

# Preuves Life Health

## Documentation :

*Je sais décrire le contexte de mon application, pour que n'importe qui soit capable de comprendre à quoi elle sert.*

Le contexte est présent dans le dossier Documentation.

## **Contexte Life Health**

Que vous cherchiez à améliorer votre qualité de sommeil, à être plus actif, ou tout simplement surveiller l'évolution de votre poids, Life Health et son capteur de pas vous accompagne vers la route d'une vie saine. Dans cette application, vous trouverez vos données personnelles que vous aurez remplies au fil de son utilisation. Elles sont faciles à comprendre, personnalisables, et entièrement exploitable.

Avec Life Health, vous avez maintenant le pouvoir d'agir sur votre qualité de vie, et de faire le bilan sur l'évolution de vos données corporelles au fil du temps.

## **Suivi de l'activité avec le compteur de pas :**

---

Suivez automatiquement votre activité quotidienne et votre déplacement journalier grâce à des informations détaillées sur votre nombre de pas.

## **Suivi du poids et de la composition corporelle :**

---

Atteignez vos objectifs de poids grâce à des informations avancées sur le poids, l'évolution du poids, l'IMC et la composition corporelle.

## **Suivi et analyse du sommeil :**

---

Contrôlez vos nuits grâce à la possibilité de rentrer la durée de vos nuits, pour ensuite suivre l'évolution de votre sommeil.

## **Système de succès :**

---

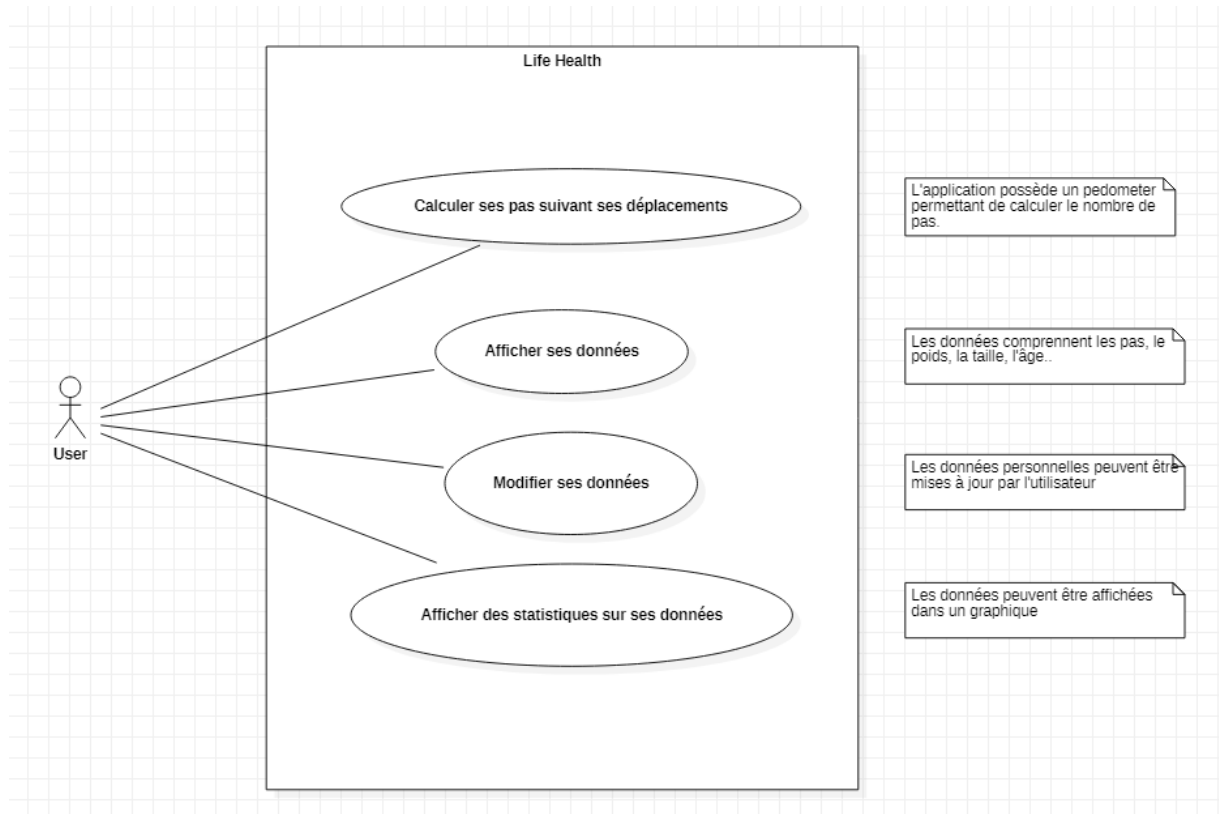
Motivez vous et atteignez vos objectifs personnels avec le système de succès disponible dans l'application.

