

Digital Care Android Integration

Document History				
Version	Date	Author	Section	Changes
1.0	10-07-2015	Deepthi Shivakumar	All	Release 2 features
1.1	10-08-2015	Deepthi Shivakumar	Modified Integration section	Modified gradle dependencies section and added more clarity for initialization steps.
1.2	30-09-2015	Deepthi Shivakumar	Modified Integration section	Modified bazaar voice related integration APIs.
1.3	04-12-2015	Deepthi Shivakumar	Modified Integration section	Modified changes for 15.2 PI release.
1.4	01-03-2016	Deepthi Shivakumar	All	Modified changes for 16.1 PI release.
1.5	07-04-2016	Ritesh Jha	All	Modified changes for 16.1 PI release.
1.6	27-04-2016	Ritesh Jha	All	HotFix on v4.1.0 (16.1 PI). URL formation logic is changed for FAQ
1.7	16-05-2016	Ritesh Jha	All	InApp FAQ feature

Author	Deepthi Shivakumar, Ritesh Jha
Approved by	Aravind.Gundumane@philips.com Bhargavi.Upadhya@philips.com
Email Id	Deepthi.Shivakumar@philips.com Ritesh.Jha@philips.com

1. INTRODUCTION	3
2. INTEGRATION	3
2.1 Maven repository Integration	3
2.2 Library Integration	3
2.3 Library versioning	4
2.4 Root gradle changes	4
2.5 Gradle dependencies	4
2.6 Proxy dependencies	4
2.7 Configuration File	4
2.8 Prerequisites	5
3. INITIALIZATION	5
4. HOW TO INVOKE DIGITAL CARE AS ACTIVITY?	6
5. HOW TO INVOKE DIGITAL CARE AS FRAGMENT?	6
6. ANDROID MANIFEST CHANGES	7
6.1 OTHER USER PERMISSIONS	8
7. DIGITAL CARE CONFIGURATION	9
7.1 Main menu configuration	10
7.2 Product Menu configuration	11
7.3 Social provider configuration	12
8. HANDLING CONSUMER CARE AS FRAGMENTS:	13
9. HANDLING CONSUMER CARE AS ACTIVITY	13
10. VERTICAL FEATURES CUSTOMIZATION	13
11. ACTION BAR CUSTOMIZATION	15
12. SUPPORTING APPS WITH OVER 65K METHODS	15
13. SUPPORTED LANGUAGES	16
14. FREQUENTLY ASKED QUESTIONS	16
A. HOW ABOUT CUSTOMISING FONTS AND SIZES?	16
15. NOTES	17

1. INTRODUCTION

This document provides an overview of integration procedure for consumer care library in android mobile applications.

Digital care library provides set of standardized consumer care support features which can be used in all Philips apps.

2. INTEGRATION

There are two ways to integrate “Digital Care” library with any Android app.

- i) **Maven repository based:** At compile time, machine has to be connected with Philips network. Do not follow section 2.2
- ii) **Library Integration:** If unable to connect with Philips network then include libraries to your root application. Do not follow section 2.4, 2.5

2.1 Maven repository (Artifactory based) Integration

The easiest and preferred way to use these components is using maven.

All dependent libraries should be downloaded from artifactory.

Artifactory Path:

<http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-release-local-android/com/philips/cdp/digitalCare/5.0.0>

If you are inside Philips network then you can directly refer “**2.5 Gradle dependencies**” section. It will automatically download all nested dependencies from artifactory.

2.2 Library Integration

Need to copy all aar files in libs folder; below are the libraries needed, Please make gradle changes

```
dependencies {
    compile fileTree(dir:'libs', include:['*.jar'])
    compile 'com.google.android.gms:play-services:6.1.+'
    compile 'com.android.support:appcompat-v7:23+'
    compile files('libs/adobeMobileLibrary-v4.5.1.jar')
    compile(name:'localeMatch-v2.0.0', ext:aar)
    compile(name:'digitalCare-v5.0.0', ext:aar)
    compile 'com.mcxiaoke.volley:library:1.0.17'
    compile(name:'prx-v2.0.0', ext:aar)
    compile(name:'uikitlib-v3.1.0', ext:aar)
    compile(name:'productselection-v1.1.1', ext:aar)
    compile(name:'gson-v2.2.2', ext:aar)
}
```

