GitHub Username: darinjswilliams@icloud.com

BnForm

Description

BnForm is an application that provides the consumer with the capacity of checking a product for recalls. A consumer can scan the Bar Code or QR Code to verify if a product has been placed on the recall list by the Consumer Product Safety Commission (CPSC). The consumer can also report product defects to the CPSC by scanning the product. The application benefits the consumer by providing information on products before purchase. The scanned information will be sent to the public database maintained by Safer Products. where the reports of harm are reviewed by the U.S Consumer Product Safety Commission agency and manufacturer of the product.

Intended User

The application is intended for all consumers who purchase products.

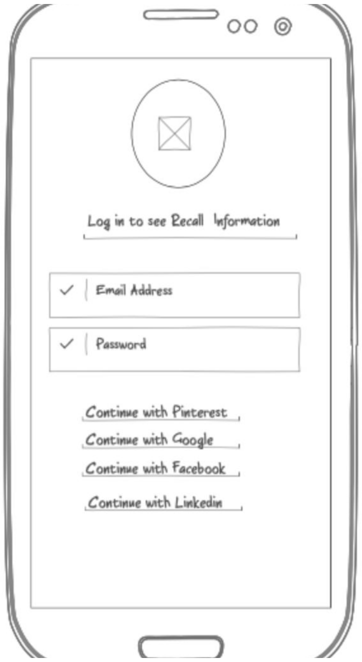
Features

List the main features of your app. For example:

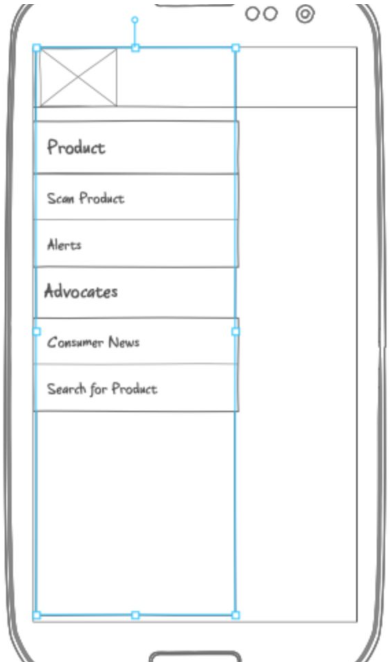
- Saves information
- Takes pictures
- Scan BarCode
- Scan QRCode
- Share product information between consumers
- Submit reports to the Consumer Product Safety Commission
- Widget

