****

**Faculty of Science and Technology**

**CST3515 Embedded Linux System**

**and Application Development**

**“Final Year Report for Individual Project ”**

**Name:** Ayman

**MISIS:** M00649153

**Module Lecturer:** Mrs. Fehmida Hussain

**Module Supervisor:** Mrs. Geethu Joy

**Campus:** Dubai

**Acknowledgement**

I would like to express my appreciation and gratitude to my supervisor, Mrs. Geethu Joy who has been guiding me throughout the course of the project.It was a pleasure to have someone willing to help me at anytime during this entire course of the academic year.

I would also like to thank the Faculty of Science and Technology at Middlesex University Dubai that allowed me to pursue my degree and provide the much needed facilities.

Finally, I would like to thank my family for their never-ending support.

# Abstract

Price comparison websites have been growing and have become very popular in the recent years, which has benefited the customers by providing information about various products available in the market and allowing them to purchase these products that provide the best offer for them. The purpose of this report is to address th problem that consumers have always complained about how some of the products could have been purchased for a cheaper price from a different store, but instead, they ended up paying a much higher price.

This report presents an application which allows the user to search for a product by scanning the bar-code number, and comparing its price with multiple stores across the city. This application allows the users to search products through different stores and helps them choose a deal based on the cost and services offered. The consumers can access all of the information needed easily, without having to visit different shops, available to them right at the tip of their fingers. It will also help the consumers to save time and money. Instead of going out to different stores to check the price of one or more products, consumers can now just sit at home and compare the best deals that suit their needs. Every project needs to have a target audience and the target audience for this project are the people who are on a limited budget or are looking for any way to save some money.

**Table of Contents**

[1. Introduction and context description 5](#_Toc22943)

[2. Literature Review 8](#_Toc14700)

[2.1 Bar-code Scanner: 10](#_Toc26299)

[2.2 Shopping Comparison Service: 12](#_Toc1946)

[2.3 Price comparison apps: 12](#_Toc15820)

[2.4 Comparison of apps: 16](#_Toc3962)

[3.3 Overall Description: 19](#_Toc20076)

[3.4 User needs: 20](#_Toc26793)

[3.5 Software Design: 21](#_Toc4390)

[3.6 System Features: 22](#_Toc31941)

[3.7 User Interface (UI): 25](#_Toc27283)

[3.8 Database design: 28](#_Toc12121)

4. Implementation: 31

[4.1 Android Studio IDE: 31](#_Toc5603)

[4.2 Program Codes Explanation: 32](#_Toc25952)

[4.3 Database: 4](#_Toc25952)3

4.4 System Screen-shots: 45

[5. Software Testing: 48](#_Toc18571)

5.1 White Box Testing : 48

5.2 Black Box Testing: 50

5.3 Usabillity Evaluation: 51

6. Conclusion: 58

[6.1 Limitations: 58](#_Toc22868)

6.2 Future Works: 59

[7. References: 61](#_Toc10891)

8. Appenedices: 63

8.1 [Survey: 63](#_Toc21726)

8.2 [Meeting Log: 66](#_Toc16876)

8.3 Ethics Form: 67

8.4 System codes: 72

1. **Introduction and context description**

Product information and price comparison apps on smart-phones play an increasing role in consumers’ purchase decision process. (Daurer, 2019) Customers are always looking for the best deals when choosing a product for themselves. They have often complained about how they could have got a product for a lower price in some other store. And in this day and age, everyone would love to save up as much as they could, and that's where product comparison apps come into play. Product comparison is when the price of the same product is compared with different outlets and among different brands. (Ferreira, 2019) With the use of bar-code scanning in price comparison, we can make it easier to search and narrow down the product. A bar-code is a machine-readable code in the form of numbers and a pattern of parallel lines of varying widths, printed on, and identifying a product. (Trujillo, 2019) These bar-codes contain details of products that are stored in a database, which helps the retailer to identify them better. With the current current crisis of COVID-19 spread across the globe, these price comparison apps can help users. In this pandemic, everybody’s initial plan is to save money wherever they can as companies have slashed pays across the globe. With the use of price comparison apps, users can now locate the cheapest product among a wide selection of stores from supermarket chains to groceries. Since users can now select and locate what they want at the comfort of their homes, it also reduces the amount of time they spend outside going from store to store to get the cheapest available product, hence, helping them by reducing the chances of being infected.

**1.1 Problem definition:**

The use of price comparison apps could make life easier for the average consumer as they wouldn't need to travel across multiple stores to check the price of a single product and then decide which store would be more feasible for them. According to a recent e-Commerce study, 36% of consumers spend 30+ minutes of comparison shopping before making a purchase, while 65% spend 16+ minutes doing so. (Blog.3dcart.com, 2019) Hence, the study shows that impulsive purchases are declining, and shoppers prefer to conduct their own sort of research before buying a product. There are multiple price comparison apps, but not all of them are available in the U.A.E. The biggest price comparison app, Shopsavvy is not available in the U.A.E. Even the ones that are available either don't allow the user to scan the product or just don't contain product details from local stores in the U.A.E. That's where the use of bar-code and price comparison can come in handy.

**1.2 Aims and Objectives:**

* The primary aim of this app is to develop a price comparison application which incorporates bar-code scanning that allows its users to compare prices from multiple stores. The main selling point of these apps is that they allow users to navigate the prices among multiple stores. By incorporating that feature, users have the opportunity to save money as the prices of multiple stores would allow users to broaden their choices.
* Since having information of multiple stores available to users using these apps, it would make shopping time-efficient as the need for people to travel to multiple stores just to find out the price of one product gets completely cut down. Now, this luxury is available to them just by swiping their screen. With the use of multiple stores, it would make sense to find a store closest to the user. They can do this by clicking on a hyperlink associated with the store which will then open up Google Maps on their phone and display the chain of super markets available near them. This will then in turn help save the time and money of the users and also help the environment to an extent by reducing their carbon footprint.
* Most product comparison apps have a very complicated and messy layout. Additional features are introduced in those apps which don’t necessarily have any important function when it comes to price comparison. To implement all the functions mentioned earlier, a simple and easy app layout will be designed which will cater to all age groups. By implementing a simple UI, users will be able to navigate through the app easily which will then improve the user experience.
* An additional feature is implemented which allows the users to manually input products if that particular product is not available in the database. This can help increase the range of the application as users will be able to add products to the database at their will.

The following sections of the report will look at the life cycle of the app development in detail. The next section in this report is the literature review which will look at similar apps that have been developed in the past and compare them with each other. The advantage of doing this is that it increases the capabilities of the project by pitting it against other similar apps. The section after that is the software specifications and design section which discusses the proposed design of the app and database as well as mentioning some of the use cases of the app and the user requirements. The target audience has to be decided so as to keep the functionalities of the app limited and simple for the user to understand and operate. The implementation part of the report briefly describes all the classes involved in the development of the app. Picture of the important parts of all the classes have been discussed. In addition to that, the testing methods used for the application, namely white box testing and black box testing, are also looked at. Finally a survey is conducted for the user evaluation so as to better understand how the user would describe their experience when using this app. The report is finally concluded with a conclusion of the entire report which describes the product in brief as well as the limitations and future works of the app. The references and appendices are also included at the end of the report.

1. **Literature Review**

The literature review goes through the results presented by previous researchers and the results were scrutinized. In order to better understand, evaluate and create a bar-code price comparison app, a literature review was conducted. In this study, current or emerging apps that have similar functionalities were compared against each other which would in turn aid in the development of the final app. To better compare the apps, a few questions need to be answered:

* Who are the competitors?
* What functionalities do their apps have?
* What are their weaknesses?

These questions will be answered by answered by going through websites which contain basic details of the apps, as well as reading customer reviews. The most logical way to improve an app is by reading the reviews posted by other users on the app of the competitors. The reviews provide a better understanding to app developers as to what needs to change in the current app.

To highlight the importance of price comparison apps and websites, a survey was conducted by “Ask Your Target Market’s” website where round 1000 respondents were asked a series of questions which gave an insight into the peoples needs:

* When asked whether it was important to get the lowest possible prices when buying food, 74% of the survey takers said that it was important to them.
* When asked whether they would consider themselves as bargain shoppers, 79% of them admitted to.
* Another 79% of the people told that they actively try to find the lowest prices of items when shopping.
* When asked whether they compare prices before making purchases 17% admitted to comparing prices from one or more stores, 41% said they compare prices most of the time,  28% do so about half the time, 10% rarely ever compare prices before purchasing and just 4% said they never do. (Pilon, 2016)

A pie chart representation of the questions and answers is shown below. A pie chart makes it easier to understand the statistics mentioned above.

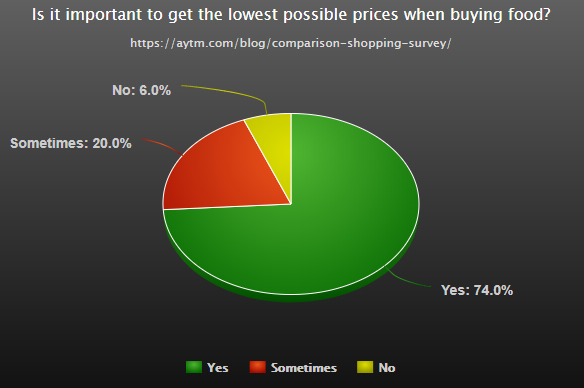


Figure 2.2

Figure 2.1

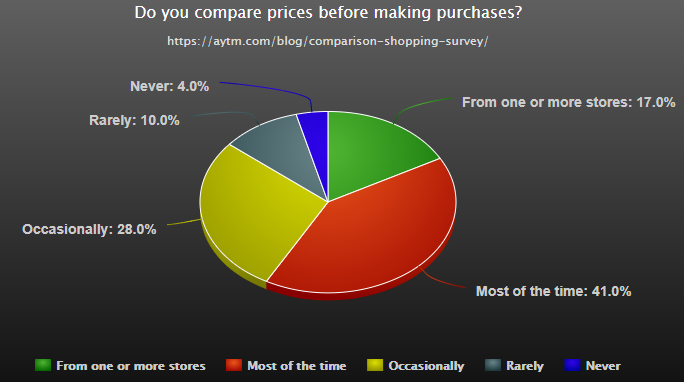
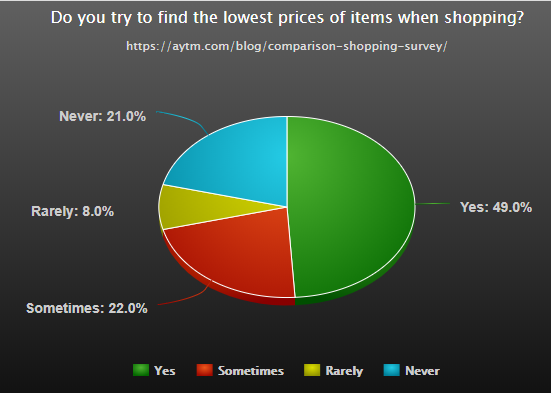


Figure 2.4

Figure 2.3

As the graphs above depict, most of the survey takers consider themselves as bargain shoppers and tend to look for the cheapest product but 83% of them don’t compare prices from more then 1 or more stores. This proves to be a serious disadvantage for them as they could possibly get the product they were looking for from some place cheaper. Since they don’t have proper access to shopping services that compare prices from a variety of stores, they tend to settle for what they get even if it is not the best deal for them. The bar-code price comparison app will try to solve this problem.

The main purpose of developing a price comparison app is to help both consumers and retailers. Consumers would be benefited as they would save up money once they compare the price of a product with different outlets and they would end up saving time by not travelling to each and every outlet physically to check for a deal that is beneficial to them. Retailers would be benefited as they could learn the ‘art of product pricing’ by analyzing how their competitors price the same products. The consumers and retailers would eventually become the main stakeholders for this product. (Ferreira, 2020)

The materials used in this part are articles from Google scholar as well as multiple websites that contain specific details about the apps discussed. Furthermore, all the apps that have been talked about below were first downloaded and installed to have a better understanding of their functionalities.

**2.1 Bar-code Scanner:**

A bar-code [is a visual pattern encoded with information](http://www.barcodesinc.com/faq/) that a machine can read. (Trujillo, 2020) They are used in a wide range of industries to store and display large amounts of product or user information in a simpler and quicker way. Traditionally, the only way to scan bar-codes would be with the help of laser scanners, but with the exponential growth of mobile technology, they are now implemented in smart phones by using its camera.

A typical bar-code is displayed below in figure 2.6.

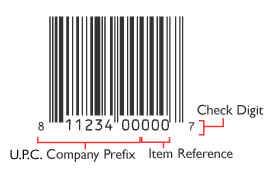


Figure 2.6

The UPC (Universal Product Code) number itself is referred to as the GTIN (Global Trade Item Number). The main purpose of UPC’s is to easily allow consumers to locate product features such as brand name, size, color, etc. The GTIN is made up of two parts: The UPC Company Prefix and the Item Reference which is a number that has been assigned to that unique product. (Barcodes Inc., 2018)

This first component,which is the UPC Company Prefix, is between 6 and 10 digits long, and is assigned by the GS1 (Global Standards One), which is an organization that maintains the global standards for bar-codes. The number of digits is determined by the number of products that will need numbers to be assigned to. So, if a company has thousands of products, the company prefix will need to be fewer digits. If the company has just a few products, the company prefix will most likely be closer to 10 digits long. This company prefix number will represent them as the manufacturer on all of their products. (Barcodes Inc., 2018)

The second component, called an "Item Reference Number", is the unique number which is used to reference a specific product. This number is not assigned by GS1 and hence it is up to the manufacturer to assign the unique Item Reference Number for the given product. (Axicon.com, 2019)

The last number is a check digit calculated from the previous 11 digits. It is not randomly assigned. The bar-code label printing software used to create the label will automatically calculate and assign the check digit. The check digit is basically used to ensure that number scanned in from a bar-code is correct.

The bar-code image is processed with the help of an open source library known as Firebase ML KIT Vision. It has the ability to process both 1D and 2D bar-code images. 1D images are the most used and are the basic UPC bar-codes as discussed above and 2D images are basically QR codes or Aztec. The app will be using the help of the bar-code scanning feature of ML KIT Vision by capturing the bar-code picture and scanning the bar-code to retrieve the raw bar-code number that is located at the bottom of the bar-code image.

**2.2 Shopping Comparison Service:**

A shopping comparison service refers to a service provided by either a website or an app, which allows the user to compare the price of a product from one specific shop with different shops. This is done by the sellers who quote their prices for a product on a shopping comparison service and then the buyers can select which product they would like to have by taking into account the distance of the store to them (Datafeedwatch.com, 2020) Since online shopping has become the most convenient way to obtain goods, the use of price comparison websites and apps has only benefited the buyers and sellers by making shopping easy. Websites such as ‘Google Shopping’ and ‘Pronto’ help users to find goods online but they have their own limitations. As most price comparison apps available on the play store and apple store, these websites don’t allow users to scan bar-codes or even compare prices from local groceries. Instead, these websites focus on large companies such as ‘Carrefour’ and ‘Lulu’. That’s one of the reasons as to why price comparison apps are being used more than websites.

Most of these apps and websites don’t sell the products directly, instead they collect information about various products and the stores they are available in and display that information on their platform. The user then has their own freedom to select which store is offering the best deal. This is why users find it more convenient to go through comparison websites first instead of directly going into a store and picking a product up. Saving money has played a huge part in the success of these services as the users would obviously want a product that is available to them for the cheapest.

