

Université de Gafsa

Institut Supérieur des Sciences Appliquées et de Technologie de GAFSA

Department de l'Informatique ET Telecommunication



Titre

Application Android pour commander les objets IOT connectés

Présenté et soutenu par :

Hiba Benkhalifa

En vue de l'obtention de

Licence Appliquée en ingénierie des systèmes informatique

Sous la Direction de : Monsieur Walid Fakheth et Monsieur Jamel Hatay

Titre & Prénom et NOM de l'encadreur

Soutenu le

Devant le jury composé de :

Président :

Rapporteur :

Membres :

2022/2023

Dédicace

A mon père

Qui m'a tout donné pour être la meilleure

A ma mère J'espère qu'elle le trouve
récompensant de tous les sacrifices qu'elle a
fait pour moi

A mes chers frères et ma sœur

Pour leur support moral

A mes amis

Remerciement

Au terme de ce projet de fin d'étude, mes vifs remerciements sont dédiés à tous ceux qui ont contribué directement ou indirectement à l'élaboration de ce projet.

J'exprime toute ma gratitude à Monsieur Walid Fakhed, mon encadreur à l'institut supérieur des études scientifique et technologique de Gafsa pour ses précieux conseils, sa patience et sa disponibilité.

J'adresse également mes sincères remerciements à Monsieur Jamel Hatay mon encadreur du stage à la société IOT Medical Technology pour ses précieux conseils, son engagement.

Je tiens aussi à remercier tout les membres de l'équipe IOT Medical Technology pour leur encouragements.

J'adresse aussi mes remerciements aux membres de jury pour avoir accepté de juger, d'évaluer ce travail

Par la même occasion, je présente mes sincères reconnaissances à tous mes enseignants qui m'ont soutenu au long de mon cursus et à tous ceux qui m'ont porté aide et soutien durant mes études.

Merci pour toute personne qui a participé de près ou loin pour l'accomplissement de ce modeste travail.

Résumé :

Ce travail est dans le cadre d'un projet de fin d'étude en vue d'obtention d'un diplôme de licence appliquée en ingénierie des systèmes informatiques à l'institut supérieur des études scientifique et technologique de Gafsa.

Ce projet été réalisé à IOT Médical technologie Notre travail consiste à développer une application mobile Android qui permet à un mobile de se connecter à un objet médical.

Mot clés : UML, Flutter, Dart, e-santé, IOTM, IOT, BLE, Bluetooth LE, BLE-bleu, l'UUID

Abstract:

This work is an end of studies project to obtain an applied license diploma in computer system engineering in the Institute of higher scientific and technological studies of Gafsa. This project is realized in IOT Medical Technology. Our project is an android mobile application for distance medical monitoring.

Keywords : UML, Flutter, Dart, e-santé, IOTM, IOT, BLE, Bluetooth LE, BLE-bleu, l'UUID

Table des matières

Introduction générale.....	1
Chapitre 1 : présentation générale du cadre de stage	2
Introduction	3
I. La présentation du cadre de société Présentation de la société	3
1. L'équipe :	3
2. Les services :	3
3. Fiche technique de la société	3
II. Cadre générale du projet.....	4
1. IoT :	4
2. Fonctionnement de IoT	5
III. Etat de l'art	6
1. Critique de l'état de l'art.....	6
2. Solution propose	6
IV. Spécification des besoins.....	7
1. Caractéristiques fonctionnelles du futur système	7
2. Les contraintes (Données à prendre en compte).....	7
3. Caractéristique non fonctionnelle du futur système	8
V. Objectifs du projet.....	8
1. Choix de l'environnement du développement.....	8
2. Choix de l'architecture de développement	9
3. Architecture MVC	10
4. Choix de l'architecture	10
Conclusion	10
Chapitre 2 : conception	11
Introduction	12

I.	Choix de methodology	12
1.	Les vues d'UML	13
2.	Les diagrammes d'UML	13
II.	Diagramme de cas d'utilisation générale.....	15
III.	Diagramme de « scan »	16
IV.	Diagramme de cas d'utilisation “connexion”	17
V.	Diagramme de cas d'utilisation “afficher les données”	18
VI.	Diagramme de séquence global	19
VII.	Diagramme de séquence scan	20
VIII.	Diagramme de séquence “connecter”	21
IX.	Diagramme de class	22
	Conclusion	23
	Chapitre 3 : Réalisation	24
	Introduction	25
I.	Environnement du travail	25
1.	Environnements matériels	25
2.	Environnement logiciel.....	25
II.	Choix de l'utilisation du flutter	27
III.	Choix de l'utilisation de BLE	28
IV.	Choix du Bluetooth LE bleu :.....	28
V.	Les Interfaces Graphiques principales:	29
	Conclusion	31
	Conclusion générale et perspectives.....	32
	Bibliographie	33
	ANNEXE	34

Liste des figures

Figure 1: les composants de l'IOT	4
Figure 2: les étapes de fonctionnement d'un scénario IOT	5
Figure 3: BLE et IOT	5
Figure 4: Architecture MVC	10
Figure 5: Diagramme de cas d'utilisation générale.....	15
Figure 6: Diagramme de scan.....	16
Figure 7: Diagramme "connexion"	17
Figure 8: Diagramme de cas d'utilisation "afficher les données"	18
Figure 9: Diagramme de séquence global	19
Figure 10: Diagramme de séquence scan	20
Figure 11: Diagramme de séquence "connecter"	21
Figure 12: Diagramme de class	22
Figure 13: Visual studio code.....	26
Figure 14: Flutter.....	26
Figure 15: Dart	26
Figure 16: Luicid chart.....	27
Figure 17: page 1 de l'application	30
Figure 18: page 2 de l'application.....	30
Figure 19: Page 3 de l'application.....	31

Liste des tableaux

Tableau 1: Les étapes de fonctionnement d'un scénario IOT.....	5
Tableau 2: Etude textuelle du diagramme de cas d'utilisation « scan ».....	16
Tableau 3: Etude textuelle du diagramme de cas d'utilisation « connexion »	17
Tableau 4: Etude textuelle du diagramme de cas d'utilisation « afficher les données ».....	18
Tableau 5: Caractéristique de l'ordinateur.....	25
Tableau 6: Caractéristique de Smart phone.....	25
Tableau 7: Caractéristique de montre intelligente T1s.....	25

Introduction générale

La santé est un élément indispensable du bien-être général, et il est primordial de prendre soin de notre corps et de notre santé pour maintenir une bonne qualité de vie. Nos jours, avec les progrès de la technologie, il est plus facile que jamais de suivre notre santé à l'aide d'applications mobiles, de montres intelligentes et d'autres appareils. Ces outils peuvent nous aider à surveiller les habitudes de sommeil, l'activité physique, la nutrition, le stress et d'autres aspects de notre santé. En surveillant notre santé, nous pouvons aussi détecter précocement les anomalies et prendre des mesures pour prévenir ou traiter les problèmes de santé dans les meilleurs délais. Cela peut nous aider à mieux vivre, à être plus productifs au travail, à améliorer la qualité de vie et à réduire les coûts des soins de santé. A l'aide des dispositifs médicaux, tels que des montres médicales, ceintures médicales...

Sont capable de réaliser des mesures médicales multiples, telles que : la tension artérielle, la fréquence cardiaque, la saturation en oxygène du sang, la variabilité de la fréquence cardiaque, la température du corps, etc.

Dans le but d'exploiter le potentiel de ces montres médicales, la société IOT-Med Tech nous a proposé que notre travail s'intégrera dans le cadre de concevoir et réaliser une application mobile pour connecter une montre médicale de la marque T1S.

Notre présent rapport est constitué de trois chapitres, le premier chapitre, intitulé "Présentation du cadre du stage", présentera la société d'accueil, la problématique et la solution proposée, et on finit par l'architecture de la solution.

Ensuite, nous aborderons le deuxième chapitre, intitulé "conception" qui présentera les besoins fonctionnels et non fonctionnels, ainsi que la conception UML.

Nous passerons ensuite au troisième chapitre, intitulé "Réalisation", qui décrit l'environnement de travail. Enfin, nous terminerons ce rapport par une conclusion générale qui résume notre travail avec des perspectives pour les futurs travaux de ce projet.

Chapitre 1 : présentation générale du cadre de stage

Introduction

Le présent chapitre définit le cadre général du projet. Nous présentons d'abord l'organisation d'accueil. Ensuite, nous allons exposer l'état de l'art accompagné d'un critique ainsi que la solution proposée. Nous arrivons à l'architecture de l'application.

I. La présentation du cadre de société Présentation de la société

IOT Medical Technology (IOT-Med-Tec) est une start-up spécialisée dans l'adaptation des équipements médicaux connectés de supervision et de consultation des patients à distance ainsi que d'autres services

1. L'équipe :

- Directeur général : Mr Chiheb Belaid
- Directeur technique : Mr Jamel Hatay
- Ingénieur assistant : Mr Younes Ben Salah

2. Les services :

La société IOT Med Tec propose des solutions pour rendre l'environnement de santé plus sûr et plus pratique pour les patients et les professionnels de la santé, telles que :

- **Télémédecine** : Rassemble les pratiques médicales autorisées ou facilitées par les télécommunications
- **Téléconsultation médicale** : Consultation auprès des médecins inscrits à l'ordre des médecins ou des professionnels de la santé : examen des dossiers médicaux ; discussion par messages avec médecin ; fixer des rendez-vous par vidéo avec médecin via ordinateur ; récupérer le traitement en ligne.
- **Dossier médical en ligne** : Sauvegarder toutes les informations de santé : les analyses, les radios, les traitements, la courbe de poids, les allergies, etc. Gérer tout l'historique de santé et partager, suivant les initiatives, avec les médecins.

