

# PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL



Année universitaire : 2022/2023

Encadré par : Pr. Mahmoud EL HAMLAOUI

▪ **Filiere :SESnum**

**Créé par : Amina AIT LAFKIH**

**Nouhaila Akhziz**

**Idriss Otmani**

# Table de matière

---

## Table des matières

PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL.....	1
.....Table de matière	
.....	2
Introduction.....	4
I. Analyse et Conception de Projet :.....	5
1. Exigences fonctionnelles et non fonctionnelles :.....	5
1.1. Exigences fonctionnelles :.....	5
1.2. Exigences non fonctionnelles :.....	6
2. Acteurs :.....	6
3. Conception et modélisation UML :.....	7
3.1. Diagramme de cas d'utilisation :.....	7
3.2. Diagramme de classe :.....	7
II. Gestion de projet :.....	8
1. Choix de la méthodologie :.....	9
La méthodologie <b>Scrum</b> est l'approche agile la plus utilisée des approches agiles existante et il est simple à comprendre.....	9
La méthodologie SCRUM est composée de quatre phases (on parle aussi de réunion) :.....	9
✓ Planification du Sprint.....	9
✓ Revue de Sprint.....	9
✓ Rétrospective de Sprint.....	9
✓ Mêlée quotidienne.....	9
2. Planification du projet :.....	9
2.1. Product backlog :.....	9
2.2. .Sprint:.....	10
2.3. Bilan de sprint :.....	11
3. Outils de collaboration :.....	15
3.1. Bitbucket :.....	15
3.2. Github :.....	16

## PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

III.	Développement et réalisation technique : .....	17
1.	<b>Hardware</b> : .....	17
	<b>MAX30102</b> : Nous allons utiliser le MAX30102 pour mesurer la fréquence cardiaque. Le MAX30102 est un capteur très polyvalent, capable de mesurer non seulement la fréquence cardiaque et le niveau d'oxygène dans le sang, mais aussi la température corporelle.....	
2.	<b>Software</b> : .....	18
IV.	Conclusion : .....	20
V.	Lien Github : .....	21

# Introduction

---

Les SMARTWATCH sont de minuscules appareils qui font beaucoup de choses sympas, et les gens les aiment vraiment ! Tout a commencé avec une entreprise appelée Pebble, et maintenant beaucoup d'autres essaient de fabriquer la meilleure montre connectée.

Une smartwatch est cool parce qu'elle est belle et qu'elle fait des choses cool. Si vous souhaitez rendre votre smartwatch encore plus cool, vous pouvez créer la vôtre avec des fonctionnalités spéciales rien que pour vous.

Mais qu'est-ce qui rend vraiment une montre intelligente cool ? C'est certainement son apparence et ses fonctionnalités. La meilleure façon d'obtenir des fonctionnalités personnalisées sur une montre intelligente est d'en concevoir et de la construire une par soi-même.

Ce projet a pour but de fabriquer une montre dont les fonctions sont de voir le temps, la date et un gyroscope. Une fonctionnalité intéressante de notre montre est qu'elle intègre un capteur de pulsations cardiaques. Cette fonctionnalité revêt une importance particulière dans le domaine médical. En mesurant en temps réel les pulsations cardiaques de l'utilisateur, la montre peut fournir des informations précieuses sur sa santé et son bien-être. Cela permet notamment de surveiller l'activité cardiaque, de détecter d'éventuelles anomalies ou variations, et d'offrir des fonctionnalités de suivi et d'analyse de la fréquence cardiaque. Cette capacité à agir comme un capteur de pulsations cardiaques fait de notre montre un outil polyvalent et utile pour ceux qui souhaitent surveiller leur santé de manière proactive.

Au bout de ce projet,, nous allons opter pour une méthode agile de planification, qui nous permettra de travailler de manière itérative et d'ajuster notre approche au fur et à mesure. Cela nous permettra de nous adapter aux éventuels changements et d'améliorer constamment notre montre intelligente tout au long du processus.

# I. Analyse et Conception de Projet :

---

## 1. Exigences fonctionnelles et non fonctionnelles :

Dans ce chapitre, nous allons décrire les besoins fonctionnels et non fonctionnels nécessaires pour la réalisation du projet. Ensuite, nous utiliserons les diagrammes de modélisation UML. L'objectif de cette étude est de proposer l'architecture du système la plus adaptée à notre solution.

### 1.1. Exigences fonctionnelles :

L'analyse et la documentation des exigences jouent un rôle essentiel dans le cycle de vie du développement logiciel. Dans le but de suivre les bonnes pratiques de documentation dans notre projet, nous avons mis en œuvre les exigences fonctionnelles suivantes :

#### ➤ **\*Simple utilisateur :**

- **Affichage de l'heure** : La montre se connecte à un serveur NTP pour obtenir l'heure et la date actuelles.
- **Affichage de date** : La montre montre le jour, le mois actuelle .
- **Affichage de pulsation cardiaque** : L'exemple affichera votre fréquence cardiaque dès que vous placerez votre doigt sur le capteur pendant suffisamment longtemps.
- **Affichage de gyroscope** : capteur de position angulaire.
- **Affichage de la météo** : Les données de température sont obtenues à partir de l'API Yahoo Weather.