2.3 Library versioning

Library version can be obtained by using below API

DigitalCareManager.getInstance().getDigitalCareLibVersion();

2.4 Root gradle changes

```
buildscript {
    repositories {
        maven { url 'http://maartens-
mini.ddns.htc.nl.philips.com:8081/artifactory/jcenter' }
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.5.0'
    }
}

allprojects {
    repositories {
        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/jcenter' }
    }
}
```

2.5 Gradle dependencies

Just by adding below gradle dependencies, digitalcare and nested possible libraries will be downloaded from artifactory. But it has to be inside Philips network.

```
dependencies {
    compile(group: 'com.philips.cdp', name: 'digitalCare', version: '5.0.0', ext:
'aar'){
        transitive = true
    }
}
```

2.6 Proxy dependencies

Gradle dependencies can get some network/proxy related issues. In order to fix this issue, we are using below proxy settings in gradle.properties of root folder.

```
systemProp.https.proxyHost=42.99.164.34
systemProp.https.proxyPort=10015
```

We are using this proxy settings locally. But Eindhoven, does not use above proxy settings.

2.7 Configuration File

1. Go to sample app's res/values folder and copy digitalcare_config.xml and place it in integrating app's res/values folder.
2. Override the values in config file and define the features as per requirement.

2.8 Prerequisites

Note: Product should be hosted at PRX server. Please read the process in doc “**ConsumerCare_Deployment**” which is shared along with other documents.

- i) Application need to know **CTN** number for each product used in app. Ex: HD8967/01. Max CTN is allowed.
- ii) Sector
- iii) Catalog.

Example: **Sector:** B2C and **catalog:** CARE (Mandatory for Consumer care)
Most of the apps will have above sector and catalog.

Note: Above information is used for backend services and mostly uses PRX system. Hence Please provide valid and complete CTN number as input otherwise consumer care features are not functional as expected.

3. INITIALIZATION

The library can be invoked both as activity and fragment.

“**DigitalCareConfigManager**” is singleton class used for communication between application and library. It exposes few public setter APIs. The application has to set few below parameters dynamically before the library is invoked. Please refer sample application.

- a. Import DigitalCareConfigManager from “com.philips.cdp.digitalcare”
- b. initializeDigitalCareLibrary(Context applicationContext)
- c. App locale (language code and country code).

```
PILLocaleManager localeManager = new PILLocaleManager(this);  
localeManager.setInputLocale(language, country);
```

Also set this whenever there is a change in language or country. Call in appropriate places.

```
API: localeManager.setInputLocale(language, country);
```

- d. Call **Tagging** related APIs as below,

```
public void setAppTaggingInputs(boolean taggingEnabled, String appId, String  
appName, String launchingPageName)
```

If app enables tagging and appid is not set, then digital care library will throw runtime exception.

Please note that appid is nothing but appsId.

Note: Until app enables tagging, consumer care component will not tag user actions.

“ADBMobileConfig.json”: This json file is tagging configuration file which has to be kept at root app asset level. Please don't forget to add this. Otherwise by default it will consider digitalcare's debugging tagging configuration.

Please follow the order of initialization as per sample app.

4. How to invoke Digital care as activity?

Please note that there is an API change for invocation. Now there is a single API to launch as activity and fragment.

invokeDigitalCare(UiLauncher uilauncher, ProductSelectionType productSelectionType)

Here UiLauncher is a base class which has 2 sub classes called ActivityLauncher and FragmentLauncher

ProductSelection type denotes the kind of inputs to support multiple products.

Please note that we currently support hardcoded list of products.

- a. Create an instance of activity launcher and insert activity orientation type and UiKitTheme as parameters. Also create productselection type as below,

```
HardcodedProductList productList = new HardcodedProductList(ctnList);
productList.setCatalog(Catalog.CARE);
productList.setSector(Sector.B2C);
ActivityLauncher uiLauncher = new
ActivityLauncher(ActivityLauncher.ActivityOrientation.SCREEN_ORIENTATION_UNSP
ECIFIED, R.style.Theme_Philips_BrightBlue_Gradient_WhiteBackground);
uiLauncher.setAnimation(R.anim.slide_in_bottom, R.anim.slide_out_bottom);
DigitalCareConfigManager.getInstance().invokeDigitalCare(uiLauncher, productList);
```

Note: ctnlist is string array of CTN numbers

5. How to invoke Digital care as Fragment?

Please note that there is an API change for invocation. Now there is a single API to launch as activity and fragment.

invokeDigitalCare(UiLauncher uilauncher, ProductSelectionType productSelectionType)

Here UiLauncher is a base class which has 2 sub classes called ActivityLauncher and FragmentLauncher

ProductSelection type denotes the kind of inputs to support multiple products.