3. Fiche technique de la société

- **Secteur d'activité** : Internet des objets médicaux
- **Date de fondation** : Mars 2020
- **Site web** : <http://www.iot-medtec.com/>
- **Adresse e-mail** : IOT.MedTec@gmail.com
- **Adresse** : 4000 Rue les Califes -Sousse – Tunisie

II. Cadre générale du projet

1. IoT :

L'Internet of Things (IoT) décrit le réseau de terminaux physiques, les « objets », qui intègrent des capteurs, des logiciels et d'autres technologies en vue de se connecter à d'autres terminaux et systèmes sur Internet et d'échanger des données avec eux. Ces terminaux peuvent aussi bien être de simples appareils domestiques que des outils industriels d'une grande complexité. Avec plus de 7 milliards de terminaux IoT connectés aujourd'hui, les experts s'attendent à ce que ce nombre passe à 10 milliards d'ici 2020 et 22 milliards d'ici 2025

Pourquoi l'Internet of Things (IoT) est-il si important ?

Ces quelques dernières années, l'IoT est devenu l'une des technologies les plus importantes du 21ème siècle. Maintenant que nous pouvons connecter des objets du quotidien (appareils électroménagers, voitures, thermostats, interphones bébés) à Internet par l'intermédiaire de terminaux intégrés, des communications sont possibles en toute fluidité entre les personnes, les processus et les objets.

Grâce à des traitements informatiques peu coûteux, au Cloud, au Big Data, à l'analytique et aux technologies mobiles, les objets physiques peuvent partager et collecter des données avec un minimum d'intervention humaine. Dans ce monde hyper connecté, les systèmes digitaux peuvent enregistrer, surveiller et ajuster chaque interaction entre les objets connectés. Le monde physique rencontre le monde digital, et ils coopèrent.

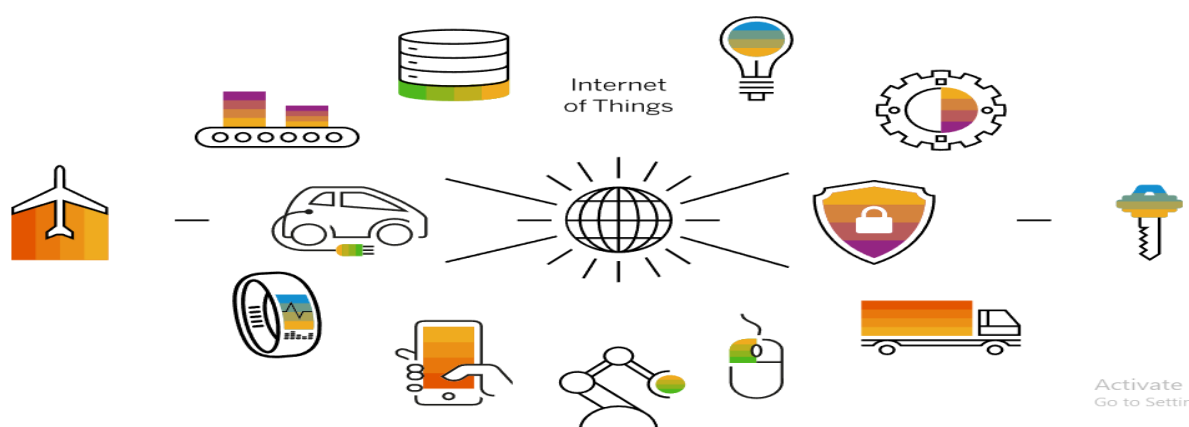


Figure 1: les composants de l'IOT

2. Fonctionnement de IoT

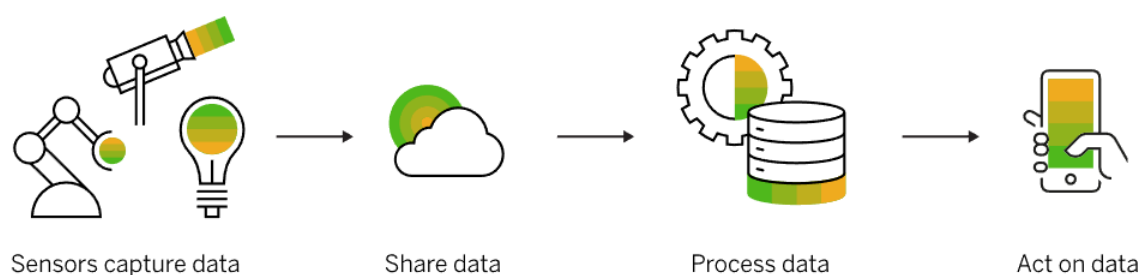


Figure 2: les étapes de fonctionnement d'un scénario IOT

Tableau 1: Les étapes de fonctionnement d'un scénario IOT

SENSING	NETWORKING	SERVICE	INTERFACE
Les "hardwares" physiques comme les capteurs intelligents	Connecte les équipements entre eux	Les technologies du "middleware" qui permet de faire communiquer entre eux "hardware" et "software"	Les plateformes qui présentent les applications aux utilisateurs finaux

Bluetooth Low Energy dans l'IoT peut supporter les appareils IoT à économiser de l'énergie en les maintenant en mode veille lorsqu'ils ne sont pas utilisés, puis laissez les utilisateurs quitter le mode lorsqu'ils sont déconnectés ou reconnectés.

Bluetooth Low Energy dans le domaine l'IoT est idéal car, au contraire des applications Bluetooth classiques, qui se reconnectent aux appareils à un moment de six secondes ou plus, Les applications IoT BLE peuvent se coupler et se reconnecter rapidement avec des appareils en six millisecondes seulement.



Figure 3: BLE et IOT

III. Etat de l'art

D'après mes recherches ils n'y a pas eu d'implémentation auparavant des applications similaires à mon projet utilisées pour suivre l'état de santé d'un patient.

Il est à remarquer qu'il existe quelque applications mobile Considérées comme application locale utilisée surtout dans le domaine du sport.

Les sportifs peuvent suivre quelque signes comme les calories consommées, le nombre de pas parcourus, rythme cardiaque...par cette application en local.

1. Critique de l'état de l'art

Notre application couvre un domaine plus vaste qui concrétise une partie importante de la télémédecine puisque le corps médicale peut suivre à distance et en temps réel l'état de santé du patient.

L'application objet de mon projet fait appel à beaucoup de réseau comme réseau IOT réseau de télécom ,réseau internet et Cloud .

L'implémentation de notre projet demande une connaissance des nouvelles technologies comme le BLE et une certaine connaissance des nouveaux outils de développement multiplateforme tel que le Framework Flutter qui se base sur le langage de programmation Dart, qui est un langage conçu et réalisé par Google.

Donc, nous somme investi pour acquérir le minimum nécessaire de toutes ces nouveautés pour pouvoir arriver a réaliser ce challenge.

2. Solution proposée

Une application mobile qui fait la connexion entre un téléphone portable et un dispositif médical.

IV. Spécification des besoins

L'utilité d'un système logiciel est déterminée par ses exigences fonctionnelles et ses caractéristiques non-fonctionnelles, ainsi que ses contraintes :

- Une exigence fonctionnelle est une exigence définissant une fonction du système à développer. Ce que le système doit faire
- Une exigence non-fonctionnelle est une exigence qui caractérise une propriété (qualité) désirée du système telle que sa performance, sa robustesse, sa convivialité, etc. ;
- Une contrainte est une restriction sur une ou plusieurs valeurs d'une partie du système ou de tout le système.

1. Caractéristiques fonctionnelles du futur système

Les caractéristiques de l'application mobile à développer sont :

- ❖ **Connexion Bluetooth LE** : La connexion BLE doit être effectuée avec succès
- ❖ **Synchronisation des données** : L'application doit pouvoir synchroniser les données collectées avec la smart Watch en assurant une mise à jour régulière.
- ❖ **Collecte des données** : L'application doit pouvoir récupérer les données de santé de la smart Watch, citant la pression artérielle, fréquence cardiaque, saturation en oxygène (services et caractéristiques fournis par le device).
- ❖ **Interfaces utilisateur** : L'utilisateur doit avoir une interface claire et compréhensible.
- ❖ **sécurité et fiabilité** : Il faut garantir la sécurité et la confidentialité des données médicales des utilisateurs.

2. Les contraintes (Données à prendre en compte)

Les contraintes sont des clefs qu'on doit les considérer dans l'élaboration et la réalisation de notre application mobile :

- Les contraintes esthétiques:
 - Donner une importance au côté design en utilisant des couleurs significatives afin d'attirer les utilisateurs de l'application mobile.

- Utilisation d'une typographie foncée sur un fond claire pour améliorer la lisibilité du contenu.

3. Caractéristique non fonctionnelle du futur système

Les besoins non fonctionnels peuvent être considérés comme des besoins fonctionnels spéciaux. Parfois, ils ne sont pas rattachés à un cas d'utilisation particulier, mais ils caractérisent tout le système (l'architecture, la sécurité, le temps de réponse, etc.) en vue de faciliter l'utilisation et améliorer les performances :

- Besoin d'utilisation : font référence aux aspects généraux de l'interface utilisateur, entoure assurer un aspect ergonomique aux interfaces ;
- Besoins de performance : décrivent les performances d'exécution du système, généralement en termes de temps de réponse ;
- Besoins de disponibilité/fiabilité : assurer que le système soit toujours bien prêt à l'emploi
- Besoins de sécurité : peuvent définir les niveaux d'accès possibles au système pour les utilisateurs du système
- Besoins physiques : définissent les configurations matérielles minimales essentielles à la progression du système
- Besoins de déploiement : décrivent la façon dont l'application sera mise en disposition de l'utilisateur final.

