduction :

Dans ce deuxième sprint, en collaboration avec le Product Owner, nous avons pris la décision de travailler sur le module de communication avec le tracker SC350MG. Cette étape s'articule autour de plusieurs phases, allant de la présentation du sprint backlog à la recherche d'informations sur le tracker, en passant par une étude comparative des données du tracker par rapport au mobile. Ensuite, nous aborderons la partie conception, réalisation et test, afin d'assurer une mise en œuvre efficace et réussie du tracker dans notre application.

4.1 Backlog du sprint 2 :

Le tableau ci-dessous représente le sprint backlog de ce sprint :

ID	User Story	ID	Tache
3.1	En tant que développeur je dois effectuer une étude sur le tracker.	3.1.1	Lire le manuel de Queclink pour extraire les messages, les informations envoyées par ces messages et déterminer le type de donnée associé à chaque information.
		3.1.2	Connaitre comment faire une communication avec le Tracker.
3.2	En tant que développeur je dois faire une communication avec le tracker.	3.2.1	Chercher ou développer un serveur TCP de simulation pour simuler les messages du tracker.
		3.2.2	Développer le client TCP en Spring Boot pour établir la connexion avec le serveur TCP et recevoir les messages en temps réel.
3.3	En tant que cycliste, je souhaite consulter la liste de mes trajets.	3.3.1	Faire une analyse comparative entre les données du trajet envoyées par le mobile et les données envoyées par le Tracker.
		3.3.2	Effectuer la rétro-ingénierie du microservice Dashboard et du Gateway, car nous allons travailler sur un microservice existant.

		3.3.3	Développer un service pour extraire les données de trajet par le traitement des messages envoyés par le tracker.
		3.3.4	Faire le calcul de la durée de trajet, le nombre de KLM parcourue ainsi que la vitesse moyenne d'un trajet.
3.4	En tant que cycliste je souhaite consulter les détails de mon trajet.	3.4.1	Faire une analyse comparative entre les données des points du trajet envoyé par le mobile et les données envoyées par le Tracker.
		3.4.2	Développer un service pour traiter les points de trajet par le traitement des messages envoyés par le tracker.

Tableau 11: Backlog Sprint 2

4.2 Recherche sur Tracker SC350MG

4.2.1 Architecture du Tracker :

Le dispositif comprend un serveur Tracker qui joue un rôle central dans la communication avec plusieurs terminaux via Internet. Voici les caractéristiques clés de ce serveur :

- Accès à Internet :** Le serveur Tracker doit être connecté à Internet pour permettre la communication avec d'autres appareils via Internet. Il doit disposer d'une connectivité réseau stable pour assurer des échanges de données fiables.
- Prise en charge des connexions TCP :** Le serveur Tracker est capable de gérer à la fois les connexions TCP (Transmission Control Protocol) et UDP (User Datagram Protocol). TCP offre une livraison fiable, ordonnée et vérifiée des données, tandis que UDP est un protocole sans connexion qui offre une communication plus légère mais sans garantie de livraison ou d'ordre des paquets. Le serveur doit être capable de basculer entre ces protocoles en fonction des besoins des terminaux.
- Réception et Envoi de Données :** Une fois qu'une connexion est établie entre un terminal (Tracker) et le serveur Tracker, le serveur doit être capable de recevoir les données envoyées par le terminal (Tracker). Il doit ensuite traiter ces données en conséquence, effectuer les opérations nécessaires et renvoyer les réponses appropriées au Tracker. Cette communication bidirectionnelle permet un échange d'informations

fluide entre le serveur et les terminaux.

4.2.2 Les types et format de messages :

Dans cette partie, nous allons nous intéresser aux messages échangés entre le tracker et le serveur Tracker :

- Les messages échangés entre le tracker et le serveur Tracker suivent un format défini par le protocole de l'interface Air Track. Chaque message, qu'il s'agisse d'une commande, d'un accusé de réception ou d'un rapport, est composé de caractères ASCII.
- Le tracker envoie des messages contenant des commandes au serveur Tracker pour effectuer diverses actions telles que la configuration, l'interrogation ou le contrôle des paramètres du tracker. En réponse à ces commandes, le serveur Tracker envoie un accusé de réception approprié.
- Selon la configuration des paramètres, le tracker peut également envoyer des rapports au serveur Tracker. Ces rapports contiennent des informations pertinentes telles que la position, la vitesse ou d'autres données spécifiques du tracker. Le serveur Tracker peut ainsi recevoir ces rapports.
- La communication entre le tracker et le serveur Tracker est bidirectionnelle, permettant un échange continu d'informations et une interaction dynamique entre les deux composants du système. Ce flux de messages joue un rôle crucial dans le fonctionnement et la gestion efficace du tracker dans divers scénarios d'utilisation.

La figure suivante résume l'échange des messages entre le backend server et le Tracker :

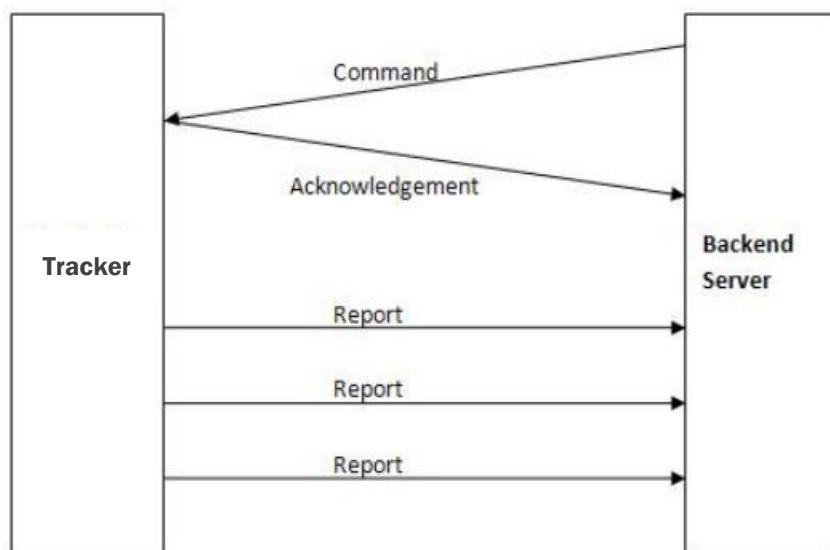


Figure 32: Types de messages envoyés par le tracker

E_Bike gestion tracker à travers TCP protocole

Dans notre cas, nous allons nous concentrer sur les messages de rapport qui suivent le format "+RESP : XXXX, param1, param2,...,paramN".

- Chaque "XXXX" représente un format de message spécifique et chaque format envoie des paramètres spécifiques. Il est possible que plusieurs formats de message envoient des paramètres communs à une échelle temporelle différente.
- Les messages de rapport, identifiés par le préfixe "+RESP", sont utilisés par le tracker pour transmettre des informations au serveur Tracker. Chaque rapport est structuré selon un format spécifique déterminé par le code "XXXX", qui différencie les différents types de rapports.
- Les paramètres "param1", "param2", ..., "paramN" contiennent les données spécifiques incluses dans le rapport. Ces paramètres peuvent varier en fonction du type de rapport et peuvent inclure des informations telles que la longitude, la latitude, la vitesse etc.

Il est important de noter que certains formats de rapport peuvent partager des paramètres communs pendant une période de temps spécifique, tandis que d'autres peuvent avoir des paramètres uniques. Par exemple, un rapport de position peut inclure des paramètres tels que la latitude et la longitude, qui peuvent également être inclus dans d'autres types de rapports.

4.2.3 Communication entre le serveur Tracker et le client web :

Dans l'architecture de notre système, le client TCP, intégré à l'écosystème Spring Boot, joue un rôle essentiel dans la réception des données en provenance du serveur Tracker.

Lorsqu'un rapport est émis par le serveur Tracker, le client TCP le reçoit en temps réel, assurant ainsi une communication fluide et instantanée entre les deux composants.

Le client TCP, agissant en tant qu'interface entre le serveur Tracker et notre application, reste constamment à l'écoute des trames de données entrantes, prêt à les traiter dès leur réception.

Une fois les données reçues, le client TCP procède à leur analyse et à leur traitement selon les spécifications de notre application à travers des service de traitement. Cela peut inclure l'extraction d'informations pertinentes, la transformation des données, ou encore leur enregistrement dans une base de données.

Grâce à cette architecture robuste et réactive, notre système est capable de gérer efficacement les flux de données en temps réel, garantissant ainsi une expérience utilisateur optimale et une prise de décision rapide et éclairée.

4.2.4 Etude comparatif

Dans cette partie, nous allons analyser et comparer les données envoyées par le tracker avec celles envoyées par le mobile, en utilisant la même table de base de données existante. Cette approche permettra de dégager des insights précieux sur la cohérence des données collectées. Cette table représente l'étude comparatif des données sur le trip dont les croix de couleur rouge signifient un champ obligatoire.

Données	Trip (mobile)	Trip (tracker)
Trip uuid	✓	✓
Nom du trajet	✓	Constante
Description du trajet	✓	Constante
Date de début	✓	✓
Date de fin	✓	✓
Niveau d'assistance moyen	✓	(A calculer)
Vitesse moyenne	✓	(A calculer)
Durée (en secondes)	✓	(À calculer)
Total des calories	✓	(A calculer)
Consommation de la batterie	✓	(A calculer)
Distance totale du trajet	✓	(À calculer)
Statut (activé ou désactivé)	✓	✓
Odomètre	✓	✓
Capacité de batterie restante	✓	✓

Tableau 12: Tableau de comparaison des données du trajet entre tracker et le mobile

Ce tableau représente la différence des données envoyées par le mobile et le tracker pour les points de trajet

Données	Trip point (mobile)	Trip point(tracker)
- type (départ / arrivée / suivi /diagnostic)	✓	✓
- date (date/heure du point)	✓	✓
Longitude	✓	✓
Latitude	✓	✓
Altitude	✓	✓
Azimut		✓

E_Bike gestion tracker à travers TCP protocole

Précision GPS	✓	✓
Calories	✓	(A calculer)
Fréquence cardiaque	✓	
Vent	✓	
Température	✓	
Assistance	✗	✓
Vitesse	✗	✓
Niveau de charge de la batterie	✗	✓
Capacité restante	✓	✓

Tableau 13: Tableau de comparaison des données point trajet entre tracker et le mobile

4.2.5 Simulation de serveur TCP :

Puisqu'il n'y a qu'un seul tracker sur lequel l'équipe embarquée travaille au siège de Tunis, et que le tracker est équipé d'une carte SIM, chaque test consomme des données mobiles, entraînant ainsi des coûts supplémentaires. Pour résoudre ce problème, nous avons choisi d'utiliser une solution de simulation.

Le simulateur joue le rôle de serveur Tracker dans le cas réel. Il reproduit le comportement du tracker et la communication avec notre application sans utiliser de données mobiles réelles. Cela nous permet de tester toutes les fonctionnalités de manière approfondie, sans encourir les coûts associés à la consommation de données mobiles. En utilisant un simulateur, nous pouvons contrôler et surveiller le flux de données, simuler différents scénarios et conditions, et effectuer des tests de manière répétée et exhaustive.

