Kenkou Integration Documentation

KenkouSDK 1.0

By Alexander Gorny/Ondej Vancak/Egor Gaydamak/Kristina Goryacheva/Jian Zhao

Version of this document: 02/10/2020

Kenkou SDK for Android

Documentation

KenkouSDK makes it easy to provide advanced stress measurement to your app. It provides an intuitive and highly customizable UI built on a powerful HRV measurement core to provide advanced Stress Management for your users.

We provide a comprehensive API, providing the ability to create complex custom integration with a few lines of code.

Version Support

- **KenkouSDK Android 1.0.0:**
  Minimum Android version 21
  Kotlin version 1.4.1

Language

KenkouSDK for Android is written in Kotlin.

Getting Started

Integrating KenkouSDK Manually

In order to be able to download all the necessary artifacts that compose the KenkouSDK, simply add the following lines to your build.gradle files:

Gradle dependencies

- <module>/build.gradle

```
dependencies {
   ...
   implementation files('lib/kenkou-measurement-sdk-1.3.aar')
}
```

Basics

Once you've integrated KenkouSDK, there are a few basics to go over before you can fully utilise the features and capabilities.

Client Authentication

We use client credentials to authenticate you as a partner product with Kenkou. These credentials are **mandatory**, and the SDK will not function without providing them.

So before continuing, make sure you have the following:

- Client Identifier
- Client Secret

You can then start using KenkouSDK - we recommend starting in your `CustomApplication` class.

Here a screenshot of code will be provided

Note:

Used permissions

Permissions for camera and internet are declared in our library manifest so you don't have to declare them yourself in case you don't use them.

Also we use internally for our network requests OkHttp. This library requires that you enable Java 8 in your builds to function properly. If you have not done so yet, then you will just need to add this snippet to your `build.gradle:`

```
android {
    ...

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
```

```
    kotlinOptions {
        jvmTarget = '1.8'
    }
}
```

All Done

KenkouSDK is now set up and ready to go.

Next Steps

With everything set up, you can now move onto the Measurement Combined Component.

Measurement Combined Component

Kenkou SDK provides Stress measurement as a combined component including measurement onboarding, measurement, question after measurement and measurement result. We provide full featured HRV analysis for scientific research and professional use, including:

- Supports wide range of ECG, PPG and RR interval data formats Accurate QRS and pulse wave detection
- Automatic artefact correction algorithm
- Automatic analysis sample generation
- Computes all commonly used time-domain, frequency-domain and nonlinear HRV analysis parameters

```
KenkouSdk.startMeasurement(this)
```

Data access

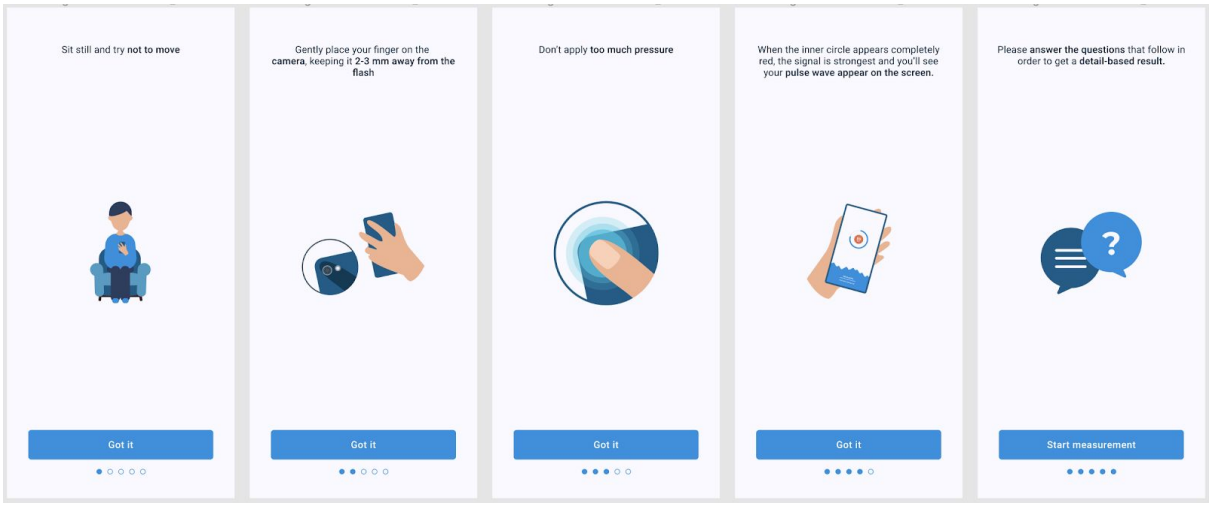
After measurement is finished, detailed stress and cardiovascular indexes would be shown in the measurement result. Current Indexes as well as indexes that we can provide in the future are listed in the reference at the end of this document. These data can be accessed from your app using the data intent which is passed to your activity as a result of the measurement.