V. Objectifs du projet

A travers ce projet, nous allons essayer de fixer les principaux objectifs à atteindre via cette application :

- Connexion Bluetooth
- Synchronisation des données
- Collecte des données (services et caractéristiques fournis par le device)

1. Choix de l'environnement du développement

Pour la réalisation de notre application, nous allons définir un environnement de développement qui facilite à l'utilisateur l'accès aux différentes fonctionnalités de l'application :

- Le système d'exploitation : Android
- Les outils de développement y compris les matériels, les langages et les logiciels : Dart, Flutter, Visual Studio Code

2. Choix de l'architecture de développement

Il existe plusieurs types d'architectures client/serveur, les plus répandus sont : Le client/serveur à 2 niveaux, le client/serveur à 3 niveaux et le MVC.

Le choix de l'architecture dépend de notre application mobile qui est généralement conforme au modèle d'architecture MVC (modèle-vue-contrôleur).

3. Architecture MVC

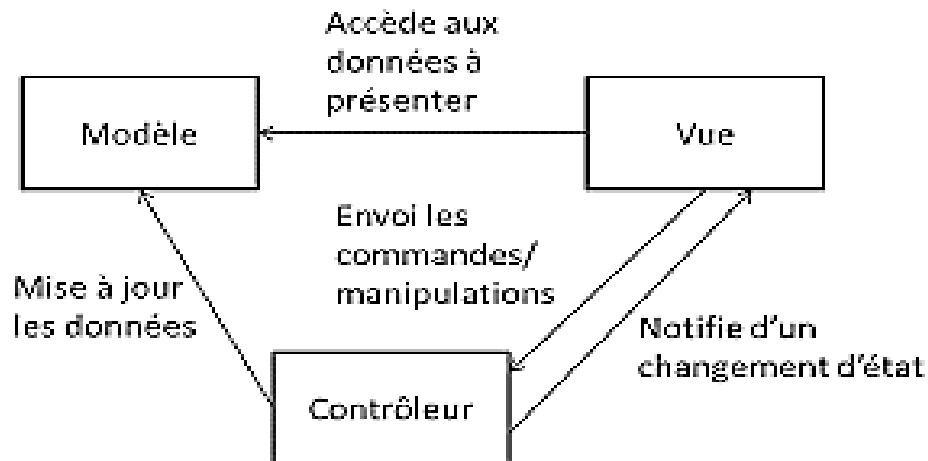


Figure 4: Architecture MVC

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôle ayant chacun un rôle précis dans l'interface. Dans l'architecture MVC, les rôles des trois entités sont les suivants.

- **Modèle** : données (accès et mise à jour)
- **Vue** : interface utilisateur (entrées et sorties)
- **Contrôleur** : gestion des événements et synchronisation

4. Choix de l'architecture

Pour notre projet, cette architecture particulière serait idéale puisqu'elle comporte une division distincte entre les données (mesures) et leur présentation (interface utilisateur). Le composant Contrôleur est responsable de la réception des entrées utilisateur de l'interface utilisateur, et le composant Modèle effectue des calculs et des prédictions en fonction de l'entrée. Enfin, le composant View est mis à jour avec les résultats du Contrôleur.

Conclusion

L'analyse des besoins et des spécifications est une phase extrêmement importante pour le cycle de vie d'un projet.

Au cours de ce chapitre nous avons discuté les exigences et les problèmes à résoudre de point de vue des besoins de l'utilisateur. Cette phase nous a permis de préparer la conception de l'application qui est notre but dans le chapitre suivant.

Chapitre 2 : conception

Introduction

La conception est essentielle dans le cycle de développement d'un projet. Elle consiste à définir des principes de gestion, aboutissant à des modèles conceptuels de données et de traitement. Cette phase permet de trouver des solutions informatiques et techniques pour mettre en œuvre et construire le système analysé.

I. Choix de méthodologie

Concevoir une application mobile n'est pas évident car il faut réfléchir à l'ensemble de tâches qu'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points aux quels on s'intéresse.

Ce type de méthode est appelé analyse. Ainsi plusieurs outils chargés de guider l'analyse ont été conçus, le plus connu d'entre eux en ce qui concerne le système UML.

UML est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ». Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

UML est l'accomplissement de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est un standard défini par l'Object Management Group (OMG).

C'est un langage formel possédant les caractéristiques suivantes :

- ✓ Un langage sans ambiguïtés
- ✓ Un langage universel pouvant servir de support pour tout langage orienté objet
- ✓ Un moyen de définir la structure d'un programme
- ✓ Une représentation visuelle permettant la communication entre les acteurs d'un même projet
- ✓ Une notation graphique simple, compréhensible même par des non informaticiens

UML est un langage qui s'appuie sur un méta modèle, un modèle de plus haut niveau qui définit les éléments d'UML (les concepts utilisables) et leur sémantique (leur signification et leur mode d'utilisation).

1. Les vues d'UML

En UML on distingue les vues suivantes :

- ✓ Vue des cas d'utilisation : c'est la description du modèle « vue » par les acteurs du système. Elle correspond aux besoins attendus par chaque acteur (c'est le QUOI et le QUI).
- ✓ Vue logique : c'est la définition du système vu de l'intérieur. Elle explique comment peuvent être satisfaits les besoins des acteurs (c'est le COMMENT).
- ✓ Vue d'implémentation : cette vue définit les dépendances entre les modules.
- ✓ Vue des processus : c'est la vue temporelle et technique, qui met en œuvre les notions de tâches concurrentes, stimuli, contrôle, synchronisation, etc.
- ✓ Vue de déploiement : cette vue décrit la position géographique et l'architecture physique de chaque élément du système (c'est l'OU).

2. Les diagrammes d'UML

a- Les diagrammes de cas d'utilisation

Ces diagrammes sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation.

Les cas d'utilisation permettent de traduire les spécifications fonctionnelles d'utilisation du système. Ils sont décrits sous la forme d'actions et de réactions le comportement du système du point de vue d'un utilisateur et représentent un ensemble de séquences d'action réalisées par le système et produisant un résultat observable pour un acteur particulier.

b- Le diagramme de séquence :

C'est la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

On montre ces interactions dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

Les périodes d'activité des classes sont symbolisées par des rectangles.

Le diagramme de séquences est un diagramme d'interaction UML. Il représente les échanges des messages entre les objets du point de vue temporel et permet de détailler un diagramme de cas d'utilisation en décrivant un scénario. Il se base sur les concepts suivants:

- Objet : description d'un objet du monde réel (une personne ou une chose).
- Message : c'est le véhicule de la communication entre les objets.

c- Diagramme de classe

Les diagrammes de classes sont la représentation de la structure statique en termes de classes et de relations. Les objets sont les instances des classes et les liens les instances des relations.

Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Les cas d'utilisation ne réalisent donc pas une partition¹ des classes du diagramme de classes. Un diagramme de classes n'est donc pas adapté (sauf cas particulier) pour détailler, décomposer, ou illustrer la réalisation d'un cas d'utilisation particulier.

II. Diagramme de cas d'utilisation générale

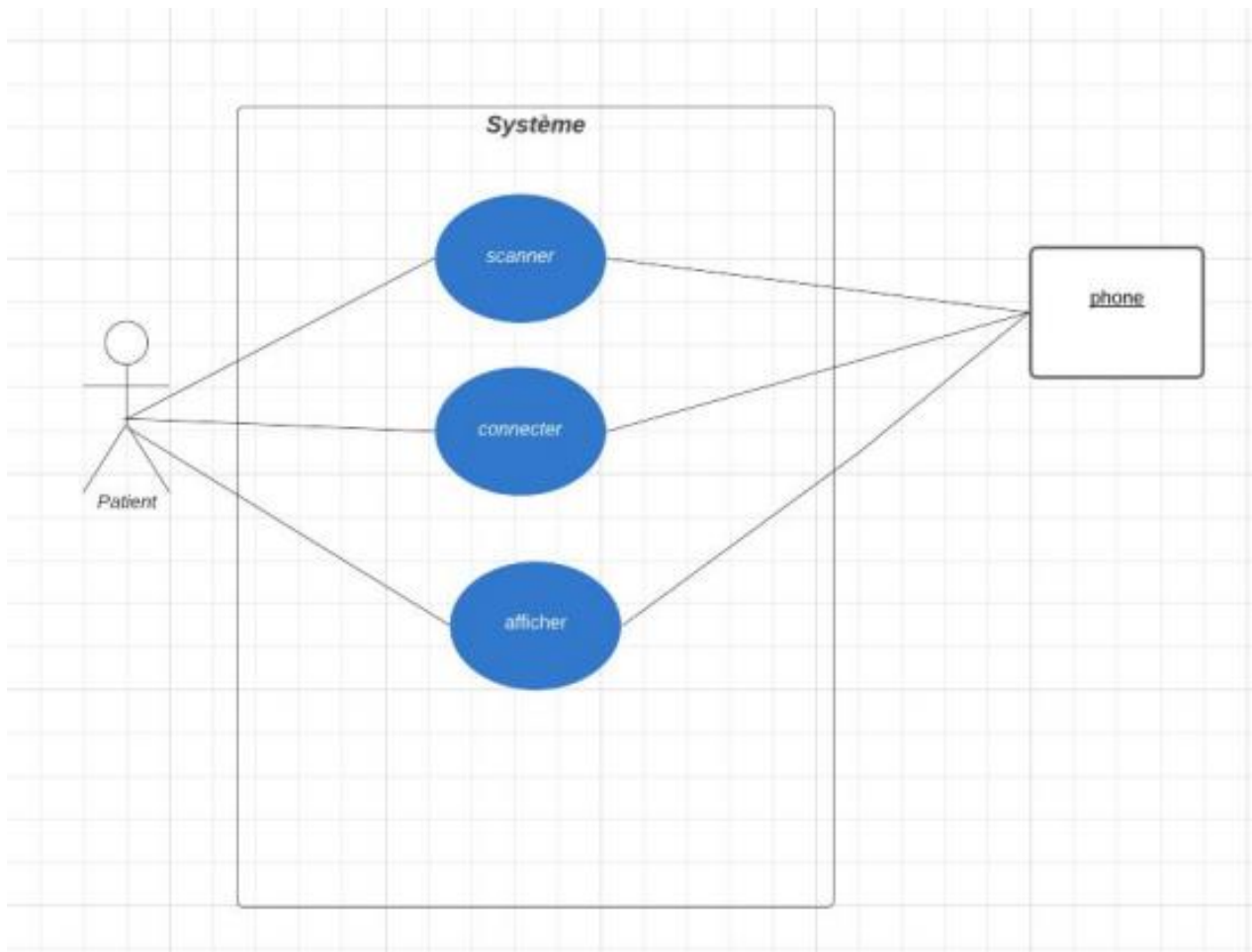


Figure 5: Diagramme de cas d'utilisation générale

Le cas d'utilisations préliminaire ou global donné par la figure 2 constitue l'ensemble des fonctionnalités offertes par notre application et permet d'exprimer les différentes possibilités d'interaction entre les acteurs et le système. Dans ce qui suit, nous allons se limiter à la description détaillée des cas d'utilisation les plus importants.