E_Bike gestion tracker à travers TCP protocole

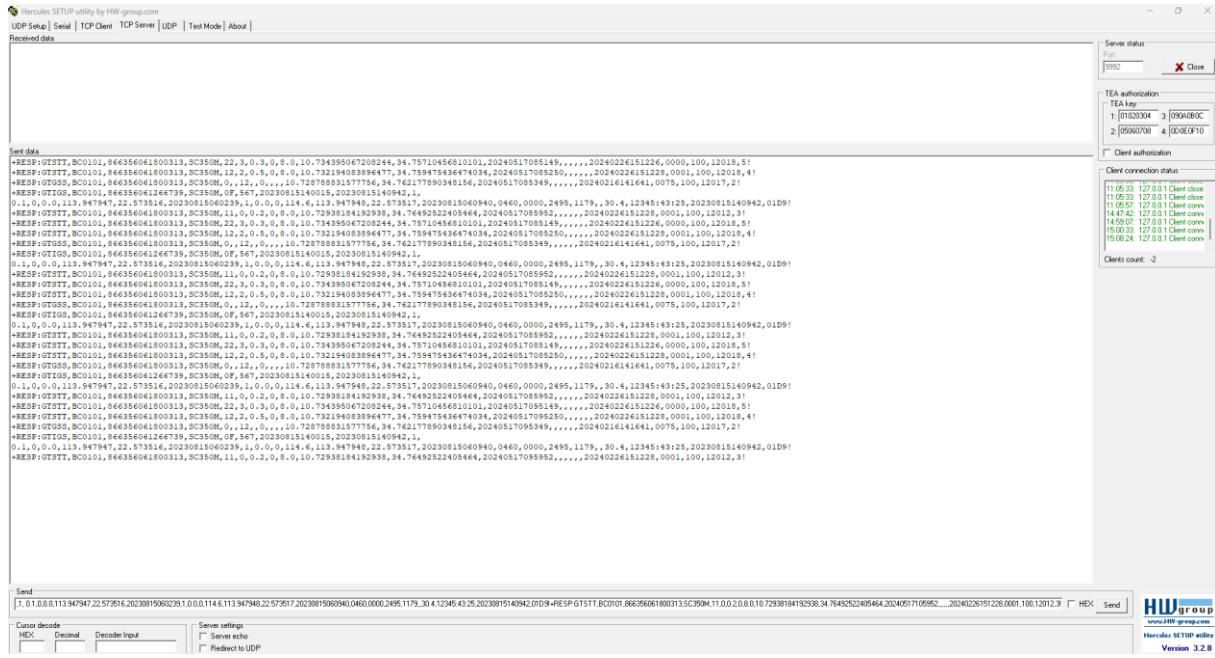


Figure 33: Logiciel de simulation

4.3 Analyse des besoins

4.3.1 Identification des acteurs

Dans ce sprint, le cycliste est l'acteur principal. Le tableau suivant identifie ses rôles.

Acteur	Rôle
Cycliste	<ul style="list-style-type: none"> Scan le QR code avec l'application mobile pour se connectée à son vélo. Faire le suivie des informations de ses vélos Consulter les informations de ses trajets Faire des réclamations en cas de besoin Consulter les statistiques de la vitesse moyenne, la durée, la distance de ses trajet effectuée par un vélo ou par tous ses vélos.

Tableau 14: Les acteurs de sprint 2

4.3.2 Diagramme de cas d'utilisation de sprint 2

La figure ci-dessous représente le diagramme de cas d'utilisation « Trajet et point de trajet »

E_Bike gestion tracker à travers TCP protocole

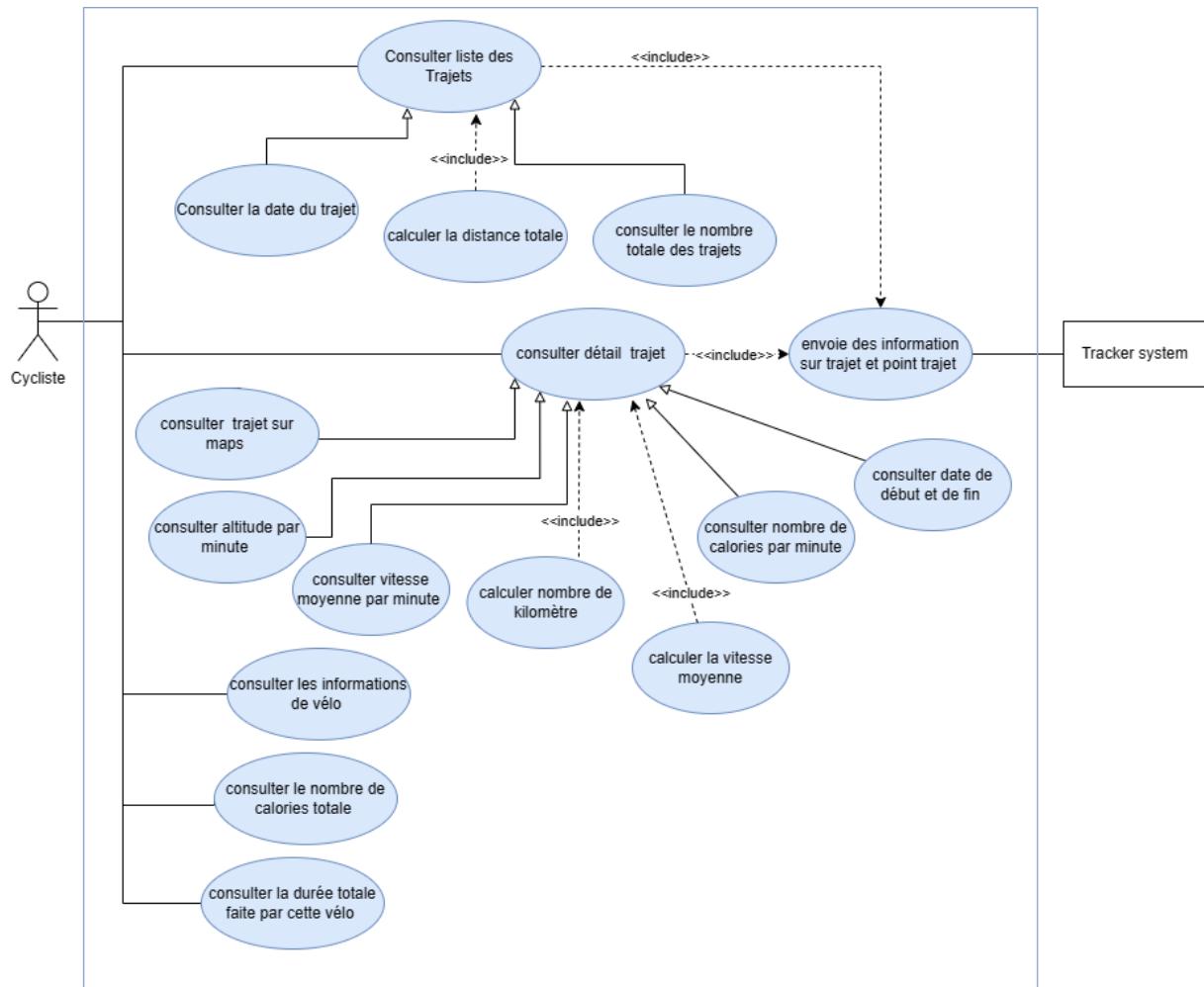


Figure 34: Use case trajet et point de trajet

4.3.3 Description textuelle des cas d'utilisation

La description textuelle suivante montre le cas d'utilisation « consulter détaille trajet »

Titre	Consulter détaille trajet
Acteur	Cycliste
Précondition	<ul style="list-style-type: none"> - Disponibilité d'une connexion internet. - déjà connecté au moins une fois au vélo. - le Tracker envoie des messages sur la position
Postcondition	Voir toutes les informations sur son trajet.
Scénario nominal	<ol style="list-style-type: none"> 1.le cycliste sélectionne un trajet. 2.le système calcule le nombre de kilomètres, la vitesse moyenne, la durée du trajet ainsi le système envoie l'altitude et la vitesse pour chaque minute et aussi la date de début et de fin du trajet. 3. le cycliste consulte les informations de détail du trajet telles que le nombre de kilomètres totale, la date de début et de fin de trajet,

E_Bike gestion tracker à travers TCP protocole

	la durée totale, la vitesse moyenne ainsi que le graphe de variation d'altitude et de vitesse tout au long du trajet 4.consulte le trajet sur une carte géographique.
Scénario d'échec	E1 : le tracker n'envoie pas des données sur le trajet. L'enchaînement E1 démarre au point 1 du scénario nominal. 1 : le système n'affiche pas ce trajet.

Tableau 15 : Description textuelle « Consulter détaille trajet »

4.4 Modélisation conceptuelle

4.4.1 Diagramme de séquence

Le diagramme de séquence suivant représente l'interaction entre le cycliste et le système pour le cas d'utilisation « consulter détail trajet ».

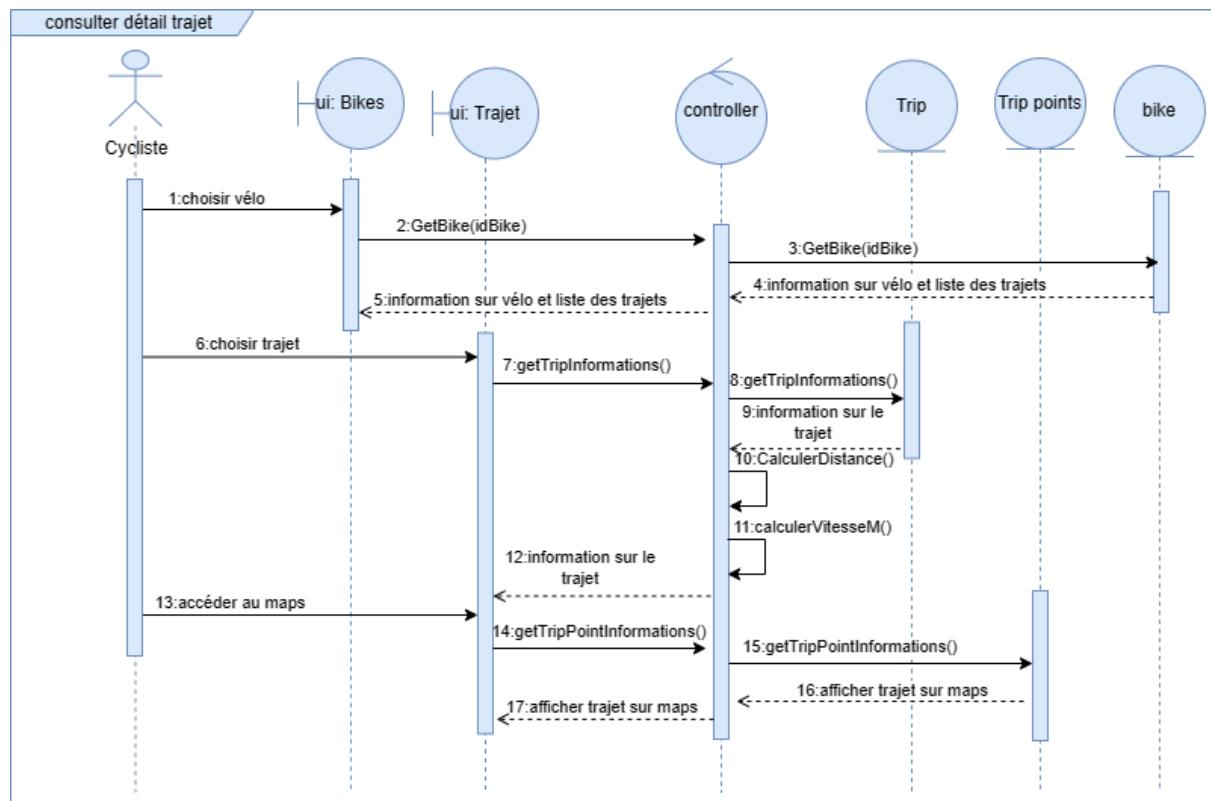


Figure 35 : Diagramme de séquence « consulter informations trajet »

Le diagramme de séquence suivant illustre l'interaction entre le Tracker, le serveur TCP, le client TCP et le système backend pour le cas d'utilisation « Envoi des informations sur le trajet et les points du trajet ».