User Interface Mocks

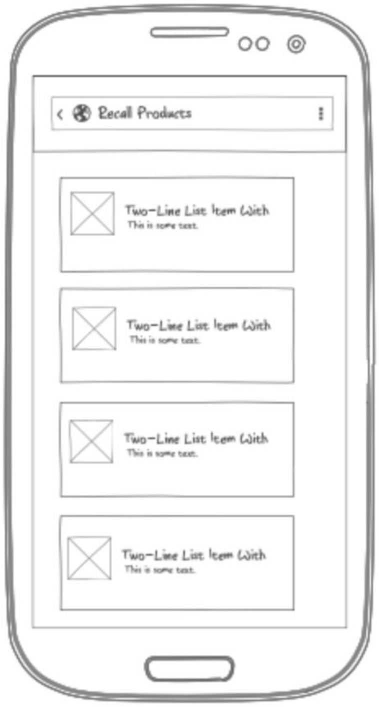
Screen 1 Login



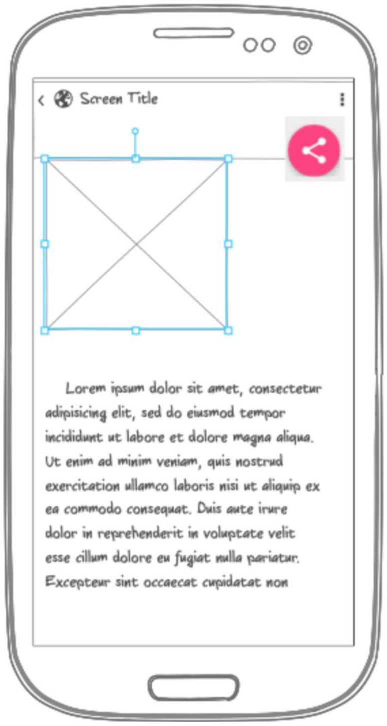
Screen 2 Navigational Draw



Screen 3 Recycler View of Product

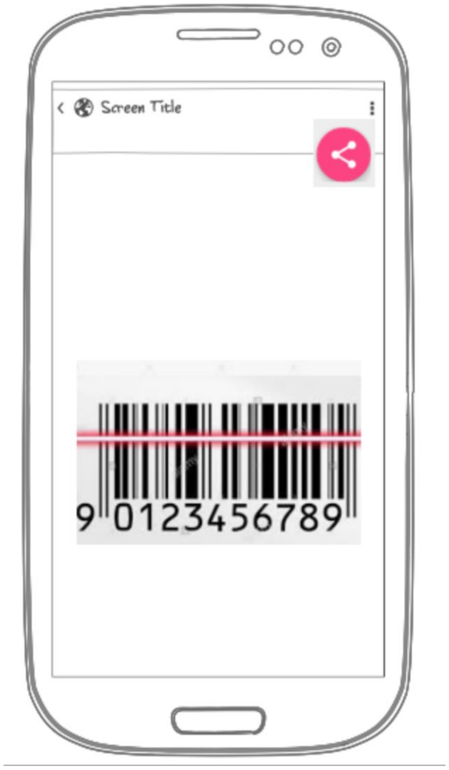


Screen 4 Detail View of Product

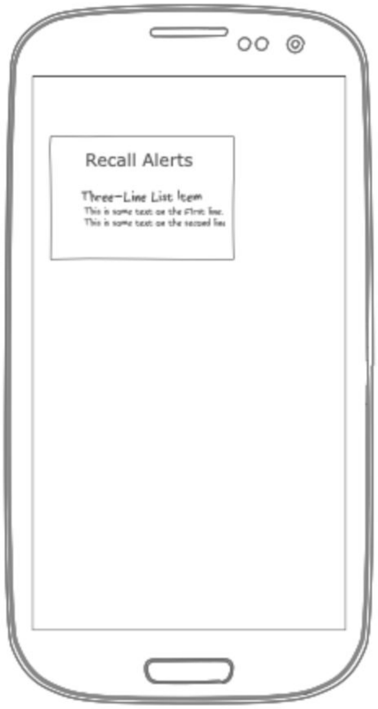


Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non

Screen 5 Scan View of Product



Screen 6 Widget View of Recall Items


Recall Alerts
Three-Line List Item
This is some text on the first line.
This is some text on the second line

Key Considerations

How will your app handle data persistence?

The application will use the room database to persist product information after the item has been scanned. The application will implement share preferences to keep track of application settings.

Describe any edge or corner cases in the UX.

Android Navigational UI components will handle the navigation of the User Interface by using Jet libraries.

- Back Button Press - User return to previous screen. If a consumer continues to press the button they will exit the application.
- Up Arrow Press - Consumer returns to previous screen
- Management of Fragments - Transition of fragments
- Sharing information by using content.

Describe any libraries you'll be using and share your reasoning for including them.

Picasso - handle the loading and caching of images.
JetPack - will handle the management of User Interfaces
Retrofit - management of web-services calls to retrieve data.
Room - utilizing room to manage data persistence
RecyclerView - handle layout of components.
LiveData - to observe lifecycle of application components
Material Design - Provide for standard visualization on application
Safety API - Consumer Safety API to access recall information.
GooglePlay Services - Barcode and QRCode Detection
Expresso - UI testing
MLKIT -BarCode API is used to scan BarCodes and QR-Codes
FireBase Analytics - The analytics is used to Log scan events and sharing events

Application	**Version**
Android Studio	4.0
Gradle	6.1.1

Java	1.8
Library	**Version**
Retrofit	2.8.1
Gson	2.8.6
Picasso	2.71828
Room	2.2.5
Material Design	1.0
Recycler View	1.1.0
Constraint Layout	1.1.3
JetPack	2.2.2
LifeCycle	2.2.0
Expresso	3.2.0
MLKIT:barcode-scanning	16.0.0
FireBase Analytics	17.4.4
API	URL
SafetyAPI	SaferProducts.gov

Describe how you will implement Google Play Services or other external services.

Google Play Services - The application will use Google Play Services to handle scanning of BarCodes and QR Codes.

Firebase Analytics - The application will use Firebase Analytics to logEvents when consumer scan products and share products.

Safety API - is an external api that communicates with the CPSC recall information. The SaferProducts.gov API provides access to all public available information visible on SaferProducts.gov.

- Incident - Incident Report Number, Incident Date, Incident Location, Incident Description
- Manufacturer - Manufacturer Name, Manufacturer Location, Manufacturer Notification Date, Manufacturer comments
- Product - Product Brand Name, Product Model Name, Product Manufactured Date, Product Purchased Date, Incident Product Description
- Retail Company - Retail Company Name, Retail Company Location
- Victim - Victim Gender, Victim Age

Next Steps: Required Tasks

Task 1: Project Setup

Implement initial configuration of application by using standards guidelines.

- Create Project Framework
- Application stores all strings in string.xml resource file
- Update Gradle with dependencies by adding all libraries as indicated in Library section
- Create github Repository
- Android Architecture Navigational UI Components
    1. Create Navigational Graph
    2. Add Navigational Host to Activity
    3. Create Fragments

Application Configuration and Libraries

Task 2: Implement UI for Each Activity and Fragment

Create Android Architectural Component

- Build UI for MainActivity
- Build UI Fragments by creating on Nav graph
    - Login Fragment
    - Registration Fragment

  - Scan Fragment
  - Scan Detail Fragment

Task 3: Create Architecture Components

Implement Google Play Services by using the Barcode API with the fragments

- Create Adapters
- Create ViewModel
- Create Repository
- Create Retrofit Client
- Create DAO’s

Task 4: Layout Configuration Task Tablet

Implement steps to include layout on tablet device

- Create layout for Tablet Devices
- Modify main activity
- Create Navigational Layout drawer
- Create Floating Action Button on Fragment Scan Detail

Task 5: Expresso User Interface - Build Variants

Create Build Variants of application

- Test Barcode scanner on different Barcode and QR-Code formats
- Create various test scenarios on Sharing Information
- Handle Error Cases
- Test on Tablet Device

Task 5: Social Media Integration

Integration social media into the application

- Pinterest
- Linkedin
- Facebook
- Google

Task 6: Layout Configuration Various Formats

Application is configured with different resource files for switching on all layouts.

Task 7: Widget Configuration

The application will use intentService to fetch data to display on Widget. AsyncTask will be used in this process

Task 8: Firebase Analytics Integration

Modify Application Level Gradle file to integrate Firebase Analytics.