Please note that we currently support hardcoded list of products.

- a. Create an instance of activity launcher and insert activity orientation type and UiKitTheme as parameters. Also create productselection type as below,

```

HardcodedProductList productList = new HardcodedProductList(ctnList);
productList.setCatalog(Catalog.CARE);
productList.setSector(Sector.B2C);

FragmentLauncher fragLauncher = new FragmentLauncher(
    this, R.id.sampleMainContainer, ActionBarClickListener);
fragLauncher.setAnimation(R.anim.slide_in_bottom, R.anim.slide_out_bottom);
DigitalCareConfigManager.getInstance().invokeDigitalCare(fragLauncher, productList);

```

Note: ctnlist is string array of CTN numbers

ActivityOrientation is an enum to specify orientation, if do not want to fix orientation, please set as "unspecified"

Supports all kind of orientation in ActivityInfo class.

ActivityOrientation is present inside DigitalCareConfigManager class.

Animation resource id can be passed in as zero if not relevant.

Animation resource id can be passed in as zero if not relevant.

6. Android Manifest Changes

1. We are using UIKit library and it expects theme to be set which is equal to or higher than appcompat theme.

```
android:theme="@style/multiprod_theme"
```

We have defined sample uikit theme in our sample application.

2. If apps face problems wrt label , theme etc during compilation, Please do the below,

Example from Registration sample app:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.philips.cdp.registration.sample">

```

```
<application
```

```

    android:name="com.philips.cdp.registration.sample.RegistrationApplication"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme"
    tools:replace="android:icon,,android:label,android:theme">

```

3. This change is required for Locale Philips Near you feature to display google maps. Please find more details below regarding debug and release keys which need to be used.

Application needs to copy below google map related meta data information in app's manifest file and **must provide new SHA key** for rendering Philips service centers on google map.

```
<meta-data
android:name="com.google.android.gms.version"
android:value="@integer/google_play_services_version" />
```

```
<meta-data
android:name="com.google.android.maps.v2.API_KEY"
android:value="AlzaSyDm1M6rUwuCe_-4pBM61QeWivE6Glu2hWM" />
<uses-feature
android:glEsVersion="0x00020000"
android:required="true" />
```

AlzaSyDm1M6rUwuCe_-4pBM61QeWivE6Glu2hWM = SHA key

Reference for SHA key generation

<https://developers.google.com/maps/documentation/android/signup>

SHA key should be generated from the system where apk is released.
SHA key along with package name should be registered.

```
//Credentials For LocatePhilips Near you feature (Google Map)
<usespermission
android:name="com.philips.cdp.sampleddigitalcareapp.permission.MAPS_RECEIVE" />
<permission

    android:name="com.philips.cdp.sampleddigitalcareapp.permission.MAPS_RECEIVE"
    android:protectionLevel="signature" />
```

Please note that it works without the above permission for debug keys but we need to add it when we use release keys. Please make sure to change package name.

6.1 Other User Permissions

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
<uses-feature android:name="android.hardware.telephony" android:required="false"/>
```


7. Digital care configuration

The application has to copy digitalcare_config.xml from sample app resource folder and create one on app side by overriding the parameters present in it. The file contains the following,