E_Bike gestion tracker à travers TCP protocole

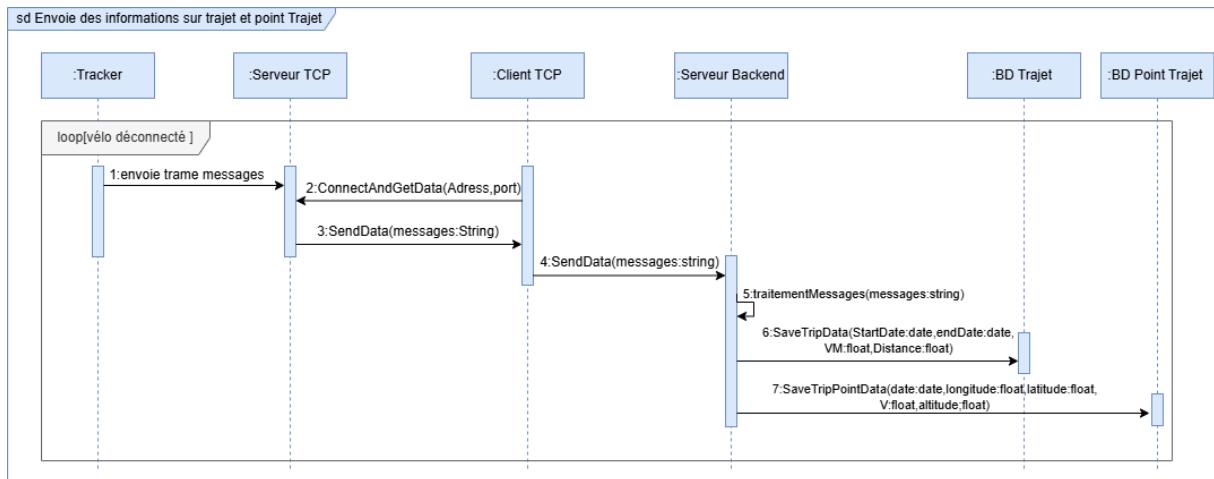


Figure 36 : Diagramme de séquence « Envoi des informations sur trajet et point Trajet »

4.4.2 Diagramme de classe

La figure suivante présente le diagramme de classe de ce sprint

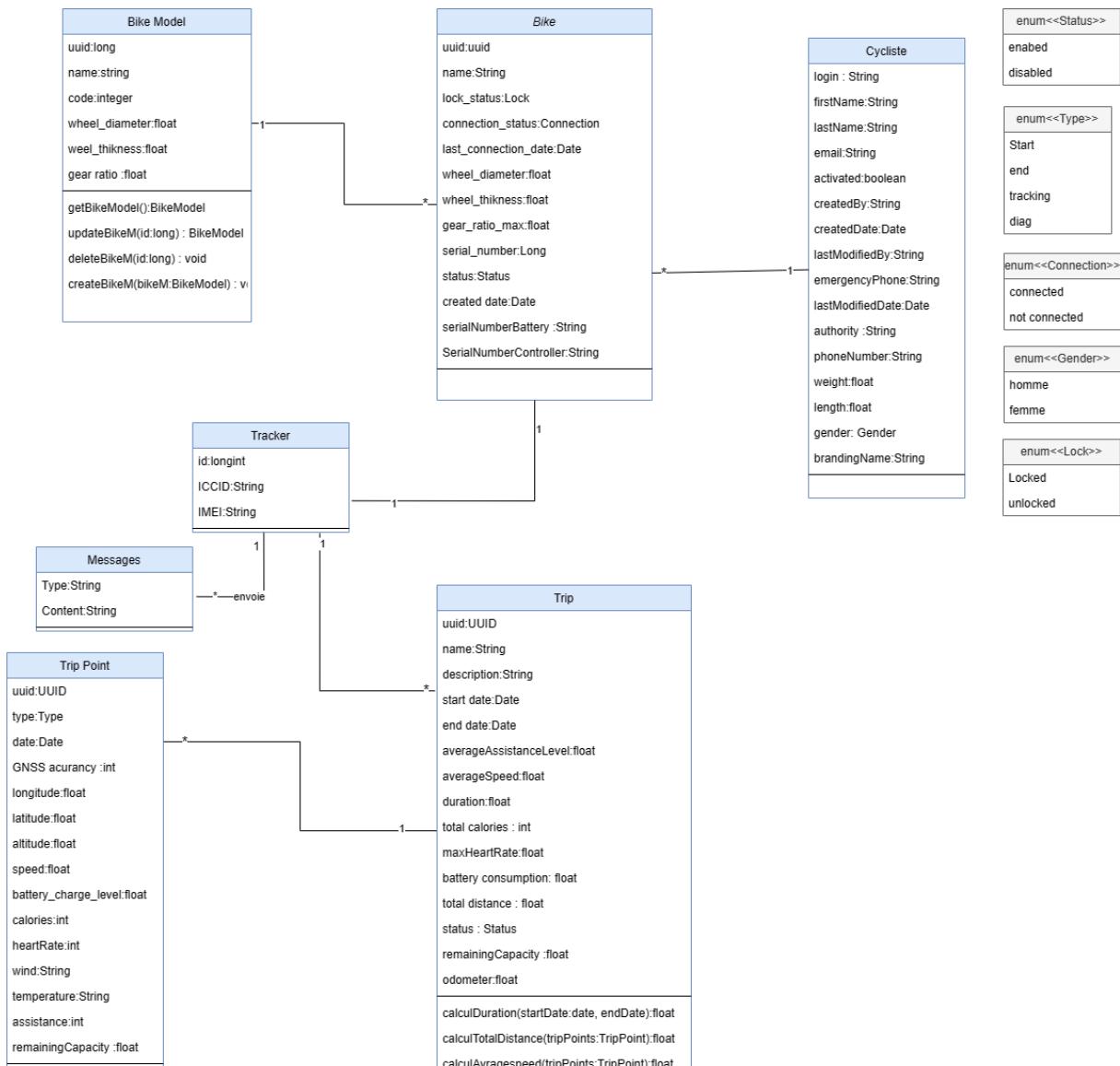


Figure 37: Diagramme de classe

4.5 Réalisation

- Liste des trajets :**

La figure suivante représente la liste des trajets effectués par le vélo. On trouve les informations telle que la date du trajet, la durée totale, la vitesse moyenne ainsi que la distance parcourue.

Parcours					
Nom	Date	Durée totale	Vitesse	Distance	
Trip using tracker1	27-avr.-2024	00:03:03	29.38Km/h	1.49Km	👁
Trip using tracker1	27-avr.-2024	00:03:03	29.38Km/h	1.49Km	👁
Trip using tracker1	27-avr.-2024	00:03:00	28.49Km/h	1.42Km	👁
Trip using tracker1	27-avr.-2024	00:03:00	28.49Km/h	1.42Km	👁

Affichage 1 - 4 de 19 items.
 «« « **1** 2 3 4 5 » »»

Figure 38: Liste te information sur les trajets

- Détaille d'un trajet :**

Les deux figures suivantes montrent les détails d'un trajet, y compris la distance parcourue, la durée, le nombre de calories brûlées, la variation d'altitude et la vitesse durant le trajet, ainsi que la représentation de ce trajet sur une carte géographique.

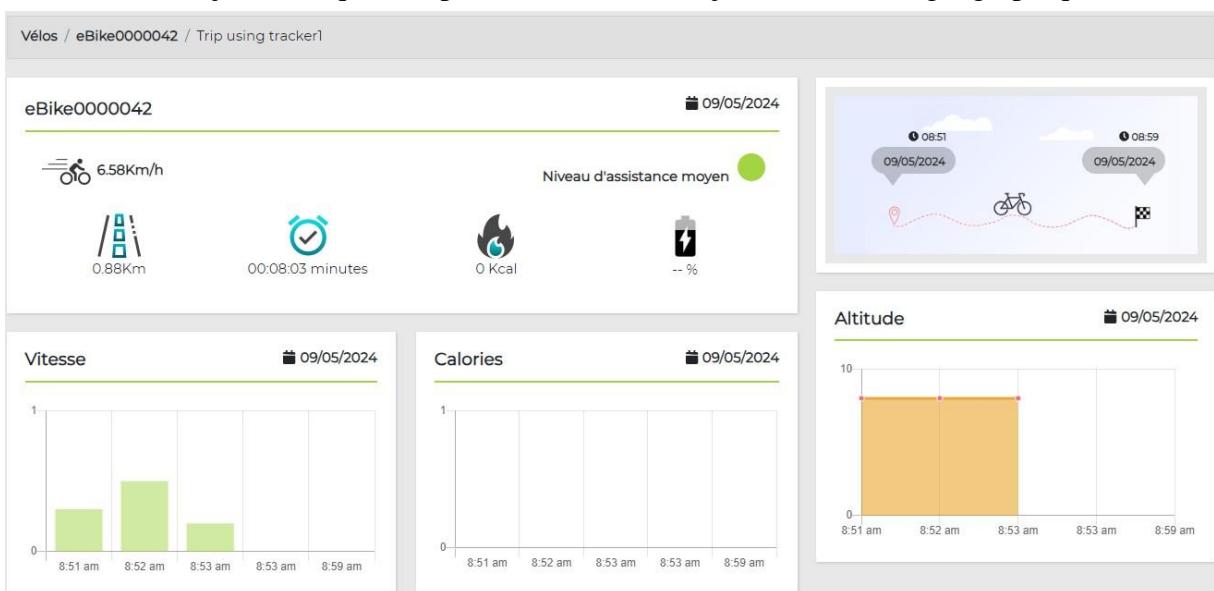


Figure 39: Détail d'un trajet

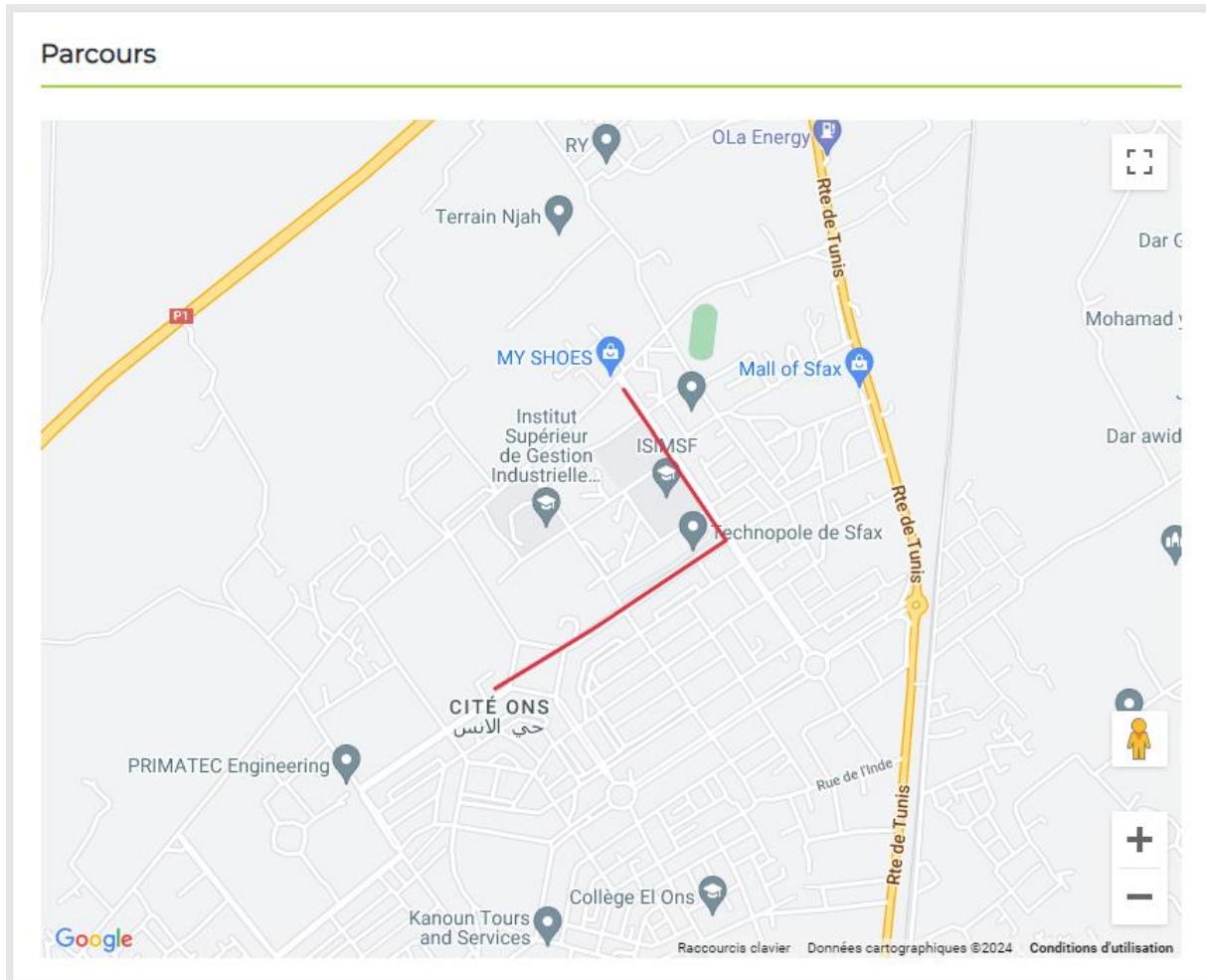


Figure 40: Représentation d'un trajet dans la carte géographique

4.6 Test

Dans cette section, nous avons décidé de tester la connexion avec le serveur simulateur TCP ainsi que l'enregistrement des Trip et des trip points à travers les messages de simulation envoyé par le tracker

- Connexion avec le serveur TCP et enregistrement des données de Trajet et points de trajet
- Pour s'assurer de la connexion du client au serveur TCP, le logiciel de simulation affiche l'heure de connexion ou de déconnexion d'un client. Le carré rouge dans la figure suivante représente un statut où le client se déconnecte, tandis que le carré bleu présente l'horaire de connexion d'un client TCP.