**2.3 Price comparison apps:**

Most price comparison apps have incorporated the use of bar-code scanning as well as the features which compare multiple products on their respective platforms. Some of these applications could allow users to only input a product and search for it or allow the user to scan the bar-code of the desired product and search for it. The popularity of these apps depends on what the buyer is most comfortable using.

Some of the most popular comparison apps are as follows:

1. ShopSavvy
2. Idealo
3. BuyVia
4. Pricena
5. Amazon Shopping

**ShopSavvy:** This is a mobile application developed by Rylan Barnes, Jason Hudgins and [Alexander Muse](https://en.wikipedia.org/wiki/Alexander_Muse" \o "Alexander Muse) and is available on both android and iOS phones. This app scans products and displays online and local stores that provide those products. In addition to that, their platform also compares prices, searches for discounts and displays user reviews for a better in-app experience. As of now, ShopSavvy is one the worlds most popular shopping application with approximately 10 million downloads. (crunchbase.com, 2019)

A brief explanation of the apps features are displayed below:

1. Scanning products: The users have the option to either enter the UPC code or directly scan the bar-code using their mobile phone camera.
2. Review: The users can also view customer reviews on products for a better shopping experience.
3. Selling: It also allows users to scan their own products and sell them via their app.
4. Deals: As discussed above, the app allows users to search and browse through deals on various products.

Although this app is the most popular shopping app, it does have its drawbacks:

1. This app does not support stores in the U.A.E.
2. It does not show the prices of grocery products.

**Idealo:** This is a price comparison service launched in the year 2000 in Germany by Martin Sinner, Albrecht von Sonntag, Christian Habermehl and later bought by the Axel Springer AG publishing company. They started off as a website and now have an app as well, with approximately 5 million downloads.

Some of its features are:

1. Scanning products: Users can search for products via text or by scanning the bar-code with their camera
2. Product information: Detailed product information is available with a brief description of the product, user ratings, images and expert reviews
3. Price alerts: Users have the option to check the price history as well as set a price alert when the product reaches the users desired price range. The price history is available in the form of a graph. The users get a notification when there is a sudden fall in the price of their desired product.
4. History: Users have the option to access past search history as well as any previous bar-code scans.

The only downside of this application is that its only available for the European market.

**BuyVia:** BuyVia is a website and app developed by Norman Fong in 2012. It allows its users to save money online and locally with the help of their price comparison features. It has approximately 5 million downloads on the apple store and play store.

What separates BuyVia from the rest is that they have a team of shoppers and researchers who constantly check for the cheapest deals and coupons which in turn could benefit the shoppers. In addition to that, they also compare the prices of hotels and restaurants in the users area.

A drawback of this app is that it is not available in the U.A.E.

**Pricena:** Pricena is a shopping app developed by Haneen Dabain in 2013. As most of the apps discussed earlier, Pricena is also a price comparison app which too allows its users to scan products with bar-codes. It has approximately 100k downloads on the play store.

Pricena only compares the prices of products that have online stores. It does not take products that are available in local grocery stores. Since it’s users can only buy from these online stores, they wouldn't know if they could get a better price at their local grocery stores. The users would have to select a product on pricena, go to a nearby shop and check if its available for cheaper there and then decide if they would like to buy from any of the online stores provided. All this proves to be a hassle and it defeats the purpose of an all round price comparison application. Although, there is a plus side to Pricena as its available in the U.A.E unlike most of the other applications discussed above.

**Amazon Shopping:** Amazon online shopping is the worlds most popular shopping platform. It first started off as a website and was developed by Jeff Bezos back in 1995. Since its conception, it has grown exponentially and not long ago, an online shopping app was developed with over 100 million downloads approximately making it the most used shopping app in the market.

Amazon shopping has the same features which most of the other apps online do but with some extra functions. Users can make use of ‘Alexa’, which is Amazon’s voice assistant, to order and track products.Shoppers can make use of this feature by tapping the microphone icon and saying “where’s my stuff?” to access open orders. They can also reorder products by saying “reorder paper towels” or “buy more batteries”. In addition to that, if the user happens to have an amazon prime account and ends up buying something from the amazon shopping app, they are entitled to further discounts as well as getting their product delivered within 24 hours to 48 hours,

Although this app is available worldwide with users enjoying benefits such as discounts and coupons, a limitation of Amazon shopping is that they don’t have details on products from local stores.

**2.4 Comparison of apps:**

A comparison of the main features that each of the apps discussed above should have has been displayed in a table below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Application** | **Supports local groceries** | **Supports bar-code scanning** | **Available in U.A.E** | **Displays location** | **Allows user to add new products to DB** |
| ShopSavvy | No | Yes | No | Yes | No |
| Idealo | No | Yes | Yes | No | No |
| BuyVia | No | Yes | No | Yes | No |
| Pricena | No | Yes | Yes | Yes | No |
| Amazon Shopping | No | Yes | Yes | No | No |

(Table 1)

The main objective of having a comparison of features is to better understand what differentiates each app from the rest. As the table suggests, all of the popular apps listed in the table above do not support grocery stores. The final app for this project will include features that have been lacking in most of these popular apps, namely, supporting prices from local groceries. All other features will remain the same with a couple of twists. For example, the apps displayed above that display the location of a store to the user don’t display a route to that certain store. A text box of the stores location is the only piece of information the user gets. Therefore, the final app will allow the user to navigate to the closest store by opening up Google maps. In addition to that, users can add products to the database for others to scan, provided they have correct information about the stores and price.

1. **Software Requirements, Specifications and Design**

One of the biggest advantages of the internet is that it has made shopping easier for the common man. Now, people don’t need to physically go and visit multiple shopping outlets to search for goods. Everything is at the reach of their finger tips. People are always looking for the best deals when they’re looking to invest in a product whether its a new fridge or just a new shampoo. Everyone would like to save up as much as they can when they’re out shopping. Normally, customers use price comparison apps to help aid them in selecting the cheapest products in the market. With a number of price comparison websites and apps available on the internet, shopping has just become a lot easier for most people.

To start of the project, software and hardware requirements have to be established so that they are assembled which enables the app to be run on it.

**3.1 Hardware Requirements:**

Since this report revolves around app development which is a software based component, the hardware requirements are limited.

* A PC is needed to write the program and execute it. The PC needs to be at least a 64-bit environment for Android Gingerbread (2.3) or higher.
* The PC needs to have enough memory as most large applications are occupy a lot of storage as well as a higher RAM capacity than most PC’s

**3.2 Software Requirements:**

* Initially, an IDE is required to code the app. Since this is an android based application, the code needs to be written in Android’s IDE which is known as Android Studio.
* A database will be required which will store all the details corresponding to the functioning of the app. A real-time database called Firestore is used as SQL commands are not required to enter details into it. The database is primarily used to enter and retrieve product details from it.
* Since this is a bar-code scanning app, a bar-code scanning SDK (Software Development Kit) needs to be implemented. For this purpose, Firebase ML KIT is implemented as it is one of the best and most easily accessible SDK’s out there.
* The Android emulator also needs to be downloaded for developers who don’t have access to any physical android device. The android emulator basically creates a virtual space for apps to be run on.

**3.3 Overall Description:**

The app built is a price comparison app that uses bar-code scanning. Although there are multiple apps similar to the one that is developed on the apple store and play store, A few tweaks are made regarding some situations in the app. As discussed in ‘Table 1’, most of the issues concerning these types of apps is as follows:

1. All of the apps discussed don’t support local groceries. Only big supermarket chains are supported. Local groceries are neglected which would prove to be a problem for a few customers as grocery stores are more often than not open longer than most supermarket chains.
2. Some of these apps aren’t available in the U.A.E. Even if they are available they either display details from online stores such as amazon, or just show details from main supermarket chains such as Carrefour and Lulu. The small supermarket chains and local groceries are neglected entirely.
3. Some of the apps don’t display locations to stores. Even if they do show the location, it just displays the name of the store and which street it is in. They don’t show all the store chains in that particular area which the user can then freely choose depending on the traffic or distance of that particular store.
4. All of the apps mentioned above don’t allow users to enter new products into the database. This feature can help increase the user experience for any future users.

My app will tackle the issues discussed above by mainly incorporating smaller supermarket chains and groceries, displaying nutrition information for the users, as well as allowing users to open up Google Maps and see the selected chain of stores closest to them.

**3.4 User needs:**

To visualize the full capability of the app, a group of users is targeted for whom the app will serve the most value. In order to conduct the research a few questions have to be answered:

* Who are these groups of individuals?
* What are their problems?
* How does this app help them solve their problem?

After carefully analyzing the questions above, the target audience has been identified for whom the app will suit best:

1. Disabled and elderly people: These groups of people find it hard to physically travel to multiple stores and look around for goods. They can just scan the product, look at the prices and select which store they would prefer going to depending on their convenience. This can then help them save time and travel less. In addition to that, they can also end up saving money as old people are mostly retired and have limited funds and disabled people tend to work low paying jobs.
2. Students: The app can target students who stay away from home since they tend to save as much as money as they can which they can spend elsewhere on other commodities such as clothes, fuel, etc. This in turn will help them to better understand the value of saving up money.
3. Unemployed person: An unemployed person would benefit from this app as they don’t have any incoming payments or income. They would need to save up as much as they can to manage and not go broke. They could use this app and would end up saving money until they are able to find a stable and permanent job.

**3.5 Software Design:**

Software design is the process by which a developer accepts and transforms the user requirements into a suitable form which is later used to develop the final app. This the most important phase of development as it links the users and the system. The figure below depicts the fundamental design of the application.

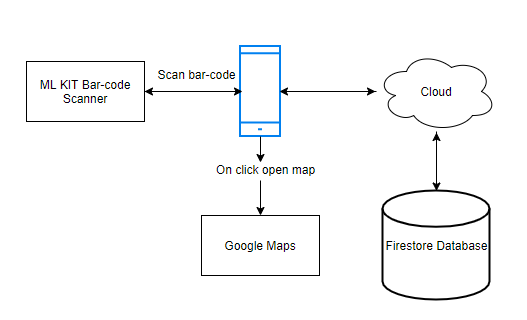


Figure 3.1

The mobile makes use of its camera and ML KIT Vision to scan a bar-code number. Once the scan is successful, the bar-code number is sent to the database, which is the Firestore Database, and all the information associated with that particular number is displayed to the user on their mobile screen. An HTML link of the stores location is also retrieved from the database which opens up Google Maps on the users phone if that link is clicked.

**3.6 System Features:**

To better understand how the app would work, a flowchart has been created which depicts its basic functionality. A flow chart basically helps to better understand and visualize all the steps of the project. It helps programmers by collecting data about every process, understanding the relationship among them and eliminating the unnecessary steps.

By creating a flow chart, I can identify all the inefficiencies and locate the parts where the app could improve.

Start/Scan product

Display an error message

Product Scanned?

No

Yes

Check in database

Display “Product unavailable”

No

Is product in database?

Yes

Display information and location

End

The flowchart shows the necessary steps the user goes through to scan and retrieve the relevant information from the app. The user starts off by pressing the “Scan” button to scan the bar-code. If the scan was unsuccessful as a result of unclear image or any other camera issues, the user gets an error message saying “unable to read bar-code” and then they can start the scanning process again. If the scan was successful,the bar-code number is sent to the database and all the necessary information regarding the product that is associated with the bar-code is sent back to the app and displayed to the user. If the bar-code number is not in the database, the user is prompted with an error message saying “bar-code not available”. The user can then manually enter the product details themselves with the second feature of the app that allows users to do so.

The app allows users to compare and search for products from different store outlets across Abu Dhabi with the use of a bar-code scanner. The app will also display the location to a nearby store by opening Google Maps.

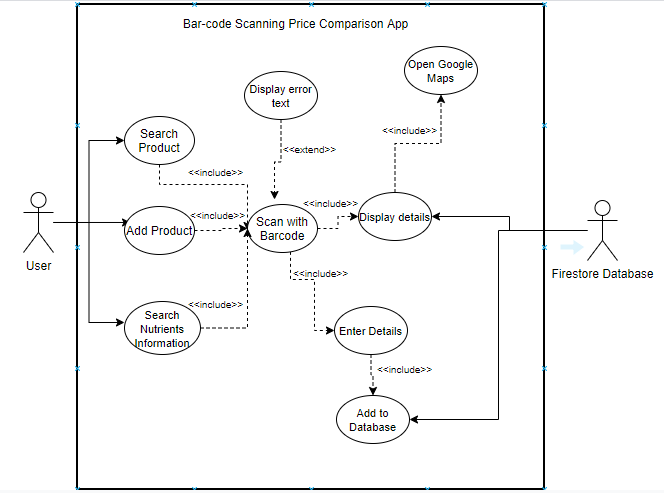


Figure 3.2

A use-case diagram has been made above which is a great tool to identify system *use cases* and its primary and secondary actors. In this context, use case represents the way by which the user can operate the system. The primary actor is the User as he or she is the one who initiates the use of the system whereas the Firestore Database is the secondary actor as it is more reactionary to what the user does. In the above image, the user initiates the use of the system by scanning a product and either adding it to the database or receiving information about that particular product. The receiving and adding information is done by the database which makes it a secondary actor as it has to wait for the users commands.

**3.7 User Interface (UI):**

After deciding the basic functionality of the app, a UI interface needs to be designed. A UI design is basically a visual representation of how the app looks. Initially a mock-up diagram of how the app is supposed to look is shown below. The diagram was made with the help of a free online tool known as “draw.io”. It allows users to design anything from UML diagrams to UI designs.

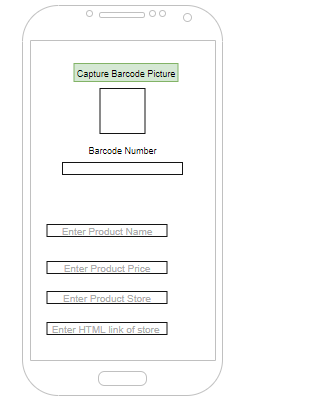
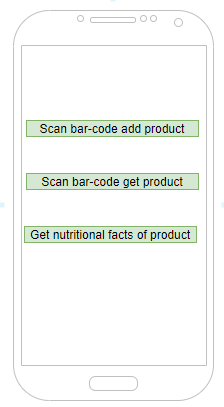


Figure 3.3

Figure 3.4

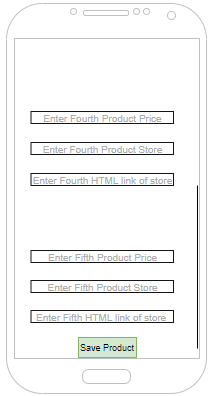


Figure 3.5

**Home Screen:** The home screen of the application is depicted at Figure 3.3 which contains three main functions that are presented as buttons, “Scan Bar-code and Add Product”, “Scan Bar-code and Get Product” and “Get nutritional facts of product”. The “Scan Bar-code and Add Product” button, as the name suggests, allows the user to scan a bar-code, enter the necessary details associated with that bar-code and add it into the Firestore database.

**Scan Bar-code and Add Product:** Once the user clicks on this button, they are taken into a different activity as shown in figure 3.4. At this screen, the user can take a picture of the bar-code by clicking the “Capture Bar-code Picture” button. Once the picture is taken, it is displayed in a box below the button as shown in the figure. Once the bar-code picture is taken, it is scanned and the number is then displayed under the “Bar-code Number” label. The user then enters all the necessary information associated with the store into text-fields. The store product name, price, store name, and an HTML link of the store location are entered in the corresponding text-fields. The screen uses “ScrollView” to navigate across the rest of the textfields as there are quite a lot of textfields to enter data into. Once the user reaches the end of the screen, they can then enter the data into the Firestore database by clicking the “Save Product” button as shown in figure 3.5.