Notre application permet a l'utilisateur de visualiser ses données vitaux on tant que des services citant son taux pression artérielle, son taux de saturation pulsée de l'hémoglobine en oxygène, sa température son rythme cardiaque sous avec leurs caractéristiques Avec l'établissement d'une connexion de son Smartphone avec son montre médicale pour effectuer ce la l'application doit accomplir trois processus avec succès

- 1) Scanner : sa veut dire chercher des appareils dont le Bluetooth est activé.
- 2) Connecter : établir une connexion avec l'appareil détecté lors du processus Scan.
- 3) Afficher : affiché les services et caractéristiques.

III. Diagramme de « scan »

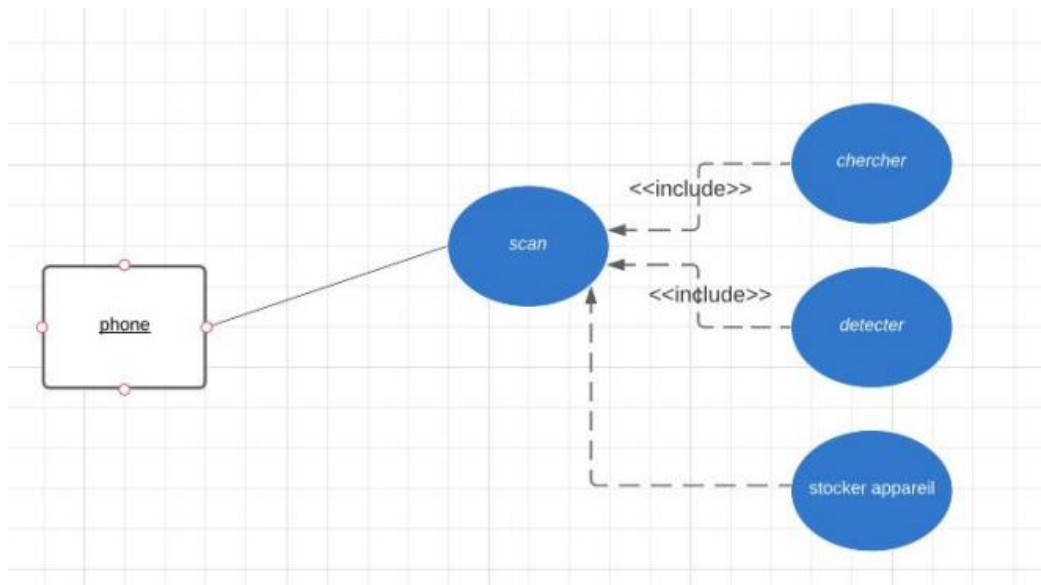


Figure 6: Diagramme de scan

L'utilisateur qui a un Smartphone initialise le scan, ce processus consiste à faire une recherche et détection des appareils dont le Bluetooth est activé.

Tableau 2: Etude textuelle du diagramme de cas d'utilisation « scan »

Cas d'utilisation	Scan
Pré condition	Bluetooth active du périphérique
Post condition	Détection des appareils
Acteur	Smartphone
Scénario nominale	[Début] Recherche des appareils disponible Détection des appareils [Fin]

IV. Diagramme de cas d'utilisation “connexion”

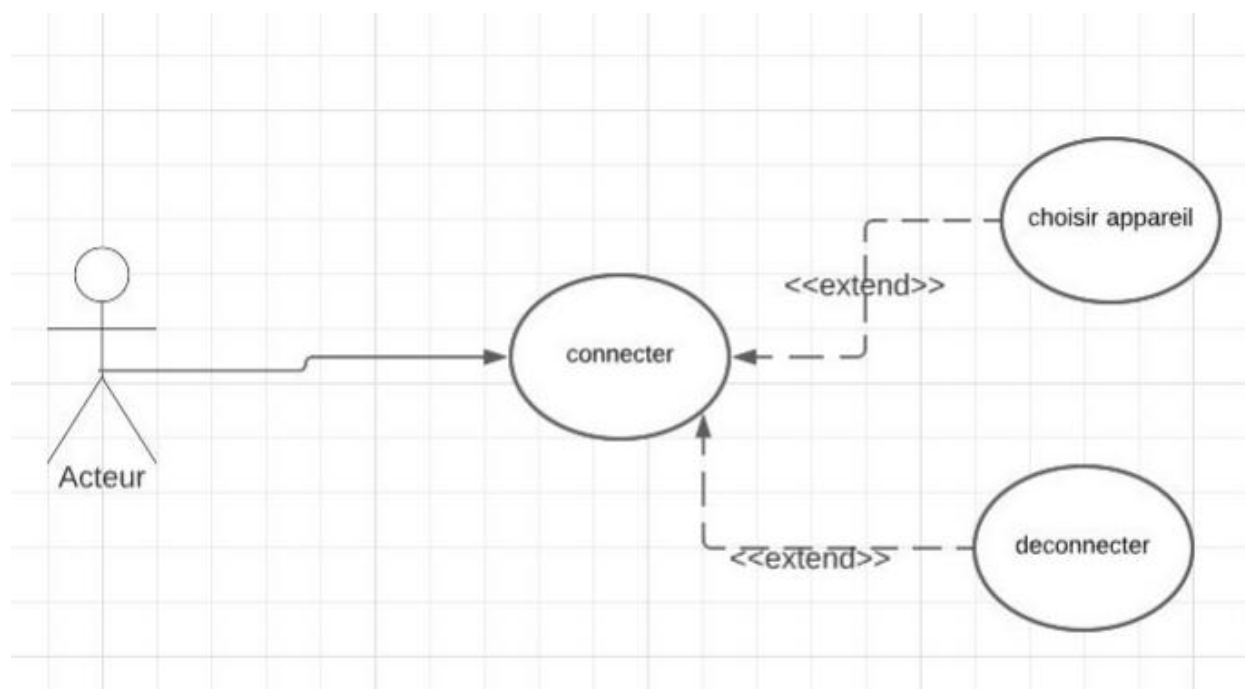


Figure 7: Diagramme “connexion”

Tableau 3: Etude textuelle du diagramme de cas d'utilisation «connexion »

Cas d'utilisation	Connecter
Acteur	Utilisateur équipé d'un Smartphone
Pré condition	Détection de l'appareil en question
Post condition	Établissement d'une connexion
Scénario nominale	[Début] Connexion avec l'appareil désiré Affichage du message “connected to device” [Fin]
Scénario alternatif	Déconnecter de l'appareil choisie en premier lieux : -mise a jour réaliser -réinitialiser la liste des dispositifs détectés -choisir un appareil et connecter a nouveau

L'utilisateur choisi l'appareil avec laquelle il veut connecter, alors la connexion est établie, s'il fait une erreur de choix il peut déconnecté par conséquent le système va gérer un mise a jour et affiche la liste des appareils disponible a nouveau.

V. Diagramme de cas d'utilisation “afficher les données”

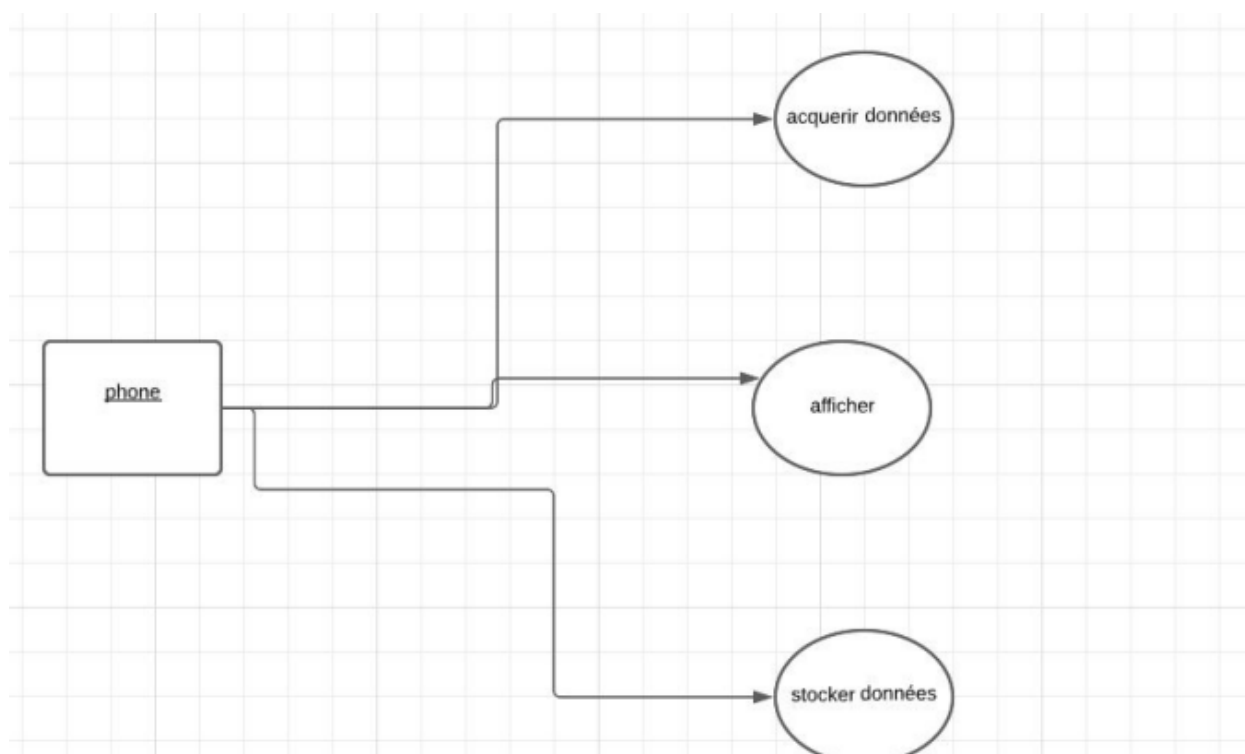


Figure 8: Diagramme de cas d'utilisation “afficher les données”