## **Simple d'utilisation :**

---

Facile à comprendre et intuitive, n'hésitez pas à abuser de l'application !

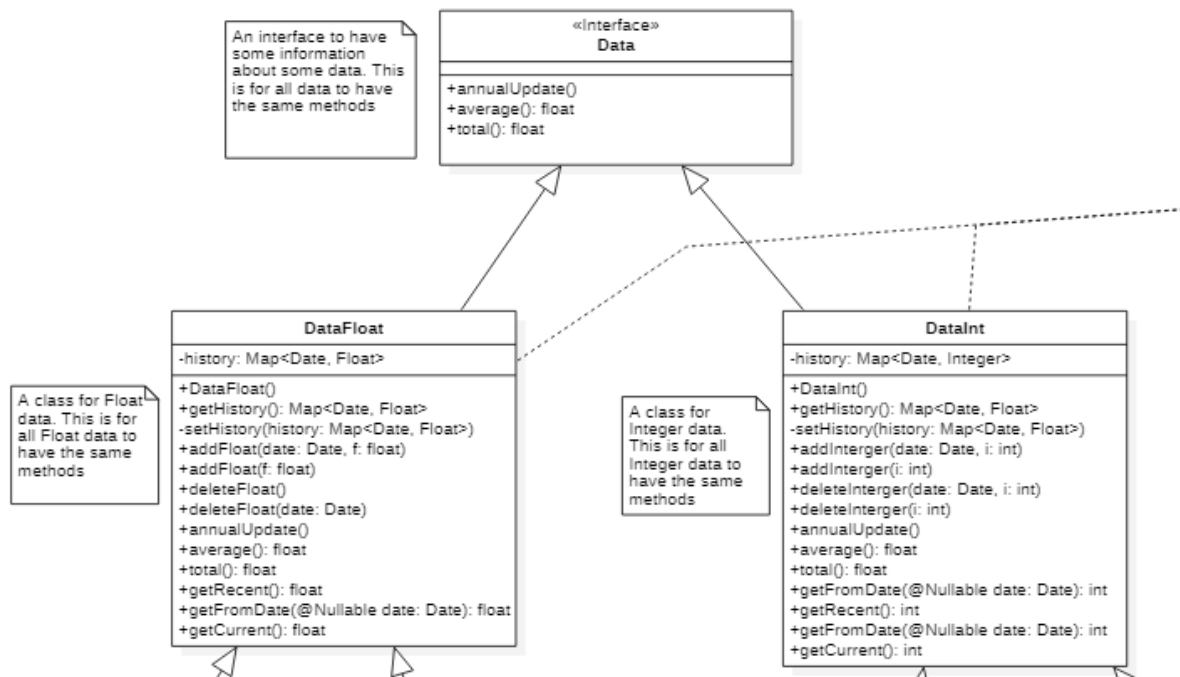
*Je sais concevoir et décrire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application.*



Le diagramme de cas d'utilisation se trouve dans le dossier Documentation.

[illegible]

*Je sais décrire mon diagramme UML en mettant en valeur et en justifiant les éléments essentiels.*



Différents commentaires sont présents sur celui-ci. Il est au complet dans le dossier Documentation.