E_Bike gestion tracker à travers TCP protocole

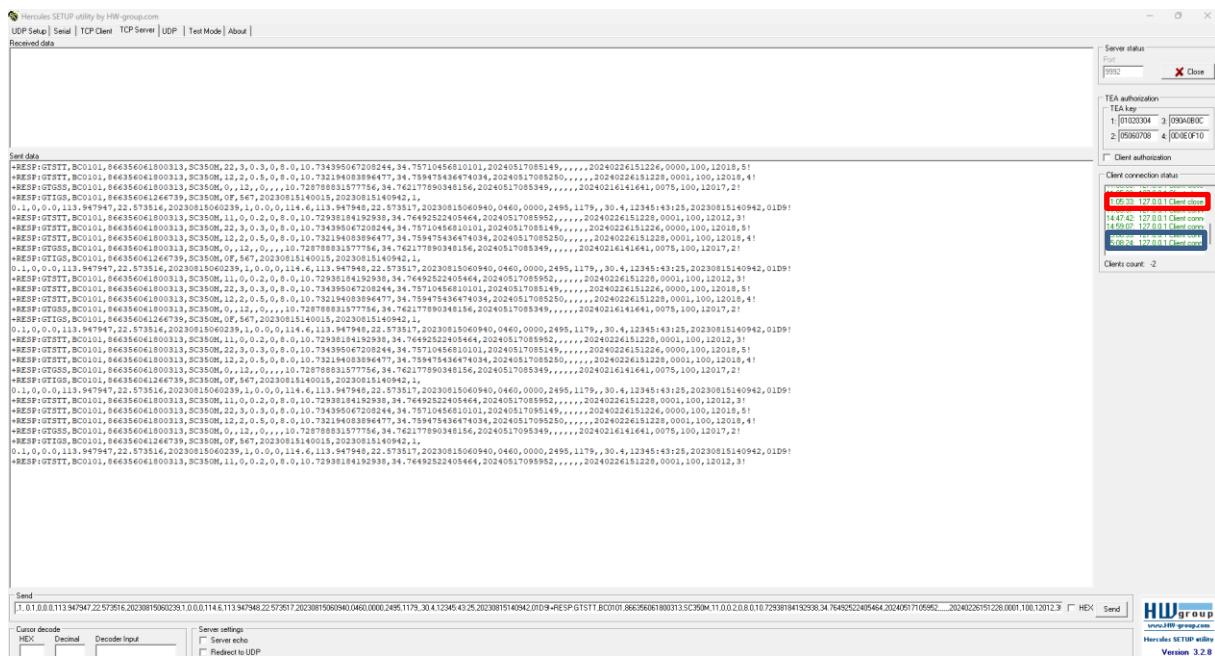


Figure 41: Connexion et déconnexion de client TCP

Nous effectuons un test d'envoi des messages simulés par le tracker et nous nous assurons de l'enregistrement des données de trajet et des points de trajet.

Cas de test	Traitement des messages et enregistrement des données d'un trajet et point de trajet
Acteur	Tracker, Serveur TCP, Système Backend
Prérequis	Le tracker est configuré avec une carte SIM et est connecté au serveur TCP. Le système backend est en fonctionnement et prêt à recevoir les données.
Scénario	<ol style="list-style-type: none"> Le tracker envoie un message de démarrage de trajet au serveur TCP. Le serveur TCP reçoit le message et le transmet au client TCP. Le système backend enregistre l'heure de début du trajet. Pendant le trajet, le tracker envoie périodiquement des messages contenant la vitesse, l'altitude et d'autres données. Le serveur TCP reçoit ces messages et les transmet au client TCP. Le système backend enregistre les données de chaque point du trajet en temps réel. À la fin du trajet, le tracker envoie un message de fin de trajet.

E_Bike gestion tracker à travers TCP protocole

	<p>8. Le serveur TCP reçoit le message et le transmet au système backend.</p> <p>9. Le système backend enregistre l'heure de fin du trajet et calcule les la distance totale, vitesse moyenne, etc.</p>
Résultat	Le système backend enregistre avec succès toutes les données du trajet, y compris les points intermédiaires, et calcule les statistiques finales du trajet.

Ces figures illustrent les résultats obtenus concernant les données de trajet et les points de trajet à l'aide de Postman, suite à l'envoi de trames de messages par le serveur TCP :

```

1 [
2   {
3     "id": 374983,
4     "uid": "449f9ea8-a5e8-4e11-a279-60249f42212b",
5     "name": "Trip using tracker",
6     "description": "no description",
7     "startDate": "2024-05-02T08:51:49+02:00",
8     "endDate": "2024-05-02T08:51:49+02:00",
9     "averageAssistanceLevel": null,
10    "averageSpeed": 10.618489295142448,
11    "duration": 403000.0,
12    "totalCalories": null,
13    "maxHeartRate": null,
14    "batteryConsumption": null,
15    "totalDistance": 1.4246471795981437,
16    "userId": "860b798b-3d94-4cfe-b78e-5eae5bdf59ef",
17    "bikeName": "eBike0000002",
18    "bikeUuid": "be9c52c-6909-4c88-8d65-f06a922ced4a",
19    "tripPoints": [],
20    "odometer": null,
21    "remainingCapacity": null
22  },
23  {
24    "id": 374982,
25    "uid": "56c2284f-fa17-47d9-81e8-bd184f39d8e9",
26    "name": "Trip using tracker",
27    "description": "no description",
28    "startDate": "2024-05-02T08:51:49+02:00",
29    "endDate": "2024-05-02T08:51:49+02:00",
30    "averageAssistanceLevel": null,
31    "averageSpeed": 10.618489295142448,
32    "duration": 403000.0,
33    "totalCalories": null,
34    "maxHeartRate": 0,
35    "batteryConsumption": null,
36    "totalDistance": 1.4246471795981437,
37    "userId": "860b798b-3d94-4cfe-b78e-5eae5bdf59ef",
38    "bikeId": 255360,
39  }
]

```

Figure 42: Test de trip avec Postman

```

1 [
2   {
3     "id": 375012,
4     "uid": "822c28c3-018b-443d-9c88-5c644980dbdd",
5     "type": "TRACING",
6     "date": "2024-05-02T08:59:52+02:00",
7     "longitude": 10.759925109816991,
8     "latitude": 34.84194214848418,
9     "altitude": 0.0,
10    "gspsAccuracy": 0,
11    "calories": 0,
12    "heartRate": 0,
13    "wind": "9",
14    "temperature": "23",
15    "speed": 0.0,
16    "assistance": 1,
17    "batteryChargeLevel": 0.0,
18    "tripId": 374983,
19    "tripUuid": "449f9ea8-a5e8-4e11-a279-60249f42212b",
20    "remainingCapacity": 0.0
21  },
22  {
23    "id": 375011,
24    "uid": "86ccca08-f410-48e3-a3e5-2647f79215d8",
25    "type": "TRACING",
26    "date": "2024-05-02T08:53:49+02:00",
27    "longitude": 10.759905312000234,
28    "latitude": 34.85319320911153,
29    "altitude": null,
30    "gspsAccuracy": 0,
31    "calories": 0,
32    "heartRate": 0,
33    "wind": "9",
34    "temperature": "23",
35    "speed": null,
36    "assistance": 1,
37    "batteryChargeLevel": 0.0,
38    "tripId": 374983,
39  }
]

```

Figure 43: Liste des trips points pour le trajet

4.7 Revue du sprint 2

À la fin du sprint, une réunion de restitution a eu lieu avec le Product Owner et le Scrum Master.

Après avoir testé l'application, ils ont demandé des rectifications, telles que :

- Le calcul de la consommation de batterie durant le trajet.
- Le calcul du nombre de calories brûlées chaque minute.
- Le calcul de l'assistance moyenne.

4.8 Conclusion

Dans ce chapitre, nous avons mené à bien notre sprint 2, en mettant en avant le module de communication avec le tracker. Suite aux modifications demandées par le Product Owner, certaines user stories seront reprises dans la prochaine itération et seront placées comme les plus prioritaires. Dans le prochain chapitre, notre attention se portera sur le module d'intégration des statistiques.

Chapitre 5 : sprint 3 : Intégration des statistiques

Introduction

Dans ce troisième sprint, en collaboration avec le Product Owner, nous avons décidé de travailler sur le module d'intégration des statistiques. Dans ce chapitre, nous commencerons par définir le backlog du produit, suivi de l'analyse des besoins, la conception, les tests et la validation.

5.1 Backlog du sprint 3

Le tableau suivant représente le sprint backlog de ce sprint

ID	User Story	ID	Tache
4.1	En tant que cycliste, je souhaite consulter le détail de mon trajet.	4.1.1	Calculé de la consommation de batterie durant le trajet.
		4.1.2	Communiquée avec le microservice UAA pour prendre les uuid des cyclistes et leur poids.
		4.1.3	Calculé de nombre de calories brûlées chaque minute.
		4.1.4	Calculé le nombre de calories totales brûlées lors du trajet.
		4.1.5	Calculé de l'assistance moyenne.
4.2	En tant que cycliste, je souhaite accéder aux statistiques de mes trajets pour un vélo spécifique, y compris la durée, la vitesse, la distance parcourue et les calories brûlées. Je souhaite pouvoir filtrer ces données par semaine, mois, année ou date personnalisée pour obtenir des informations pertinentes pour ce vélo particulier.	4.2.1	Etudié de la partie statistique de l'existant.
		4.2.2	Traitées les données des trajets pour faire l'enregistrement des données des statistiques dans la table BikeUserStatistique.
		4.2.3	Travaillé sur le filtre de la semaine dernière pour faire la moyenne des vitesses la durée totale et la distance totale des trajets qui ont la même date, user et vélo.
		4.2.4	Travailler sur le filtre de mois dernier pour faire la moyenne des vitesses la durée totale et la distance

E_Bike gestion tracker à travers TCP protocole

			totale des trajets qui ont la même date, user et vélo
	4.2.5		Travailler sur le filtre de l'année dernière pour faire la moyenne des vitesses la durée totale et la distance totale des trajets qui sont effectuée le même mois, par le même user et vélo.
	4.2.6		Travailler sur le filtre de date personnalisée qui permet au cycliste de filtrer la moyenne des vitesses, la durée totale et la distance totale selon la période des dates.
4.3	En tant que cycliste je souhaite accéder à d'autres statistiques générales qui englobent l'ensemble des vélos que je possède, telles que le nombre total de kilomètres parcourus, le temps total passé à vélo et le nombre total de calories brûlées sur l'ensemble de mes trajets.	4.3.1	Etudié de la partie statistique de l'existant.
		4.3.2	Traité les données des trajets pour faire l'enregistrement des données des statistiques dans la table UserStatistique.
		4.3.3	Travailler sur le filtre de la semaine dernier pour faire la moyenne des vitesses la durée totale et la distance totale des trajets qui sont effectuée par la même date et user
		4.3.4	Travailler sur le filtre de mois dernier pour faire la moyenne des vitesses la durée totale et la distance totale des trajets qui ont la même date et user.
		4.3.5	Travailler sur le filtre de l'année dernière pour faire la moyenne des vitesses la durée totale et la distance

		totale des trajets qui sont effectuée le même mois, par le même user.
--	--	---

Tableau 16: Backlog Sprint 3

5.2 Analyse des besoins

5.2.1 Identification des acteurs

Le cycliste est l'acteur principale de ce sprint le tableau suivant identifie ses rôles

Acteur	Rôle
Cycliste	<ul style="list-style-type: none"> • Scan le QR code avec l'application mobile pour se connecter à son vélo. • Faire le suivi des informations de ses vélos • Consulter les informations de ses trajets • Faire des réclamations en cas de besoin • Consulter les statistiques de la vitesse moyenne, la durée, la distance de ses trajet effectuée par un vélo ou par tous ses vélos.

