Digital care

Design Document

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 0.1 | 28 Nov 2014 | Kaushik C S | All | Initial version |
| 0.2 | 5 Dec 2014 | Kaushik C S | All | Incorporated Review comments on the initial version from Aravind G |
| 0.3 | 8 Dec 2014 | Kaushik C S | All | Incorporated Review comments on v0.2 from Aravind G |

|  |  |
| --- | --- |
| **Author** | Kaushik C S |
| **Approved by** |  |
| **Email Id** | [Kaushik.murthy@philips.com](mailto:Kaushik.murthy@philips.com) |

CONTENTS

1. DEFINITIONS & ABBREVATIONS 3

2. INTRODUCTION 3

2.1 Purpose 3

2.2 Scope 3

2.3 Target Audience 3

2.4 References 3

3. ARCHITECTURAL ANALYSIS 4

3.1 Product Overview 4

3.2 System Context 4

3.2.1 App to PRX communication 4

3.2.2 App to Janrain communication 4

3.3 Design Method 5

4. STATIC DESIGN 6

4.1 Layering Model 6

4.1.1 View 6

4.1.2 Controller 6

4.1.3 Model 6

5. Decomposition / Sub Systems 7

5.1 Configuration Manager 7

5.2 User Registration 8

5.3 Product Registration 9

6. UI/UX Design 10

6.1 Android Design 10

# DEFINITIONS & ABBREVATIONS

|  |  |
| --- | --- |
| DI | Digital Innovation |
| MVC | Model View Controller |
| CDLS | Contact Data Lookup service |

# INTRODUCTION

This document provides architectural and top level design for Digital Care application (termed as library as well) developed on Android and iOS. The various building blocks of the application are depicted in the document.

## Purpose

This document serves as the base document for designing the individual component and implementation.

## Scope

This document covers the design aspects of Digital Care mobile app.

## Target Audience

*PIC – DI development team*

## References

|  |  |
| --- | --- |
| Design | Care in apps – design specifications v1.pdf |
| Overview – Feature | Digital consumer care app functionality overview\_updated.xlsx |

# ARCHITECTURAL ANALYSIS

## Product Overview

The main functionality of this product/library is to cater common features used across different applications. This application/library can be reused by other project with minimal development changes.

## System Context

The following diagram shows the Eco-System of Digital Care application.

The Digital Care application will talk to different services as shown in the below picture (Fig. 3.2).



Figure 3.2: Digital Care Eco-System

### App to PRX communication

Digital care application will communicate with PRX web services for fetching “Contact Us” (CDLS API) information of a product and also for registering the product.

The communication method used is “http”.

### App to Janrain communication

Janrain is used to user registration/login and product registration. Janrain provides SDK which exposes the interfaces to communicate with Janrain server.

## Design Method

The design used Object Oriented concepts which are supported by Android and iOS.

The app follows the layered approach. Following rules are enforced for how the layers interact with each other.

* **One way interaction (Top to down)**: Higher level layers can interact with layers below, but lower layers cannot interact with layers above. This is to avoid circular dependencies between layers. We will be using observer pattern to notify changes in the model back to the user interface
* **Strict interaction**: Layers can only interact with layers below. This will make sure that modifications in one layer will only affect layers above.

The focus of the design is to reduce make this App/library reusable across various products.

We are de-coupling the different features in the Application in order to increase the reusability and reduce the size of the application by removing unused features.

# STATIC DESIGN

## Layering Model

The figure 4.1 below shows the top level structure of Digital care library. It is a three layer architecture that conforms to MVC architecture pattern. User Interface part includes either an activity or a fragment or a **custom** view.

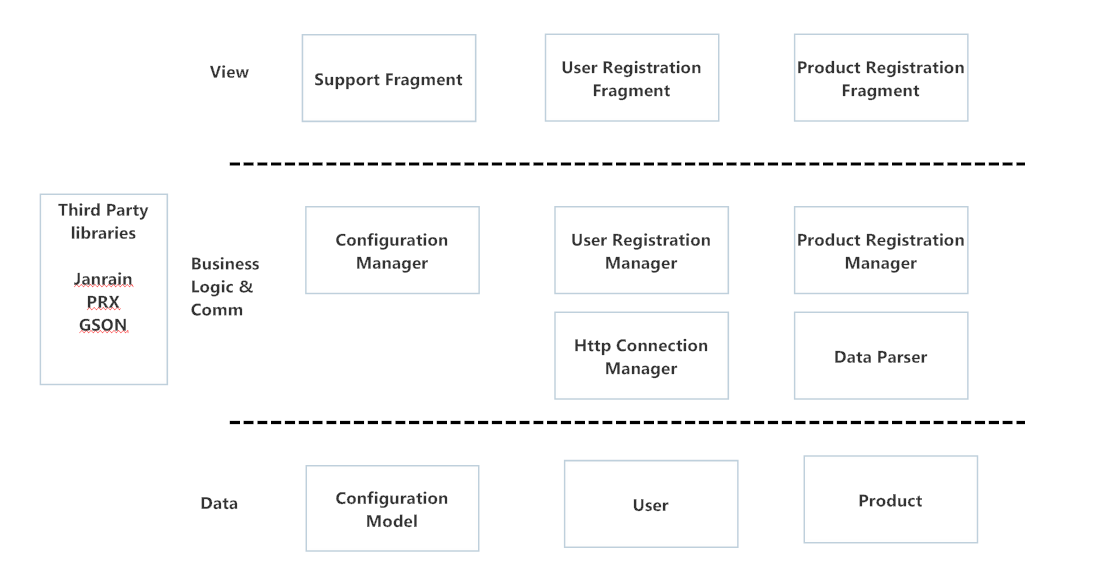


Figure 4.1: High level software decomposition

### View

View is the visual representation of data on the screen. Each screen in Digital Care app/library is considered as view or fragment or activity.