Code :

*Je sais utiliser les Intent pour faire communiquer deux activités.*

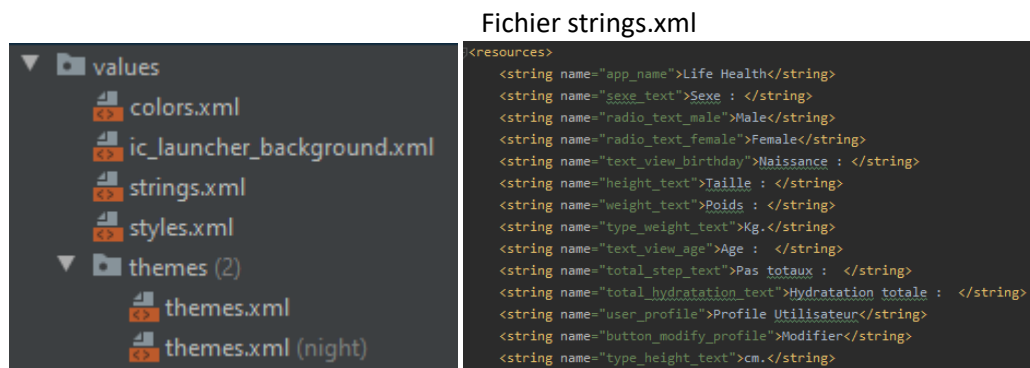
*Je sais développer en utilisant le SDK le plus bas possible.*

Fichier Gradle

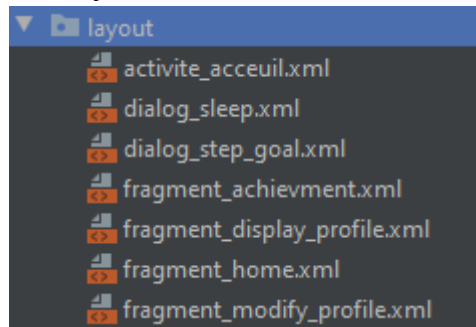
```
defaultConfig {
    applicationId "iut.cours.lifehealth"
    minSdkVersion 16
    targetSdkVersion 16
    versionCode 1
    versionName "1.0"

    testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
}
```

*Je sais distinguer mes ressources en utilisant les qualifier.*



*Je sais faire des vues xml en utilisant layouts et composants adéquats.*



Fichier fragment\_display\_profile.xml dans res/drawable/layout

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/constraint_layout_content_profile"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#555454"
    android:layout_marginTop="15dp"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:paddingStart="15dp"
    android:paddingEnd="15dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/constraint_layout_profile">

    <LinearLayout
        android:id="@+id/linear_layout_content_sexe"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"
        android:background="@drawable/drawable_layout_profile_type"
        android:paddingTop="3dp"
        android:paddingBottom="3dp"
        app:layout_constraintBottom_toTopOf="@id/linear_layout_content_age"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.517"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:id="@+id/text_view_sexe"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Sexe :"
            android:textColor="@color/black"
            android:textSize="24sp"
            android:layout_marginStart="10dp"/>
    </LinearLayout>
</ConstraintLayout>
```

Nous avons séparé différents layout, et utilisons des ConstraintLayout pour avoir des vues responsives.

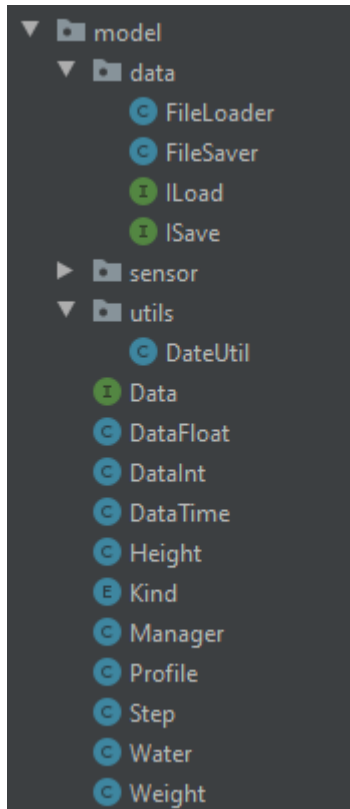
*Je sais coder proprement mes activités, en m'assurant qu'elles ne font que relayer les*

## événements