#### ➤ **\*Admin: (quelques fonctionnalités supplémentaires) :**

- **Contrôle d'autres appareils** : La montre se connecte au serveur Blynk et utilise sa fonction de pont pour contrôler directement d'autres modules relais ESP8266. Cela permet à l'utilisateur de contrôler à distance des appareils en appuyant sur les boutons de la montre.

## 1.2. Exigences non fonctionnelles :

Pour mettre en place une solution adéquate aux attentes des concepteurs des architectures dynamiques, on doit prendre en considération les contraintes qui peuvent caractériser ce système. Notre application doit nécessairement assurer les besoins suivants :

### + Fiabilité :

Les résultats apportés par l'application doivent être fiables et reflètent effectivement l'état de la base au moment de son interrogation.

### + Disponibilité :

L'application doit être disponible à tout instant pour être utilisée, l'accès aux données doit être permanent et sans coupure, et doit être facilement accessible via n'importe quel ordinateur dans le réseau de l'entreprise.

### + Performance :

- L'application doit supporter un grand nombre de connexions simultanées.
- Le lancement de l'application et le temps de réponse aux actions de l'utilisateur doivent être optimisés.

### + Sécurité :

- Les données et les informations entrées par l'utilisateur doivent être bien sécurisées.

### + Extensibilité :

Dans le cadre de ce travail, l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

## 2. Acteurs :

**1. Utilisateur :** L'utilisateur est la personne qui porte la smartwatch et interagit avec ses fonctionnalités. L'utilisateur peut utiliser la smartwatch pour afficher l'heure, les notifications, les données de santé, contrôler des applications, etc.

**2. Développeurs d'applications :** Les développeurs d'applications sont responsables de la création et de la programmation des applications spécifiques à la smartwatch. Ils conçoivent et développent des applications qui tirent parti des fonctionnalités de la montre, telles que le suivi de la condition

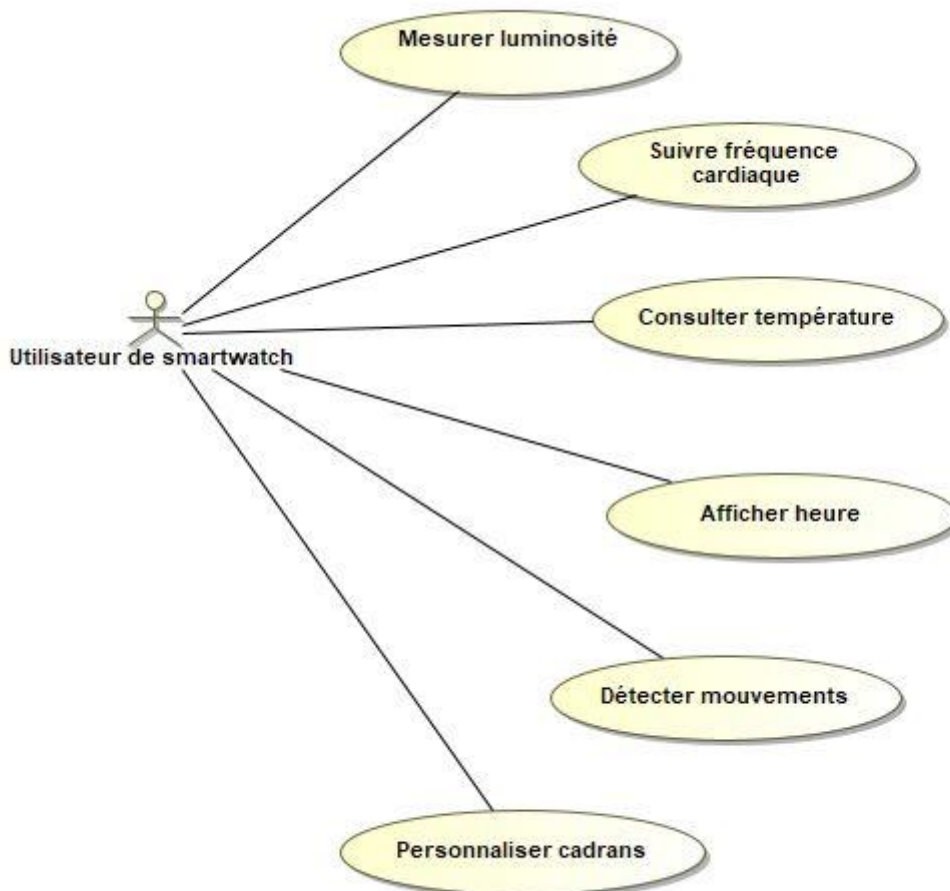
## PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

physique, la gestion des appels, les applications de productivité, les jeux-Donc , ce smartwatch doit etre programmable sans fils , avec d autre mots ,on peut ajouter des nouveaux exigences selon l utilisateur.