Tableau 17: Identification des acteurs

5.2.2 Diagramme de cas d'utilisation

La figure suivante montre le diagramme de cas d'utilisation de ce sprint

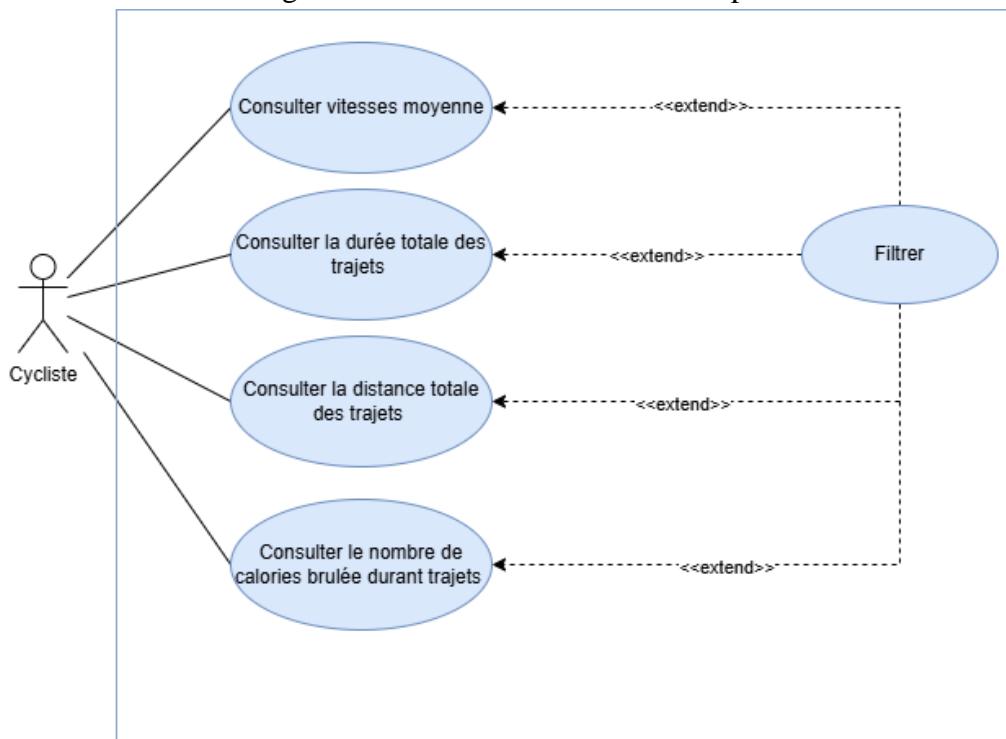


Figure 44: Use case sprint trois

5.2.3 Description textuelle des cas d'utilisation

Le cas d'utilisation suivant montre le cas d'utilisation « consulter vitesses moyenne »

Titre	Consulter vitesses moyenne
Acteur	Cycliste
Précondition	<ul style="list-style-type: none"> - Disponibilité d'une connexion internet. - déjà effectuée au moins un trajet par l'un des vélos.
Postcondition	Voir tous les statistiques de la vitesses moyenne.
Scénario nominal	<ol style="list-style-type: none"> 1. Le cycliste accède à son Dashboard et clique sur le premier filtre "Vitesse" puis sur le deuxième filtre "La semaine dernière". 2. Le système calcule la vitesse moyenne des trajets effectués par tous les vélos du cycliste durant la semaine dernière et affiche les statistiques de la vitesse moyenne des trajets par date. 3. Le cycliste clique sur le deuxième filtre "Le mois dernier". 4. Le système calcule la vitesse moyenne des trajets effectués par tous les vélos du cycliste durant le mois dernier et affiche les statistiques de la vitesse moyenne des trajets par date. 5. Le cycliste clique sur le deuxième filtre "L'année dernière". 6. Le système calcule la vitesse moyenne des trajets effectués par tous les vélos du cycliste durant l'année dernière et affiche les statistiques de la vitesse moyenne des trajets par mois. 7. Le cycliste sélectionne un vélo, clique sur le premier filtre "Vitesse" puis sur le deuxième filtre "La semaine dernière". 8. Le système calcule la vitesse moyenne des trajets effectués par le vélo sélectionné durant la semaine dernière et affiche les statistiques de la vitesse moyenne des trajets par date. 9. Le cycliste clique sur le premier filtre "Vitesse" puis sur le deuxième filtre "L'année dernière". 10. Le système calcule la vitesse moyenne des trajets effectués par le vélo sélectionné durant l'année dernière et affiche les statistiques de la vitesse moyenne des trajets par mois. 11. Le cycliste clique sur le premier filtre "Vitesse" puis sur le deuxième filtre "Date personnalisée".

E_Bike gestion tracker à travers TCP protocole

	12. Le système calcule la vitesse moyenne des trajets effectués par le vélo sélectionné durant la période sélectionnée et affiche les statistiques de la vitesse moyenne des trajets pour cette période.
Scénario d'échec	<p>E1 : Aucun trajet n'a été effectué la semaine dernière. Le scénario échoue au point 1 du scénario nominal. 1 : le système n'affiche pas des statistiques.</p> <p>E2 : Aucun trajet n'a été effectué le mois dernier. Le scénario échoue au point 3 du scénario nominal. 2 : le système n'affiche pas des statistiques.</p> <p>E3 : Aucun trajet n'a été effectué l'année dernière. Le scénario échoue au point 5 du scénario nominal. 3: le système n'affiche pas des statistiques.</p> <p>E4 : Aucun trajet n'a été effectué la semaine dernière par le vélo sélectionné. Le scénario échoue au point 7 du scénario nominal. 1 : le système n'affiche pas des statistiques.</p> <p>E5 : Aucun trajet n'a été effectué l'année dernière par le vélo sélectionné. Le scénario échoue au point 9 du scénario nominal. 1 : le système n'affiche pas des statistiques.</p> <p>E6 : Aucun trajet n'a été effectué durant la période sélectionnée par le vélo choisi. Le scénario échoue au point 11 du scénario nominal. 1 : le système n'affiche pas des statistiques.</p> <p>E7 : Le cycliste sélectionne une date future lors de la sélection d'une date personnalisée. Le scénario échoue au point 11 du scénario nominal. 1 : le système affiche une erreur.</p>

Tableau 18: Description textuelle « Consulter vitesse moyenne »

5.3 Modélisation conceptuelle

5.3.1 Diagramme de séquence

Le diagramme de séquence suivant présente l'interaction entre le cycliste et le système pour consulter les statistiques

E_Bike gestion tracker à travers TCP protocole

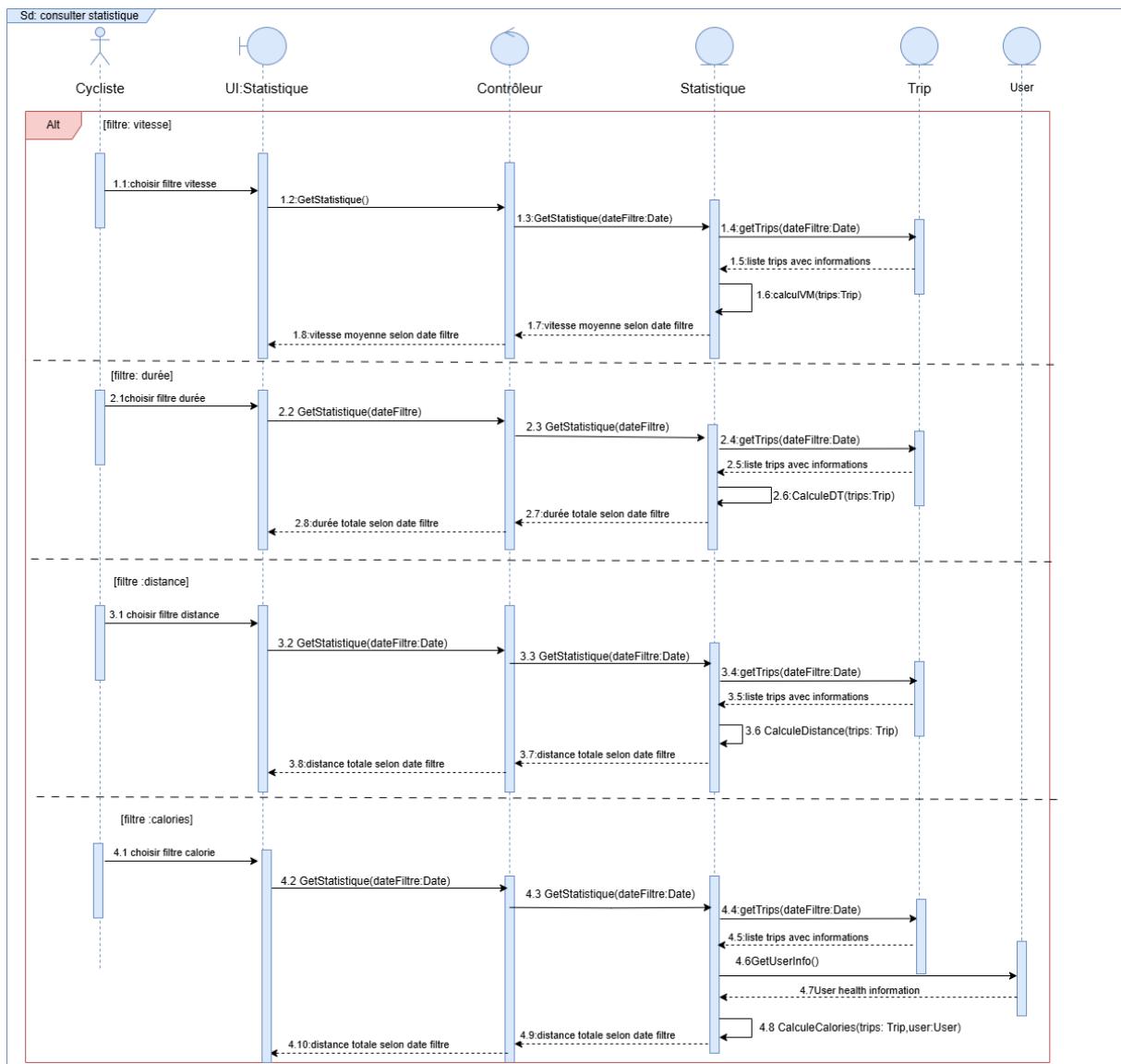


Figure 45: Diagramme de séquence consulter statistiques

Le Diagramme de séquence suivant présente l'interaction entre le cycliste et le système pour consulter les statistiques des trajets effectués par le vélo sélectionné

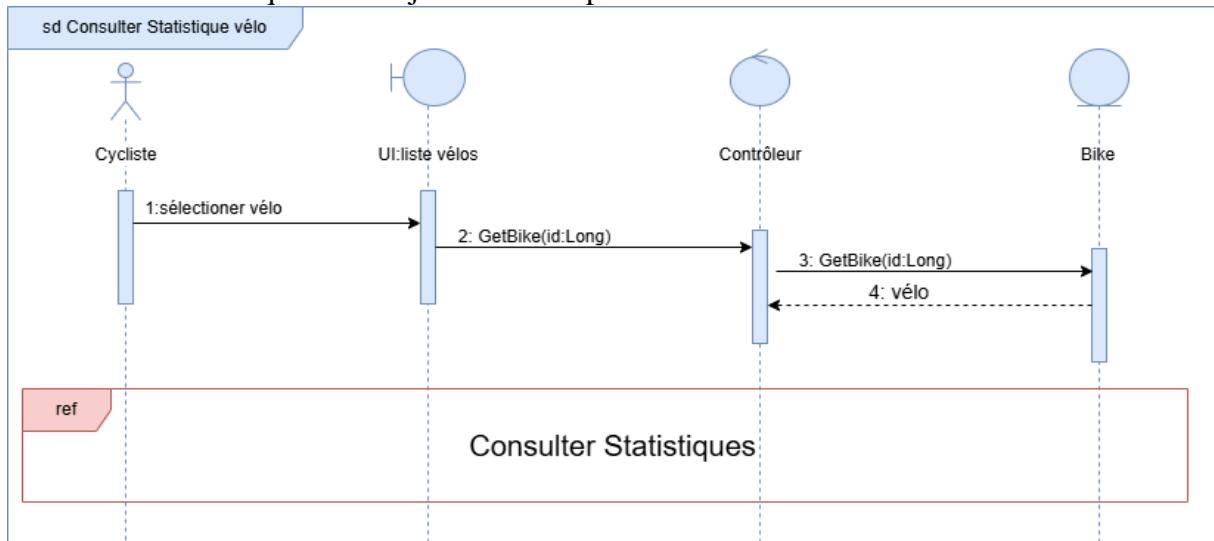


Figure 46: Diagramme de séquence statistique vélo

E_Bike gestion tracker à travers TCP protocole

5.3.2 Diagramme de classe

Le diagramme de classe ci-dessous modélise les relations entre le cycliste, ses vélos, ses trajets et ses points de trajet, et les statistiques qui peuvent être obtenues à partir de ces données. Les statistiques peuvent être calculées pour un vélo spécifique ou globalement pour tous les vélos d'un cycliste.