**Scan Bar-code and Get Product:** Once the user clicks on this button, they are taken into a different activity as shown in figure 3.6. At this screen, the user takes a picture of the bar-code and it is then scanned and displayed to user under the “Bar-code Number” label. The bar-code number is then sent to the database and all the information related to the number is retrieved and displayed next to their respective labels. Only the HTML link of the location of the store will be displayed below its label as the link is quite long and it wont be possible to accommodate it adjacent to its respective label. Even this screen uses “ScrollView” to navigate to the bottom of the screen.

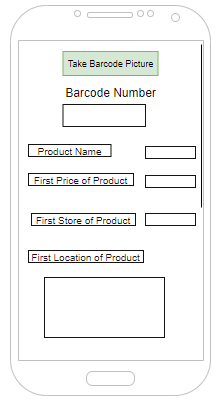


Figure 3.6

**Get Nutritional Facts of Product:** Once the user clicks on this button, they are taken into a different activity as shown in figure 3.7. At this screen, At this screen, the user takes a picture of the bar-code and it is then scanned. The bar-code number is then sent to the database and it looks for nutritional detail that is associated with that number. Once found, the details are sent to the app and displayed to the user. The name of the product associated with the bar-code is displayed directly under “Scan bar-code” button and all the nutritional information is displayed under the “Nutrition facts” heading.

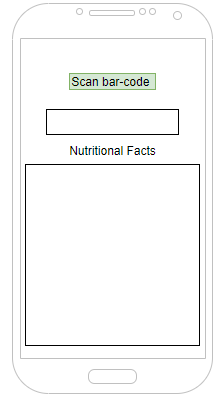


Figure 3.7

**3.8 Database design:**

A database is used to store all the important information of the bar-code price comparison app such as the names, prices, stores and locations of all the products. The app will update and retrieve information from the database as per the users requests. Since the app will require a lot of information and needs to update quickly, it will be using Firestore as it is a database which is real-time and it does not require SQL commands to enter data to the it. It keeps data in sync across client apps through real-time listeners and offers offline support for mobile and web so it is possible to build responsive apps that work regardless of network latency or Internet connectivity. (Cloud Firestore | Firebase, 2020)

Data in Firestore is in put in documents, which is then organized into collections. Collections contain all the necessary information required for the app. Users can either query documents and retrieve all the information from all the collections or they can query specific collections and retrieve data from them.

The Firestore database design is shown below. This just a representation of what the database will look like. More products are added into the database.

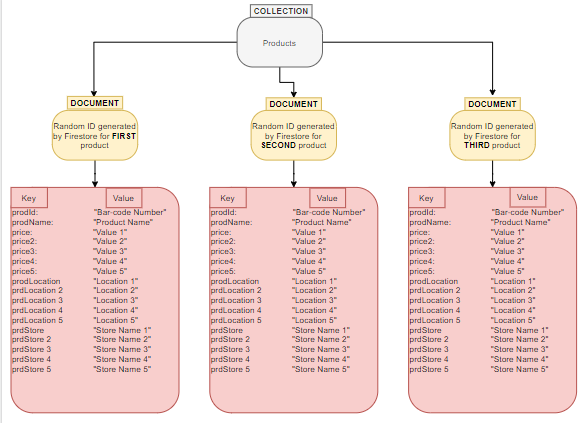


Figure 3.8

The database contains a collection called “Products” with multiple documents that contain details of products. Each document is associated with a different bar-code number. However, each of these documents has the same “key” whereas the value changes depending on the product.

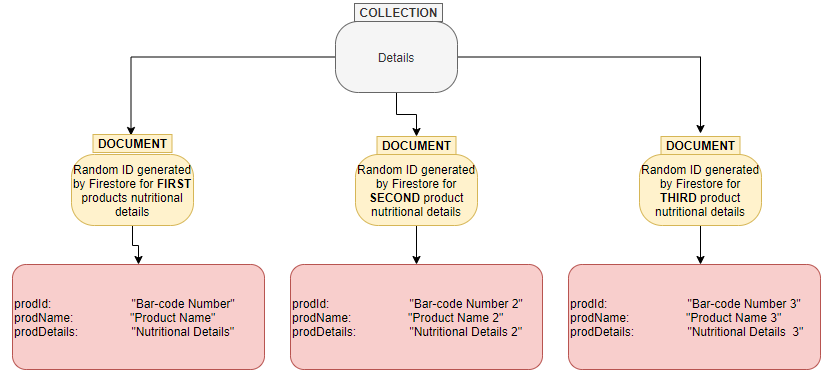


Figure 3.9

The database also contains another document as shown in the figure 3.9, called “Details”. All of these documents will display nutritional details of a specific bar-code number which is associated with a product. Each document is associated with a different bar-code number. However, each of these documents has the same “key” whereas the value changes depending on the product.

1. **Implementation**

The implementation is the section of the report which discusses the execution of the system as well as providing information about the IDE (Integrated Development Environment) used. The classes used in the application will be explained with the testing and user evaluation discussed. The database will also be shown which contains all the necessary information for the app to run.

**4.1 Android Studio IDE:**

Android studio is the official IDE for Google’s Android application development. It is based on the [IntelliJ IDEA](https://www.theserverside.com/definition/IntellJ-IDEA), a [Java](https://www.theserverside.com/definition/Java) integrated development environment for software, and incorporates its code editing and developer tools. (Rouse, 2018) Programs can be written in two languages in Android Studio, Kotlin and Java. To use features such as text detection or in this reports case, bar-code scanning, “gradle” files have to be implemented and downloaded to the project which are external files that aid the developer in their process of app development. The program can be executed with the help of the Android Emulator which is a virtual extension of an android mobile. The emulator basically creates a virtual environment for an app to be downloaded on if the programmer does not have an android device. Once the app has been downloaded to a device, the “Logcat” which is a command-line tool that allows the programmer to view a log of system messages including stack traces when the system throws an error and messages that have been written with the “Log” class . One of the main advantages of using Android Studio is that it provides a robust static analysis framework and includes over 280 different lint checks across the entirety of your app. Lint is basically a tool which analyzes the code to display any programming errors, bugs, suspicious constructs and other such errors. Additionally, it provides several quick fixes that help address issues in various categories, such as performance, security, and correctness, with a single click. Furthermore, android studio is very user friendly and what makes it that is its sleek design and its intelligent code editor. As the user types a piece of code, Android Studio displays suggestions in a drop-down list which the programmer can insert by pressing the “Tab” button.

**4.2 Program Codes Explanation:**

All the classes and files used in this project are explained in the next part. A general picture of what these classes and files look like will also be displayed here which will provide the reader with a basic idea of that particular class or file. Namely, two types of files will be discussed:

* Java class files: The codes used to execute the integral part of the app such as bar-code scanning and database queries are written in these files.
* XML files: The codes which describe the layout of the files are written here. All the designing part of the app happens here from adding labels and buttons to the colour schema of the app.

When the program is executed, these files are run together which then provide the user with the final experience of the app.

1. **activity\_main.xml:**

This file is responsible for displaying the Home screen of the app. Buttons are used which transport the user to the selected activity. Under these buttons, the “TextView” codes are used which have a brief description of what happens when the user clicks on these button. Instead of traditional buttons, the code “ImageButton” is used which allows programmers to put up images which function the same as buttons. It gives the program a clean look. The buttons are given a unique ID which are “add\_prd”, “get\_prd” and “btn\_details” and are associated with the “Add Product”, “Scan Product” and “Nutritional Details” buttons respectively. They also have an “onClick” function which makes the button clickable.

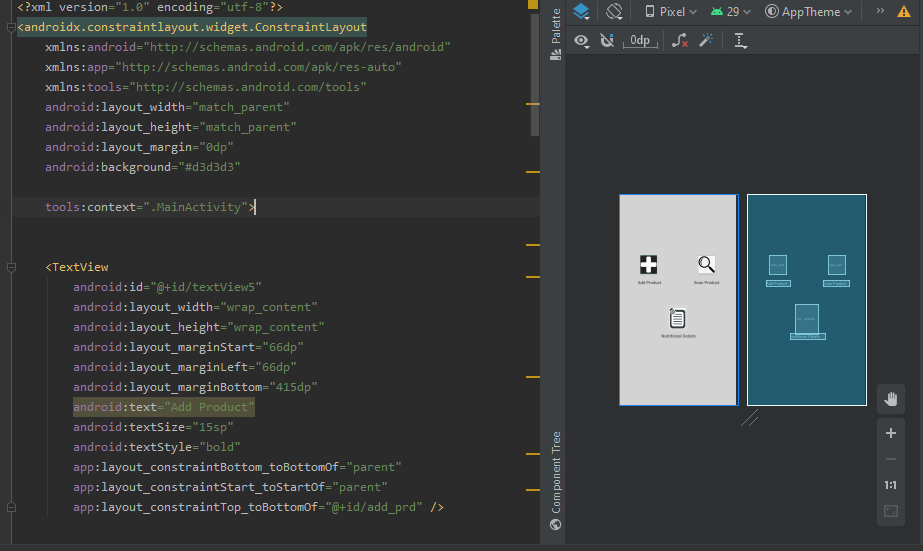


Figure 4.1

1. **MainActivity:**

The MainActivity is responsible for handling the “Home” page of the app. The “onClick” function of the buttons declared in the activity\_main.xml file are to execute the following methods:

* The “add\_prd’ button executes the “addProduct(View v)” method which takes users to the “ProductAdditionActivity” class.
* The “get\_prd” button executes the “getProduct(View v)” methods which takes users to the “ProductReaderActivity” class.
* The “btn\_details” button executes the “getDetails(View v)” method which takes users to the “ProductDetails” class.

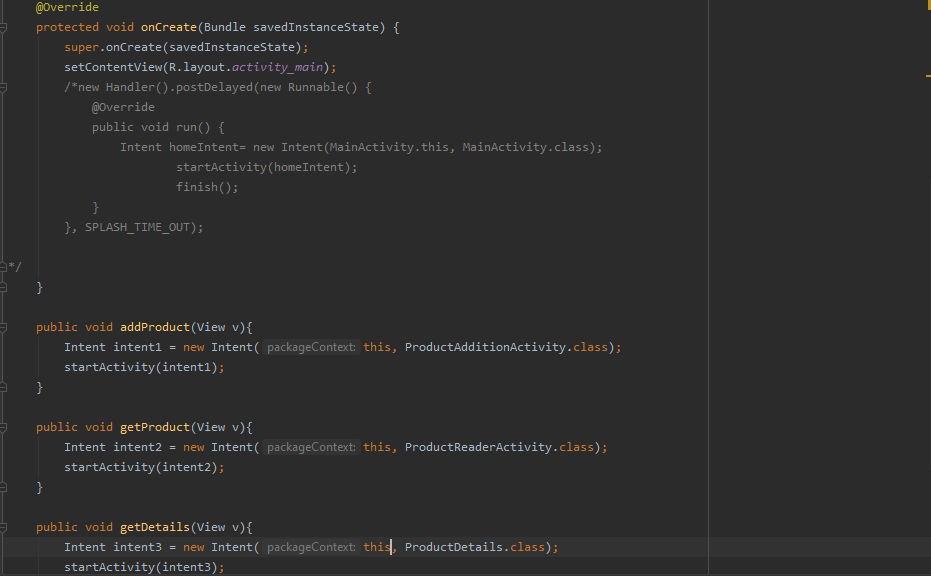


Figure 4.2

1. **ProductUtil:**

This class is responsible for the scanning the bar-code when the image is taken. The bar-code is detected by passing the bit image to the readBarcodeValueTask() method. The bar-code is then sent to the setBarcodeValue() method to be retrieved as a raw value. The raw value is then assigned across the classes for further use.

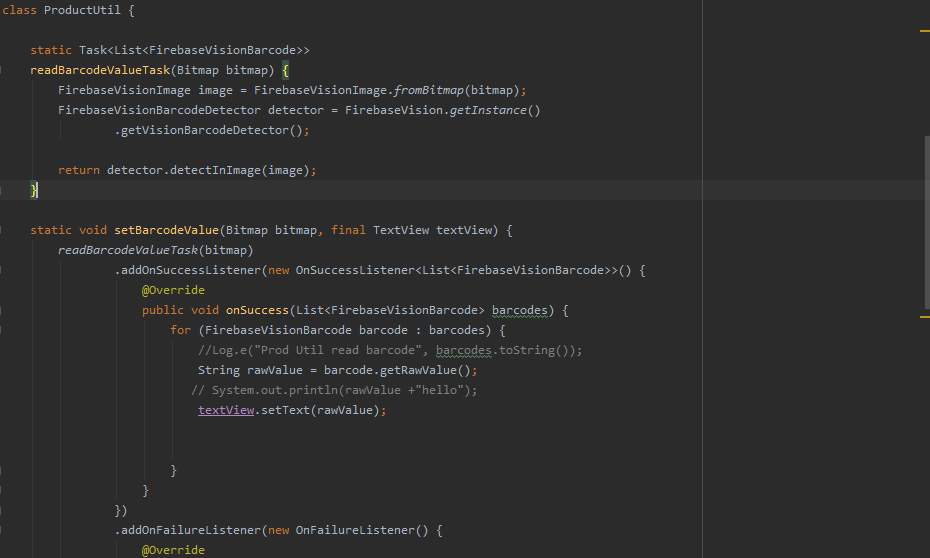


Figure 4.3

1. **ProductBaseActivity:**

This class mainly handles the camera part of the app. The method captureBarcodePic(View v) opens the camera up and allows the user to take a picture. Once the picture has been taken, the onActivityResult() method in any of ProductAdditionActivity, ProductReaderActivity or ProductDetails classes gets called, depending on what the user is doing, which then gets the bar-code value. Errors are also thrown if the bar-code is not readable.



Figure 4.4

1. **product\_addition.xml:**

This file is responsible for displaying the “Add Product” part of the app. The user can scan a bar-code, enter its details and save them to the database. Buttons, text-fields, and labels will be used primarily in this file. The button to scan the bar-code has a unique ID called “capture\_barcode” which has an onClick method attached to it called “captureBarcodePic”, which is in the ProductBase class. Text-fields are also used which allow the users to enter the product details manually. The details are then sent to the database via a save button which has a unique ID called “save\_prd” and it has an onClick method attached to it called “saveProduct” which saves the details that are entered into the text-fields, to the database.

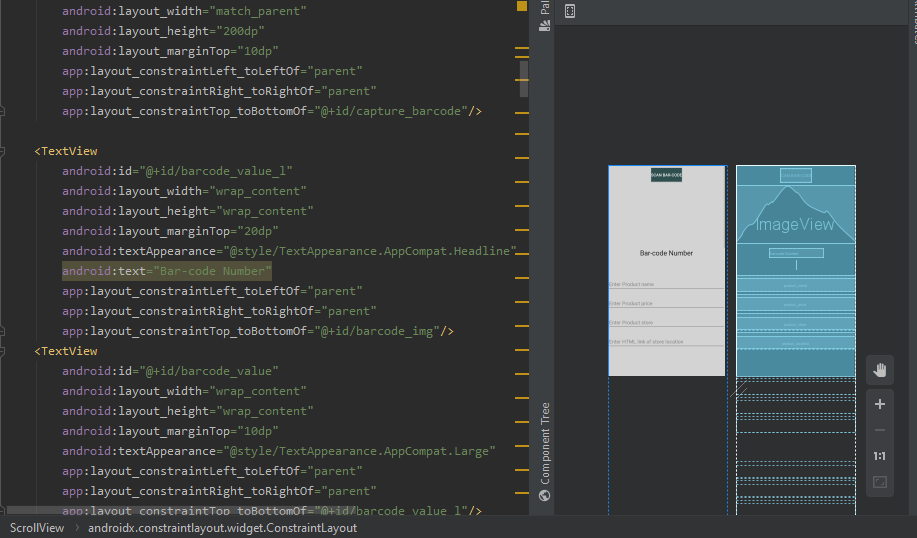


Figure 4.5

1. **Product:**

This class contains the setters and getters for all the variables that are defined. The setters will be used to enter the details into the database whereas the getters will be used to retrieve the information from the database.