### 3. Conception et modélisation UML :

#### 3.1. Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation est un des diagrammes UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel.

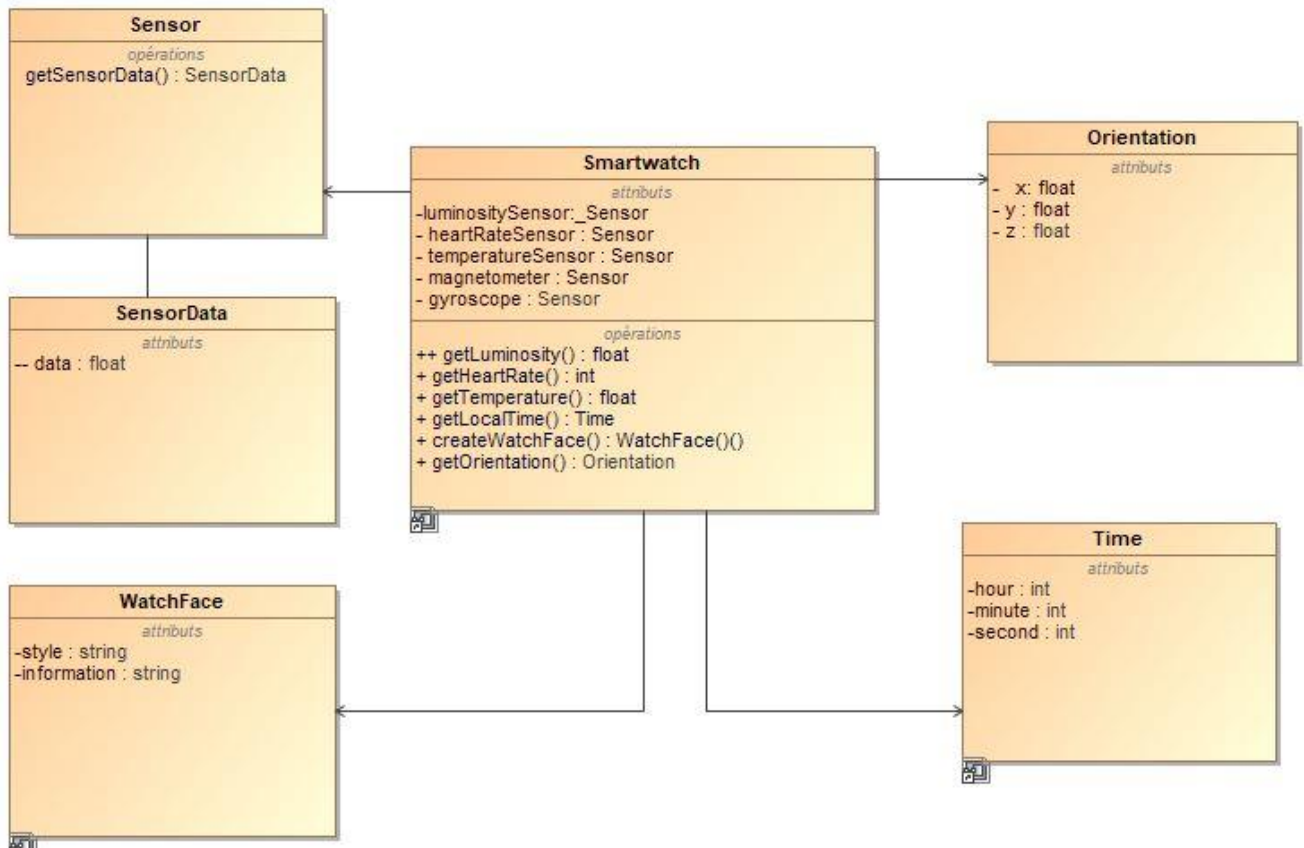


#### 3.2. Diagramme de classe :

Le diagramme de classe est l'un des outils essentiels de la modélisation orientée objet. Il permet de représenter visuellement la structure et les relations entre les classes d'un système logiciel. Le diagramme de classe offre une vue statique du système, en mettant l'accent sur les entités principales, leurs attributs et leurs interactions. Il aide les développeurs et les concepteurs à comprendre la conception globale du système, à communiquer efficacement entre eux et à détecter les éventuels problèmes de conception. Grâce à ses éléments tels que les classes, les associations, les héritages et les multiplicités, le diagramme de classe offre une représentation claire et concise des

## PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

entités du système et de leurs relations, fournissant ainsi une base solide pour la mise en œuvre du logiciel.



Dans ce chapitre, nous avons présenté les différents besoins. Tout d'abord, nous avons commencé par l'identification des besoins fonctionnels et non fonctionnels. Ensuite, nous avons procédé à l'explication et à l'illustration de chaque fonctionnalité en se basant sur les diagrammes UML qui constituent la base sur laquelle s'appuie tout projet.

Le chapitre suivant présente les outils utilisés afin de réaliser ce projet.

## II. Gestion de projet :



## 1. Choix de la méthodologie :

La méthodologie **Scrum** est l'approche agile la plus utilisée des approches agiles existante et il est simple à comprendre.

