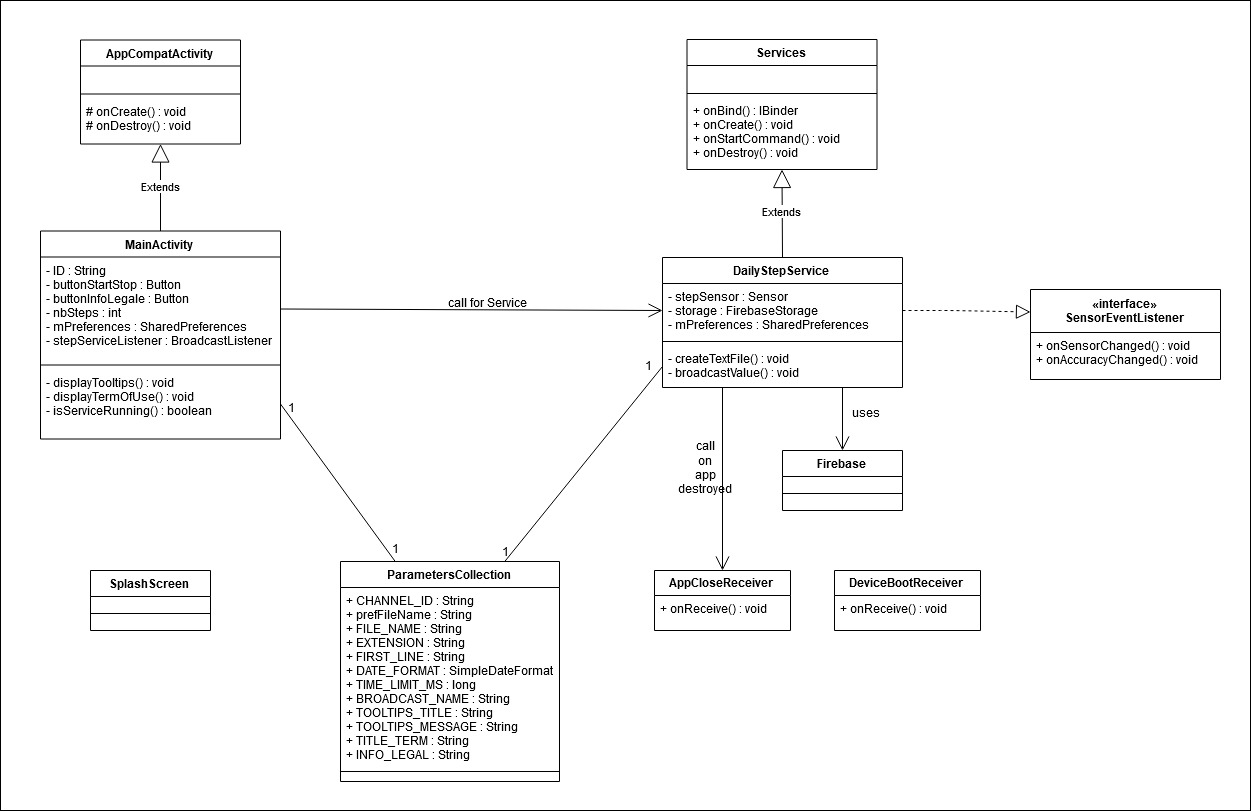
1. **StepAnalyzer**

*StepAnalyzer,* using the Step Detector sensor introduced since Android 4.4 KitKat’s, is an improved version of Step Counter application which collects informations about users’ moving rhythm : it measures time gap between two consecutive steps during a walking/running session. Psychological-wise, moving rhythm is affected by walker’s mental health. Thus, this application can show if there is any variability regarding an user’s moving rhythm during a given screening period.



About its architecture, *StepAnalyzer*’s measuring task, file creation and uploading to server are taken charge of by Android Foreground Services with two *BroadcastReceiver* : one that checks if the application is closed and other if the device is rebooted. Both will guarantee that the *StepAnalyzer* Service will be relaunched after the detected event.