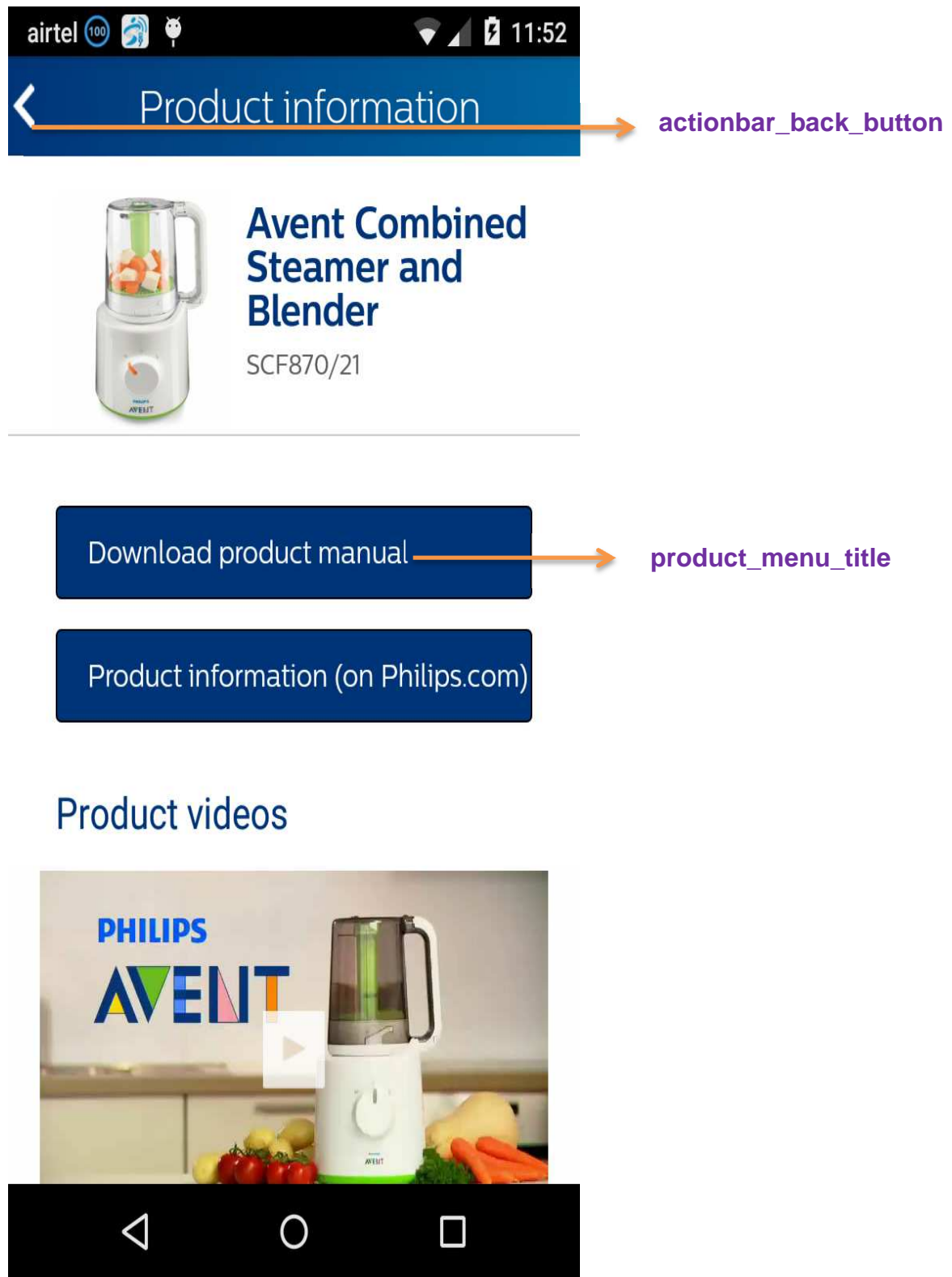
- a. UI resources like themes, styles, background colors and action bar customization.
- b. Facebook & Twitter Philips product page url which is app/product specific.
- c. An array is defined for customizing main menu title, main menu icons, product menu title and social provider list. Each menu contains default features supported by Digital Care library. App can include only the required features by adding/deleting/modifying the array items.
- d. Each title string in above arrays is expected to be a resource key and not the value.
- e. Production environment is introduced as Boolean parameter. It allows to set either as production or test environment. It is applicable for all the backend systems which we use.

