Integration Document- Tele-Health Services (Android) – PI-17.4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 1.0 | 4/08/2017 | Spoorti Hallur | All | Release for 17.4.1 |
| 1.2 | 5/09/2017 | Spoorti Hallur | All | Release for 17.4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Administrative Information | | | |
| Project Name | CoCo Telehealth-services | Project Identification | NA |
| Scrum Master | Mahabaleshwara Adiga | Date | 5/09/2017 |
| Quality Leader | Vinay Nair | Form Filled by | Spoorti Hallur |
| Name  Address  Phone  Fax | All Projects in CDP2 | | |

|  |  |
| --- | --- |
| **Authorization** | |
| Quality Leader Signature | Scrum Master Signature |
| Vinay Nair | Mahabaleshwara Adiga |

**For Code:**

**Product : Telehealth Services**

**Release Version :** **v0.0.1 (Android)**

**Release Date: 30-July-2017**

User manual (Optional): NA Customer Version ID (Optional): NA

Contents

**[Authorization](#_Toc494201869)** [1](#_Toc494201869)

[**Introduction** 3](#_Toc494201870)

[**Prerequisites** 3](#_Toc494201871)

[**Library Integration** 4](#_Toc494201872)

[**Maven repository Integration** 4](#_Toc494201873)

[**Library Integration** 4](#_Toc494201874)

[Telehealth Dependency to be added by Propositions 6](#_Toc494201875)

[**User Registration** 6](#_Toc494201876)

[**Telehealth standard micro-app interfaces to be implemented by Verticals are as follows** 6](#_Toc494201877)

[Reference **APP** link 6](#_Toc494201878)

[**Notes** 6](#_Toc494201879)

# **Introduction**

Telehealth services is built as a common-component (micro-app). The purpose of this micro-app is to facilitate end-to-end telehealth consultation the user. Telehealth brings in a unique F2F experience for the user for consulting the doctor remotely.

The micro-app is build using a 3rd party Library (Amwell). Amwell SDK provides the necessary tools to build the micro-app and gives the control over the app for look and feel of app and the way the app operates.

This document provides an overview of integrating Telehealth Services Microapp in Android mobile application.

**Source Path:**

http://tfsemea1.ta.philips.com:8080/tfs/TPC\_Region24/CDP2/\_git/ths-android-telehealth-app

# **Prerequisites**

1. Vertical project is configured for Android Studio
2. Setting->Developer Options->Don’t Keep Activities should be unchecked. [If Developer Mode is on]
3. Android API version should on 21



# **Library Integration**

### **Maven repository Integration**

The easiest and preferred way to use these components is using maven. Only we need to add maven repositories in app build gradle:

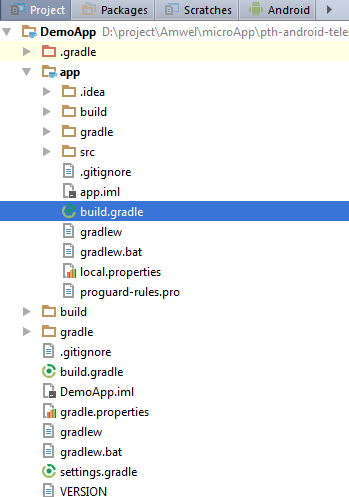
allprojects {  
 repositories {  
 maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/jcenter' }  
 maven {  
 url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/ext-release-local'  
 }  
 maven {  
 url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-release-local-android'  
 }  
 maven {  
 url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-stage-local-android'  
 }  
 flatDir {  
 dirs 'libs'  
 }  
 }  
}

# **Library Integration**

1. Checkout the code from above path wherein developer can find sample app which depends on Telehealth Services library’s aar file :

http://tfsemea1.ta.philips.com:8080/tfs/TPC\_Region24/CDP2/\_git/pth-android-telehealth-app

1. Telehealth Services needs other libraries to build which are as below
2. User registration
3. App-Infra
4. Please make sure all these are added to application with the latest versions along with Telehealth library aar file under libs folder.
5. Clean project
6. Refer Telehealth Services dependencies along with .aar files in build.gradle as mentioned below :



**Note**: Since Telehealths Android was built with

**com.android.support:appcompat-v7:25.3.1**

So vertical app support libs should be greater or equal to mentioned versions



### Telehealth Dependency to be added by Propositions

   def thsModule = module(group: 'com.philips.cdp', name: 'telehealth', version: versions.ths\_telehealth+ objcdp.getVersionSuffix(), ext: 'aar')

                compile(dp ? project(':ths') : thsModule) {

                                transitive = true

                }

# **User Registration**

User Registration HSDP flow is used to get HSDP token. The token along with the uuid of the user are used for authentication with Amwell server

### **Telehealth standard micro-app interfaces to be implemented by Verticals are as follows**

1. The below files are supposed to be added in assets folder
   1. ADBMobileConfig.json
   2. AppConfig.json
   3. logging.properties
2. Propositions must initialize App-Infra in the Application class and inject to telehealth services as a dependency through microapp interface
3. Propositions must initialize Registration and launch Amwell component post the User Login. This is mandatory else the application crashes
4. Propositions must implement ActionBarListener for handling the back navigation
5. Propositions must handle Back navigation only when micro-app passes “true” through handleBackEvent
6. Use the below code for launching the TeleHealth Componant

PTHMicroAppLaunchInput = new THSMicroAppLaunchInput("Launch Uapp Input"); PTHMicroAppInterface = new THSMicroAppInterface();  
PTHMicroAppInterface.init(new THSMicroAppDependencies(((THSDemoApplication) this.getApplicationContext()).getAppInfra()), new THSMicroAppSettings(this.getApplicationContext()));  
PTHMicroAppInterface.launch(fragmentLauncher, PTHMicroAppLaunchInput);

1. **For more details please refer the Demo-App of Telehealth services or Reference app**.

### Reference **APP** link

<http://tfsemea1.ta.philips.com:8080/tfs/TPC_Region24/CDP2/_git/rap-android-reference-app>

# 

# **Notes**

1. Registration is developed as separate library project. App is expected to invoke registration library before calling Telehealth component.
2. Library should be initialized as per document or sample app otherwise library would throw runtime exception.
3. Please refer sample application for more details
4. All dependencies can be referred as it is.
5. Configuration can be followed [all sample configurations are provided in sample]