When the service is launched, the application starts to measure user’s moving rhythm (ie. time gap between every two steps) and stores them into a *.csv* tabular file. In order not to affect the data visualisation, everytime users have stopped walking for more than a certain time in milliseconds (defined as constant *TIME\_LIMIT\_MS*), the application will upload the current file containing all the recorded measure to the server, then create a new file and continue recording data into it. The Cloud Storage server is Google Firebase.

This application also features a shared preference file containing user’s ID and application’s working status so they will be kept throughout application’s launch.

1. **Constants & Parameters**

|  |  |
| --- | --- |
| Name | Explanation & Value |
| CHANNEL\_ID | A string constant containing the ID for the Foreground Notification Channel, used in the Foreground Service. This constant is used in *DailyStepService.java*  Value : “ForegroundServiceChannel” |
| sharedPrefFile | A string constant containing the Shared Preferences File name where *MainActivity* will store user’s ID on it in order to be extracted by *DailyStepService* and for future launch. Also store application’s working state.  Value : “StepAnalyzerPreferences” |
| FILE\_NAME | Used for file creation.  Value : “DailyStep” |
| EXTENSION | A string constant containing the file’s extension  Value : “.csv” |
| FIRST\_LINE | A string constant containing the first line to be inserted during file creation.  Value : “Timestamp; Delta\n” |
| df2 | A simple date format constant that will convert a given date in milliseconds to hour, minute and second. Use to mark a step’s captured moment.  Value : SimpleDateFormat(“hh:mm:ss”) |
| dfYear | A simple date format constant that convert given date into year. Used to manipulate Firebase Storage’s directory.  Value : SimpleDateFormat(“YYYY”) |
| dfMonth | A simple date format constant that convert given date into month. Used to manipulate Firebase Storage’s directory.  Value : SimpleDateFormat(“MMM”) |
| dfDay | A simple date format constant that convert given date into day. Used to manipulate Firebase Storage’s directory.  Value : SimpleDateFormat(“d”) |
| dfHour | A simple date format constant that convert given date into hour. Used to name recorded file.  Value : SimpleDateFormat(“hh-mm-ss”) |
| TIME\_LIMIT\_MS | Time threshold above which when a time gap is calculated, the current file will be uploaded and a new file will be created. Unit is milliseconds.  Value : 5000 |
| ACTION\_CUSTOM\_BROADCAST | A string constant containing the Broadcast’s name, used to create a broadcast and send it (in *DailyStepService)*, and to create a broadcast listener and listen to that broadcast (in *MainActivity)*  Value : BuildConfig.APPLICATION\_ID + ".ACTION\_CUSTOM\_BROADCAST" |
| ZERO\_REPORT\_LATENCY | Max report latency parameter used in registering listener for the step sensor.  Value : 0 |
| TOOLTIPS\_TITLE | A string constant containing the tooltips dialog’s title. Used in *MainActivity*.  Value : "Welcome to Step Analyzer !" |
| TOOLTIPS\_MESSAGE | A string constant containing the tooltips message. Used in *MainActivity.*  Value : "In order to use this app, " +  "please first fill correctly the ID field with your given ID " +  "before pressing the Start button.\n\n" +  "This app works well on Background task, and the Service will be relaunched " +  "if app is manually turned off, or your device is rebooted.\n\n" +  "Thanks for your patience ! Click anywhere outside this box to use Step Analyzer."; |
| TITLE\_LEGAL | A string constant containing the term of use dialog’s title. Used in *MainActivity*.  Value : “Term of Use” |
| INFO\_LEGAL | A string constant containing the term of use. Used in *MainActivity*  Value : "You can read about term of use here" |