### Controller

Controller does data handling between the view and the model.

### Model

Model is the layer that consists of data.

# Decomposition / Sub Systems

The layered design is decomposed into following major paradigms.

## Configuration Manager

Below picture (Fig 5.1) depicts how the configuration manager interacts with user interface.

Configuration Manager is a static class which will have the Application configuration model.

Digital care library/APP will have an XML or a JSON file stored in the assets folder or project folder.

User interface will talk to Configuration Manager to get the APP configuration model.

 Fig 5.1: App level Configuration Handling Flow

Examples file of configuration json

{

“appconfiguration” : {

“featurelist”: [“contact us”,”product registration”,”faq”,”locate us”],

“ratethisappurl”:” <https://play.google.com/store/apps/details?id=com.philips.cl.di.dev.pa>",

“product name”:”Smart Air”

}

}

## User Registration

User Registration flow has been depicted in the below picture 5.2.

User Registration Manager will be the interface between the User interface and the Registration API.

User Registration manager will talk to Registration API (Horizontal component) which will internally talk to Janrain library. Observer pattern will be used to talk to the upper layer.



Fig 5.2 User Registration flow

## Product Registration

User interface talk to Registration API library which in turn calls PRX web services (HTTP) for registering the product and store the product registration information in Janrain.

Registration API will return the success or failure back to the User interface.



Fig 5.3 Product Registration flow

# UI/UX Design

## Android Design

Below diagram (Fig 6.1) represents the UI/UX design approach for designing the screens which can be configured by a developer.

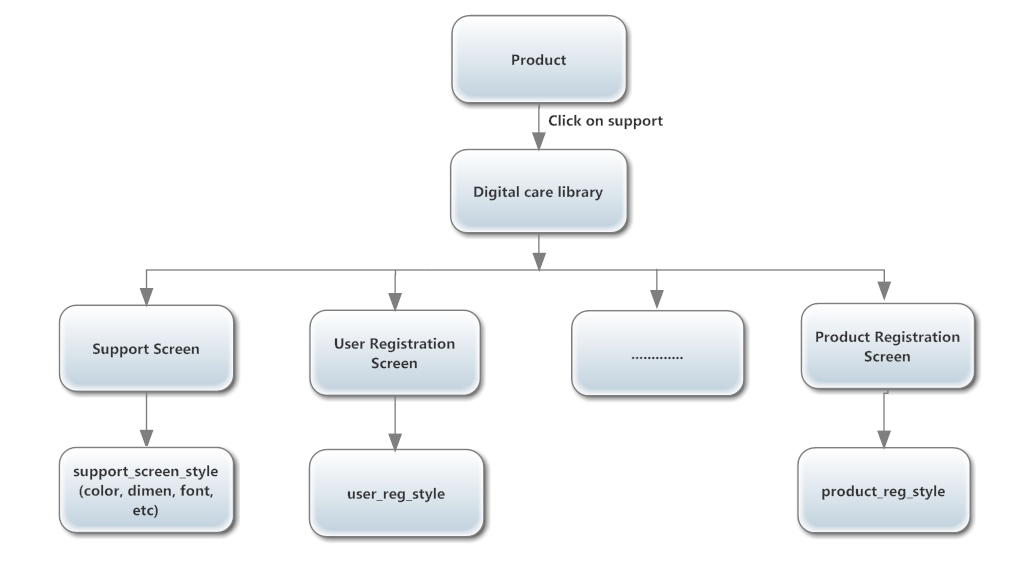


Figure 6.1: Package level UI Handling Flow

Each product will have a support screen; on click of the support screen Digital Care library will get invoked.

We will be having a parent style which is defined for the application. In the parent\_style.xml we specify each style for button, text view, list view and the components used in the Application.

If the button style (Background color or font) across the application/library differs from screen to screen, then we will be having different custom styles defined for each of the screens. Each custom style will extend the parent style. We have represented this pictorially in the below picture 6.1.1

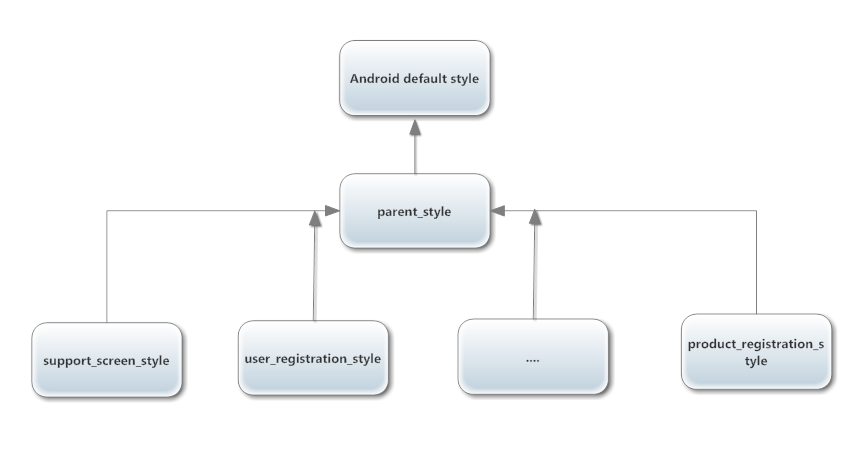


Figure 6.1.1 Digital care style hierarchy