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == KenkouSdk.MEASUREMENT && resultCode == RESULT_OK)
{
        val result = KenkouSdk.getMeasurementResult(data)
    }
}
```

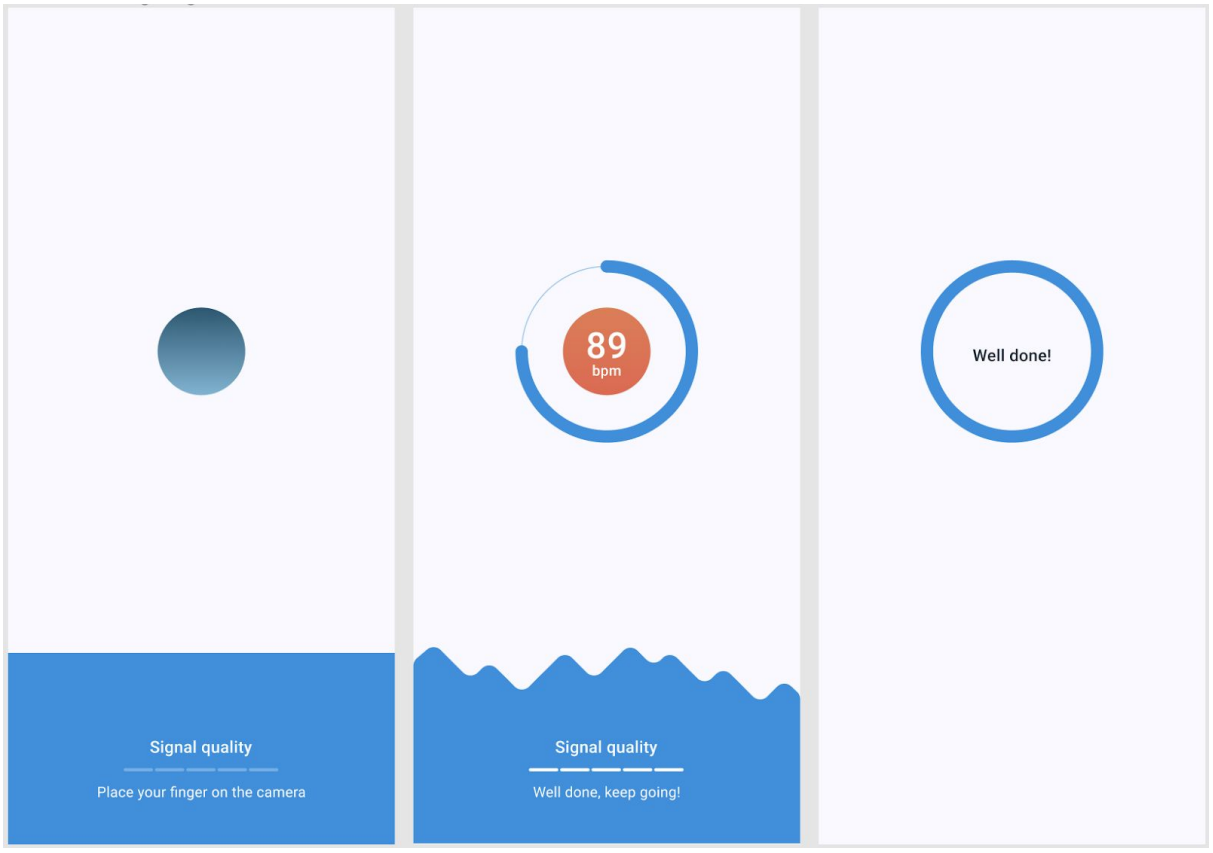
Color/Font

KenkouSDK provides customized color, shape and font options built in every UI component, designed to provide all the flexibility you need to make Kenkou feel at home in your app.

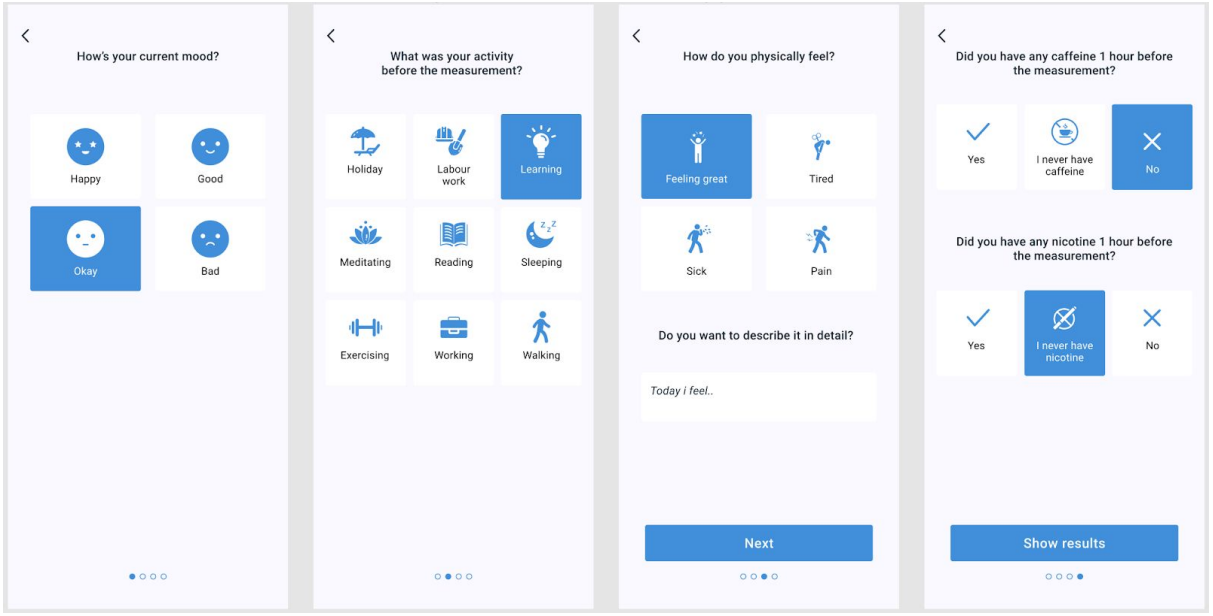
Measurement onboarding screen



Measurement screen



Questions after measurement screen



Contents

- KenkouColorBook
    - ColorPrimary
    - ColorSurface
    - ColorOnSurface
    - ColorBackground
    - ColorOnBackground
    - ColorValue
        - ColorA
        - ColorB
        - ColorC
        - ColorD
    - ColorOnValue
- KenkouFontBook

Color

Colors are derived from the application theme, using the color attributes as defined by Material components. In order for the user interface to be clear make sure that your application theme defines following colors.