1. **Important methods**

|  |  |  |
| --- | --- | --- |
| Methods | Input, output | Functionality |
| MainActivity.java | | |
| isMyServiceRunning() | - Input : a service class  - Output : boolean | Checks whether that service class is running and returns true if it is running, false otherwise. Used to ensure that only instant of service is launched at a time. |
| displayTooltips() | None | Create a dialog containing tooltips when application is launched. |
| displayTermOfUse() | None | Create a dialog containing term of use. A button is created for this purpose. |
| onCreate() | - Input : a saved instance state | Overridden method from AppCompatActivity class. Everything relied to the application (UI, functionalities, service call, etc.) are implemented inside this method. |
| onDestroy() | None | Overridden method from AppCompatActivity class. Evoked when application is closed/destroyed. |
| DailyStepService.java | | |
| createTextFile() | - Input : user ID, current date, recorded data | Create a .csv file containing the recorded data and named with the current date (so the file’s unicity is guaranteed) that is temporarily saved in the internal memory; then upload it to the Firebase Server using the given user ID. |
| broadcastValue() | None | Send the number of steps counted to MainActivity in order to show them in the UI |
| createNotificationChannel() | None | Create a notification channel used for Foreground Service Notification |
| onBind() | - Input : an Intent  - Output : IBinder | Overridden method that binds the service with the system. |
| onCreate() | None | Overridden method that initializes all attributes at the beginning of the services. |
| onDestroy() | None | Overridden method that calls for AppCloseReceiver broadcast that listens to application’s closing states when closed. Will guarantee that the service is relaunch when the application is closed |
| onStartCommand() | - Input : an Intent, a flags, a start ID, all are system-defined | Overridden method where the main service’s code is executed. |
| onSensorChanged() | - Input : an event where a change is detected | Overridden method from the SensorEventListener interface which listens to sensor’s change. In this case, an event corresponds to when a step is detected, and this method will calculate the time gap between two recent steps and store it into recorded data. If the time gap is higher than the allowed time limit, createTextFile() will be called. |
| onAccuracyChanged() |  | A method that must be redefined when implementing SensorEventListener interface. It has no impact on our application’s case. |

1. **Background Task Architecture’s interaction with different smartphone model**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GalaxyS6  (Android 7.0) | HuaweiY6  (Android 9.1) | Pixel3  (Android 9.1) | K4 Dual  (Android 5.0) |
| WorkManager - PeriodicWorkRequest | Multiple tasks are created and executed parallely. Periodic task stays after long time and after phone reboot. | Same execution characteristic as S6. Periodic task disappears after a long time and after phone reboots. | Periodic task disappears after a short while and after phone reboots. | Crash because of lack of sensor. |
| Services | Works very well. Task stay even when app closed or phone rebooted, as the service will be automatically recreated after destroyed. | Does not work, as since Android Oreo (version 8), launching a service from Broadcast Receiver is prohibited.  => Switched to an architecture which will use Services for Android version 7 or lower, and Foreground Services otherwise. | |
| Foreground Services | Already work with Services. | Works as well as Services, with differences being task makes CPU awake, runs in foreground and creates a notification for user when service (re)created. | |
| JobIntentServices | Creates a thread that works for some minutes then ends itself. No method to relaunch services after services ended or phone rebooted. | | |

***Remark*** : In an attempt to save more battery, as well as protect users’ privacy more effectively, Google has adopted the new background task policy. Since Android 8.0, every background task should only be running for a short while (which explains WorkManager and JobIntentServices’ behaviour), by example Transaction, Downloading from an online server, etc. If that task needs to be executed consistently, which is our Step Analyzer’s case, it has to be switched to Foreground Services with a Notification Channel that notify when that foreground service is launched.

***Additional technical aspect :***

* This application features a splash screen with IMT Mines Alès’ logo on it. To make this splash screen appear before the main screen, define it in *AndroidManifest.xml* as such:

<activity android:name=".SplashScreen">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
</activity>  
  
<activity android:name=".MainActivity"></activity>

* In order to be able to work correctly, two permissions must be defined in *AndroidManifest.xml* : RECEIVE\_BOOT\_COMPLETED to allow DeviceBootReceiver to listen to device’s reboot status, and FOREGROUND\_SERVICE to allow foreground service to work.
* The *DailyStepService* also needs to be defined here.
* Firebase Storage’s usage required its package to be declared in the GradleScripts/build.gradle(app)/*dependencies* section, including com.google.firebase:firebase-storage:<latest\_version> and com.google.firebase:firebase-auth:<latest\_version>.