```
45     private String fragmentTag = "HomeFragment";
46
47
48 /**
49  * the onCreate method of the principal activity
50  * This is here that the the default view is set on the HomeFragment
51  * or the current fragment if the device rotate,
52  * the bottom navigation is got view and the listener navigation is attached,
53  * the manager is instantiate to bind the model to the view,
54  * the sensor is instantiate and begin to register with the Service system
55  *
56  * @param savedInstanceState
57  */
58 @Override
59 protected void onCreate(@Nullable Bundle savedInstanceState) {
60
61     super.onCreate(savedInstanceState);
62
63     this.manager = new Manager(getApplicationContext());
64
65     setContentView(R.layout.activite_acceuil);
66
67     //Commencer a detecter
68
69     sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
70
71     registerDetector();
72
73     startService(new Intent( packageContext: ActiviteAccueil.this,Pedometer.class));
74
75     if(savedInstanceState != null) {
76
77         fragment=getSupportFragmentManager().getFragment(savedInstanceState, key: "Key");
78
79     }
80
81     loadFragment(fragment);
82
83     //getting
84
85     BottomNavigationView navigation = findViewById(R.id.navigation);
86
87     navigation.setOnNavigationItemSelectedListener(this);
88
89 }
```

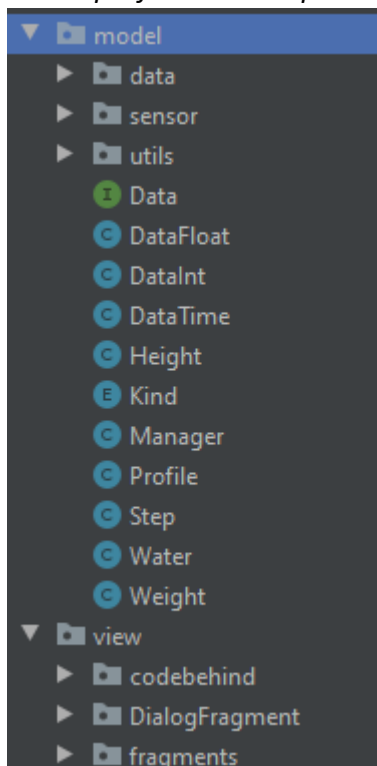
Classe ActiviteAccueil dans view/codebehind

*Je sais coder une application en ayant un véritable métier*



La partie modèle est le métier, avec par exemple le manager, qui gère un profil pour l'utilisateur.

*Je sais parfaitement séparer vue et modèle*





```

.java x ActiviteAccueil.java x dialog_sleep.xml x dialog_step_goal.x
*/
@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    this.manager = new Manager(getApplicationContext());

```

Les vues utilisent le manager, mais ne font pas les tâches du manager.

*Je maîtrise le cycle de vie de mon application*

Classe ActiviteAccueil dans view/codebehind

```

@Override
protected void onStart() {
    super.onStart();
    manager.loadManagerProfile();
}

```

```

@Override
protected void onStop() {
    manager.saveManagerProfile();
    stopService(new Intent( packageContext: ActiviteAccueil.this,Pedometer.class));
    super.onStop();
}

```

Lorsque nous avons un onStop, nous mettons également un onStart, pour, par exemple ici, sauvegarder et charger le manager.

*Je sais utiliser le findViewById à bon escient*  
Classe DisplayProfileFragment view/fragments

```
@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

    manager = ((ActiviteAccueil) view.getContext()).getManager();

    //Kind
    ((TextView) view.findViewById(R.id.text_view_sexe_user)).setText(manager.getProfile().getKind().toString());

    //Age
    ((TextView) view.findViewById(R.id.text_view_age_user)).setText(String.valueOf(manager.getProfile().getAge()));

    //Height
    ((TextView) view.findViewById(R.id.text_view_height_user)).setText(String.valueOf(manager.getProfile().getHeight().getRecent()));

    //Weight
    ((TextView) view.findViewById(R.id.text_view_weight_user)).setText(String.valueOf(manager.getProfile().getWeight().getRecent()));

    //Steps
    ((TextView) view.findViewById(R.id.text_view_step_user)).setText(String.valueOf(manager.getProfile().getSteps().total()));

    //Hydratation
    ((TextView) view.findViewById(R.id.text_view_hydration_user)).setText(String.valueOf(manager.getProfile().getHydratation().total()));