La méthodologie SCRUM est composée de quatre phases (on parle aussi de réunion) :

- ✓ Planification du Sprint.
- ✓ Revue de Sprint.
- ✓ Rétrospective de Sprint.
- ✓ Mêlée quotidienne.

La planification du sprint : Il s'agit d'une réunion permettant la définition des objectifs du sprint.

La revue de sprint : Il s'agit d'une réunion permettant de faire un point sur le sprint.

La Rétrospective de Sprint : Il permet de faire un point sur le sprint en lui-même (productivité, efficacité, qualité...) afin de pouvoir s'améliorer pour les prochains sprints.

La Mêlée quotidienne : Il est organisé, chaque jour, une réunion d'avancement afin de s'assurer que les objectifs du sprint seront tenus.

## 2. Planification du projet :

### 2.1. Product backlog :

Afin de planifier nos users stories on utilise Jira :



# PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

Smartwatch3  
Projet logiciel

PLANIFICATION

- Feuille de route
- Backlog**
- Tableau

DÉVELOPPEMENT

- Code
- Pages de projet
- Ajouter un raccourci
- Paramètres du projet

Vous faites partie d'un projet géré par l'équipe

En savoir plus

Projets / Smartwatch3

## Backlog

Q AL NA OI Epic Analyses

▼ Backlog (10 tickets) 0 0 0 Créer un sprint

SMRTWTCH3-1	Afficher l'heure actuelle sur la Smartwatch	À FAIRE	AL
SMRTWTCH3-2	afficher la date et le mois actuels sur la smartwatch	À FAIRE	NA
SMRTWTCH3-3	Mesurer ma fréquence cardiaque à l'aide du capteur MAX30102 de la smartwatch	À FAIRE	OI
SMRTWTCH3-4	Mesurer ma température corporelle à l'aide du capteur MAX30102 de la smartwatch	À FAIRE	AL
SMRTWTCH3-5	Afficher ma fréquence cardiaque en temps réel sur l'écran de la smartwatch	À FAIRE	AL
SMRTWTCH3-6	utiliser la boussole numérique de la smartwatch pour connaître ma direction (Nord, Sud, Est...	À FAIRE	NA
SMRTWTCH3-7	afficher les informations de localisation en utilisant le gyroscope de la smartwatch.	À FAIRE	NA
SMRTWTCH3-8	utiliser les entrées tactiles de la smartwatch pour interagir avec l'interface utilisateur	À FAIRE	OI
SMRTWTCH3-9	Obtenir les informations météorologiques en temps réel sur la smartwatch	À FAIRE	AL
SMRTWTCH3-10	contrôler d'autres appareils connectés en utilisant la fonction de contrôle à distance de la ...	À FAIRE	OI

+ Créer un ticket

## 2.2. .Sprint:

### ➤ Sprint 1 :

## Backlog

Q AL NA OI Epic Analyses

▼ Tableau Sprint 1 10 mai – 17 mai (2 tickets) 0 0 0 Démarrer un sprint

SMRTWTCH3-1	Afficher l'heure actuelle sur la Smartwatch	-	À FAIRE	AL
SMRTWTCH3-2	afficher la date et le mois actuels sur la smartwatch	-	À FAIRE	NA

+ Créer un ticket

### ➤ Sprint 2 :

## PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

▼ Tableau Sprint 2 10 mai – 17 mai (2 tickets) 0 0 0 Démarrer un sprint ...

SMRTWTCH3-9	Obtenir les informations météorologiques en temps réel sur la smartwatch	À FAIRE	AL
SMRTWTCH3-8	utiliser les entrées tactiles de la smartwatch pour interagir avec l'interface utilisateur	À FAIRE	OI

+ Créer un ticket

### ➤ Sprint 3 :

▼ Tableau Sprint 3 17 mai – 24 mai (3 tickets) 0 0 0 Démarrer un sprint ...

SMRTWTCH3-3	Mesurer ma fréquence cardiaque à l'aide du capteur MAX30102 de la smartwatch	À FAIRE	OI
SMRTWTCH3-4	Mesurer ma température corporelle à l'aide du capteur MAX30102 de la smartwatch	À FAIRE	AL
SMRTWTCH3-5	Afficher ma fréquence cardiaque en temps réel sur l'écran de la smartwatch	À FAIRE	AL

+ Créer un ticket

### ➤ Sprint 4:

▼ Tableau Sprint 4 24 mai – 31 mai (3 tickets) 0 0 0 Démarrer un sprint ...

