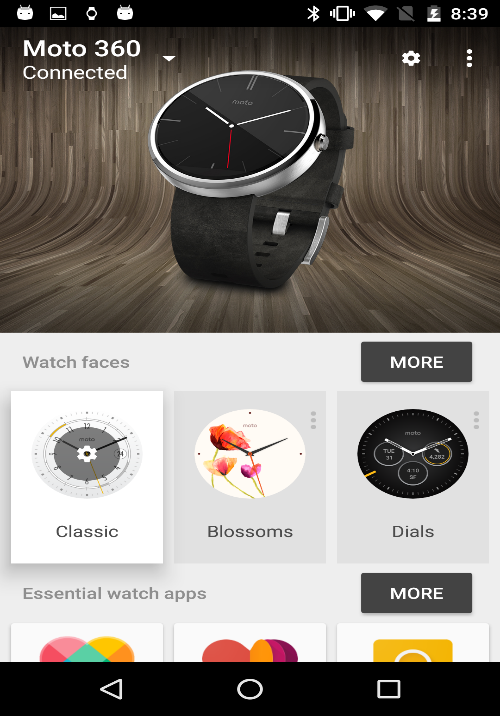
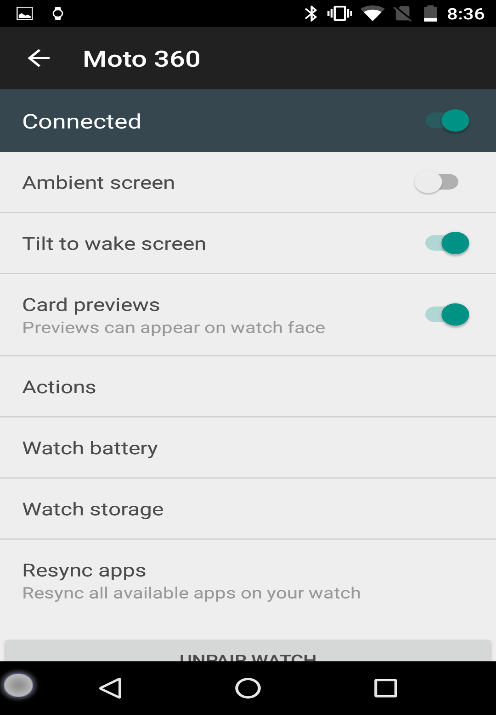
Android Wear Application

Initial Steps to Test the application.

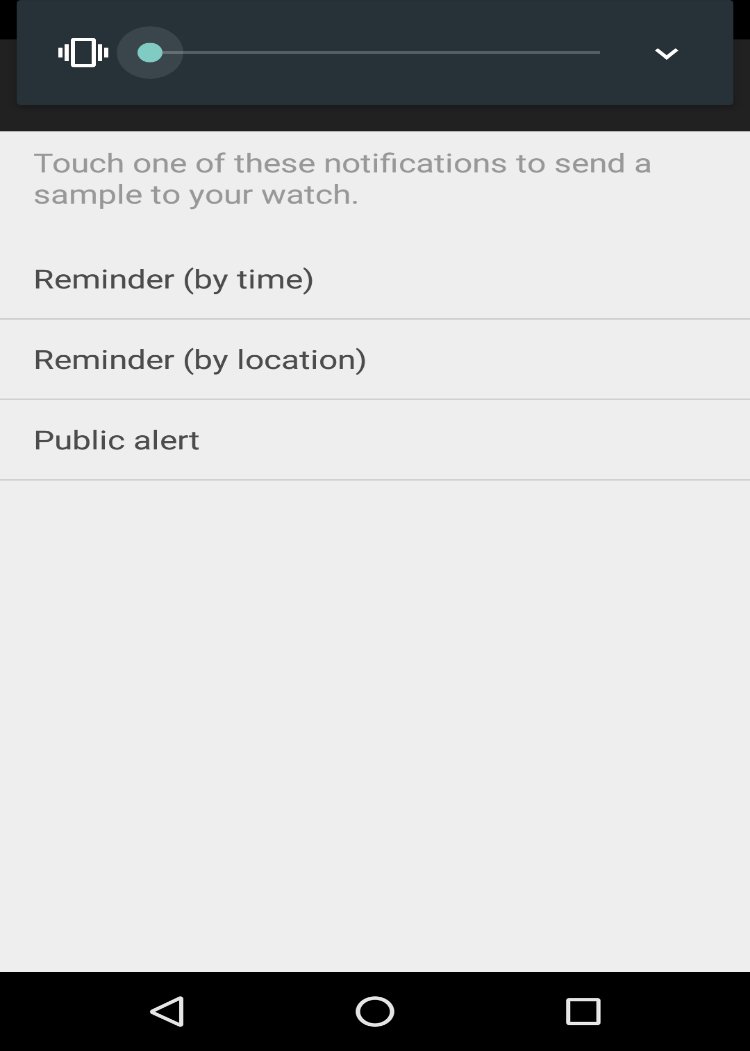
1. Pair the Watch and Mobile. Once paired the watch its show like this.