The digitalcare_config.xml on the library side will have default values. In case the app does not override, all the resources would be picked from library. Main menu configuration

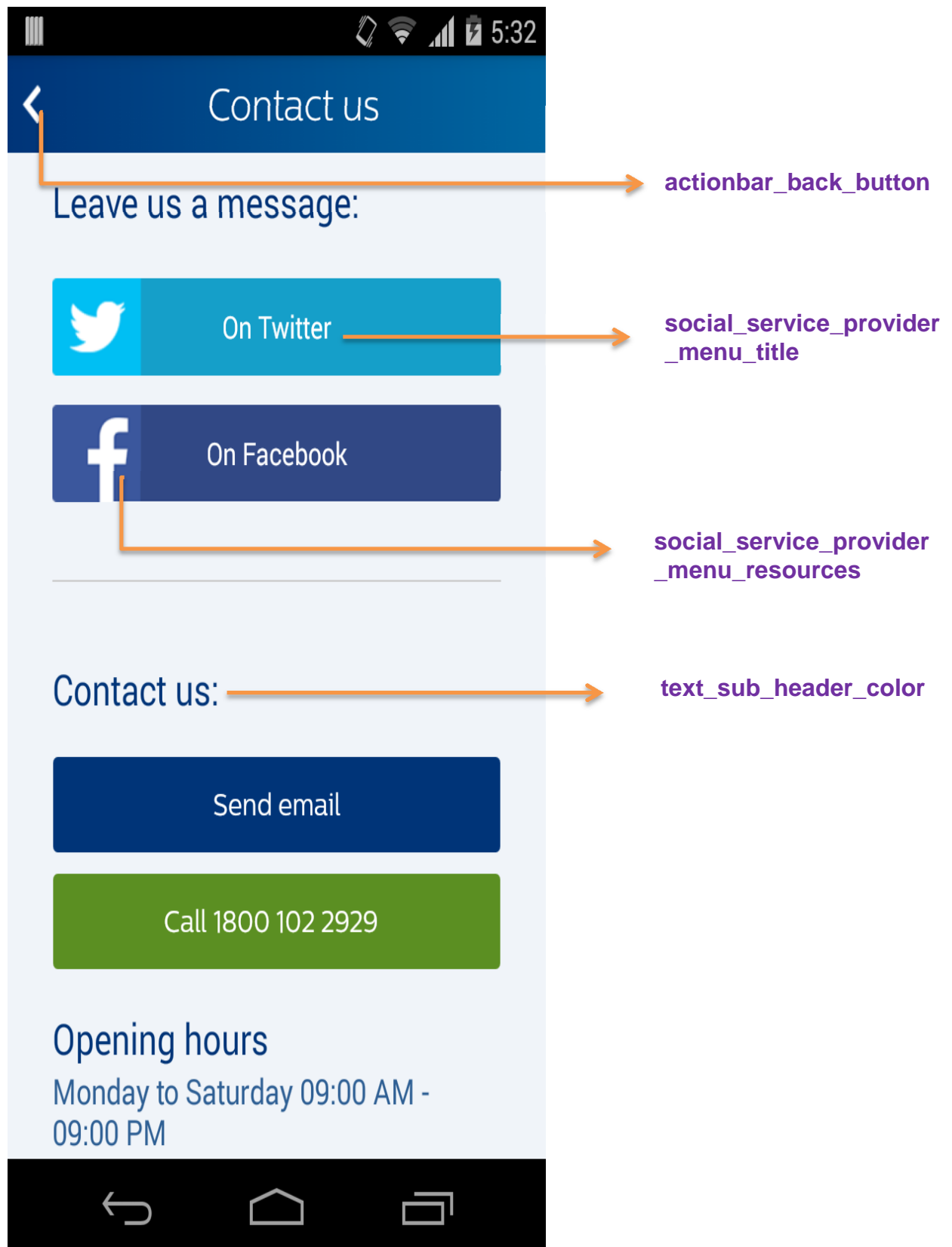
7.1 Main menu configuration



7.2 Product menu configuration



7.3 Social provider configuration



8. Handling consumer care as fragments:

If it is invoked as fragment, action bar has to be handled by verticals because they have control.

- a. Adding cc fragments will be taken care by library.
We define one interface called “ActionBarUpdateListener” and it will have method called `updateActionBar (String titleActionBar, Boolean hamburgerIconAvaialable)`.
- b. We invoke this callback and verticals have to implement this listener and update the action bar accordingly.
- c. On click of back arrow app has the control again and it is responsible for popping out the fragment from the stack. Consumer care has no action over it.
- d. Handling hamburger menu icon and anything related to action bar is the responsibility of the app.