SMRTWTCH3-10	contrôler d'autres appareils connectés en utilisant la fonction de contrôle à distance de...	-	À FAIRE	OI
SMRTWTCH3-6	utiliser la boussole numérique de la smartwatch pour connaître ma direction (Nord, Sud, Est...		À FAIRE	NA
SMRTWTCH3-7	afficher les informations de localisation en utilisant le gyroscope de la smartwatch.		À FAIRE	NA

+ Créer un ticket

## 2.3. Bilan de sprint :

# PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

## Sprint 1 :

AL
NA
+

REGROUPER PAR Aucun Ar

A FAIRE 2 TICKETS

Afficher l'heure actuelle sur la Smartwatch

SMRTWTCH3-1 AL

afficher la date et le mois actuels sur la smartwatch

SMRTWTCH3-2 NA

EN COURS

FINI ✓

## Sprint 2 :

### Jour 1 :

AL
OI
+

REGROUPER PAR Aucun Analyses

A FAIRE 2 TICKETS

Obtenir les informations météorologiques en temps réel sur la smartwatch

SMRTWTCH3-9 AL

utiliser les entrées tactiles de la smartwatch pour interagir avec l'interface utilisateur

SMRTWTCH3-8 OI

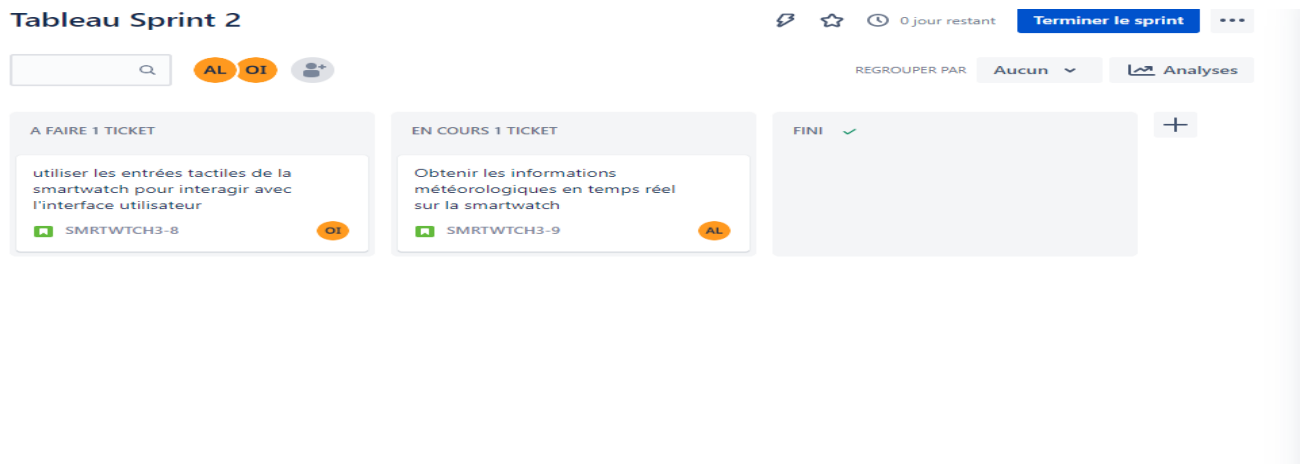
EN COURS

FINI ✓

# PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

## Jour 2 :

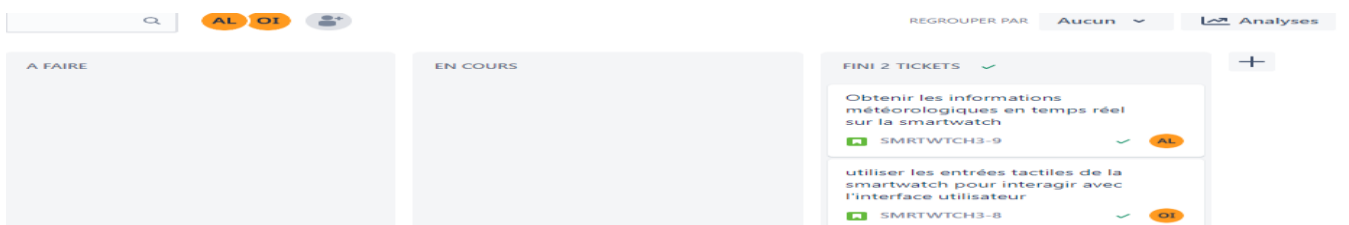
### Tableau Sprint 2



## Jour 4 :



## Jour 5 :



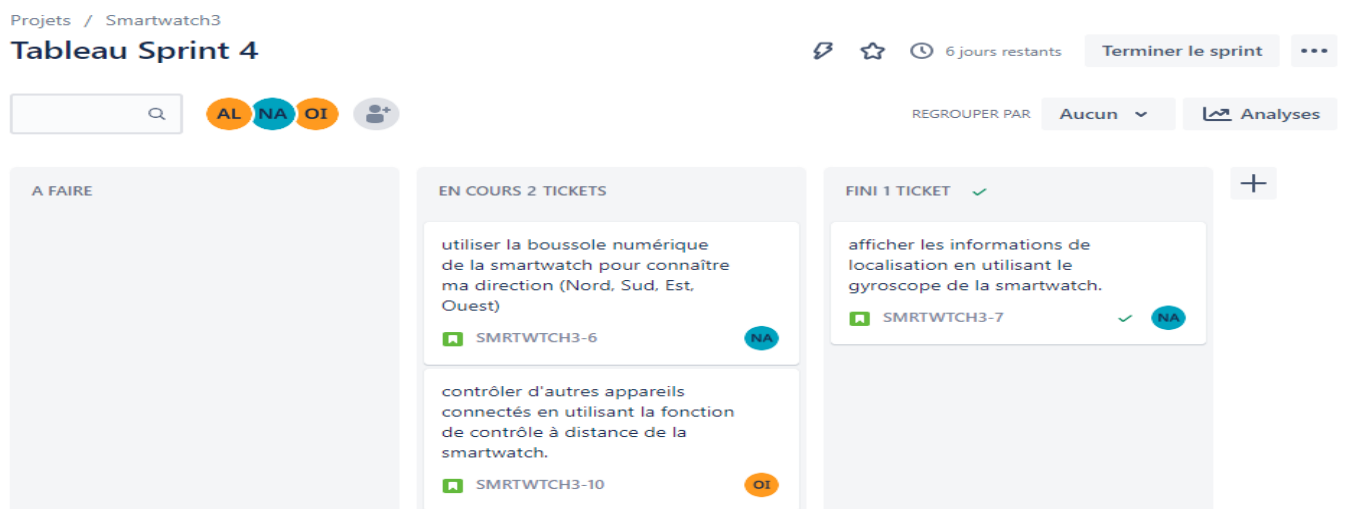
# PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

De même pour les autres sprints :

## Sprint 3 :



## Sprint 4 :



### 3. Outils de collaboration :

#### 3.1. Bitbucket :



Source: <https://bitbucket.org/>

Bitbucket est un service web d'hébergement et de gestion de développement logiciel utilisant des logiciels de gestion de versions Git. Git est un gestionnaire de version de fichier qui permet d'enregistrer toutes les modifications sur un fichier, permettant de revenir en arrière en cas d'erreur sur celui-ci.