Tableau 4: Etude textuelle du diagramme de cas d'utilisation « afficher les données »

Cas d'utilisation	Afficher des données
Acteur	Smartphone
Pré condition	On a connecté a un appareil choisi
Post condition	Affichage des données
Scénario nominale	[Début] Acquérir les données afficher les données à l'utilisateur [Fin]

VI. Diagramme de séquence global

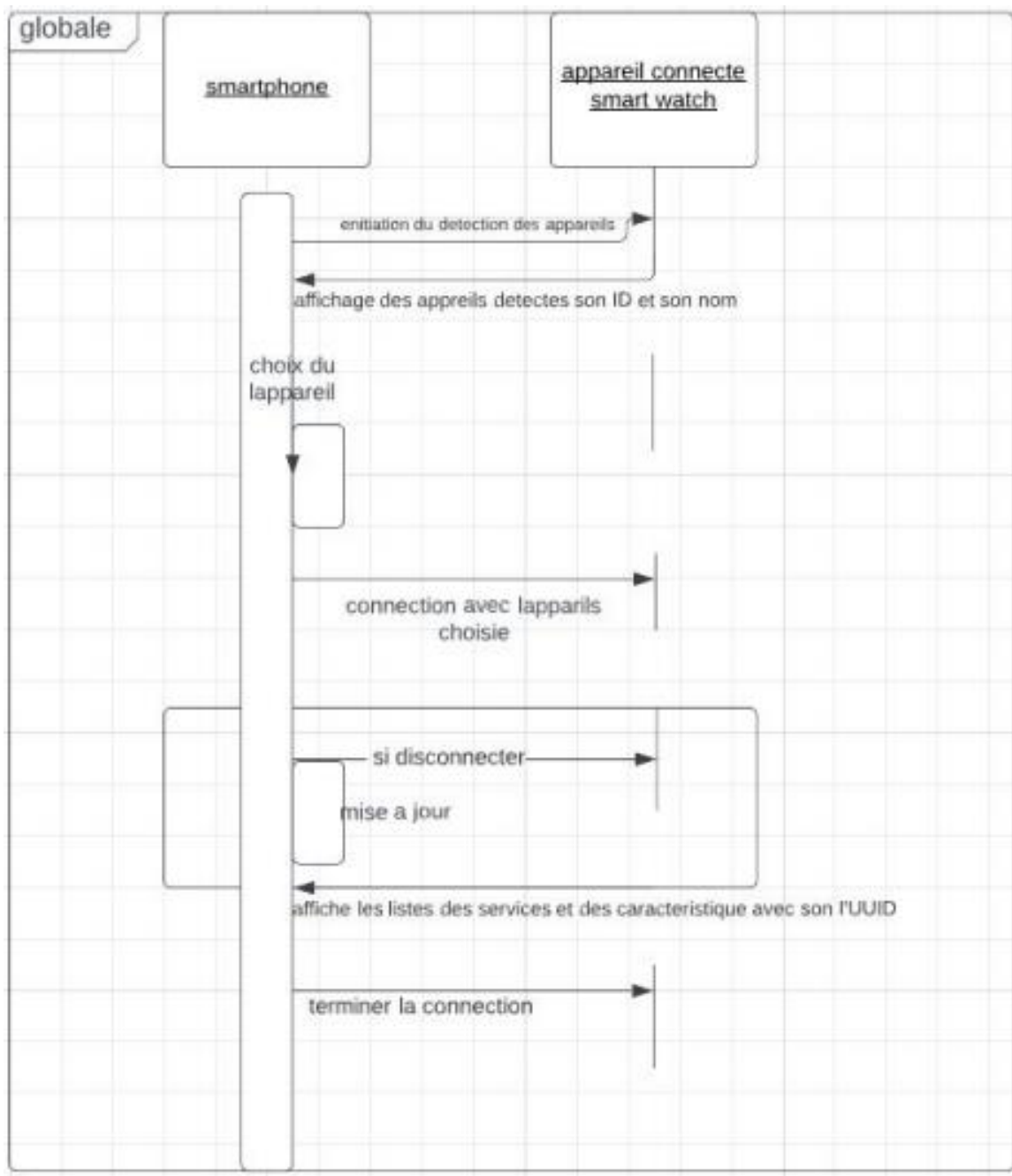


Figure 9: Diagramme de séquence global

Description : L'utilisateur commande l'initiation de détection des appareils pour établir la connexion avec son dispositif désiré, s'il y a des appareils disponibles. En d'autres termes, si le Bluetooth est activé, l'application affiche leurs noms et ID. L'utilisateur choisit son appareil et clique sur «connect» comme conséquence la connexion est établie, dans ce cas l'utilisateur peut voir ses mesures sous forme de services et caractéristiques selon notre projet et en fin l'utilisateur peut terminer la connexion en faisant un retour à la page précédente.

Si l'utilisateur fait une erreur et connecte a un autre appareil il peut déconnecté et le système fait un mise a jour et affiche de nouveau la liste des appareils disponible.

VII. Diagramme de séquence scan

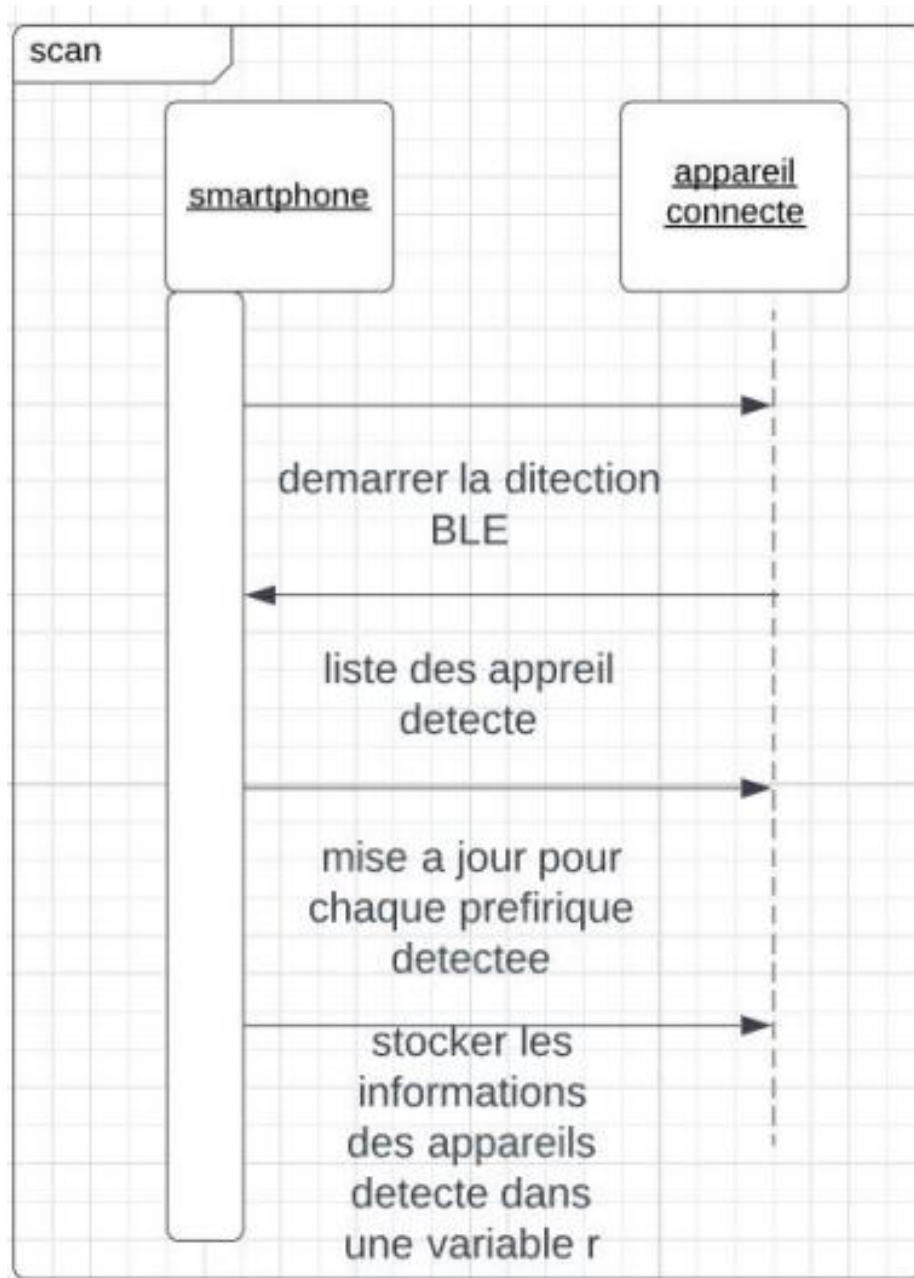


Figure 10: Diagramme de séquence scan

Description : Ce diagramme explique le démarche du processus Scan .En premier lieux l'application commence par la détection des appareils pour simplifie on dire fait la recherche des appareils disponible, On second lieux elle affiche la liste de ses appareils et a chaque fois un nouveau appareil est détecté le système fais sont mise a jour et les stocker.