    ((Button) view.findViewById(R.id.modify_button_user_profile)).setOnClickListener(this::clickModifyProfileButton);
}
```

Ceux-ci nous permettent d'insérer les données souhaitées où nous le voulons.

*Je sais gérer les permissions dynamiques de mon application*

*Je sais gérer la persistance légère de mon application*

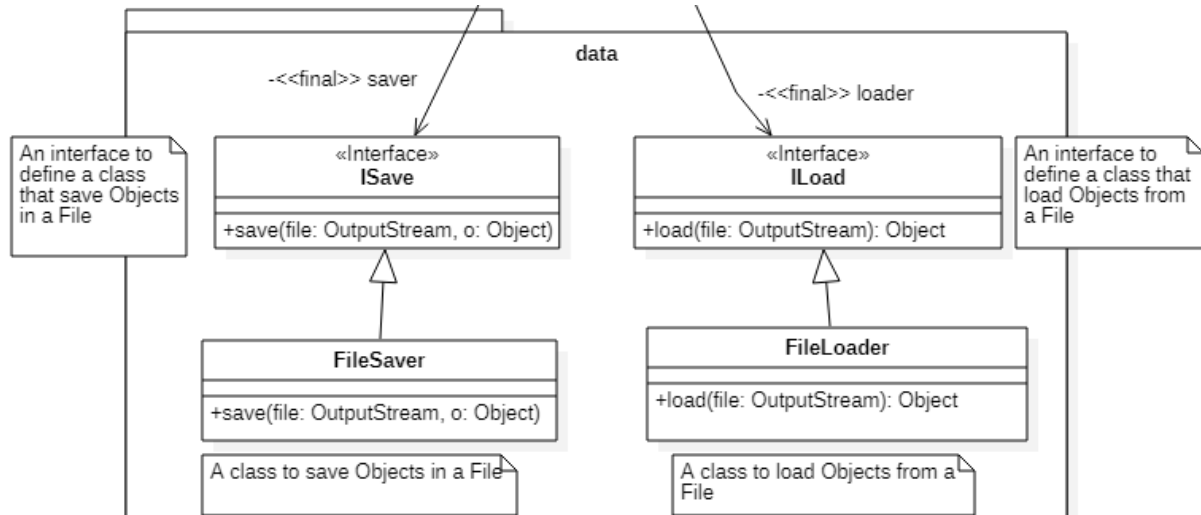
```
if(savedInstanceState != null) {

    fragment=getSupportFragmentManager().getFragment(savedInstanceState, key: "Key");

}
```

Ici, on fait la persistance légère du fragment qui était en cours d'affichage

*Je sais gérer la persistance profonde de mon application*



Classe Manager dans model :

```

/**
 * The method to save the Profile of the Manager in a file
 * @return Return true if the profile as been correctly save, false otherwise
 */
public boolean saveManagerProfile() {
    try(FileOutputStream fos = context.openFileOutput(FILE_PROFILE, Context.MODE_PRIVATE)) {
        // File file = new File(context.getFilesDir(), FILE_PROFILE);
        saveProfile(fos, getProfile());
    } catch (FileNotFoundException e) {
        return false;
    } catch (IOException e) {
        e.printStackTrace();
    }
    return true;
}

/**
 * The method to Load the Profile in the Manager from a file
 */
public void loadManagerProfile() {
    Profile profile = null;

    try {
        Log.d( tag: "LOADMYMANAGER", msg: "Ici loading manager : loadManagerProfile");
        profile = loadProfile(context.openFileInput(FILE_PROFILE));
    } catch (FileNotFoundException ignored) { }

    if(profile == null) {
        profile = new Profile();
    }

    profile.annualUpdate();
    setProfile(profile);
}

```

Nous sauvegarder le profil dans un fichier du téléphone.

*Je sais afficher une collection de données*

```
@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);
    manager = ((ActiviteAccueil) getActivity()).getManager();

    listViewStep = (RecyclerView) view.findViewById(R.id.la_list_step);
    listViewStep.setLayoutManager(new LinearLayoutManager(this.getContext()));
    listViewStep.setAdapter(new AdapterStepView(manager.getProfile().getSteps().getHistory()));

    close = ((ImageView) view.findViewById(R.id.step_dialog_closing));
    close.setOnClickListener(this::clickStepClosing);
}
```

Extrait de la classe StepGoalDialog pour instancier la RecyclerView et l'adapter à l'affichage de la Map.

Voir aussi les classes AdapterStepView/StepViesHolder

*Je sais coder mon propre adaptateur*

```
/**
 * a method for instanciate the view of a cell of the display
 * @param parent the parent to get the context
 * @param viewType the view
 * @return the a new Holder with the cell
 */
@NonNull
@Override
public RecyclerView.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    LinearLayout stepLayout = (LinearLayout)LayoutInflater.from(parent.getContext()).inflate(R.layout.step_view_cell, parent, attachToRoot: fa
    return new StepViewHolder(stepLayout);
}