L'avantage de ce service est de pouvoir travailler de manière collaborative distribuée.

En effet, chacun peut travailler sur ses tâches en parallèle et ensuite l'envoyer sur la plateforme. Le second avantage, Bitbucket favorise le travail à distance car les fichiers sont hébergés sur le serveur et accessibles gratuitement. Finalement, nous avons choisi Bitbucket car il offre des dépôts privés et publics illimités et il est gratuit pour les petites équipes de 5 personnes. À l'opposé de GitHub, qui impose le paiement par utilisateurs pour les dépôts privés. De plus bitbucket a une puissante intégration avec Jira pour un suivi des tâches.

The screenshot shows the Bitbucket web interface for a repository named 'smartwatch'. The top navigation bar includes 'Bitbucket', 'Your work', 'Pull requests', 'Repositories', 'Projects', 'People', 'More', and a 'Create' button. A search bar is on the right. The left sidebar shows the repository structure: 'smartwatch' (selected), 'Source', 'Commits', 'Branches', 'Pull requests', 'Pipelines', 'Deployments', 'Jira issues', 'Security', 'Downloads', and 'Repository settings'. The main content area shows the 'smartwatch' repository with a description: 'Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, add a description to your repository.' Below this is a file browser showing the 'master' branch with a table of files:

Name	Size	Last commit	Message
.gitignore	624 B	1 hour ago	Initial commit
README.md	1.42 KB	41 minutes ago	README.md edited online with Bitbucket

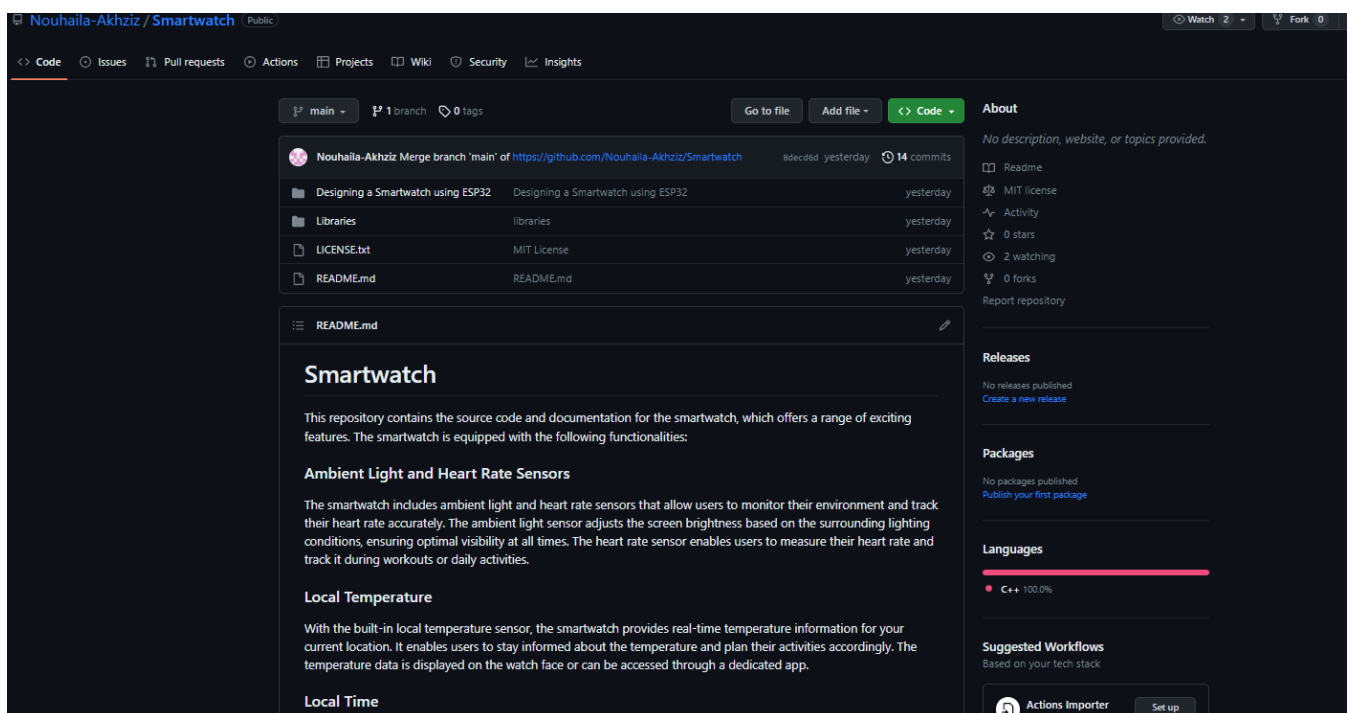
Below the file list is the 'README.md' content, which describes a smartwatch project and its features, including heart rate monitoring and Jira integration.