Figure 47: Diagramme de classe

5.4 Réalisation

Dans cette partie, nous présentons plusieurs captures d'écran afin d'illustrer la réalisation des différentes tâches du sprint

- Modification lors de rétrospective du sprint précédent :

Lors de la rétrospective du sprint précédent, le Product Owner a demandé les modifications suivantes. La figure ci-jointe montre les détails des modifications demandées, incluant l'assistance moyenne, le nombre de calories brûlées, ainsi que la consommation de charge durant le trajet.

E_Bike gestion tracker à travers TCP protocole

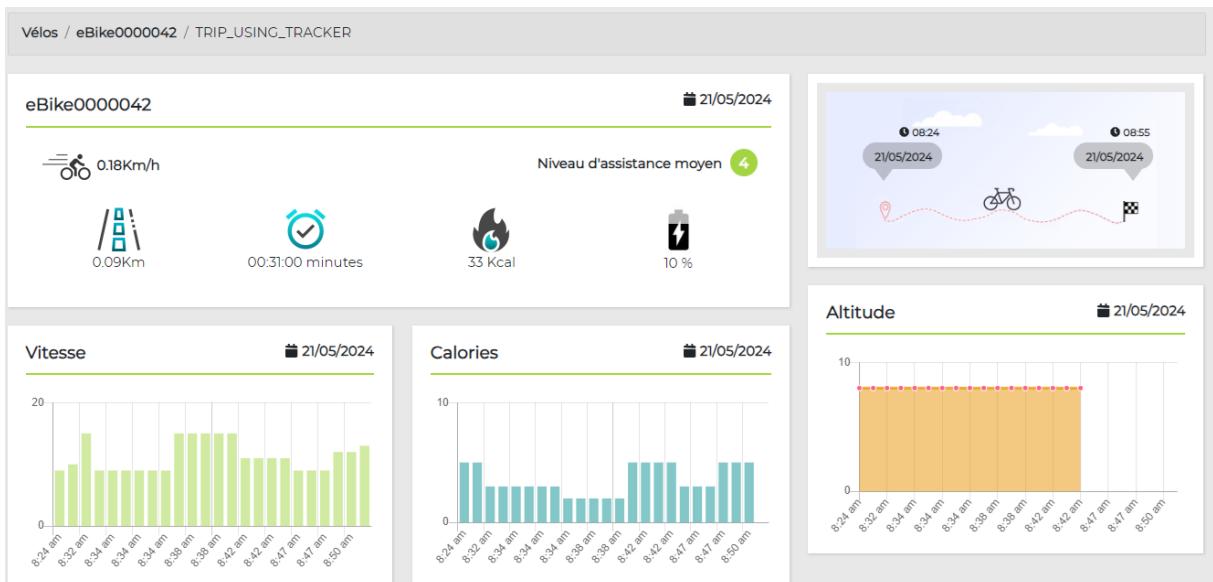


Figure 48:Détaille d'un trajet

- Statistique d'un cycliste pour un seul vélo :**

La figure suivante présente la liste des vélos utilisés par un cycliste.

The figure shows a list of bicycles managed by the user. The table includes columns for Nom, Dernière Connexion, Statut, Code Pin, Connectivité, and Rôle. The bicycles listed are:

Nom	Dernière Connexion	Statut	Code Pin	Connectivité	Rôle
eBike0000042	21/05/24 08:24	Connecté		Connecté	Conducteur
eBike0000025_MID	26/01/24 11:54	Non connecté	●	Non connecté	Propriétaire
eBike0000017_badra	06/12/23 10:29	Non connecté	●	Non connecté	Conducteur
eBike0000043	25/10/23 13:32	Non connecté	●	Non connecté	Propriétaire

Figure 49: Liste des vélos

E_Bike gestion tracker à travers TCP protocole

La figure suivante présente la liste des trajets réalisés par le cycliste avec le vélo « eBike0000042 ».

Parcours					
Nom	Date	Durée totale	Vitesse	Distance	
TRIP_USING_TRACKER	21-mai-2024	00:31:00	0.18Km/h	0.09Km	
TRIP_USING_TRACKER	19-mai-2024	00:08:30	0.65Km/h	0.09Km	
TRIP_USING_TRACKER	19-mai-2024	00:08:30	0.65Km/h	0.09Km	
TRIP_USING_TRACKER	19-mai-2024	00:08:30	0.09Km/h	0.01Km	

Figure 50: Liste des trajets pour un vélo

La figure suivante présente les statistiques de la vitesse moyenne pour les trajets réalisés la semaine dernière.

Calcule pour les trajets de 19 mai 2024 : $(0.09 + 0.65 + 0.65) / 3 = 0.46$

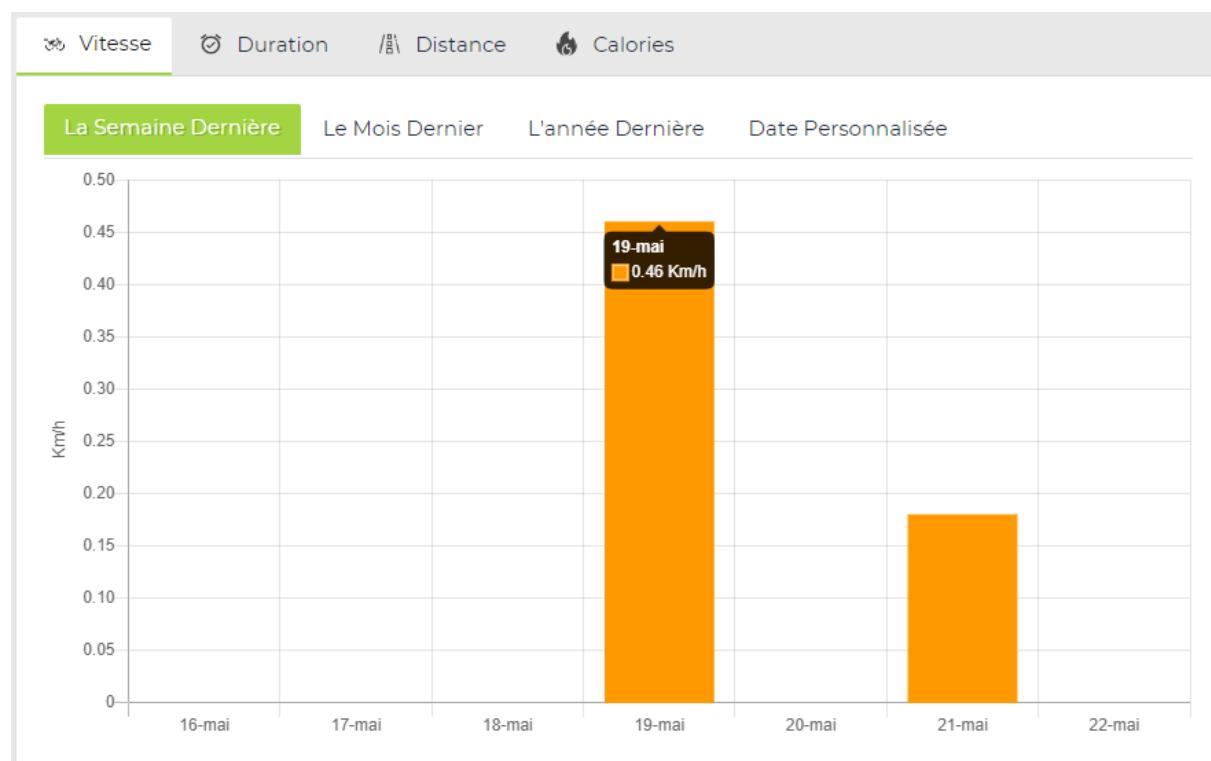


Figure 51: Statistique de la vitesse moyenne d'un vélo

E_Bike gestion tracker à travers TCP protocole

La figure suivante présente la durée des trajets réalisés le mois dernier, en minutes. Elle indique pour chaque date la durée totale effectuée par un cycliste avec un vélo.

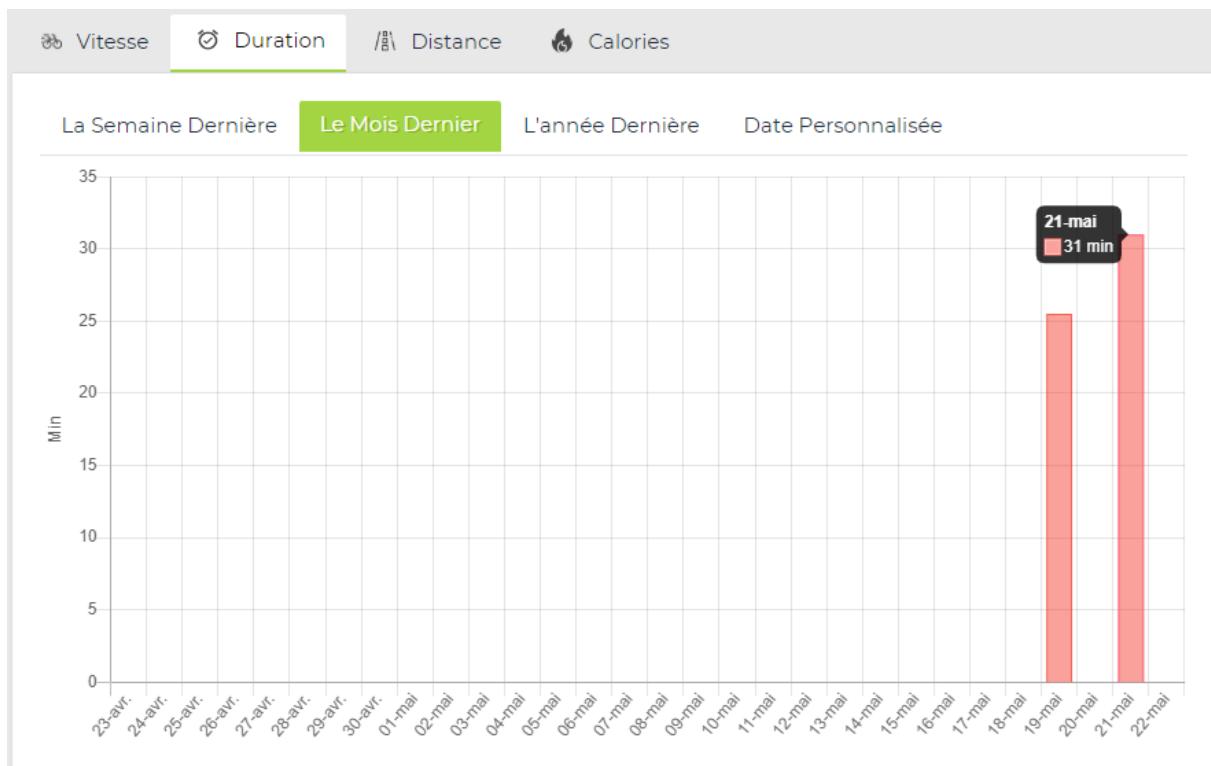


Figure 52: Statistique pour la durée des trajets

La figure suivante montre la distance totale en kilomètres parcourue chaque mois au cours de l'année dernière. Elle fournit des détails précis sur les distances mensuelles réalisées par le cycliste, permettant une analyse approfondie de son activité sur l'ensemble de l'année.