VIII. Diagramme de séquence “connecter”

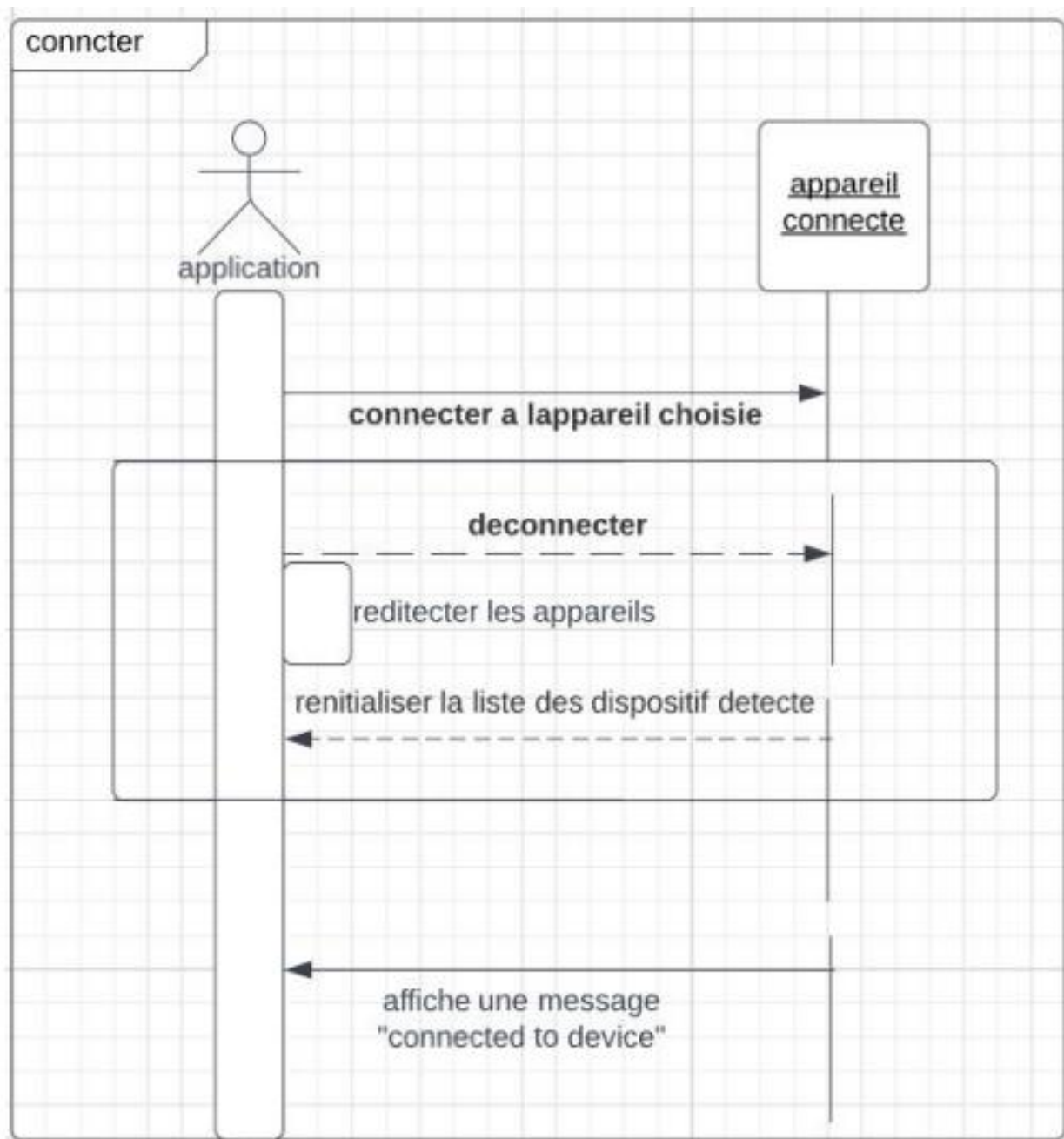


Figure 11: Diagramme de séquence “connecter”

L'utilisateur connecte avec son appareil désiré et l'application lui affiche un message «connected to device». Ou l'utilisateur peut également se déconnecter, dans ce cas le système va subir une mise à jour et affiche de nouveau la liste des appareils détectés.

IX. Diagramme de class

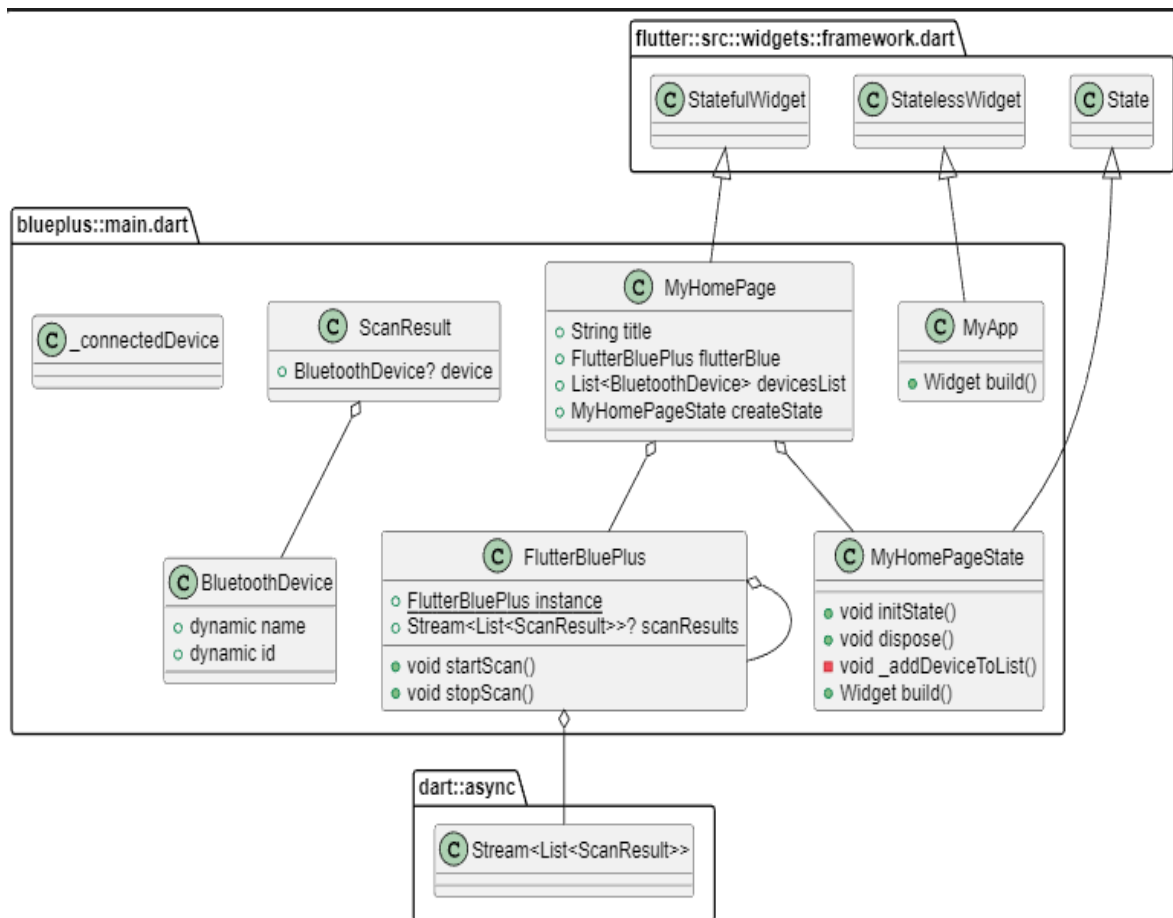


Figure 12: Diagramme de class

Ce diagramme représente les composant clé pour que une application mobile puisse afficher des donnés mesuré par une smartwatch qui est connecté avec la Technology BLE avec le Smartphone de l'utilisateur.

Classe qui affiche la data collecté et extrait par la smart Watch, elle a des attributs privé.

Classe qui est responsable de connecté la montre médicale en utilisant le BLE.

Classe qui représente l'utilisateur : sont nom, prénom.

Classe représente le Smartphone de l'utilisateur.

Classe qui représente les capteur du montre médicale.

Conclusion

Dans ce chapitre on a présenté les besoins fonctionnels et non fonctionnels pour notre application en plus on a vu les diagrammes de cas d'utilisation, des séquences et de classe, accompagnée par une description d'acteurs et des cas d'utilisations.

Dans le chapitre suivant on va expliquer dans quel endroit matériel et logiciels on a effectué le travail.

Chapitre 3 : Réalisation

Introduction

Une fois l'étape de conception terminée, nous commençons, avec ce chapitre, la partie mise en œuvre, qui est une phase finale pour développer notre application.

On décrit l'environnement de développement matériel et logiciel, Nous montrons ensuite quelques captures d'écrans comme exemple illustratif de notre application.