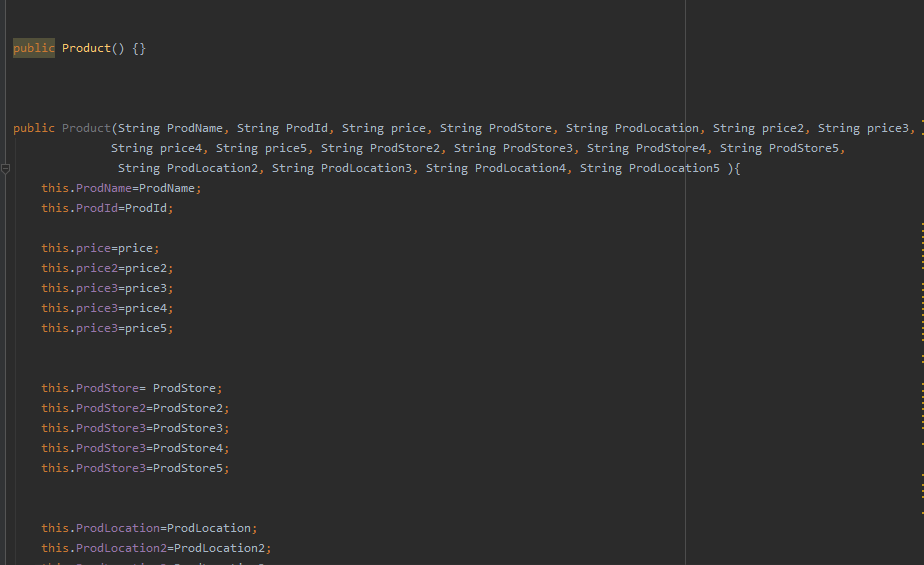


Figure 4.6

1. **ProductAdditionActivity:**

This class handles the users request to add the product details into the database. This class is an extension of the ProductBase class which is the main class for handling the camera and scanner functions. All the variables used are declared in the beginning. The onCreate() method sets the view and initializes the variables with their respective text-fields from the xml. The createProductObj() initializes the setters to add the specific product detail into the database. The addProductToDB() method handles the database part by creating a collection and storing the details in that collection path. Since no document name has been specified, Firestore automatically assigns a random document ID for that specific bar-code. The saveProduct() method is assigned with the “save\_prd” button. Once the button is pressed, saveProduct() is executed. To make sure none of the details entered into the text-fields remain once the button is clicked, the refreshUi() method makes the text-fields empty.

1. **product\_reader.xml:**

This file is responsible for displaying the “Add Product” part of the app. The user can scan a bar-code and the program will find the details of the product linked with that bar-code number. The button to scan the bar-code is the same as the one used in the “product\_addition.xml” file with the same unique ID as “capture\_barcode” which has an onClick method attached to it called “captureBarcodePic”. Text-views are used here to display the details from the database.

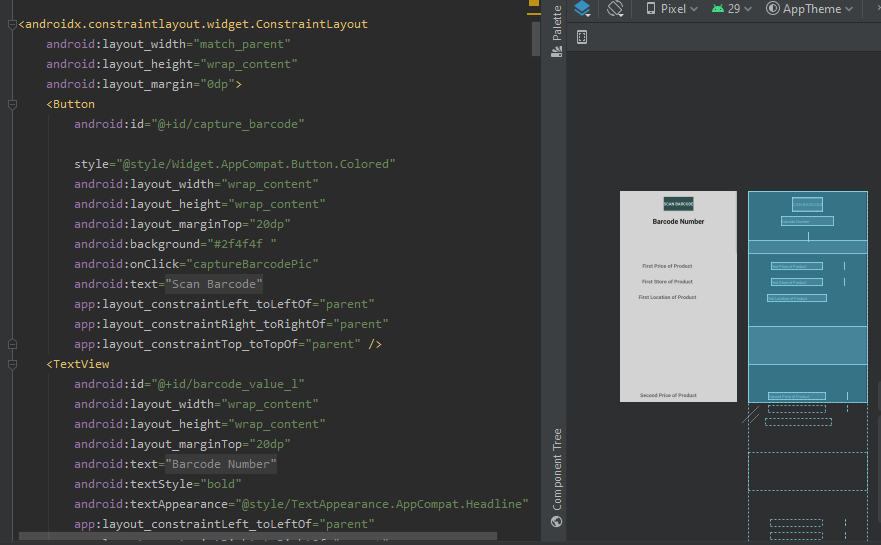


Figure 4.7

1. **ProductReaderActivity:**

This class handles the users request to scan a product and display its details from the database. This class is an extension of the ProductBase class which is the main class for handling the camera and scanner functions. All the variables used are declared in the beginning. The onCreate() method sets the view and initializes the variables with their respective text-fields from the xml. The getProductfromDB() method handles the database part by going through the “products” collection and retrieving the details associated with the bar-code number. This class also contains a method which checks whether the bar-code is clear and can be read or not which is the readProductfromDb() method. If the size of the bar-code is “0”, which makes it unreadable, the program throws an error. If it is readable, the getProductFromDB() method is executed and the bar-code details are displayed to the user with the help of the displayProductDetails() method which uses the “set” functions declared in the “Product” class to assign the details on the users screen.

1. **product\_details.xml:**

This file is responsible for displaying the “Nutritional Details” part of the app. The user can scan a bar-code and the program will find the nutritional details of the product linked with that bar-code number. The button to scan the bar-code is the same as the one used in the “product\_addition.xml” and “product\_reader.xml” files with the same unique ID as “capture\_barcode” which has an onClick method attached to it called “captureBarcodePic”. Text-views are used here to display the details from the database.

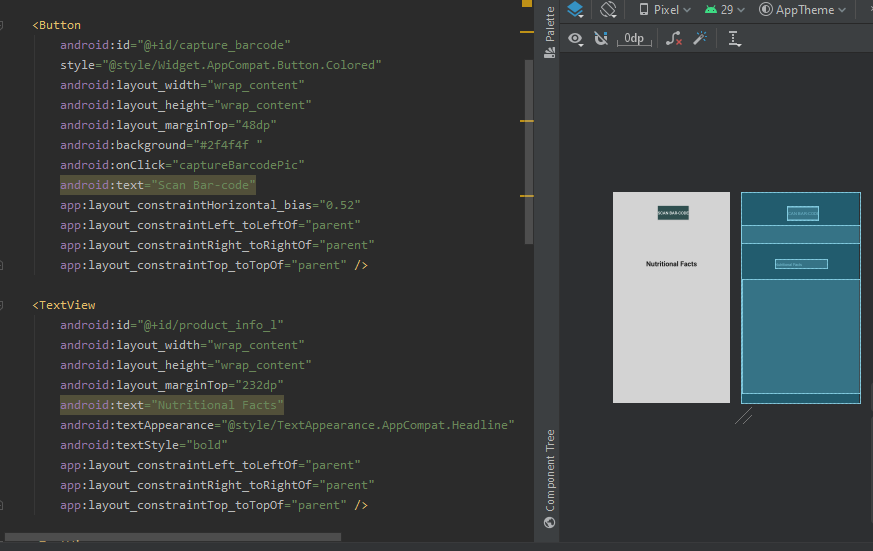


Figure 4.8

1. **Details:**

This class contains the “setters” and “getters” for all the variables that are defined. The “setters” will be used to enter the details into the database whereas the “getters” will be used to retrieve the information from the database. What differentiates this class from the “Product” class is that the “Product” class contains information specifically to the price comparison whereas the “Details” class contains information specifically related to the nutritional details of the product.



Figure 4.9

1. **ProductDetails:**

This class handles the users request to scan a product and display its nutritional details from the database. This class is an extension of the ProductBase class which is the main class for handling the camera and scanner functions. All the variables used are declared in the beginning. The onCreate() method sets the view and initializes the variables with their respective text-fields from the xml. The getProductDetails() method handles the database part by going through the “details” collection and retrieving the nutritional information associated with the bar-code number. This class also contains a method which checks whether the bar-code is clear and can be read or not which is the readProductfromDb1() method. If the size of the bar-code is “0”, which makes it unreadable, the program throws an error. If it is readable, the getProductDetails() method is executed and the bar-code details are displayed to the user with the help of the displayProductDetails() method which uses the “set” functions declared in the “Details” class to assign the nutritional facts on the users screen.

1. **splash\_activity.xml:**

This file is responsible for the display of the splash screen. A splash screen is a graphical control element which consists of a window that contain an image or a logo. The image appears at the start of the app which makes the splash screen the first screen the user interacts with. For the purpose of this app, an image will be used with the apps name at the top. The image will be displayed with the use of “ImageView” whereas the apps name will be displayed using “TextView”. A progress bar is also implemented at the bottom of the screen in the shape of a spinning wheel.

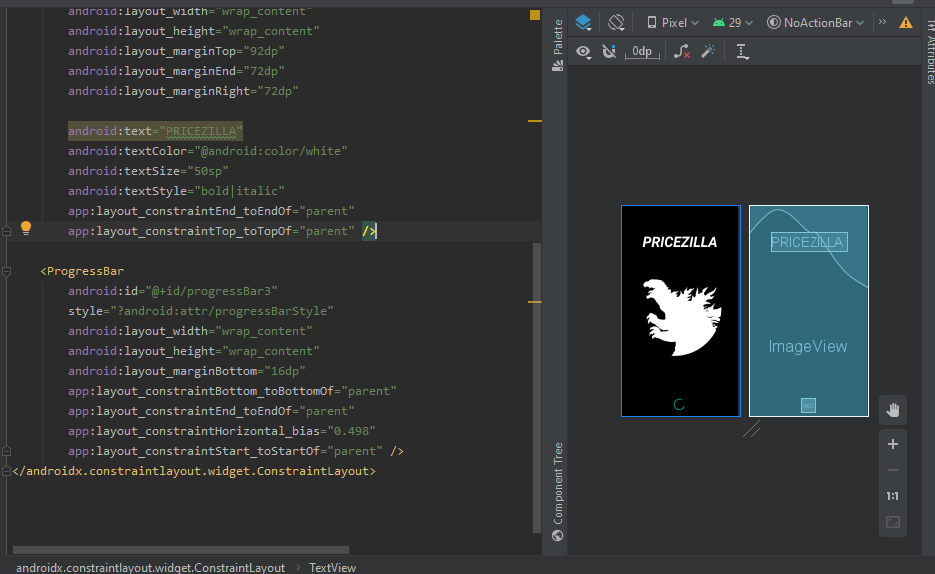


Figure 4.10

1. **SplashActivity:**

This activity is responsible for handling the splash screen. The content is set to the “splash\_activity” file. A *SPLASH\_TIME\_OUT* function is initialized with a time of 3000 milliseconds. This depicts that the splash screen should change to the home screen after exactly 3000 milliseconds or 3 seconds.

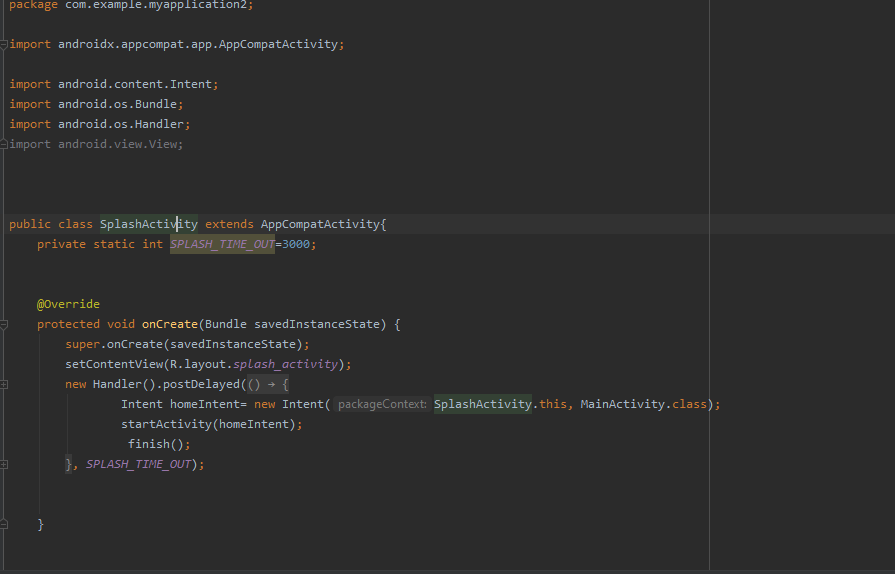


Figure 4.11

**4.3 Database:**

The database is used to store all the details needed for the app to run. Two collection will be used, namely, “prdoucts” and details. Their explanations are given below and the final database is shown below:

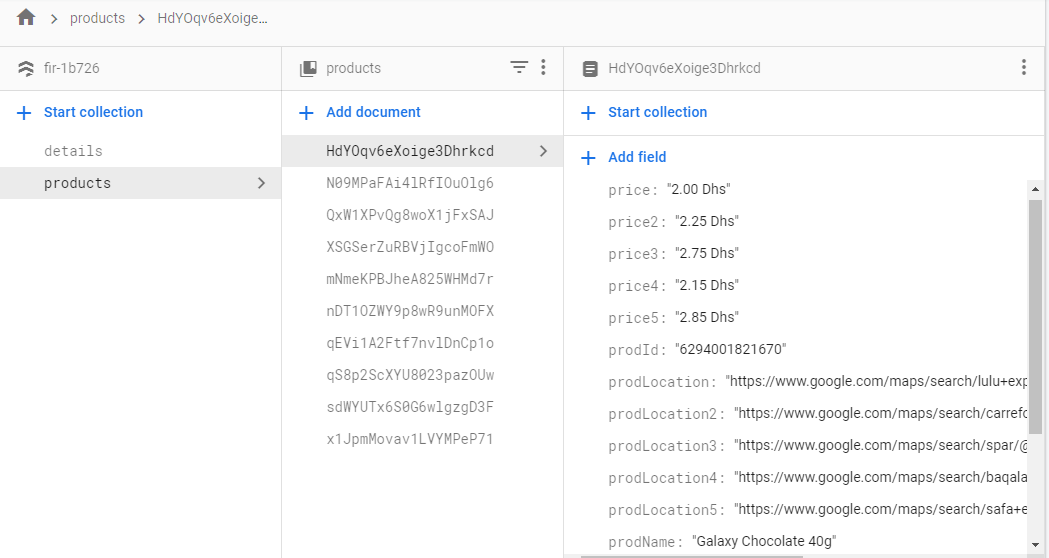


Figure 4.12

A common collection called ‘products’ is used which will contain all the information of all the necessary products. In the collection “products” documents are created with a random ID which contain all the information related to one specific bar-code. As the above figure 4.12 depicts, the document “HdYOqv6eXoige3Dhrkcd” contains all the information related to the bar-code number “6294001821670”. Once a user scans a product, the bar-code number is queried and all the information related to it such as the different prices, stores and locations are displayed to the user.