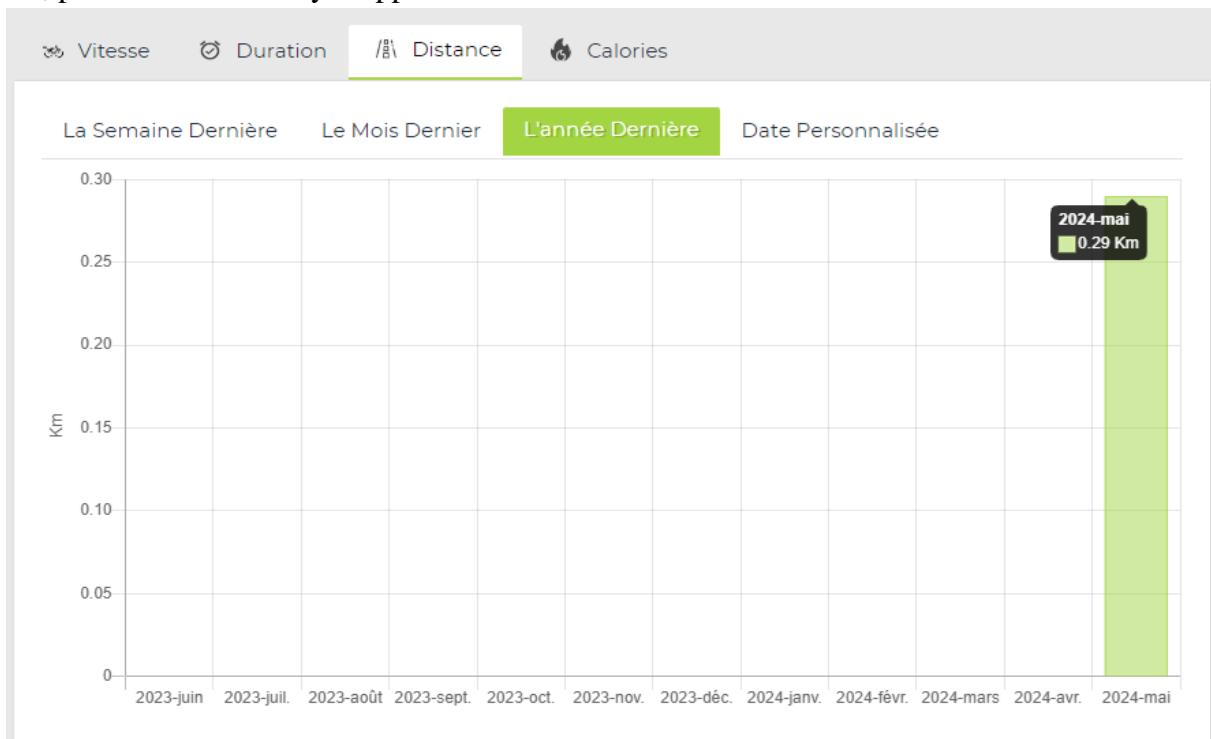


Figure 53: Statistique pour distance des trajets

E_Bike gestion tracker à travers TCP protocole

La figure ci-dessous illustre le nombre de calories brûlées montre le nombre en utilisant le filtre « date personnalisée ».

Pour le calcul de nombre de calories pour chaque trip point :

Pour le calcul de nombre de calories chaque minute :

- deltaVaule : la différence entre le niveau de batterie actuelle et précédente
- MET : équivalent métabolique de la tâche qui dépend de la vitesse
- Effort : $((\text{MET} * 3.5) * \text{poids de cycliste}) / 200$
- batteryValue = $(\text{deltaValue} * \text{Masse molaire de l'acide gras} * \text{densité de la graisse corporelle}) / (1000*60)$
- Calories = effort – (batteryValue * (batteryHealth/100))

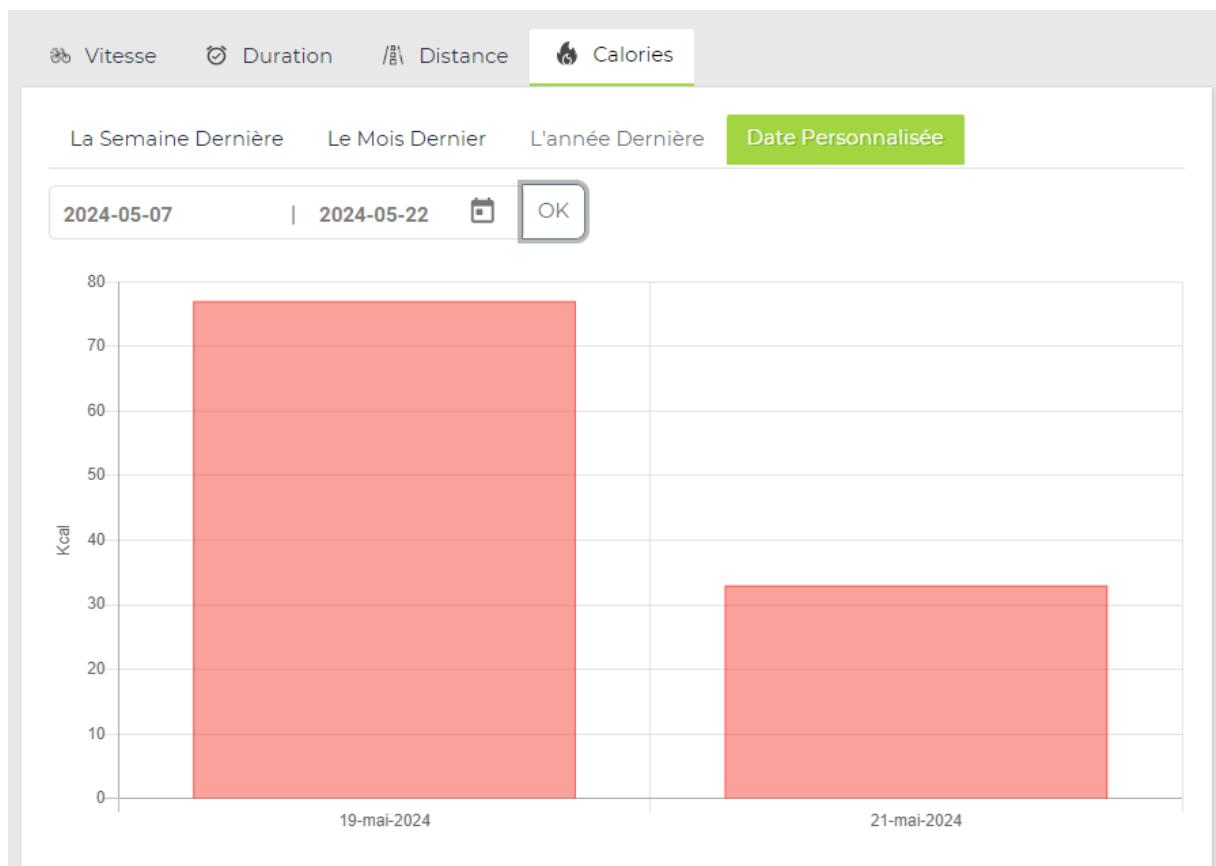


Figure 55: Statistiques de nombre de calories brûlées

- **Statistique pour la totalité des vélos d'un cycliste**

La figure suivante présente la vitesse moyenne en kilomètre par heure de l'ensemble des trajets effectués par un cycliste pour le mois dernier, détaillant les variations de vitesse pour chaque jour durant cette période.

E_Bike gestion tracker à travers TCP protocole

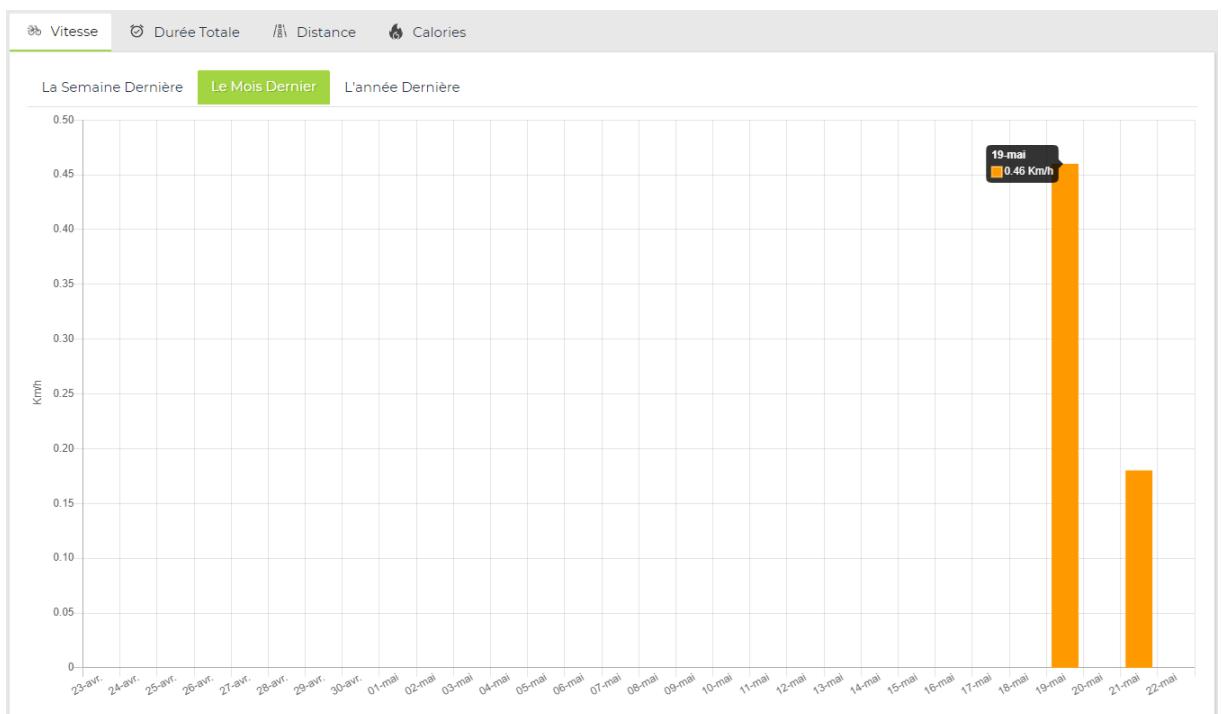


Figure 56: Statistique de la vitesse moyenne pour totalité des vélos

La figure suivante présente le nombre de calories brûlées en Kcal de l'ensemble des trajets effectués par un cycliste pour chaque mois au cours de l'année dernière.