### 3.2. Github :



Source: <https://github.com/>

GitHub est un service d'hébergement Open-Source, permettant aux programmeurs et aux développeurs de partager le code informatique de leurs projets afin de travailler dessus de façon collaborative. On peut le considérer comme un Cloud dédié au code informatique. Il s'agit d'une plateforme Open-Source et un réseau social pour les développeurs, comme il a une traçabilité complète des modifications...





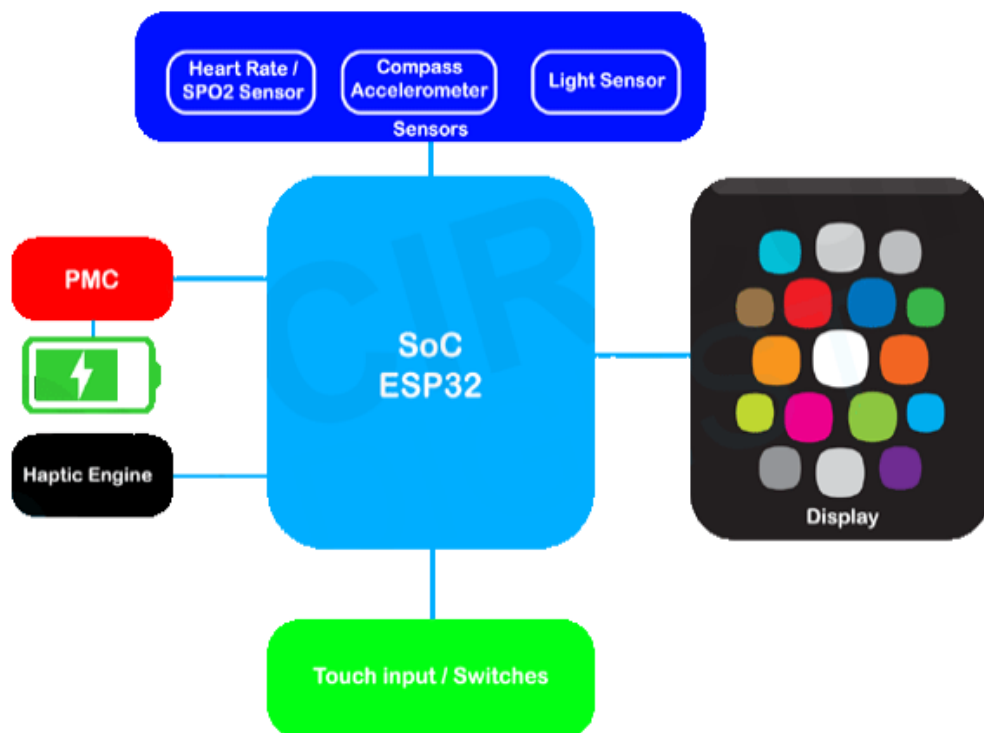
# III. Développement et réalisation technique :

---

Au cours de ce chapitre nous allons décrire l'architecture du système, et nous passerons de la formalisation du problème dégagé et analysé dans les chapitres précédents à sa conception.

## 1. Hardware :

L'image ci-dessous représente le schéma bloc de base de la smartwatch que nous allons réaliser.



**La smartwatch** est composée des blocs de base suivants. Nous discuterons des fonctions de chaque bloc et les composants exacts utilisés seront déterminés dans les parties suivantes.

**SoC :** Dans un système embarqué, le cerveau sera un microcontrôleur ou un microprocesseur. Pour le projet de smartwatch, nous avons sélectionné l'ESP32.

## PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

**Affichage :** J'ai considéré deux affichages pour le projet : un écran TFT rond de 1,28 pouce avec un contrôleur GC9A01 et un écran rectangulaire aux coins arrondis de 1,69 pouce avec un contrôleur ST7789V. Ce sont tous deux d'excellents écrans IPS avec une résolution de 240x240 pixels (GC9A01) et 240x280 pixels (ST7789V). J'ai choisi l'écran plus grand avec une résolution plus élevée.

**Circuit de gestion de l'alimentation (PMC) :** Le PMC comprendra tous les circuits liés à l'alimentation, y compris le chargeur de batterie et le convertisseur élévateur-réducteur.

**Capteurs :** Les capteurs de base que nous prévoyons d'intégrer dans le projet sont le capteur de fréquence cardiaque/SPO2, le capteur de boussole et le capteur de lumière ambiante...

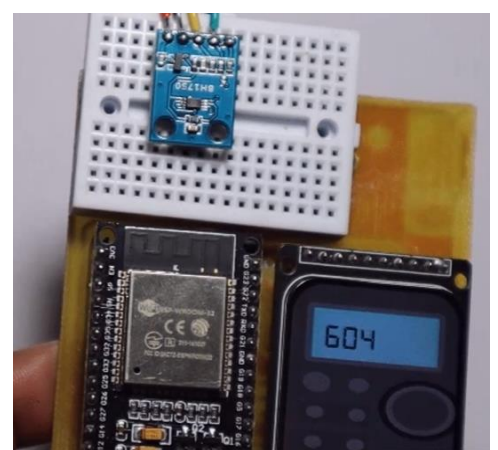
**HMC5883L/QMC5883L :** Étant donné que le capteur peut indiquer les points cardinaux géographiques (Nord, Sud, Est et Ouest), nous pouvons l'utiliser comme une boussole numérique.

**MAX30102 :** Nous allons utiliser le MAX30102 pour mesurer la fréquence cardiaque. Le MAX30102 est un capteur très polyvalent, capable de mesurer non seulement la fréquence cardiaque et le niveau d'oxygène dans le sang, mais aussi la température corporelle.

**Entrées tactiles / Interrupteurs :** L'ESP32 dispose de 10 entrées tactiles. Nous pouvons les utiliser ou utiliser des boutons physiques pour les entrées utilisateur.

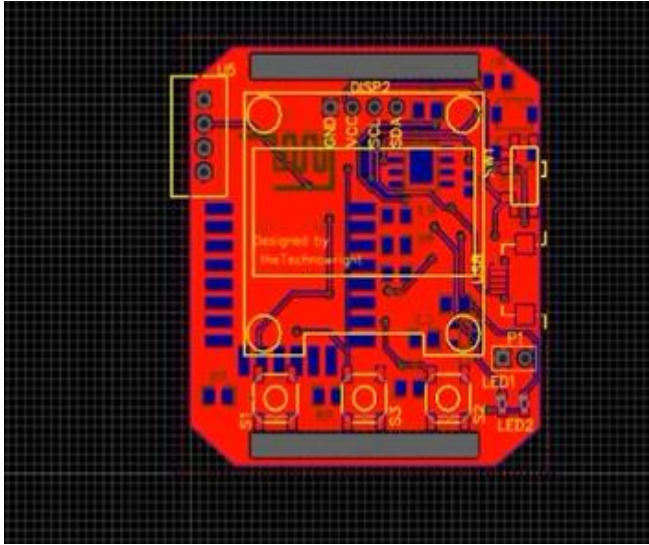
## 2. Software :

Ensuite, nous écrivons le code qui connecte notre ESP32 aux capteurs pour récupérer les données relatives au rythme sanguin, à la température corporelle, à la localisation grâce au gyroscope, à l'heure en temps réel et à la température. Cela nous permet de constater que notre système fonctionne correctement.



## PROJET DE PROCESSUS DE DÉVELOPPEMENT LOGICIEL

Maintenant, nous pouvons implémenter le code dans notre ESP32, effectuer le routage des composants et finalement concevoir notre circuit imprimé. Ainsi, nous pourrions finaliser la réalisation de notre smartwatch en intégrant tous les éléments nécessaires de manière optimale.



## IV. Conclusion :

---

À travers ce projet de smartwatch, nous avons acquis une compréhension approfondie du déroulement et du processus de réalisation d'un projet avant sa finalisation. En choisissant l'application SMARTWATCH et en travaillant selon la méthode Scrum, nous avons pu facilement comprendre les besoins des utilisateurs et ce que les développeurs devaient satisfaire. De plus, ce projet nous a offert l'opportunité de travailler en groupe, simulant ainsi le fonctionnement d'une équipe au sein d'une entreprise. Chacun de nous a appris à gérer efficacement les tâches qui lui étaient assignées tout en respectant au mieux le Product Backlog. Cette expérience a renforcé nos compétences en gestion de projet et notre capacité à collaborer de manière efficace dans un environnement de travail.

## V. Lien Github :

---

<https://github.com/Nouhaila-Akhziz/Smartwatch>