9. Handling consumer care as activity

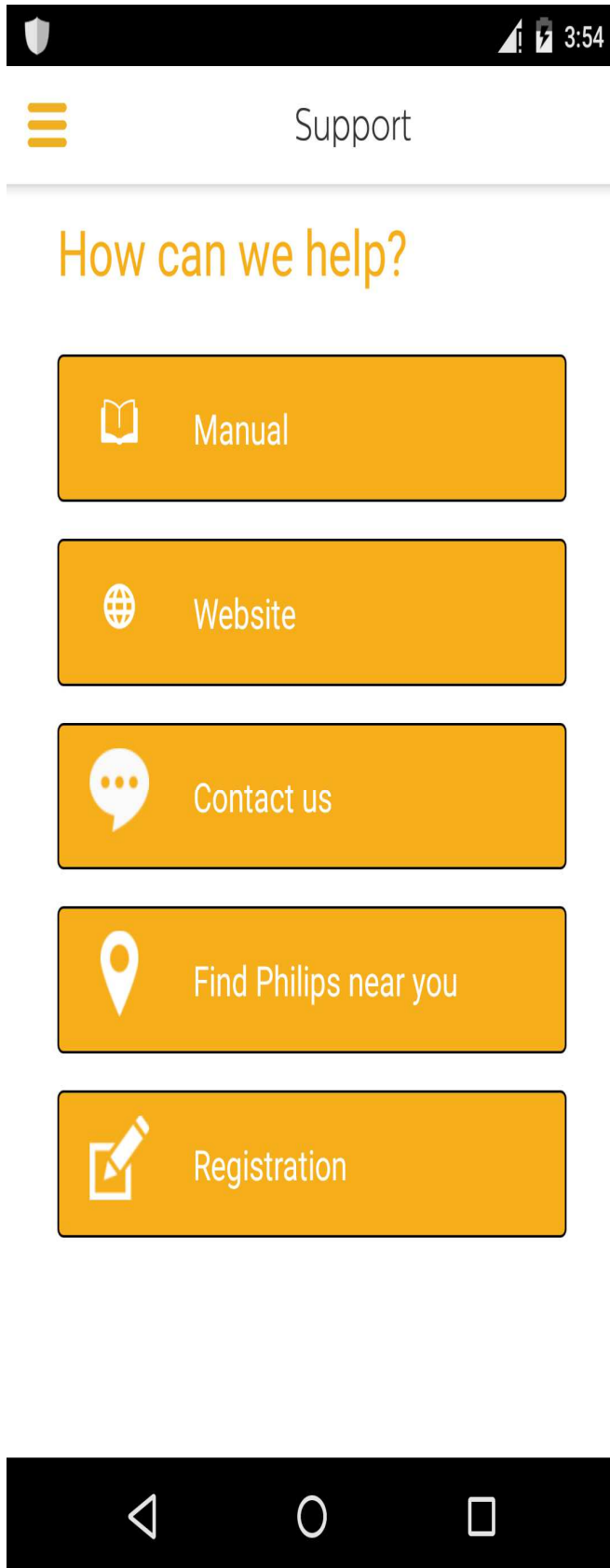
If it is invoked as activity, then CC will be launched as full screen and also responsible for updating action bar. Action bar customization happens depending on the configuration set by app in config xml file

10. Vertical features customization

The consumer care features are highly customizable; each vertical can add any specific button inside support menu if it is really required.

- a. App can add its own button by passing valid string for button title and icon resource name in configuration file.
- b. Application has to implement interfaces like MainMenuListener, ProductMenuListener, and SocialProviderListener.
 - i. MainMenuListener – Invoked on click of each button on support home page. This helps in configuring features on main/home page.
 - ii. ProductMenuListener – Invoked on click of each button on product info page. This helps in configuring features on product info page.
 - iii. SocialProviderListener – Invoked on click of each social provider buttons in Contact us page. This helps in configuring list of social list providers. CC by default supports Facebook and Twitter.
- c. App is required to implement all these to take control on app side for the menu items which is added by verticals. For digital care menu item, library is supposed to take action and app should not ideally override this. If app returns “true” then it means that app will take action on click of that button and library will not perform any operation in that case otherwise library will take action if app returns “false”.