1. Once you installed the APK file in your Mobile you have Resync the app to the watch.

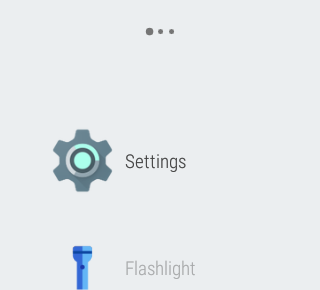


1. You have test the sample notification from mobile to watch. Because the watch and mobile having proper connection then only we have receive workout values from watch to mobile. The Android Wear App having menu (Try out watch notification).

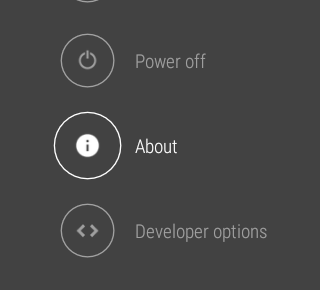


**Enable Developer Option in Android Wear:**

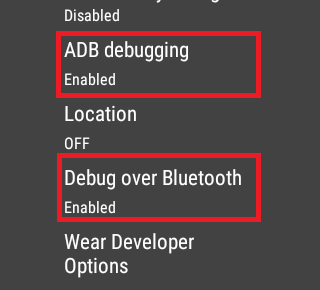
Step1: Please click the Settings.

****

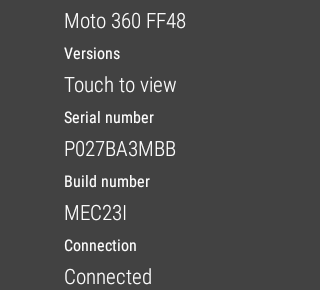
Step2: If the Developer option enables please click “developer option”. Otherwise please click “About”.



Step3: Please enable the following option.



Step4: If not developer please click Build Number 7 times and enable developer option.

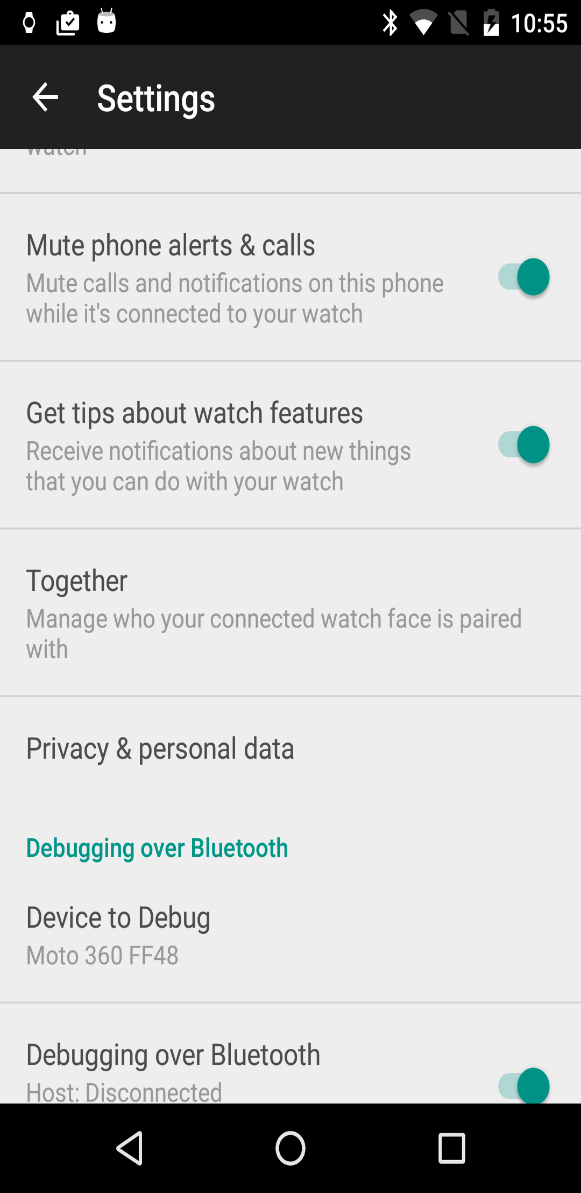


**Development Testing Steps:**

1. Import the Project to Android Studio.
2. Enable the Developer option In Mobile device.
3. Enable Developer option in Watch also (Note: you can debug the Android wear through the Bluetooth debugging).
4. Enable the following option (Debugging over Bluetooth) in Android Wear App.