I. Environnement du travail

1. Environnements matériels

a. Ordinateur :

Tableau 5: Caractéristique de l'ordinateur

CPU	AMD Ryzen 555000 U Radeon graphics 2.10GHZ
RAM	8GB
Disque Dur	MVMe-SSD 500GB

b. Smart phone :

Tableau 6: Caractéristique de Smart phone

CPU	Snapdragon
RAM	6GO
système d'exploitation	android 12
version Bluetooth	5.0, A2DP, LE

c. Montre médicale ou montre intelligente T1s

Tableau 7: Caractéristique de montre intelligente T1s

version du BLE	5.0
batterie	280 mAh
capteur (mise en jeu dans notre application)	PPG,spo2

2. Environnement logiciel

Dans cette partie nous allons présenter l'environnement logiciel pour la réalisation de notre solution

a) Visual studio code (version : 1.0)

Visual studio code ou VS Code est un éditeur de code open source développé par Microsoft en 2015 supportant un très grand nombre de langages.

Grâce à des extensions. Il supporte l'auto complétion, la coloration syntaxique, le débogage, et les commandes git.



Figure 13: Visual studio code

b) Flutter

Flutter est un Framework open source développé par Google en 2017 pour créer des applications multiplateformes compilées nativement à partir d'un seul code source.



Figure 14: Flutter

c) Dart :

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web. Il été créer pour remplacer le langage de programmation JavaScript.



Figure 15: Dart

d) Luicid chart :

Application web qui permet aux utilisateurs de dessiner et partager des logigrammes professionnels en leur offrant des modèles pour type de diagramme.



Figure 16: Luicid chart

e) android studio

Le plateforme android studio est un outil nécessaire pour pouvoir développer des applications sous le système d'exploitation qui est créé par Google. Dans l'installation de notre IDE, les packages sont toutes les fonctionnalités d'android sont intégrées dans flutter sous forme de plugin (SDK) android.



Figure 17: android studio

f) Git

Git est un système de contrôle de version qui a été inventé et développé par Linus Torvalds, également connu pour l'invention du noyau Linux, en 2005.

Il s'agit d'un outil de développement qui aide une équipe de développeurs à gérer les changements apportés au code source au fil du temps.

Les logiciels de contrôle de version gardent une trace de chaque changement apporté au code dans un type spécial de base de données.



Figure 18: Git

II. Choix de l'utilisation du flutter

- Produit de Google donc plus sécurisé

- Rapide
- Répond tous les exigences de notre application : réactivité et performance
- Multiplateforme : il permet de développer des applications mobiles avec IOS et android avec une seule base du code et le web

III. Choix de l'utilisation de BLE

BLE est Bluetooth (Low Energy) est un protocole de communication sans fil qui permet des communications entre des périphériques électroniques comme Smartphone et smartwatch à faible puissance à courte portée et en temps réel.

IV. Choix du Bluetooth LE bleu :

Flutter Blue est la bibliothèque la plus utilisée pour la communication Bluetooth en Flutter, peut être Pour les raisons suivantes :

- Mise à jour régulière : Flutter Blue est régulièrement mis à jour pour inclure de nouvelles fonctionnalités et corriger les bogues. Cela assure une compatibilité et une stabilité à long terme.
- Compatibilité multiplateforme : Flutter Blue est compatible avec les plateformes iOS et Android, ce qui en fait un choix pratique pour les développeurs d'applications mobiles multiplateformes.
- Fonctionnalités étendues : Flutter Blue offre de nombreuses fonctionnalités pour la communication avec des périphériques Bluetooth, telles que la découverte des périphériques, la connexion aux périphériques, la lecture et l'écriture de caractéristiques, la notification des changements de caractéristiques, la recherche de services, etc.

Choix de l'utilisation du Visual studio code

- Il supporte le git donc nous pouvons faire tout le travail git à partir de VS studio code. Il a une bonne interface graphique avec diverses options de thème
- Les points forts de VS code est la possibilité de travailler sur plusieurs onglets simultanément ainsi que plusieurs fenêtres, de même c'est très léger et très productif. Aussi ce logiciel est livré avec beaucoup de personnalisation et de support. Ainsi l'accès directement au terminal (le point faible d'Atom).

V. Les Interfaces Graphiques principales:

Dès que l'application est lancée, la page scanning permet à l'utilisateur de scanner les appareils disponibles en cliquant sur l'icône du Bluetooth pour établir la connexion entre lui et sa montre ou appareil désirée.

❖ Page 1 de l'application

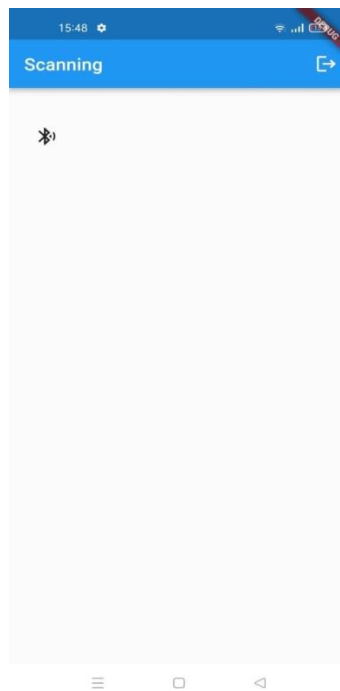


Figure 19: page 1 de l'application

On trouve l'icône du Bluetooth affiché pour commencer le scan

❖ Page 2 de l'application

On trouve les appareils disponibles à connecter

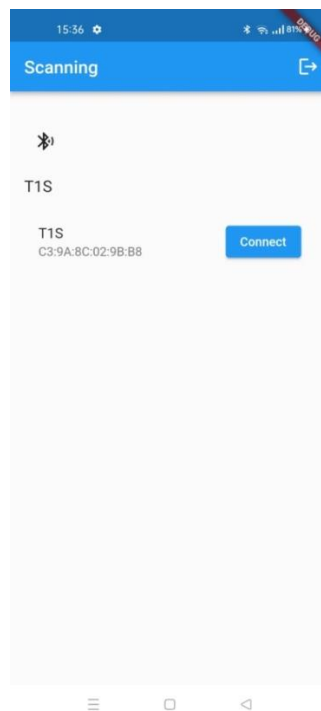


Figure 20: page 2 de l'application

❖ Page 3 de l'application

Qui va afficher les 4 services et leurs caractères.

Mesure de la température : [AB,00,04,FF,86,80,01]

Mesure du rythme cardiaque : [AB,00,04,FF,31,09,01]

Mesure du spo2 : [AB 00,04,FF,31,17,01]

Mesure de la pression artérielle : [AB 00,04,FF,31,33,01]

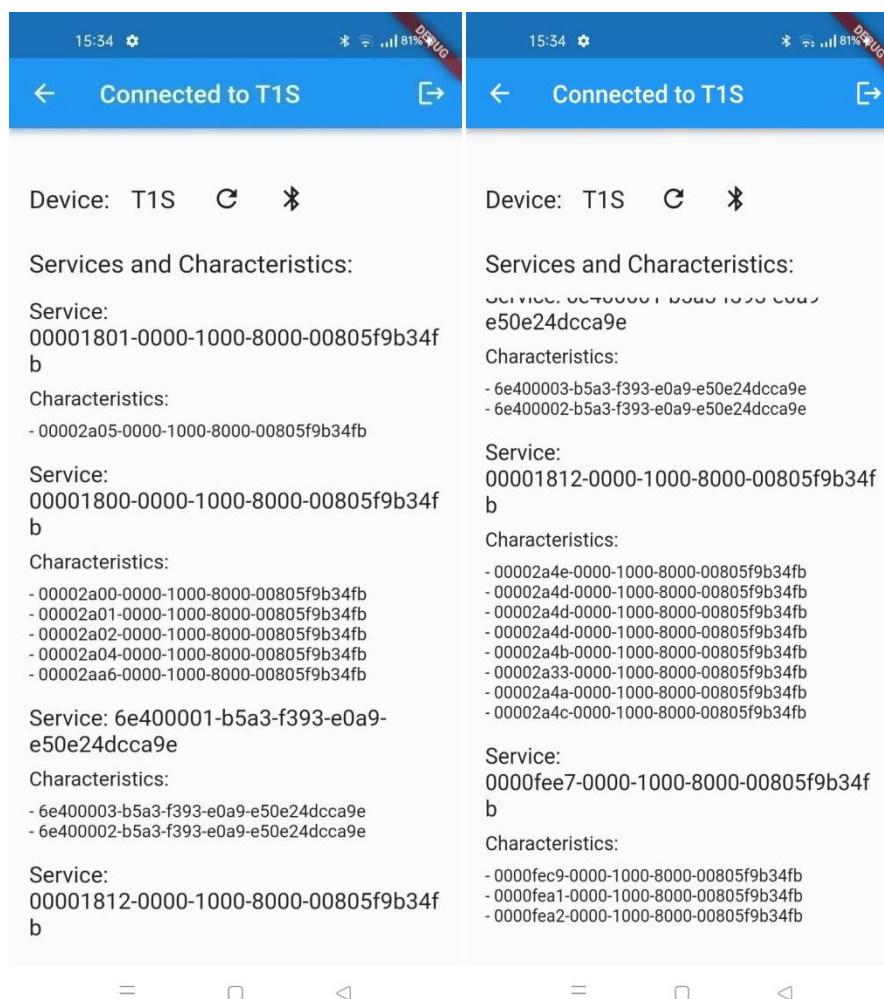


Figure 21: Page 3 de l'application

Conclusion

Dans ce chapitre on a spécifié les environnements du travail en justifiant nos choix et on a montré les interfaces graphiques principales de notre application.

Conclusion générale et perspectives

Dans ce rapport, on a dévoilé différentes étapes de la conception et la réalisation de notre application. Pour concevoir ce travail on a présenté premièrement le cadre de ce projet, puis on a analysé l'état de l'art et on a dégagé des déficiences existantes.