Example: Air Fryer customization



11. Action bar customization

Here is how we customize navigation/action bar in consumer care module,

1. We allow customization of both Hamburger menu icon and back arrow button in navigation bar.
2. Projects which uses navigation bar will go for above option.
3. Projects where in navigation bar is not used, will have to do the following
 - a. Replace hamburger menu icon with back arrow icon.
 - b. Set navigation bar background color same as screen background color.
 - c. Sets a Boolean in config file which says hamburger menu title required as "False", in this case navigation bar title will not be shown in consumer care.
4. Projects which uses spring board, replaces hamburger menu with spring board icon.

12. Supporting apps with Over 65K Methods

This is special case if app which has more than 65K methods do follow below link for reference.

<https://developer.android.com/tools/building/multidex.html>

According to this do following changes in gradle and application class

In Gradle file:

```
android {  
    compileSdkVersion 23  
    buildToolsVersion "23.0.1"  
  
    defaultConfig {  
        ...  
        minSdkVersion 14  
        targetSdkVersion 23  
        ...  
    }  
  
    // Enabling multidex support.  
    multiDexEnabled true  
}  
  
dependencies {  
    compile 'com.android.support:multidex:1.0.1'  
}
```

In Application Class:

```
@Override  
public void onCreate() {  
    MultiDex.install(this);  
    Super.onCreate();  
}
```

13. Supported Languages

English	German	French (FR)	Simplified Chinese	Traditional Chinese	Portuguese (EU)
Russian	Arabic	Japanese	Dutch	Malay	Montenegrin
French-CA	Romanian	Spanish Mexican	Ukrainian	Portuguese (BR)	Serbian
Gaelic	Greek	Hebrew	Hindi	Finnish	Swedish
Italian	Polish	Spanish	Korean	Czech	English UK
Norwegian	Albanian	Bosnian	Bulgarian	Croatian	Estonian
Hungarian	Indonesian	Kazakh	Latvian	Lithuanian	Macedonian
Slovak	Slovenian	Spanish (AR)	Thai	Vietnamese	Danish
Farsi	Turkish	Hongkong-China			

14. Frequently asked Questions

a. How about customising fonts and sizes?

We use Central Sans Font as per Philips design guidelines. Both fonts and sizes cannot be configured as of now.

b. Can we add our own buttons inside consumer care?

Yes. However we have restricted this to main menu screen, product menu and list of social service providers list.

c. Does Consumer care support App Tagging?

Yes. Please make sure you call enableTagging and provide apps ID. Please check sample app for more details. Unless App calls enableTagging () API tagging will not be enabled inside consumer care.

d. Where do I find my product information?

Each business has to provide complete ctn number including slash

Ex: HD9240/90

With this please check below link which helps in finding out product tree information like category, sub category etc. please add relevant ctn number

- a. http://www.philips.co.uk/prx/product/B2C/en_GB/CONSUMER/products/HD9240/90.summary
- b. http://nlvu077.gdc1.ce.philips.com:9080/repobrowser/catalogBrowser.jsp?catalogid=catalog_CL_CONSUMER&catalogType=CONSUMER&country=CL&language=es

Please replace locale to check whether product data is available in corresponding countries.

e. Does Consumer care support multiple products?

Yes it supports multiple products from 16.1 release which has version 4.0.0.

Currently app can send hardcoded list of products as input after which consumer care will take care of selection of product.

f. How to fix below fatal exception?

com.android.build.api.transform.TransformException:

com.android.builder.packaging.DuplicateFileException: Duplicate files copied in APK lib/mips/libblasV8.so

Or any duplication related exception, the duplication library can be added to packageOption.

```
android {  
    packagingOptions {  
        exclude 'lib/armeabi-v7a/libblasV8.so'  
    }  
}  
  
android {  
    packagingOptions {  
        exclude 'lib/arm64-v8a/librsjni.so'  
        pickFirst 'lib/arm64-v8a/libRSSupport.so'  
    }  
}
```

15. Notes

- a. Registration is developed as separate library project. App is expected to invoke registration library on click of registration button inside consumer care support page.

- b. Library should be initialized as per document or sample app otherwise library would throw runtime exception.
- c. Developer option “don’t keep activities” should be disabled.
- d. Please refer sample application for more details.