**Reference Link:**

1. [**http://developer.android.com/training/wearables/apps/bt-debugging.html**](http://developer.android.com/training/wearables/apps/bt-debugging.html)
2. [**https://medium.com/@jcdelvalle/android-wear-bluetooth-debugging-quickly-guide-ef279b84169c#.lavbvami1**](https://medium.com/@jcdelvalle/android-wear-bluetooth-debugging-quickly-guide-ef279b84169c#.lavbvami1)
3. Please connect your mobile to your PC/laptop.
4. Once you connected your device type the following comments in Android Studio Terminal.

****

**Command list:**

1. adb forward tcp:4444 localabstract:/adb-hub
2. adb connect localhost:4444/adb connect 127.0.0.1:4444
3. Once you entered the following commends in terminal you will get notification in your Android Wear. The notification belongs to the device authorization. So pleas click ok.
4. adb devices
5. adb -s localhost:4444 shell
6. pm list packages
7. adb uninstall [package name]/ adb -s localhost: 4444 uninstall [package\_name]

Backend Its Firebase.

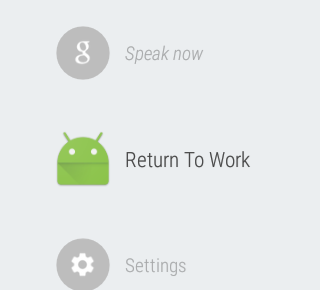
* Go to the <https://www.firebase.com/>
* Create the backend Application.
* The Application helpful to store your Application data’s.

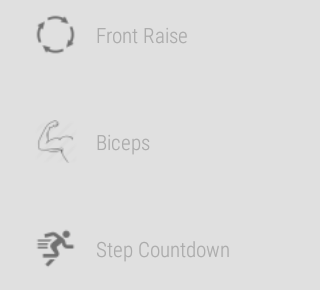
**Sample Application URL:**

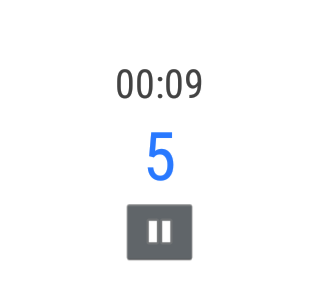
URL: <https://returntowork.firebaseio.com/>

1. Android Wear Screens.

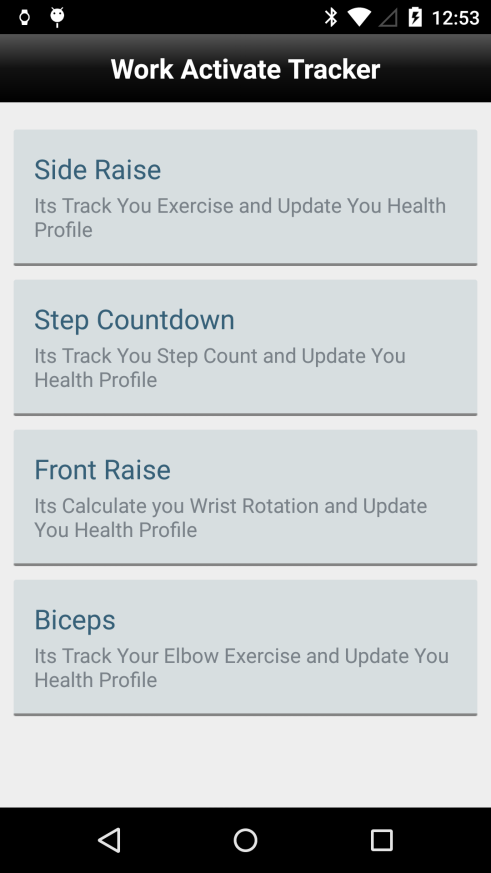
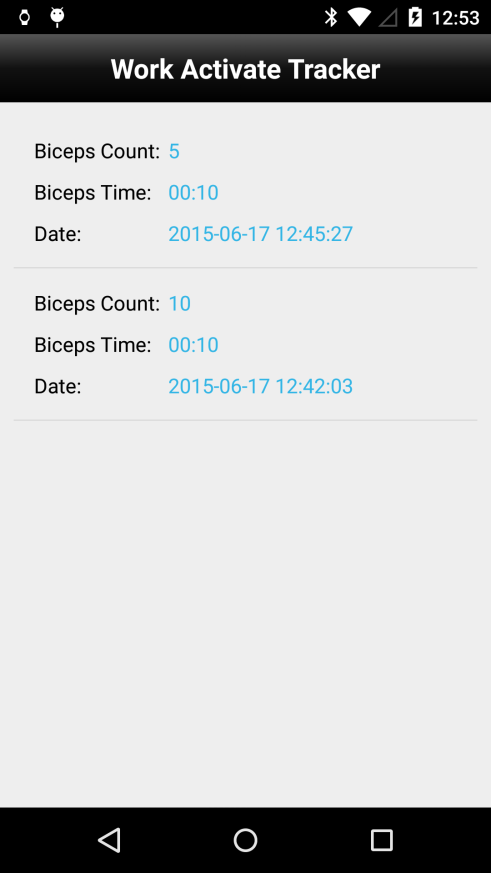
* SDK Version 22.
* We developed the Application both Round and Square Screens(Note: We are not tested in Square watch faces).
* We used “WatchViewStub” for both screen resolution.