/**
 * a method to bind the list to the modele
 * @param holder the recycler view
 * @param position the index of the lists
 */
@Override
public void onBindViewHolder(@NonNull RecyclerView.ViewHolder holder, int position) {
    ((StepViewHolder)holder).getTextViewDate().setText(DateUtil.getStringFromDate(date.get(position)));
    ((StepViewHolder)holder).getTextViewStep().setText(step.get(position).toString());
}
```

Extrait de la classe AdapterStepView

## Je maîtrise l'usage des fragments

```
@Override
public boolean onNavigationItemSelected(@NonNull MenuItem item) {

    switch (item.getItemId()) {

        case R.id.home:

            fragment = new HomeFragment();
            fragmentTag = "HomeFragment";
            break;

        case R.id.profile:

            fragment = new DisplayProfileFragment();
            fragmentTag = "DisplayProfileFragment";
            fragment = new ModifyProfileFragment();

            break;

        case R.id.bluetooth:

            fragment = new AchievementFragment();
            fragmentTag = "AchievementFragment";
            break;

    }

}
```

Ici, on a le gestionnaire de navigation utilisant les fragments

## Je maîtrise l'utilisation de Git

root @ master URL : [https://forge.clermont-universite.fr/git/android\\_eliass\\_pesce\\_2020\\_2021](https://forge.clermont-universite.fr/git/android_eliass_pesce_2020_2021) | Stat

Nom				
Code				
Documentation				
.gitignore				

Dernières révisions

#	Date	Auteur	Commentaire
ea100a50	28/03/2021 00:06	Emrick PESCE	[DOC] Ajout de certaines modifications dans le diagramme de classe. Modification du diagramme de cas d'utilisation
b3d39306	27/03/2021 23:56	Emrick PESCE	[DEV] Ajout l'annualUpdate au profil et au Manager
4daea3df	27/03/2021 23:52	Emrick PESCE	[DEV] Modification de la classe DateTime pour qu'elle puisse être utilisée pour les données de sommeil, avec les heures et minutes
ed7afbea	27/03/2021 23:01	Emrick PESCE	[DEV] Correction de la persistance
f58f07b9	27/03/2021 22:34	guilhemillas	[DEV] ajout service pour ne pas faire tourner le sensor dans le thread principal
2d2089b3	27/03/2021 22:26	guilhemillas	[DEV] ajout service pour ne pas faire tourner le sensor dans le thread principal
521284ed	25/03/2021 00:42	guilhemillas	[DOC] Diagramme de cas, sketches et contexte fais en début de projet
0e7f93d4	24/03/2021 23:57	guilhemillas	[DEV] divers essais dans la sauvegarde profonde
3d50e193	20/03/2021 17:52	Emrick PESCE	[DOC] Modification du diagramme de classe pour ajouter les modifications récentes et le package view
d8e8fc08	20/03/2021 17:51	Emrick PESCE	[DEV] Correction de différents petites choses qui n'allaient pas. Déplacement des packages sensor et data dans le package model

root @ Guilhem

URL : [https://forge.clermont-universite.fr/git/android\\_elias\\_pesce\\_2020\\_2021](https://forge.clermont-universite.fr/git/android_elias_pesce_2020_2021) | [Statistiques](#) | [Branch](#)

Nom

Taille

Code

Documentation

.gitignore

Dernières révisions

#	Date	Auteur	Commentaire
<a href="#">d7a076f7</a>	28/03/2021 12:47	Emrick PESCE	Merge branch 'Emrick' into Guilhem
<a href="#">1659a280</a>	28/03/2021 12:44	Emrick PESCE	[DOC] Contexte en .odt renommé
<a href="#">7673499a</a>	28/03/2021 12:42	Emrick PESCE	[DOC] Passage du diag de cas dans un nouveau fichier, pour séparer les fichiers des diagrammes
<a href="#">a0d9e1a6</a>	28/03/2021 12:37	Emrick PESCE	[DOC] J'ai renommé l'image du sketch
<a href="#">6318f987</a>	28/03/2021 12:36	Emrick PESCE	[DEV] Modification du contexte pour changer quelques points mineurs. Ajout d'un contexte en .md pour peut-être le faire en markdown
<a href="#">d6383fcf</a>	28/03/2021 11:15	Emrick PESCE	[DEV] Essai de mettre en place le sleep qui se met à jour, mais fonctionne pas
<a href="#">526c1c46</a>	28/03/2021 10:00	guilhemillas	[DEV] debut ajout heure de sommeil et ajout getRecentHours/Minues dans DateTime
<a href="#">b7df7fe4</a>	28/03/2021 00:17	guilhemillas	Merge branch 'master' into Guilhem
<a href="#">d9a5c955</a>	28/03/2021 00:17	guilhemillas	[DEV] changement de la fermeture Service au dans le cycle de vie de l'application, et documentation

root @ Emrick

URL : [https://forge.clermont-universite.fr/git/android\\_elias\\_pesce\\_2020\\_2021](https://forge.clermont-universite.fr/git/android_elias_pesce_2020_2021) | [Stat](#)

Nom

Taille

Code

Documentation

.gitignore

Dernières révisions

#	Date	Auteur	Commentaire
<a href="#">1659a280</a>	28/03/2021 12:44	Emrick PESCE	[DOC] Contexte en .odt renommé
<a href="#">7673499a</a>	28/03/2021 12:42	Emrick PESCE	[DOC] Passage du diag de cas dans un nouveau fichier, pour séparer les fichiers des diagrammes
<a href="#">a0d9e1a6</a>	28/03/2021 12:37	Emrick PESCE	[DOC] J'ai renommé l'image du sketch
<a href="#">6318f987</a>	28/03/2021 12:36	Emrick PESCE	[DEV] Modification du contexte pour changer quelques points mineurs. Ajout d'un contexte en .md pour peut-être le faire en markdown
<a href="#">ea100a50</a>	28/03/2021 00:06	Emrick PESCE	[DOC] Ajout de certaines modifications dans le diagramme de classe. Modification du diagramme de cas d'utilisation
<a href="#">b3d39306</a>	27/03/2021 23:56	Emrick PESCE	[DEV] Ajout de l'annualUpdate au profil et au Manager
<a href="#">4daea3df</a>	27/03/2021 23:52	Emrick PESCE	[DEV] Modification de la classe DateTime pour qu'elle puisse être utilisée pour les données de sommeil, avec les heures et minutes
<a href="#">ed7afbea</a>	27/03/2021 23:01	Emrick PESCE	[DEV] Correction de la persistance

On peut voir que l’on a chacun une branche personnelle où on code, et on merge le tout sur la branche principale master, ou sur nos branches si on a des changements qu’on ne veut pas forcément directement sur le master.

## Application :

*Je sais développer une application sans utiliser de librairies externes*

*Je sais développer une application publiable sur le store.*

À venir

## Je sais utiliser l'accéléromètre et ou le gyroscope