```
<item name="colorPrimary">#2688E5</item>
<item name="colorOnPrimary">#ffffff</item>

<item name="android:colorBackground">#e5e5e5</item>
<item name="colorOnBackground">#121212</item>

<item name="colorSurface">#ffffff</item>
<item name="colorOnSurface">#121212</item>
```

ColorPrimary

ColorPrimary that is applied to button, page slider, pulse wave background, measurement progress indicator, icons tint on questions after measurement screens. The default primary color code is #2688E5.

ColorOnPrimary

ColorOnPrimary that is applied to text on buttons, text on pulse wave, Signal quality indicator, text on selected cards. The default ColorOnPrimary is #FFFFFF.

ColorBackground

ColorBackground that is applied to every screen background. The default primary ColorBackground is #E5E5E5.

ColorOnBackground

ColorOnBackground that is applied to texts on the background, navigation elements, chips on the measurement details screen. The default ColorOnBackground is #121212.

ColorSurface

ColorSurface that is applied to cards background and text field background. The default ColorSurface is #FFFFFF.

ColorOnSurface

ColorOnSurface is applied to text on unselected cards and text in the text field. The default secondary color is #121212.

ColorValue

ColorValue refers to the color indicating the level of stress index/recovery ability/heart rate and mood on the measurement result screen. Four ColorValue are set in the current version (ColorA/ColorB/ColorC/ColorD). You can change these by overriding them in your resources.

Measurement result screen

ColorA applied to measurement values showing “low stress index/excellent recovery ability/HR normal/mood happy; The default ColorA is #31BFDE.

ColorB applied to measurement values showing medium stress index/good recovery ability/HR medium/mood good; The default ColorB is #36CB83.

ColorC applied to measurement values showing high stress index/okay recovery ability/HR high/mood okay; The default ColorC is #FFAE10.

ColorD applied to measurement values showing very high stress index/poor recovery ability/HR very high/mood bad; The default ColorD is#FF6C4C.

ColorOnValue

ColorOnValue that is applied to texts on measurement result values. The default ColorOnValue is #FFFFFF.

Examples:

```
<color name="color_value_a">#31bfde</color>
<color name="color_value_b">#36cb83</color>
<color name="color_value_c">#ffb629</color>
<color name="color_value_d">#ff6c4c</color>
<color name="color_on_value">#ffffff</color>
```

Typography

`kenkouFontBook` provides a collection of fonts that can be customized to provide text styles to your Kenkou experience. The font book will default to system font. The fonts we recommend to use include Lato, Roboto, Open Sans, SF pro and Helvetica neu.

Examples:

Here a screenshot of code will be provided

Troubleshooting

Logging

To debug SDK functionality you might want to enable logging. The SDK provides extensive logging if enabled.

Example how to enable logging upon application start.

Here a screenshot of code will be provided

Code minification

If you are planning to minify and obfuscate your app code that integrates our SDK, you will need to add these lines to your preferred `proguard-rules` file in addition to any other rules that might be already in place.

Here a screenshot of code will be provided

Reference

Parameter	unit	Definition	comment
Cardio			
HR	bpm	Heart rate	
HRmean	bpm	Average heart rate	
HRmax	bpm	Maximum heart rate	
HRmin	bpm	Minimum heart rate	
HRV			
RR	ms	weighted average of RR intervals	
RMSSD	ms	Square root of the squared mean of the sum of all differences in successive RR intervals	RMSSD expresses how much the heart rate changes from one heartbeat to the next. Indicator of parasympathetic activity. Error-prone with artifacts and arrhythmias.
SDNN	ms	Standard deviation of all RR intervals of a measurement (total variability)	The SDNN is the "gold standard" for medical stratification of cardiac risk when recorded over a 24h period, it is more accurate when calculated over 24h than during shorter periods monitored.
PNN50	ms	Percentage of consecutive RR intervals that differ from each other by more than 50ms.	Indicator of parasympathetic activity.
stress index	ms^-1	Baevsky’s stress index	Mathematical description of the histogram (see also BR. M. Baevsky. Methodical recommendations use kardivar system for determination of the stress level and estimation of the body adaptability standards of measurements and physiological interpretation. 2009)
normalized	0-100	normalized version of stress index	normalized version of previous quantity ranging from 0 to 100

stress index			
SD1	ms	Standard deviation of the orthogonal distances of the RRi / RRi + 1 points to the transverse diameter of the ellipse	Width of the point cloud; more sensitive to rapid, higher frequency changes in heart rate.
SD2	ms	the standard deviation along the line-of-identity in the Poincaré plot	Length of the point cloud; quantifies the long-term HRV.
PNS	0-100	Parasympathetic Nervous System index	Parasympathetic nervous system activity compared to normal resting values
histogram		aggregated distribution of RR intervals in 50 ms bins	normalised RR interval histogram with bin width 50 msec