1. Android Wear Companion Mobile APP.

* The Firebase application link mentioned in “RTWFireBase.Java” in db package.

Suppose if you choose the Biceps exercise in Android Wear should be open the biceps Activity in companion App. We consider 10 sec is one session. So every 10 second the session will be automatically stopped. We update the backend (Firebase) from the companion App.

1. APK Files.

<https://www.dropbox.com/s/d6hb7aq5vhmbp71/mobile-release.apk?dl=0>

1. Android Wear Source Files.

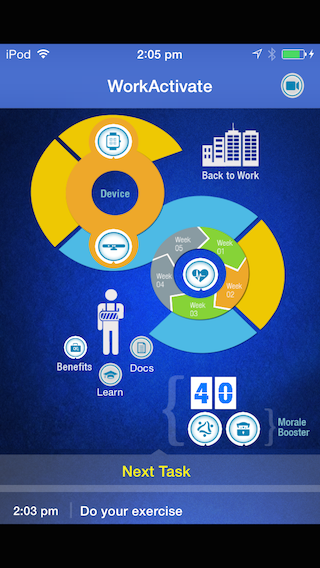
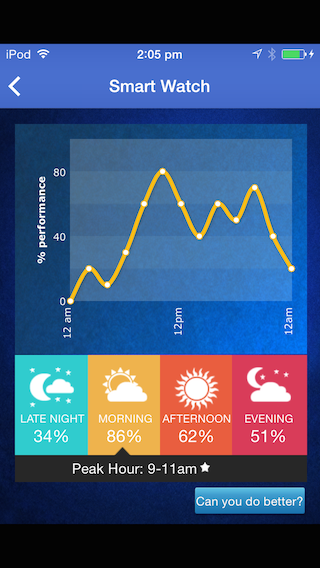
<https://www.dropbox.com/s/7njt68vtgx82ym9/RTWCWear.zip?dl=0>

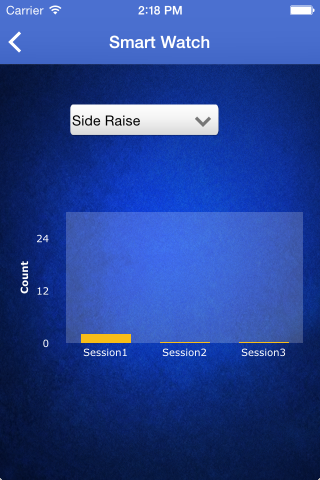
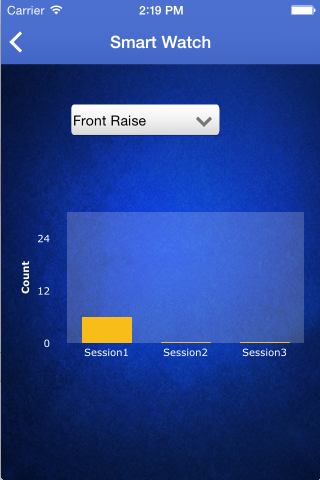
1. IOS APP Screens.

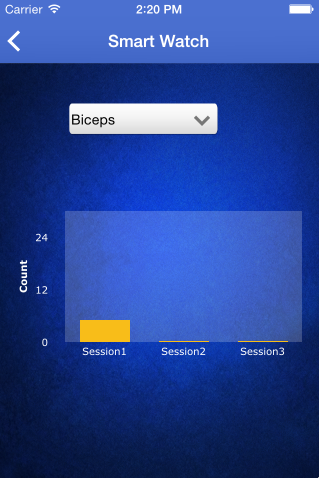
Step1: Click the Smart Watch icon to the below Screens.

Step2: Click the “Can You Do Better” button in second screens.

Step3: Choose you exercise and see your progress.



1. IOS Application Source Code.

<https://www.dropbox.com/s/hrfs9u9hqg8tm1k/RTWCC_code.zip?dl=0>

1. IPA File for the IOS Application.

<https://www.dropbox.com/s/qief44lx2logq8i/RTWCC.zip?dl=0>