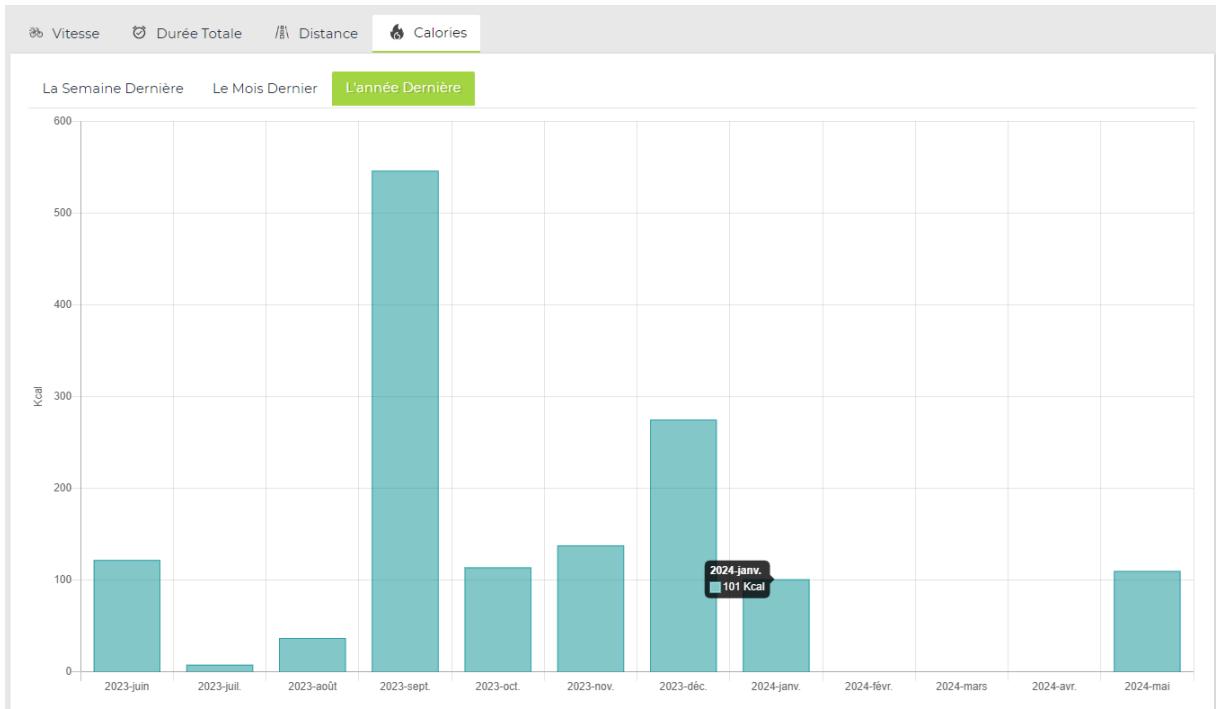


Figure 57: Nombre de calories brûlée pour tous les vélos

5.5 Test

Dans cette section, nous avons entrepris de tester le cas de test Consultée statistiques d'un cycliste avec un vélo spécifique pour le filtre date personnalisée

E_Bike gestion tracker à travers TCP protocole

Cas de test	Consultée statistique d'un vélo de cycliste par le filtre date personnalisé
Acteur	Cycliste
Prérequis	Authentifier Réalisé des trajet
Scénario	1.Le cycliste choisit un intervalle de date. 2.Le système vérifie l'intervalle temporel sélectionné. 3.Le système vérifie l'existence de uuid de vélo pour le cycliste. 4.Le système calcule la vitesse moyenne, la durée totale, le nombre de kilomètre total et le nombre de calories totale pour chaque date dans l'intervalle
Résultat	Le système affiche les statistiques de chaque date dans la période sélectionnée

Cette figure illustre le résultat obtenu avec Postman pour la période allant du 13 mai 2024 au 22 mai 2024.

```

1 [
2   {
3     "id": null,
4     "uuid": null,
5     "totalDistance": 0.1960035108006924,
6     "totalDuration": 1530000.0,
7     "totalCalories": 77,
8     "maxSpeedRate": null,
9     "minSpeedRate": null,
10    "averageSpeedRate": 0.4630670842557468,
11    "date": "2024-05-19",
12    "bikeUuid": null,
13    "year": null,
14    "month": null,
15    "userUuid": null,
16    "bikeId": null,
17    "bikeName": null
18  },
19  {
20    "id": null,
21    "uuid": null,
22    "totalDistance": 0.09199658281892366,
23    "totalDuration": 1860000.0,
24    "totalCalories": 33,
25    "maxSpeedRate": null,
26    "minSpeedRate": null,
27    "averageSpeedRate": 0.1780579022301748,
28    "date": "2024-05-21"
29  }
]

```

Figure 58: Test unitaire filtre « date personnalisée »

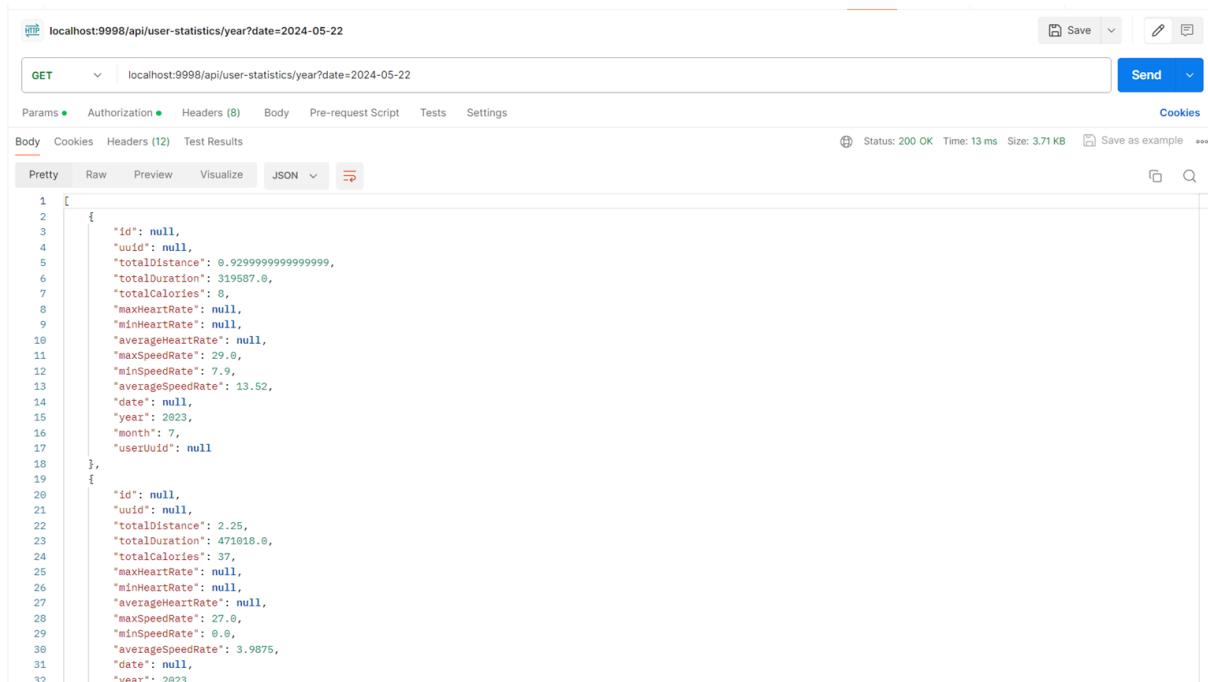
Le cas de test suivant présente les statistiques consultées pour l'ensemble des vélos avec le filtre "année précédente".

Cas de test	Consultée statistique de cycliste par le filtre année dernière
Acteur	Cycliste

E_Bike gestion tracker à travers TCP protocole

Prérequis	Authentifié Réalisé des trajets
Scénario	1. Le système récupère la date actuelle. 2. Le système calcule la vitesse moyenne, la durée totale, le nombre total de kilomètres parcourus et le nombre total de calories brûlées pour l'ensemble des vélos du cycliste.
Résultat	Le système affiche les statistiques de chaque mois de l'année précédente

Ces figures illustrent le résultat obtenu avec Postman pour la période allant du 22 mai 2024 jusqu'à un an en arrière.

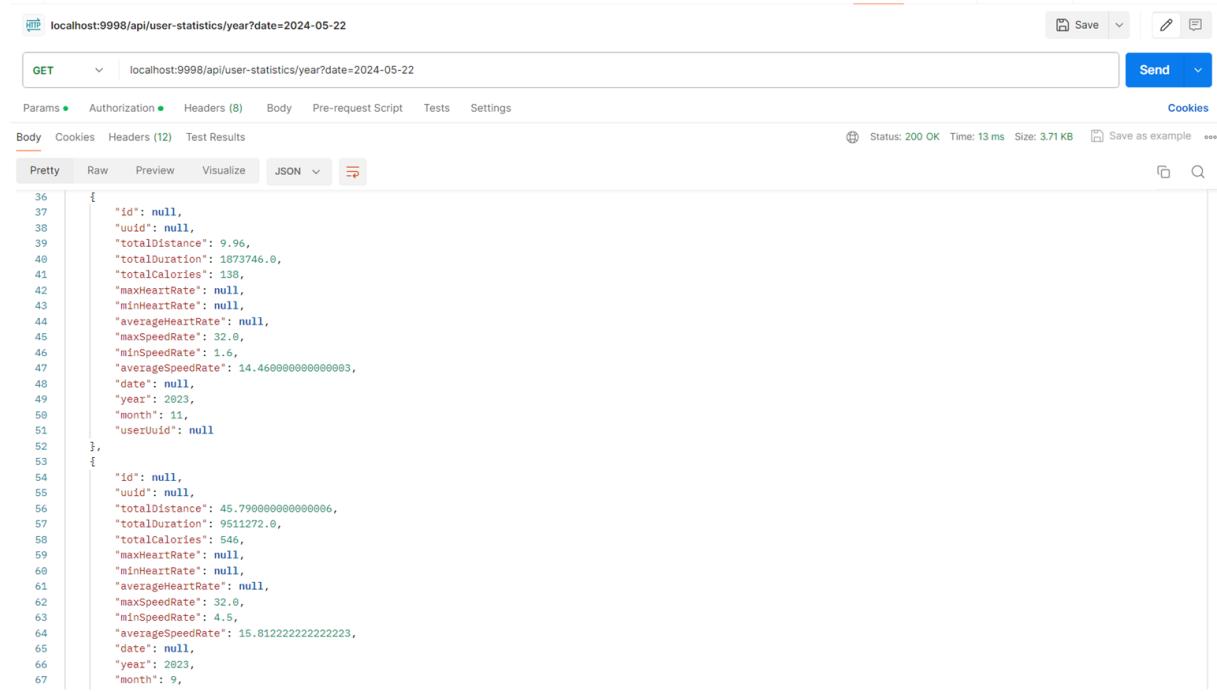


```

1 [
2   {
3     "id": null,
4     "uuid": null,
5     "totalDistance": 0.9299999999999999,
6     "totalDuration": 319587.0,
7     "totalCalories": 8,
8     "maxHeartRate": null,
9     "minHeartRate": null,
10    "averageHeartRate": null,
11    "maxSpeedRate": 29.8,
12    "minSpeedRate": 7.9,
13    "averageSpeedRate": 13.52,
14    "date": null,
15    "year": 2023,
16    "month": 7,
17    "userUuid": null
18  },
19  {
20    "id": null,
21    "uuid": null,
22    "totalDistance": 2.25,
23    "totalDuration": 471018.0,
24    "totalCalories": 37,
25    "maxHeartRate": null,
26    "minHeartRate": null,
27    "averageHeartRate": null,
28    "maxSpeedRate": 27.0,
29    "minSpeedRate": 0.0,
30    "averageSpeedRate": 3.9875,
31    "date": null,
32    "year": 2023,
  }
]
  
```

Figure 59: Test unitaire pour le filtre année dernière

E_Bike gestion tracker à travers TCP protocole



The screenshot shows a browser-based API testing interface. The URL is `localhost:9998/api/user-statistics/year?date=2024-05-22`. The response is a JSON object containing two entries, each representing a year's worth of data. The data includes total distance, total duration, total calories, heart rate metrics, speed metrics, date, year, and month. The JSON is displayed in a "Pretty" format with line numbers on the left.

```
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
```

```
{
  "id": null,
  "uuid": null,
  "totalDistance": 9.96,
  "totalDuration": 1873746.0,
  "totalCalories": 138,
  "maxHeartRate": null,
  "minHeartRate": null,
  "averageHeartRate": null,
  "maxSpeedRate": 32.0,
  "minSpeedRate": 1.6,
  "averageSpeedRate": 14.46000000000003,
  "date": null,
  "year": 2023,
  "month": 11,
  "userUuid": null
},
{
  "id": null,
  "uuid": null,
  "totalDistance": 45.79000000000006,
  "totalDuration": 951272.0,
  "totalCalories": 546,
  "maxHeartRate": null,
  "minHeartRate": null,
  "averageHeartRate": null,
  "maxSpeedRate": 32.0,
  "minSpeedRate": 4.5,
  "averageSpeedRate": 15.81222222222223,
  "date": null,
  "year": 2023,
  "month": 9
}
```

Figure 60: Suite test unitaire pour le filtre année dernière

5.6 Revue du sprint 3

Lors de la réunion de rétrospective de sprint, après une démonstration de l'application, le Product Owner a confirmé les fonctionnalités développées lors de ce sprint.

5.7 Conclusion

Dans cette conclusion de chapitre, nous reconnaissons les réalisations accomplies dans le cadre du dernier sprint, notamment l'implémentation réussie du module de statistiques qui a satisfait les attentes du Product Owner. Cependant, il est important de souligner qu'il reste des opportunités d'amélioration pour consolider davantage cette application web.