```
/**
 *A class which use the accelerometer sensor data to create a pedometer
 * It implement SensorEventListener, to get the data of the accelerometer, and extends Service to run the pedometer
 * on an other thread that the principal thread
 */
public class Pedometer extends Service implements SensorEventListener {

    private double magnitudePrevious = 0;
    private OnPedometerChangeListener onPedometerChangeListener;

    /**
     * a method for the SensorEventListener which is called every time it detected a new value
     * @param event the data matrix of a three-dimensional vector representing acceleration along each device axis
     * Here is the calculation of the subtract between the previous and the new magnitude , when the sensor detect a new value.
     * More the result is high, less the pedometer is sensible. If the result is high enough, that call subscribed classes of the OnPedometerChangeListener
     */
    @Override
    public void onSensorChanged(SensorEvent event) {

        if (event != null) {
            float x_acceleration = event.values[0];
            float y_acceleration = event.values[1];
            float z_acceleration = event.values[2];

            double magnitude = Math.sqrt(x_acceleration * x_acceleration + y_acceleration * y_acceleration + z_acceleration * z_acceleration);
            double magnitudeDelta = magnitude - magnitudePrevious;
            magnitudePrevious = magnitude;

            if (magnitudeDelta > 5) {
                onPedometerChangeListener.onPedometerChanged();
            }
        }
    }
}
```

Classe Pedometer dans model/sensor