Requirement document and notes alongside

App Features:

* ***MVP*** **design pattern** to isolate UI layer(Android specific) with java specific code
* **Singleton** Design using Enums
* **Parcelable** Beans to pass through activities
* Recycler view and Card View to display list of products
* Database management : **Room persistence** library
* Android 8.0 **Adaptive App Icons**, Android 7.0 round icons with default app icons
* **Dagger 2** for data injection
* **Butterknife** for views binding
* Build flavours and variants:
* Three build flavours : **Debug**(signed with debug keystore), **Beta** and **Release**(signed with release keystore)
* Two build dimensions : **color** and **country**
* Build flavors : Combination of **color** (pale, grey) and **country** (India, USA)
* Variant Filter in gradle with ability not to generate selective builds like beta build for India
* Signing config in app folder with auto signing and ability to generate all combination of build flavours and variants.

PS: This is only for demo that the release keystore and credentials are with the app code. In production code, it is never a good practice to keep them with code.

* **Localization** support for Hindi and English. US builds are English and India builds are hindi. Although if the user chose to, they may switch based on the system language between English and Hindi. App defaults to english in case system language is not supported.
* Programming style followed includes consistent naming conventions.

- xml ui elements are named and ui elements in code are prefixed with chorthand camelcase with the name of the view to its role like a textview for name goes like tv\_name.

- Instance variables and parameters are not prefied with mInstance or pParameter as IDE does the stuff differentiating them pretty well.

* Grey Product flavour loads grey resources, pale loads pale resources.
* Landscape mode and portrait mode are supported with different layouts for different orientations in product list view.
* Ability to pre-populate the debug and beta builds with dummy data of thre products so release builds are not affected.

**Requirements** :

Create a new app that showcases the following.

1. Creation of the model.  :  done using Room Persistance

2. Create mock data for the models using json. Show case at least 3 mock products :  Done for debug and beta builds

3. Use of Persistent Storage to store the model. Implement the following. :  Done using Room Persistence

1. INSERT

2. SELECT

3. UPDATE

4. Have the initial screen contain at minimum buttons that are labeled “Show Product” and “Create Product”.

 I have taken a little liberty here making initial screen contain list of products and add product button. The next screen contains a product details screen where user may add or update a product.

Features not implemented include:

* Product Colors
* Stores
* Product Image
* Deletion of product