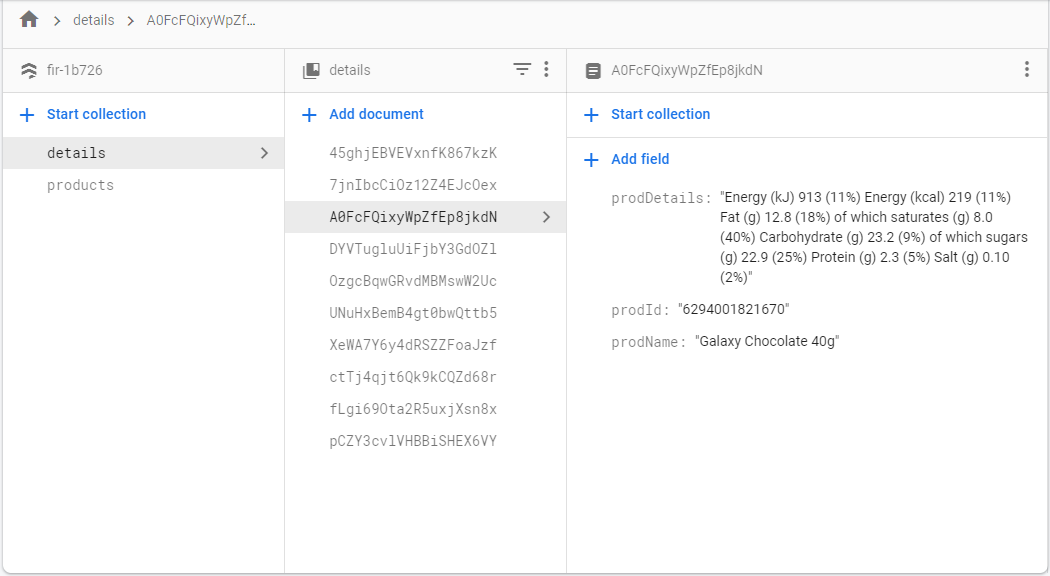


Figure 4.13

As shown in the figure 4.13, the database also contains another collection called “details” which contains nutritional information of a specific product and its bar-code number. Once the user scans the product, all the nutritional information associated with that bar-code is displayed on the screen.

Ten more documents are created in both “products” and “details” which contain different bar-code numbers. All the other documents in the collection have the same fields, which is prodId, prodDetails and so on, but different values, such as Carrefour, 2.00Dhs, and so on.

**4.4 System Screen-shots**

Once the codes have been written, they have to be run to generate the APK of the app. The final screen-shots of the system are shown below:

**Splash Screen: Home Page:**

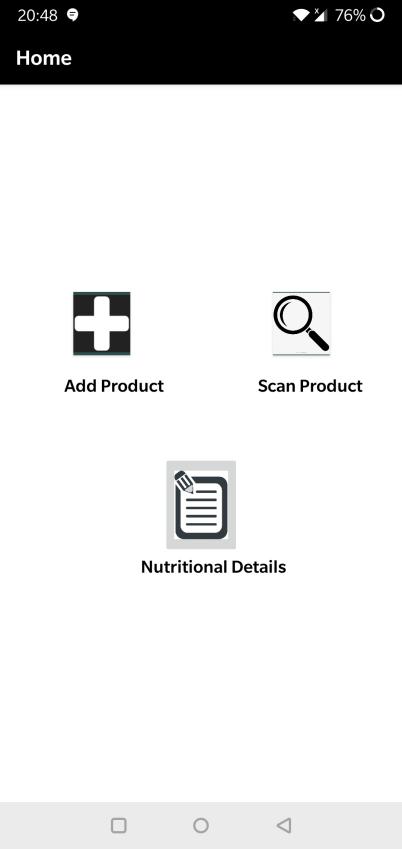
 

Figure 4.15

Figure 4.14

**Add Product:**

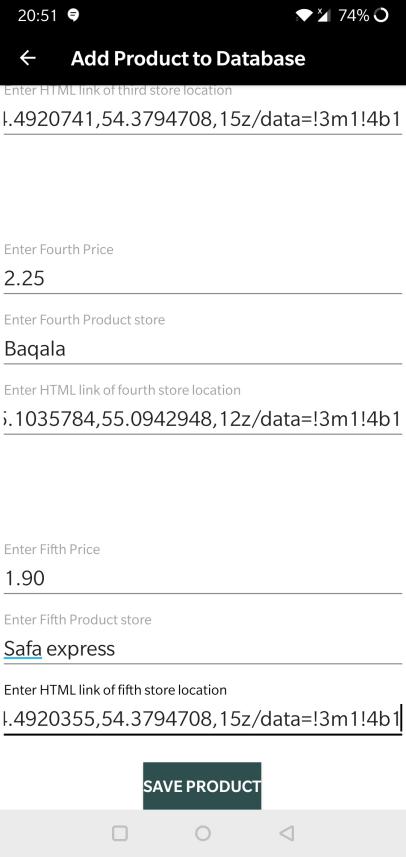
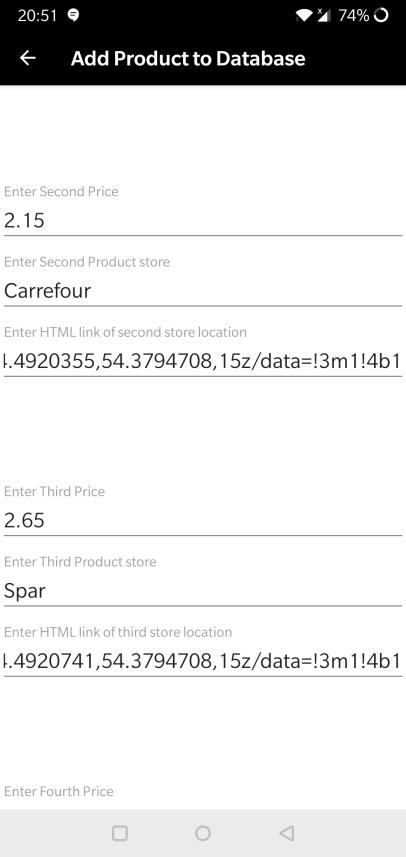
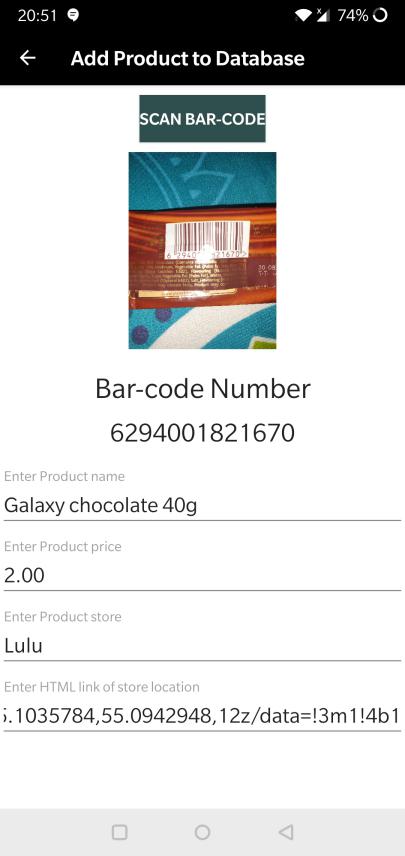


Figure 4.17

Figure 4.18

Figure 4.16

**Scan Product:**

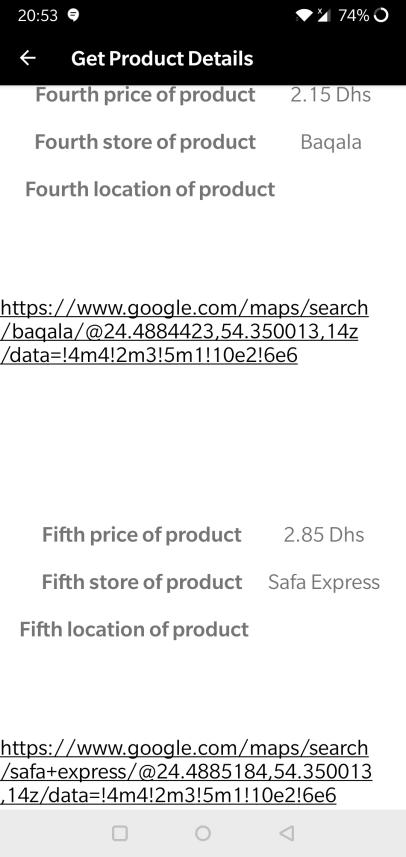
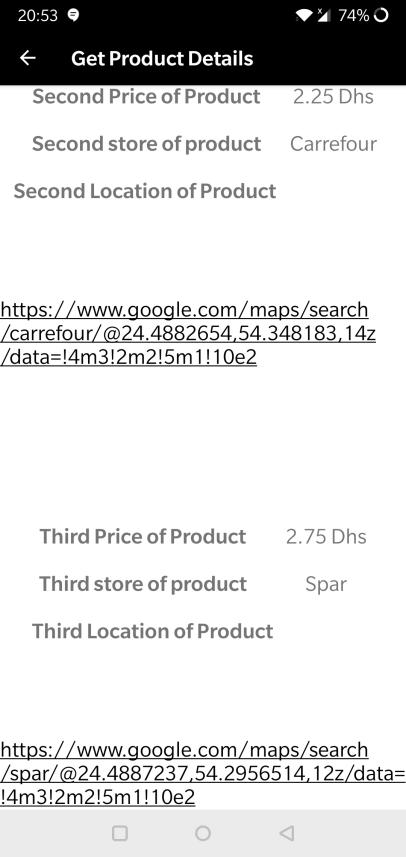
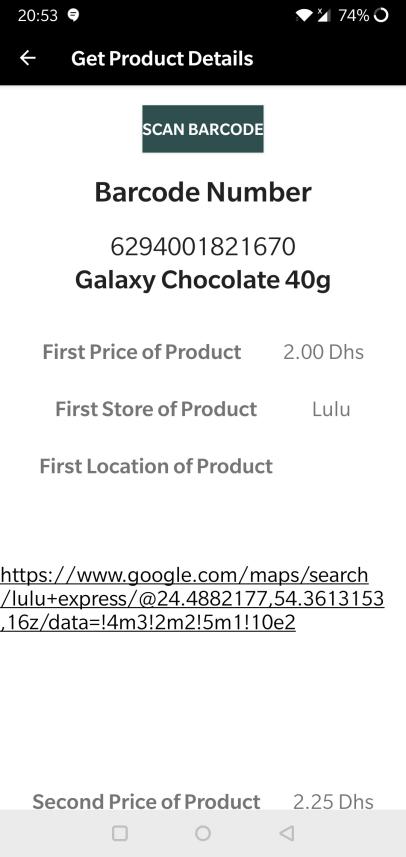


Figure 4.21

Figure 4.20

Figure 4.19

**Get Nutrition:**

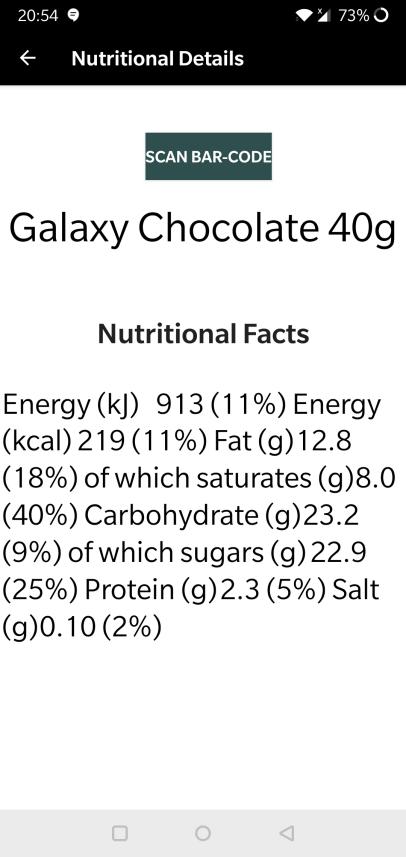


Figure 4.22

1. **Software Testing**

Software testing is the process by which the functionality of a software application is evaluated to identify defects and to check whether the software application has met its specified requirements. (Rajkumar, 2020) There are two approaches for software testing that are used in this project:

* White Box Testing
* Black Box Testing

**5.1 White Box Testing:**

White box testing, also known as clear box testing or glass box testing, is the evaluation on the inner workings of an application and revolves around internal testing. Its primary focus is to strengthen security, examining the flow of input and output through the application, and improving design and usability.

All the methods used in the program are executed and their goals and result are displayed in the table below:

|  |  |  |
| --- | --- | --- |
| **Method** | **Objective** | **Achieved Goal?** |
| onCreate() | Initialize and set the view of the activity | Yes |
| addProduct() | Open the ProductAdditionActivity class on click | Yes |
| getProduct() | Open the ProductReaderActivity class on click | Yes |
| getDetails() | Open the ProductDetails class on click | Yes |
| createProductObj() | Call objects of the variables declared | Yes |
| addProductToDb() | Create a collection and save the details which are entered | Yes |
| refreshUi() | Remove the previous entered details | Yes |
| onActivityResult() | Get bar-code value, display the value and image | Yes |
| captureBarcodePic() | Open up camera to capture the bar-code picture | Yes |
| getProductDetails() | Get the nutritional details of bar-code from database | Yes |
| displayProductDetails() | Display details recieved from getProductDetails() | Yes |
| readProductFromDb1() | Read bar-code number and send the value to getProductDetails() | Yes |
| getProductfromDB() | Get the relevant bar-code details of the bar-code from database | Yes |
| displayProductDetails() | Display details recieved from getProductfromDB() | Yes |
| readProductFromDb() | Read bar-code number and send the value to getProductfromDB() | Yes |
| readBarcodeValueTask() | Pass bar-code image to detect bar-code | Yes |
| setBarcodeValue() | Retrieve bar-code number from image | Yes |
| getUprightImage() | Display bar-code image as upright | Yes |

Table 2

As the table above suggests, all the methods have achieved the goal they set out to do with little to no faults.

**5.2 Black Box Testing:**

In black box testing, the functionality of the software under test is evaluated without considering the internal code structure, implementation details or knowledge of the internal paths of the software. This type of testing is based entirely on the software requirements and specifications. The table described below highlights the functionalities of the app and whether they were achieved or not.

|  |  |  |
| --- | --- | --- |
| **Function** | **Objective** | **Achieved Goal?** |
| Home Page | Select the add to database button | Yes |
| Select the scan bar-code button | Yes |
| Select the nutritional details button | Yes |
| Bar-code Scanner | Scan the bar-code | Yes |
| Get the bar-code image | Yes |
| Get the bar-code number | Yes |
| Product Search | Search with the scanned bar-code number | Yes |
| Send bar-code number to database | Yes |
| Product Add | Enter details related to the bar-code | Yes |
| Add bar-code details to database | Yes |
| Enter Google Maps HTML link of store | Yes |
| Product Details | Get product details | Yes |
| Get nutritional details | Yes |
| Get HTML link of store location | Yes |
| Product Location | Clickable HTML link | Yes |
|  | On click open Google Maps | Yes |

Table 3

As the table suggests, all the functions are working as expected.

**5.3 Usability Evaluation:**

Usability evaluation focuses on how well users can adapt to and learn the app to achieve their goals. Basically, it refers to the experience an external user has when using the proposed system. The user has to be kept in mind throughout the development of the app as it needs to cater to the people who are actually going to make use of it. Some of the user needs that need to be met are as follows:

* A clear understanding of the architecture and navigation of the app so as to not confuse the users.
* Developing an app which a user who has not been exposed to the user interface can learn and accomplish tasks.
* Users must be able to accomplish their tasks quickly and not waste time looking around for what they want in the app.
* The app must be memorable to entice the user and other potential users to use the app in the future.
* Help users recognize and recover from errors by displaying a toast message describing the error and providing a solution to the user for it.