En second, on a montré la phase de conception. Finalement, on a traité toutes les phases nécessaires à la réalisation de cette application dont l'objectif est de permet aux utilisateurs de collecter et de suivre facilement leurs données de santé en toute crédibilité et en temps réel.

Enfin cette application peut être améliorée en ajoutant plus de fonctionnalité comme migrer l'application vers le serveur Cloud de la société, ce qui nous permettra d'attirer plus les utilisateurs, De plus, nous avons l'intention d'intégrer des modèles d'apprentissage automatique 'Machine Learning ' dans l'application. Le modèle analysera les mesures prises par l'utilisateur. Grâce à l'intelligence artificielle, l'application pourra interpréter ces données et fournir une évaluation de la santé du patient.

Cela permettra aux utilisateurs de recevoir des informations précises sur leur santé et de détecter d'éventuels risques. Pour finir un design plus moderne dans le but d'une expérience utilisateur plus intuitive.

Il est important de dire que Ce stage nous donner l'occasion de mettre en pratique nos compétences acquises au cours de nos études universitaires et parallèlement d'acquérir de nouvelles connaissances dans le domaine du développement mobile. En outre on a trouvé des contraintes que nous étions obligés à les surpasser pour l'accomplissement de ce projet comme le développement avec une langage de programmation complètement nouvelle, la découverte du monde IOT et ses technologies de transmission comme Bluetooth. C'est aussi un défi qui nous a permis d'évoluer en termes de gestion du temps et d'intelligence émotionnelle.

Bibliographie

<https://flutter.dev/> : définition du flutter

<https://www.spiceworks.com/tech/iot/articles/what-is-bluetooth-le/> : information BLE

<https://www.fondationdelavenir.org/e-sante-definition/> : actualité de e-commerce

<https://www.oracle.com/fr/internet-of-things/what-is-iot/> : définition de l'IOT

ANNEXE

Etape de réalisation

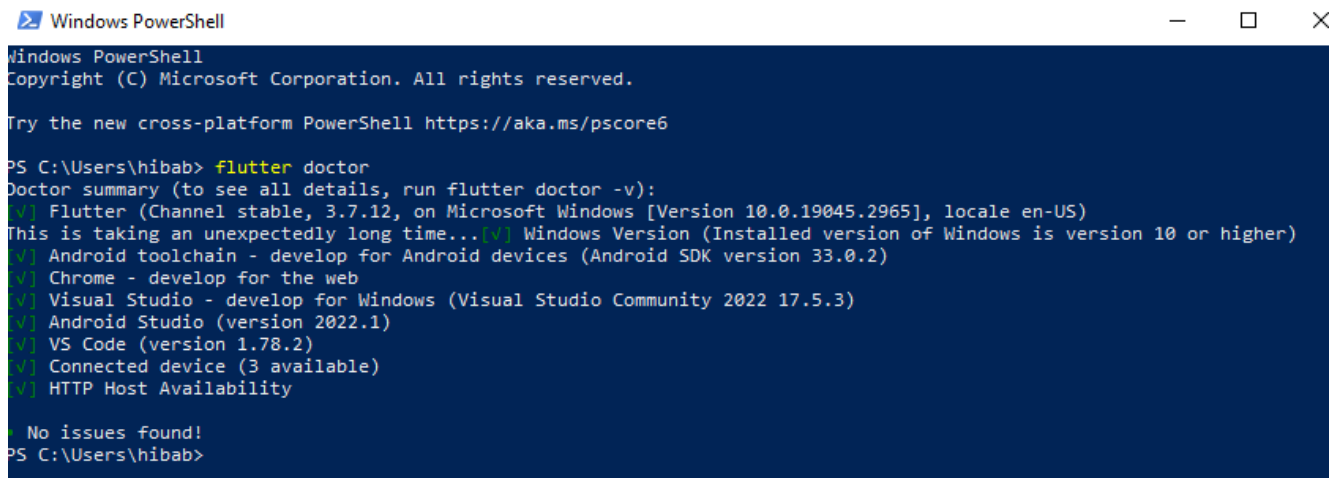
-Installation du flutter

Installation du flutter SDK

Installation du Dart SDK

Mise a jour du son localisation

Tester la commande Flutter Doctor pour vérifier s'il est bien installé.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\hibab> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.7.12, on Microsoft Windows [Version 10.0.19045.2965], locale en-US)
This is taking an unexpectedly long time...[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Community 2022 17.5.3)
[✓] Android Studio (version 2022.1)
[✓] VS Code (version 1.78.2)
[✓] Connected device (3 available)
[✓] HTTP Host Availability

No issues found!
PS C:\Users\hibab>
```

-Android set up

Installation du l'android studio

Installation du SDK du l'android studio

-Set up de l'appareil android utilisé (Smartphone)

Changement du mode d'utilisation a développeur

Autorisé la débogage sans fil (USB)

Installation du Google USB Driver

Tester la commande flutter devices dans le terminale pour vérifier que Flutter reconnait mon appareil

```
Command Prompt - Flutter d
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\21692>Flutter devices
4 connected devices:

RMX3491 (mobile) • e9034d6d • android-arm64 • Android 11 (API 30)
Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.22621.1702]
Chrome (web) • chrome • web-javascript • Google Chrome 113.0.5672.93
Edge (web) • edge • web-javascript • Microsoft Edge 112.0.1722.68

C:\Users\21692>
```

-Installation du powershell

Pour automatisé la gestion du system

-Installation du Git

Pour traquer tout les fichiers de mon projet lors de son évolution.

Explication du code

-Script du scanning des appareils

Cette fonction utilise la bibliothèque FlutterBlue pour détecter les dispositifs Bluetooth et mettre à jour la liste des dispositifs en temps réel.

```
void _scanForDevices() {  
  flutterBlue.startScan();  
  flutterBlue.scanResults.listen((results) {  
    setState(() {  
      for (ScanResult r in results) {  
        if (!_devices.contains(r.device)) {  
          _devices.add(r.device);  
        }  
      }  
    });  
  });  
}
```

setState() { ... }:: cette ligne met à jour l'état de la page 1 en ajoutant les dispositifs Bluetooth détectés.

For (ScanResult r in results) { ... }:: cette ligne de code utilise une boucle pour parcourir chaque résultat de la détection des dispositifs Bluetooth détectés. La variable r contient les informations du périphérique Bluetooth détecté.

if (!_devices.contains(r.device)) { _devices.add(r.device); }:: cette ligne de code vérifie si le dispositif Bluetooth détecté n'est pas déjà présent dans la liste « devices ». Si c'est le cas, le dispositif est ajouté à la liste.

-Script du connections avec l'appareil choisie

cette partie du code établit une connexion Bluetooth avec l'appareil sélectionné et navigue vers la page 2 de l'application en passant le dispositif sélectionné en tant que paramètre.

```
void _connectToDevice(BuildContext context, BluetoothDevice device) async {  
  await device.connect();  
  setState(() {  
    _selectedDevice = device;  
  });  
  Navigator.push(  
    context,  
    MaterialPageRoute(  
      builder: (context) => Page2(device: _selectedDevice!)), // MaterialPageRoute  
    )  
}  
  
void _disconnectFromDevice() async {  
  await _selectedDevice?.disconnect();  
  setState(() {  
    _selectedDevice = null;  
    _devices.clear();  
  });  
}
```

if (device != null) { ... }: cette ligne de code vérifie que le dispositif sélectionné n'est pas nul (nul). On autre terme il ya une détection d'une appareil ..

await device.connect(); cette ligne de code établit une connexion avec le dispositif Bluetooth sélectionné

setState() { _selectedDevice = device; }); permet de garder une trace du dispositif sélectionné même si l'état de la page change.

Navigator.push(...); cette ligne de code utilise l'objet Navigator pour naviguer vers la page 2 de l'application en passant le appareil choisie en tant que paramètre.

La classe MaterialPageRoute est utilisée pour spécifier la nouvelle route à afficher et la classe Page2 est utilisée pour créer la nouvelle page.

Cette fonction _disconnectFromDevice() est utilisée afin de se déconnecter d'un dispositif Bluetooth connecté auparavant, mettre à jour l'état de la page 1 de l'application et mettre à jour la liste des dispositifs détectés.

Script d'affichage des données détecté du dispositif choisie

```

), // Text
Expanded(
child: ListView.builder(
  itemCount: _devices.length,
  itemBuilder: (context, index) {
    var name = _devices[index].name;
    var id = _devices[index].id.toString();

    return ListTile(
      title: Text(name),
      subtitle: Text(id),
      trailing: ElevatedButton(
        onPressed: () =>
          | _connectToDevice(context, _devices[index]),
        child: const Text('Connect'),
      )); // ElevatedButton // ListTile
    }, // ListView.builder
  ), // Expanded
),
// Column
/ Padding // Scaffold

```

Ce block de code expose une liste des services et de caractéristiques pour un périphérique Bluetooth connecté. Il contient plusieurs éléments visuels :

Une ligne d'en-tête avec le nom du périphérique connecté et une icône Bluetooth.

Un bouton de mise à jour qui permet de redécouvrir les services.

Une section intitulée "Services and Characteristics" avec une liste de services. Pour chaque service, un titre avec l'UUID du service, une liste de caractéristiques pour ce service.

On utilise une `ListView.builder` pour afficher la liste de services et de caractéristiques. Pour chaque service, une nouvelle colonne est créée avec le titre et la liste de caractéristiques associées. Les caractéristiques sont affichées en utilisant la méthode `map()`, qui transforme chaque caractéristique en un widget `Text` avec l'UUID de la caractéristique.

Les Plugin utilisées dans ce travail

Plugin du PlantUML

Plugin du Dart

Plugin de FlutterBlue

Plugin de l'android studio