Android Developer Test Documentation

The project is to provide ability for customer to search products, display the list product, and on upon product selection from the list show the product detail.

The customers enters the search word and on submit, the product list will be fetch from the back end through API endpoints. The product list is shown to the user in list format. The first 20 records are fetched from the back end. When customer scroll down and at the end of the last item, the next set of 20 records are fetch from the back end.

The list will show product image and product description. The customer can select the product from the list and on selection the product details is shown.

In addition to above features, the app allows the feature for customer to search product by scanning the EAN-13 format product barcode. When the customer scan the barcode the product details are shown to the user.

The above app is developed using Kotlin programming language. The application architecture is is MVVM design.

**Architecture**

**Model**

**Repository**

View Model

View

The following libraries are used in the application.

The Dagger is used for dependency injection in the project,

He Retrofit and okhttp is used for network api calls.

Glide is used to load the image.

Gson is used to convert the json string to model calls.

Zxing is used to achieve barcode scanning,

The Expresso testing libraries for testing.

**Components**

**UI:** The fragment is implemented for UI screen. The fragments are loaded at run time.

Activity:

MainActivity: This activity consists of two buttons. One for product search and another is for barcode scanning.

Product search activity: The product search fragment is attached to the activity at run time. The customer enters the search keyword in the search view. The product list is fetched from the backend for the given search key word. The list of products is shown in recycler view. The elements are product image and product description is shown in the recycler view.

The product detail fragment shows the product details on product selection in the recycler view. When the customer selects the product the barcode is passed to backend to fetch the product details.

The product detail fragment is reused when customer scans the product bar code. The bar code is passed to backend to fetch the product details from the back end.

Barcode scan activity scans the product barcode and pass the value to the product detail fragment.

Permission Activity is implemented to prompt the customer to allow camera access for the app.

**Repository:**

UserRepository is used to fetch user id from the backend. The user id is stored in the shared preference for API calls.

ProductRepository is used to fetch the product list and product detail from the back end.

**Service:**

WarehouseService is the network api end point service class.

**Injection:**

NetworkModule: This class provides the retrofit, and service object as dependency injection.

AndroidModule: This class provides the share preference, UserRepository, ProductRepository and SharedPreferencesFacade objects as dependency injection.

**Facades:**

SharedPreferencesFacade: This class is used id to store and retrieve user id from the storage.