(Usability Evaluation Basics | Usability.gov, 2020)

To gather information as to how the users have adapted to the app, a survey was conducted which contained the following questions for the user to answer:

* How likely are you to use an app to find cheaper prices/ deals of a product? \*
* How do you like the app design? \*
* What did you like MOST about the design? \*
* What did you like LEAST about the design \*
* What functions of the app do you like the MOST? \*
* What functions of the app do you like the LEAST? \*
* Are there any additional features you would like to see incorporated into the app? \*
* How was the overall interface of the app? \*
* How likely are you to use the app? \*
* What do you think Pricezilla could improve on? \*

The questions were made compulsory for all the survey takers so as to not just skip the questions and submit a blank survey. Their responses are shared below:

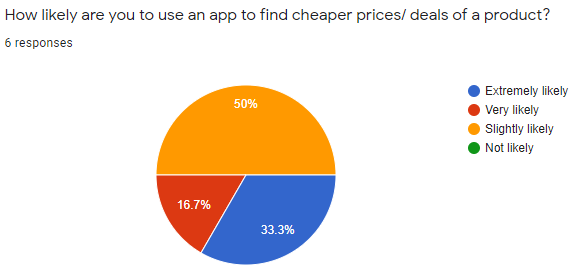


Figure 5.1

In figure 5.1, all the participants said it was likely that they would use an app to find cheaper and better deals of a product.

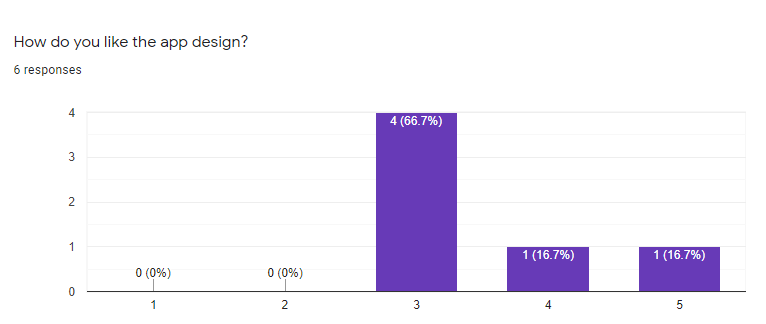


Figure 5.2

In figure 5.2, on a scale of 1 to 5 with 1 being the worst and 5 being the best, all the participants were somewhat satisfied or happy with the design of the app.

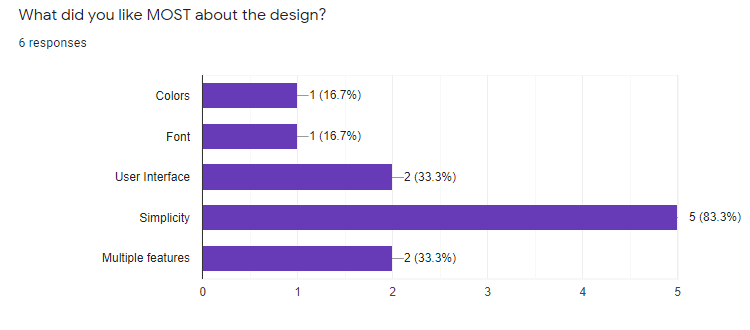


Figure 5.3

In figure 5.3, users had the option to select multiple answers in this question with most of the participants liking the simplicity of the app which meets the users needs described in the previous section.

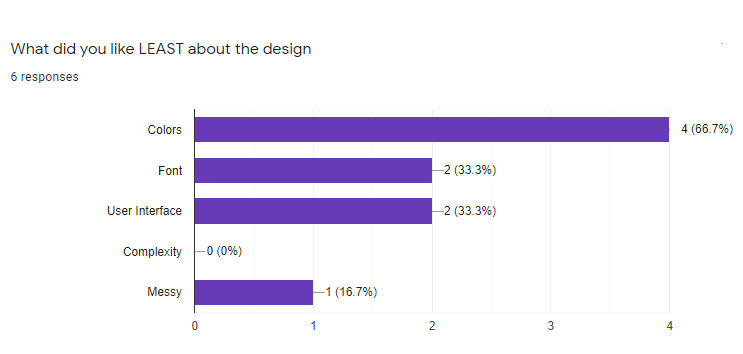


Figure 5.4

In figure 5.4, users had the option to select multiple answers in this question with most of the participants disliking the colors of the app design. On the positive side, none of the participants found the app hard to maneuver.

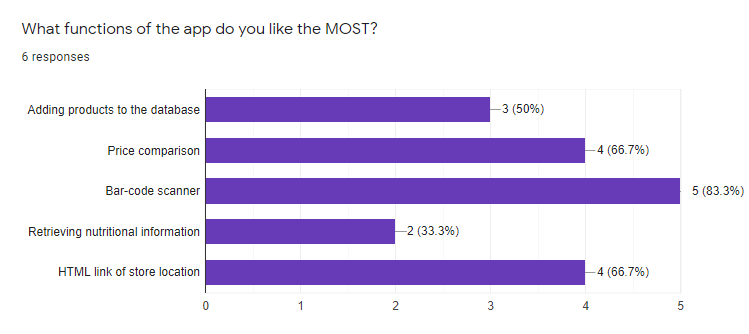


Figure 5.5

In figure 5.5, most of the users were satisfied with the bar-code scanner. The nutritional information didn’t seem to be a hit with the participants.

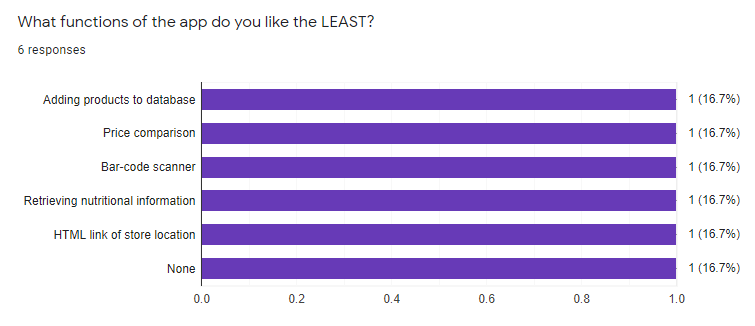


Figure 5.6

In figure 5.6, since the participants had the opportunity to select multiple options for this question, thy chose not to indicating they were satisfied with the app apart from a discrepancies here and there.

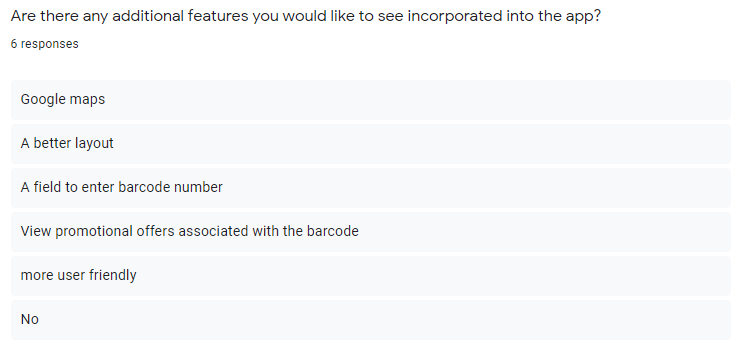


Figure 5.7

In figure 5.7, all the participants had different answers for this question. This gives the programmer an idea as to what to incorporate or change in the app for it to be a success.

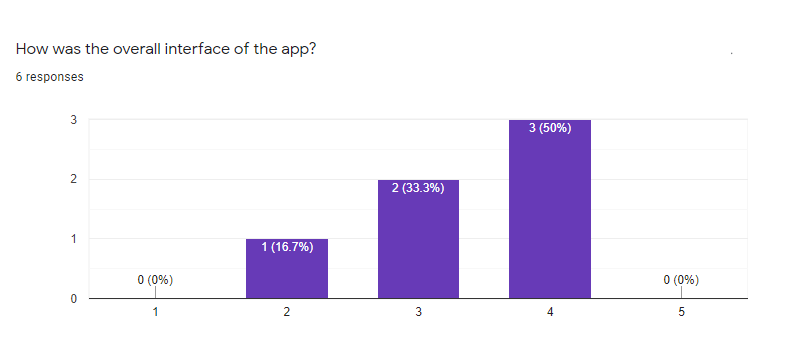


Figure 5.8

In figure 5.8, the participants seemed to be quite neutral about the app. The app didn’t exceed their expectations nor did it make it hard for them to use.

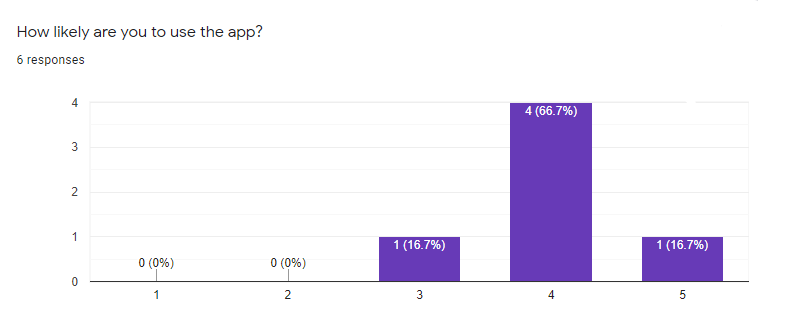


Figure 5.9

In figure 5.9, all the participants are likely to use the app in the future if they’re criticisms have been accounted for.



Figure 5.10

In figure 5.10, the app would mainly need to improve it’s layout and design, as well as the bar-code scanning aspect of it.

1. **Conclusion:**

In conclusion, the app was implemented successfully with all the coding components working perfectly with no errors. The app was implemented with the intent of helping consumers save money while shopping by comparing the price of a product among many outlets. The process of comparing the process is done with the help of a bar-code scanner by selecting a product, scanning its bar-code number and retrieving its pertinent information. The user also has the advantage of getting the information of all the stores near the their area by clicking an HTML hyperlink which opens up Google Maps and displays it to them. The user then has the freedom to select what store they would prefer going to. In addition to that the user also has an opportunity to add a product to the database. This is done when a particular product is not available and the user would like to add it to the database for any future users to find. Finally, a feature which allows the user to get nutritional information about the product has also been established. The user can scan the bar-code number and all the nutritional details associated with it is displayed to them on their screen. Overall, the goals set out in the beginning have been achieved. Although the system works fine, there still are some limitations with reference to its functionalities. This section will focus on the limitation of the app as well as any additional features that could be used to develop it further.

**6.1 Limitations:**

Limitations of a project are critical reflection’s of ones work. The functions that would need to be addressed and improved for the app to be a successful software have been stated below:

* Bar-code scanner: From the survey conducted in the Implementation section and from personal use and testing of the application, the bar-code scanner needs to be improved. Although it does scan the bar-code when its picture is taken, it needs to be under the right circumstances. If there is an issue with lighting or if the image captured isn’t clear, the bar-code number isn’t extracted from the image and the user is prompted to take the picture again.
* Database: If a user adds the same product to the database, there is no method which tells the user that the product is in the database.
* Layout: As indicated by most survey participants, the layout isn’t great. The colour and design is nominal. However, the users had reported that the splash screen was very good and captivating.
* Map: The app only allows the user to click on an HTML link and open Google maps. There is no method by which the user can find out the distance from their location to the store in the same app without having to leave and open up Google Maps.
* Search Function: This app is purely a bar-code scanner and hence it does not allow users to search products by name. This is done mainly to avoid any conflict of the product details in the database.

These complications could not have been met due to lack of time and resources. Hence, these complications will be addressed and solved some time in the future.

**Future Works:**

To develop this app in the future and increase the scale of users using the app, the app will need to implement some additional features to remain up to date with the growing needs of the market.

* A search bar needs to be implemented which allows the user to enter the bar-code number and search for details associated to it. This search bar can be used when the bar-code scanner ceases to work due to some unforeseen error. The search bar could also be used to search for the product name directly.
* Instead of having an HTML link which opens the app, Google Maps, up on the users phone, the app could have Google Maps integrated into it which would make the app unique.
* A save list function could be implemented. Users would select a product from the searched products and then save it into a checklist for future reference or use.
* A great function which would aid users in shopping would be a price alert function. User could select a product they like and have a notification turned on for that specific product for when its price drops. This would be useful for customers who are looking for certain products at a reduced cost.
* A user review system could be implemented which would allow the users to post positive and negative feedback of the app. This would help the programmer by giving them an idea of what needs to be improved in the app or what functions of the app are working perfectly. This would also help other users looking for bar-code scanning price comparison apps as they would know what other users feel about this specific app.

**7. References:**

Trujillo, P. (2020). *The Difference Beween a Barcode Scanner and a Mobile Computer*. [online] Wasp Buzz. Available at: http://www.waspbarcode.com/buzz/barcode-scanner-or-mobile-computer/ [Accessed 10 Jan. 2020].

Axicon.com. (2019). *Check digit calculator | Check your barcode*. [online] Available at: http://www.axicon.com/checkdigitcalculator.html [Accessed 10 Jan. 2020].

Barcodes Inc. (2018). *Barcoding Frequently Asked Questions (FAQ)*. [online] Available at: https://www.barcodesinc.com/faq/ [Accessed 10 Jan. 2020].

Upccode.net. (2020). *GS1 US Barcodes - Need GS1 US Issued Official & Legal UPC & EAN Bar Codes?*. [online] Available at: https://www.upccode.net/upc-guide/gs1-US.html [Accessed 10 Jan. 2020]

Shopify. (2020). *Universal Product Code (UPC) Definition - What is Universal Product Code (UPC)*. [online] Available at: https://www.shopify.com/encyclopedia/universal-product-code-upc [Accessed 10 Jan. 2020].

*Pathak, Bhavik. (2010). A SURVEY OF THE COMPARISON SHOPPING AGENT-BASED DECISION SUPPORT SYSTEMS*. Journal of Electronic Commerce Research. 11 [Accessed 10 Jan. 2020].

Datafeedwatch.com. (2020). *What are Comparison Shopping Engines?*. [online] Available at: https://www.datafeedwatch.com/academy/comparison-shopping-engines [Accessed 10 Jan. 2020].

crunchbase.com. (2019). *ShopSavvy*. [online] Available at: https://www.crunchbase.com/organization/shopsavvy#section-overview [Accessed 10 Jan. 2020].

Ferreira, N. (2020). *25+ Best Price Comparison Websites and Apps to Compare Prices Online*. [online] Oberlo. Available at: https://www.oberlo.com/blog/25-best-price-comparison-websites [Accessed 10 Jan. 2020].

Daurer, S. (2019). *AIS Electronic Library (AISeL) - Wirtschaftsinformatik 2017 Proceedings: Consumer Preferences for Product Information and Price Comparison Apps*. [online] Aisel.aisnet.org. Available at: https://aisel.aisnet.org/wi2017/track10/paper/5/

Blog.3dcart.com. (2019). *Make Shopping Easy with the Product Comparison App*. [online] Available at: https://blog.3dcart.com/blog/make-shopping-easy-with-the-product-comparison-app [Accessed 24 Oct. 2019].

Rouse, M., 2018. *What Is Android Studio? - Definition From Whatis.Com*. [online] SearchMobileComputing. Available at: <https://searchmobilecomputing.techtarget.com/definition/Android-Studio> [Accessed 30 April 2020].

Pilon, A., 2016. *Comparison Shopping Stats: Consumers Get Info From Big Stores | AYTM*. [online] Aytm.com. Available at: <https://aytm.com/blog/comparison-shopping-survey/> [Accessed 30 April 2020].

Firebase. 2020. *Cloud Firestore  |  Firebase*. [online] Available at: <https://firebase.google.com/docs/firestore> [Accessed 30 April 2020].

Rajkumar, 2020. *What Is Software Testing – Definition, Types, Methods, Approaches*. [online] Softwaretestingmaterial.com. Available at: <https://www.softwaretestingmaterial.com/software-testing/> [Accessed 30 April 2020].

Usability.gov. 2020. *Usability Evaluation Basics | Usability.Gov*. [online] Available at: <https://www.usability.gov/what-and-why/usability-evaluation.html> [Accessed 30 April 2020].

**8. Appendices**

**8.1 Survey:**

The survey conducted for the user evaluation is given below:

1. How likely are you to use an app to find cheaper prices/ deals of a product? \*

Extremely likely

Very likely

Slightly likely

Not likely

1. How do you like the app design?

1 (Worst)

2

3

4

5 (Excellent)

1. What did you like MOST about the design? \*

Colors

Font

User Interface

Simplicity

Multiple features

Other: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What did you like LEAST about the design

Colors

Font

User Interface

Complexity

Messy

Other:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What functions of the app do you like the MOST?

Adding products to the database

Price comparison

Bar-code scanner

Retrieving nutritional information

HTML link of store location

Other:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. What functions of the app do you like the LEAST?

Adding products to database

Price comparison

Bar-code scanner

Retrieving nutritional information

HTML link of store location

Other:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Are there any additional features you would like to see incorporated into the app?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. How was the overall interface of the app?

1 (Worst)

2

3

4

5 (Excellent)

1. How likely are you to use the app?

1 (Not Likely)

2

3

4

5 (Extremely Likely)

1. What do you think Pricezilla could improve on?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**8.2 Meeting Log:**

A meeting log has been kept which depicts the number of times I have met or had contact with my supervisor throughout the course of the year.

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Place** | **Agenda of Meeting** | **Approval** |
| 22/10/2019 | Middlesex University Dubai, Block 16, Room 305. | The final year project topic was discussed. | G.J |
| 16/12/2019 | Middlesex University Dubai, Block 16, Room 305. | Some doubts on the literature topic were discussed. | G.J |
| 23/2/2019 | Middlesex University Dubai, Block 16, Room 305. | The progress if the final year application was discussed. | G.J |
| 8/3/2019 | Email | Issues regarding the bar-code scanner were addressed | G.J |
| 21/3/2019 | Email | Deadline of the code and report was addressed | G.J |
| 3/4/2019 | Email | Details about the final report and an issue regarding the database no adding details into it were addressed. | G.J |
| 5/4/2019 | Email | Issue regarding the database was addressed and fixed. | G.J |
| 14/4/2019 | Email | Details of the final report as well as the follow up of my app were discussed | G.J |
| 27/4/2019 | Email | Details regarding the discussion of classes in my report was cleared | G.J |
| 28/4/2019 | Email | Review of my report | G.J |

**8.3 MIDDLESEX UNIVERSITY Faculty of Science and Technology RESEARCH ETHICS APPROVAL FORM**

*NOTES ON COMPLETING THIS FORM:*

The purpose of this form is to help **both staff and students** in their pursuit of ethical research methodologies and procedures. It is your responsibility to ensure that you have considered all relevant ethical issues.

*No fieldwork or other method of data collection should begin until this form has been completed and, where necessary, approval has been given by the Research Ethics Committee, Dr Carlisle George is the representative of the Ethics Committee. His email address is c.george@mdx.ac.uk. His office is in Town Hall of Hendon Campus.*

In completing the form, please give as much detail as possible. If a question is not applicable, please indicate by marking N/A.

Students should discuss and complete the form with their supervisors/tutors.

In relation to collaborations with external partners and applications to external bodies, the following principles should be borne in mind when completing this form:

partner institutions and researchers are expected to accept ethical standards which are equivalent to or higher than those applied by Middlesex University

where partner institutions require higher ethical standards than those currently applied by the

CST3590

Middlesex University, Middlesex staff and students should adhere to them

where funding bodies or other external agencies lay down minimum ethical standards for research, these should be complied with by Middlesex staff.

1. **NAME OF RESEARCHER:…………** Ayman
2. **PROGRAMME OF STUDY:……………**BEng Computer Systems Engineering
3. **NAME OF SUPERVISOR/TUTOR:…………………**Mrs. Geethu Joy
4. **NAMES OF ANY RESEARCH COLLABORATORS:……………**N/A
5. **ARE YOU UNDERGRADUATE**? [delete asappropriate].
6. **PROPOSED TITLE OF RESEARCH PROJECT:…………………**Bar-code Price Comparison App
7. ***BRIEF* DESCRIPTION OF THE MAIN AIMS OF THE STUDY**:………… To create a bar-code scanning price comparison apps to help consumers while shopping.
8. **HAVE YOU READ AND UNDERSTOOD THE UNIVERSITY’S *CODE OF* *PRACTICE FOR RESEARCH: PRINCIPLES AND PROCEDURES*?**YES
9. **WILL YOU COLLECT PRIMARY DATA (FOR EXAMPLE, THROUGH INTERVIEWS)?** YES

**IF NO, PLEASE GO TO 18.BELOW AND SIGN THE FORM**.*ALL**STUDENTS MUST ALSO HAVE THE FORM COUNTERSIGNED BY THEIR SUPERVISOR/TUTOR.*

**10. WILL YOUR RESEARCH INVOLVE:**

1. CONDUCTING INTERVIEWS? **NO.**

IF YES, STATE WITH WHOM…….

1. PARTICIPANT OBSERVATION? **YES**
2. USE OF QUESTIONNAIRE(S)WHICH YOU HAVE DESIGNED? **YES**

d) FOCUS GROUPS

**NO**

e) OBSERVATION

**NO**

f) OTHER FORMS OF PEOPLE INFORMATION GATHERING

**NO**

CST3590

1. **WILL YOU OBTAIN WRITTEN INFORMED CONSENT DIRECTLY FROM RESEARCH PARTICIPANTS?……………. NO**
2. **DO YOU INTEND TO OFFER INCENTIVES TO RESEARCH**

**PARTICIPANTS?…………….…NO**

IF YES, PLEASE SPECIFY..

1. **WILL YOU INFORM PARTICIPANTS OF THEIR RIGHT TO WITHDRAW FROM THE RESEARCH AT ANY TIME?………… NO**
2. **WILL YOU GUARANTEE CONFIDENTIALITY OF INFORMATION TO PARTICIPANTS?…………………… YES**

If no, please give reasons

1. **WILL YOU GUARANTEE ANONYMITY TO PARTICIPANTS?…… YES** If no, please give reasons
2. **DOES YOUR RESEARCH METHODOLOGY RAISE ANY SAFETY/LEGAL ISSUES FOR YOU OR YOUR PARTICIPANTS? ……….NO**

**IF YES TO EITHER, PLEASE SPECIFY…**

1. **DO YOU HAVE ANY ETHICAL CONCERNS ABOUT THIS RESEARCH PROJECT? [FOR EXAMPLE,ABOUT FINANCIAL SUPPORT,CONFLICTS**

**OF INTEREST ETC]………………………….NO**

IF YES, PLEASE SPECIFY…

1. **STUDENT DECLARATION**

THE INFORMATION GIVEN ON THIS FORM IS TRUE TO THE BEST OF MY KNOWLEDGE. I WILL USE THESE METHODS IN MY RESEARCH UNLESS I RENEGOTIATE ANY CHANGES WITH MY SUPERVISOR/ TUTOR.

CST3590

30/4/2020

Ayman

STUDENT SIGNATURE……………………………………… DATE …………………………

[THE FOLLOWING STATEMENTS ARE FOR THE SUPERVISOR/TUTOR TO DELETE AS APPROPRIATE.

1. I HAVE READ THE INFORMATION SUPPLIED ON THIS FORM AND DO NOT THINK THAT IT RAISES ANY ISSUES THAT NEED TO BE CONSIDERED BY *Ethics Committee*.
2. I HAVE READ THE INFORMATION SUPPLIED ON THIS FORM AND

HAVE REFERRED/WILL REFER THE PROPOSAL TO *Ethics Committee* FOR THEIR CONSIDERATION.

c) THIS PROPOSAL HAS BEEN APPROVED BY *Ethics Committee*.

G.J

30/4/2020

**SIGNATURE OF SUPERVISOR/LECTURER** ………………………… DATE …………………

---------------------------------------------------------

1. **STAFF DECLARATION**

**THE INFORMATION GIVEN ON THIS FORM IS TRUE TO THE BEST OF MY KNOWLEDGE. I HAVE TAKEN NOTE OF APPROPRIATE GUIDELINES IN DESIGNING MY RESEARCH AND APPRECIATGE THE NEED TO KEEP ETHICAL ISSUES UNDER REVIEW DURING THE RESEARCH PROCESS.**

30/4/2020

G.J

**SIGNATURE OF LECTURER**………………… DATE …………………

[THE FOLLOWING STATEMENTS SHOULD BE DELETED AS APPROPRIATE.]

1. I DO NOT THINK THAT THIS PROPOSED RESEARCH RAISES ANY ISSUES THAT NEED TO BE CONSIDERED BY *Ethics Committee*.
2. I HAVE REFERRED/WILL REFER THIS RESEARCH PROPOSAL TO

*Ethics Committee* FOR THEIR CONSIDERATION.

c) THIS PROPOSAL HAS BEEN APPROVED BY *Ethics Committee*.

G.J

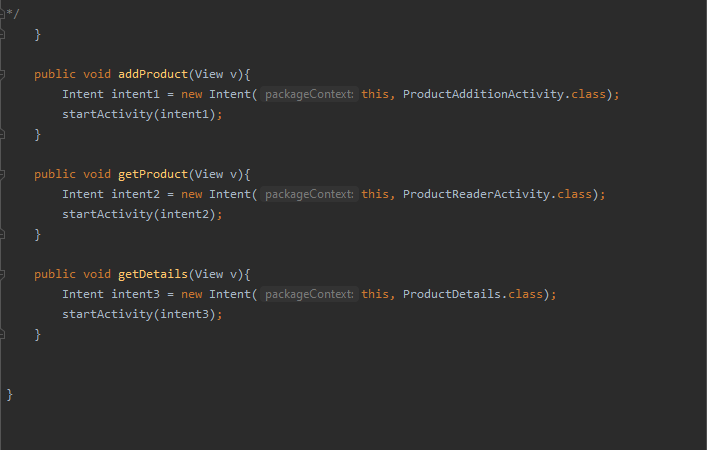
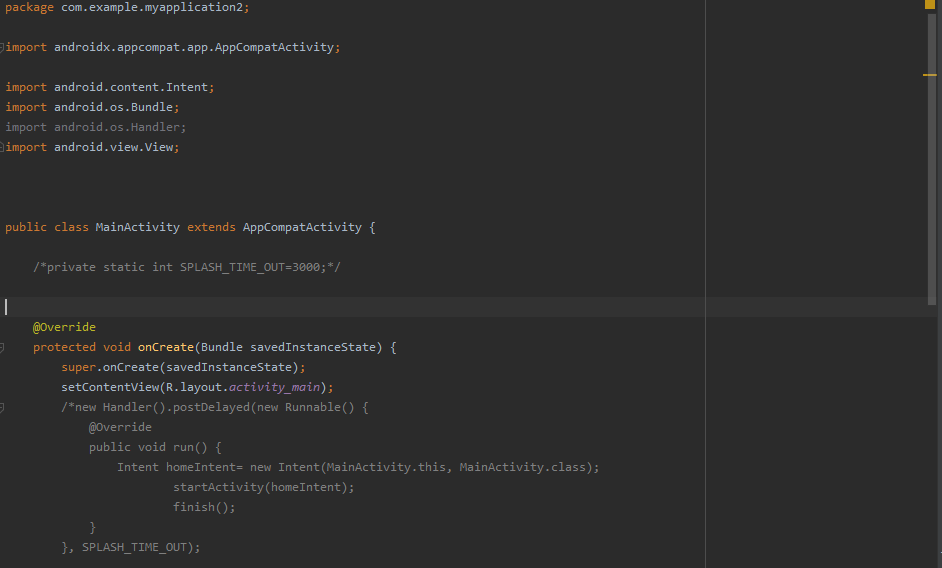
**SIGNATURE OF ACADEMIC GROUP RESEARCH LEADER**…………… DATE……………

30/4/2020

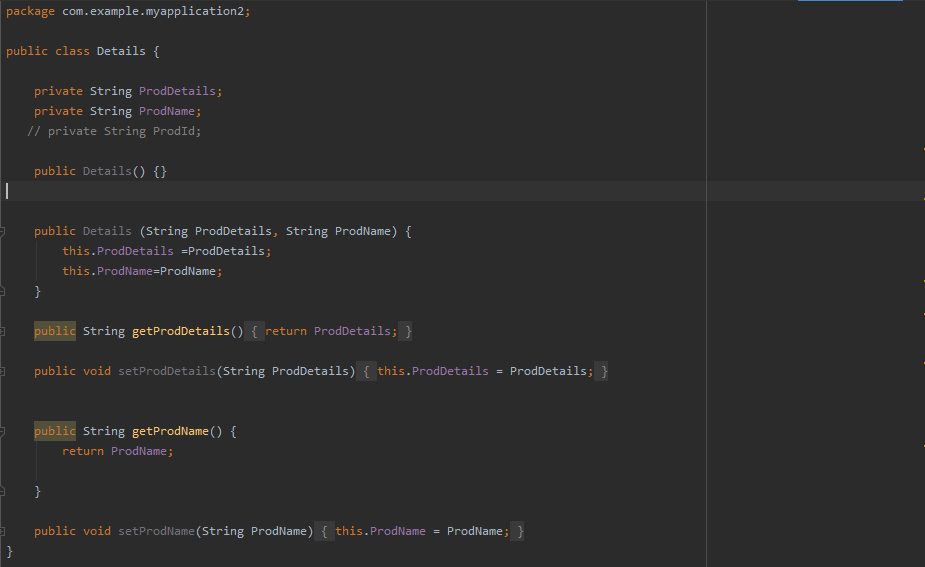
**8.4 Program codes:**

Images of all the codes used in the development of this app have been shown below.

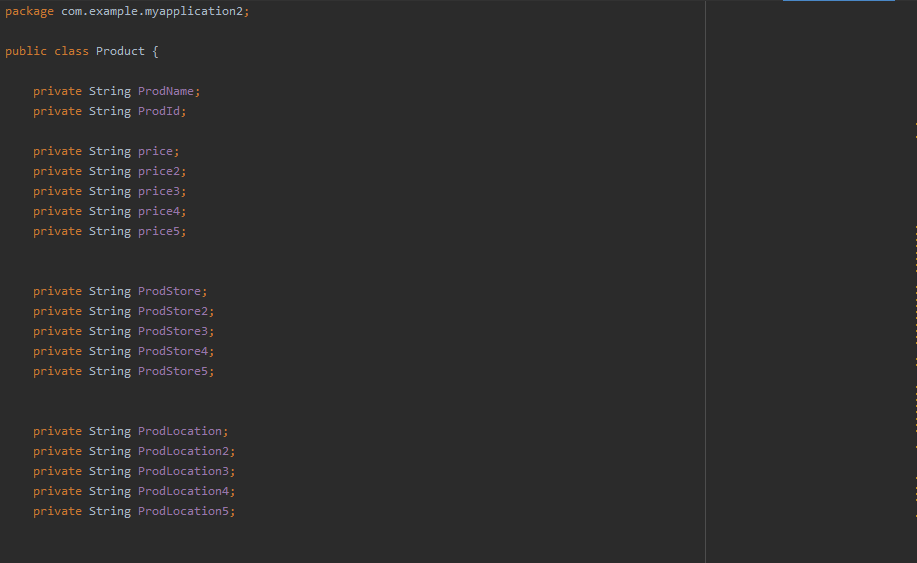
**Main Activity:**

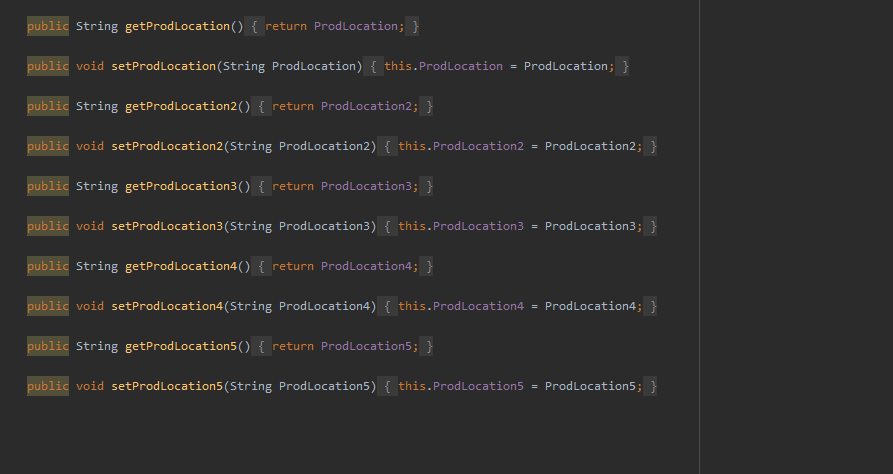
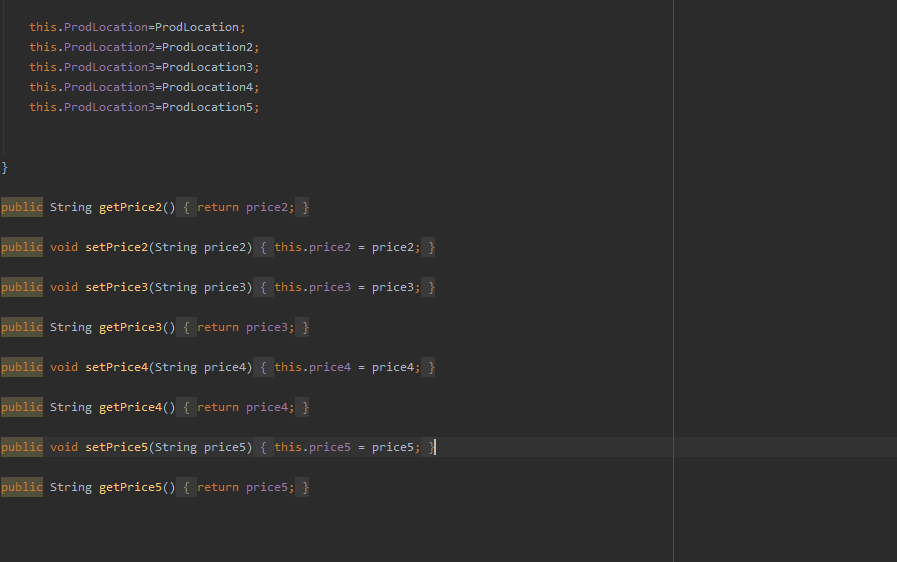
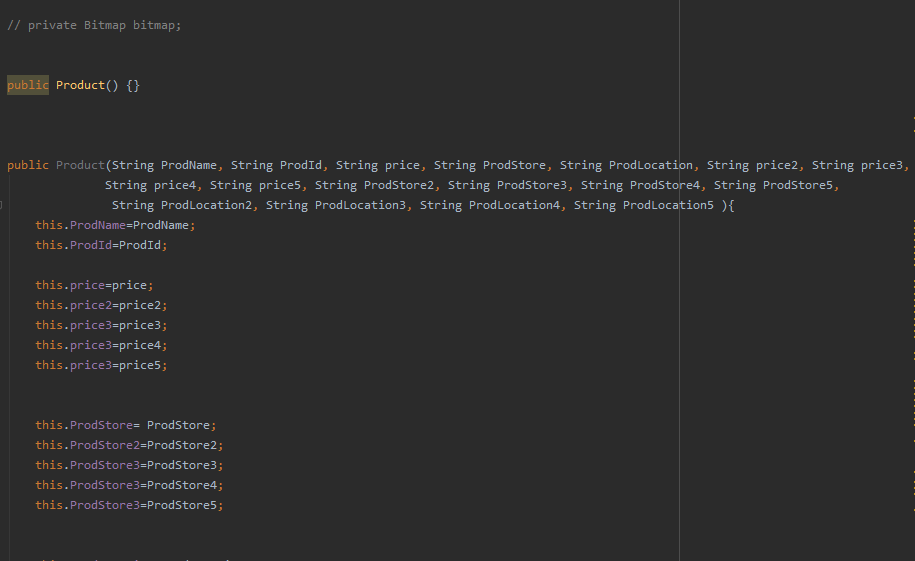


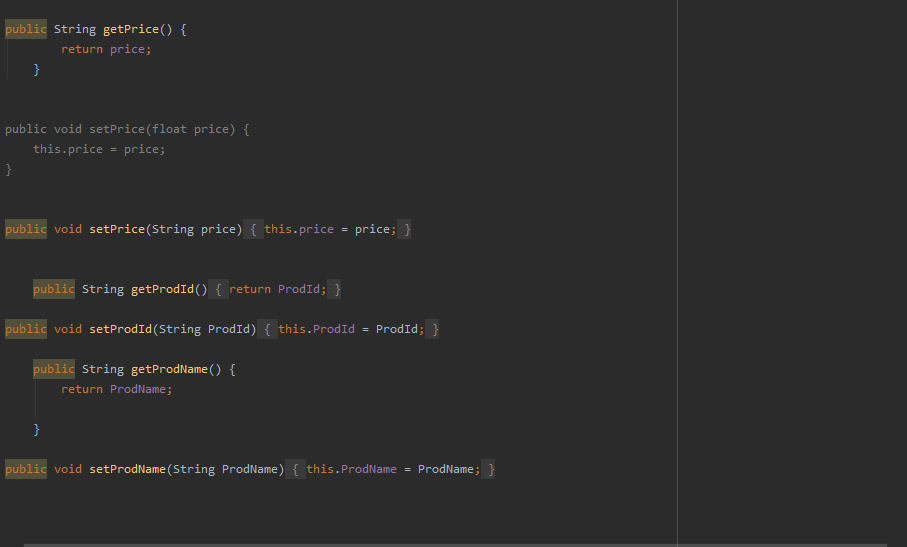
**Details:**



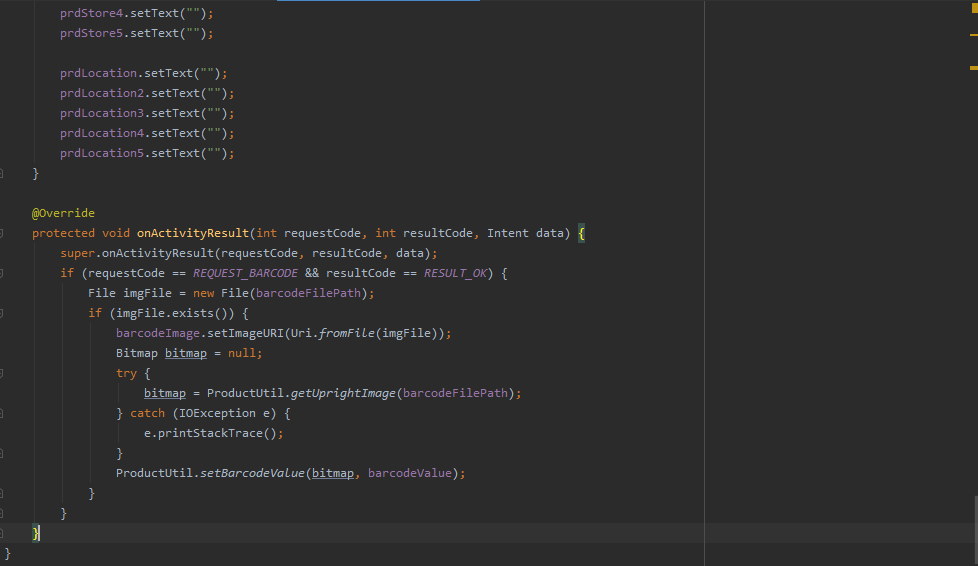
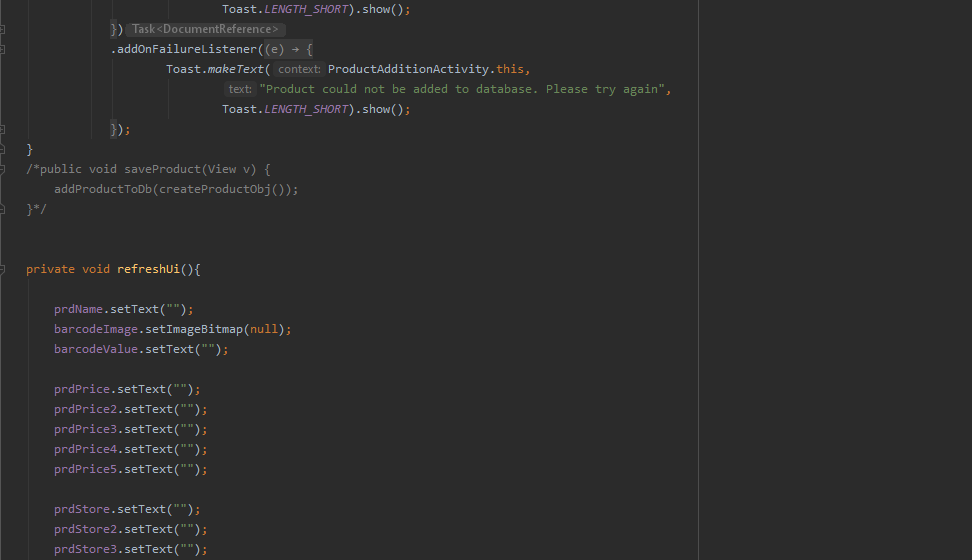
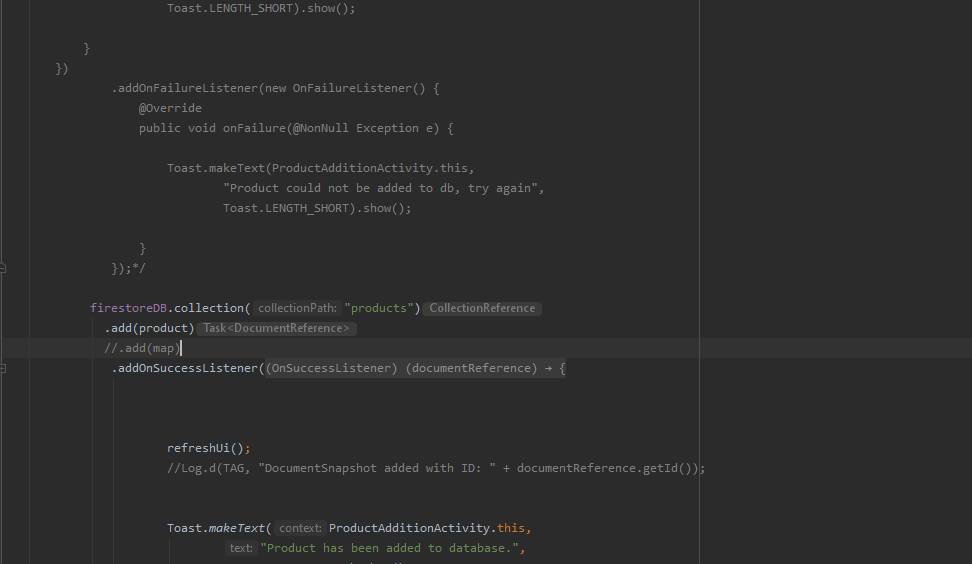
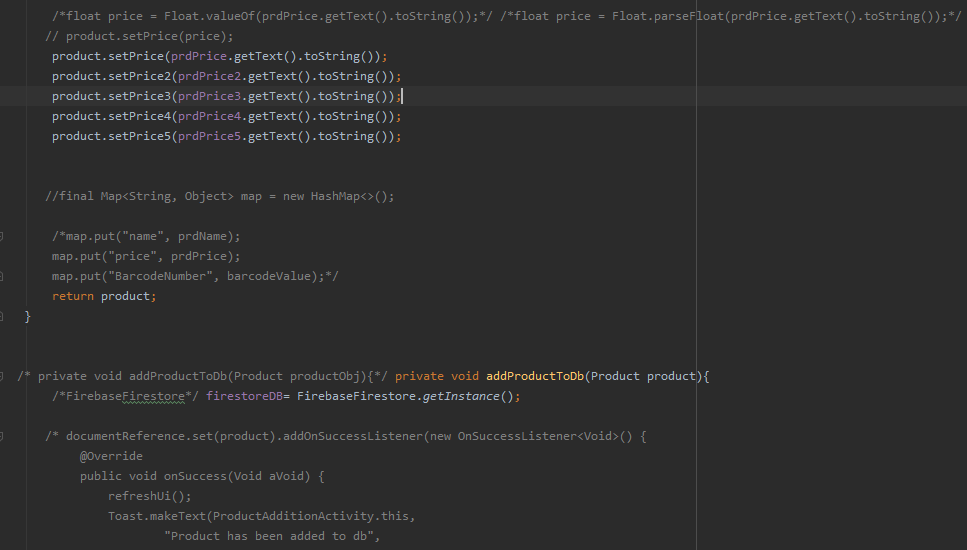
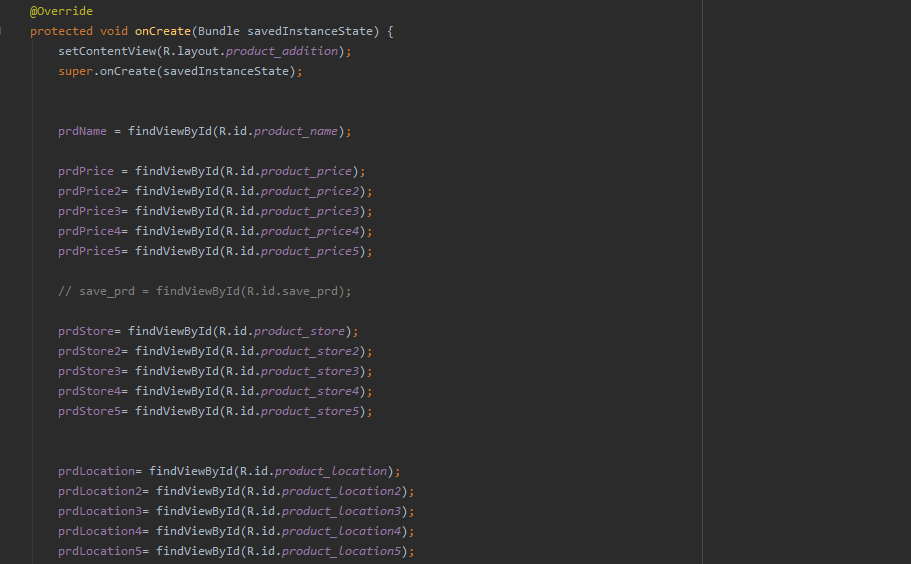
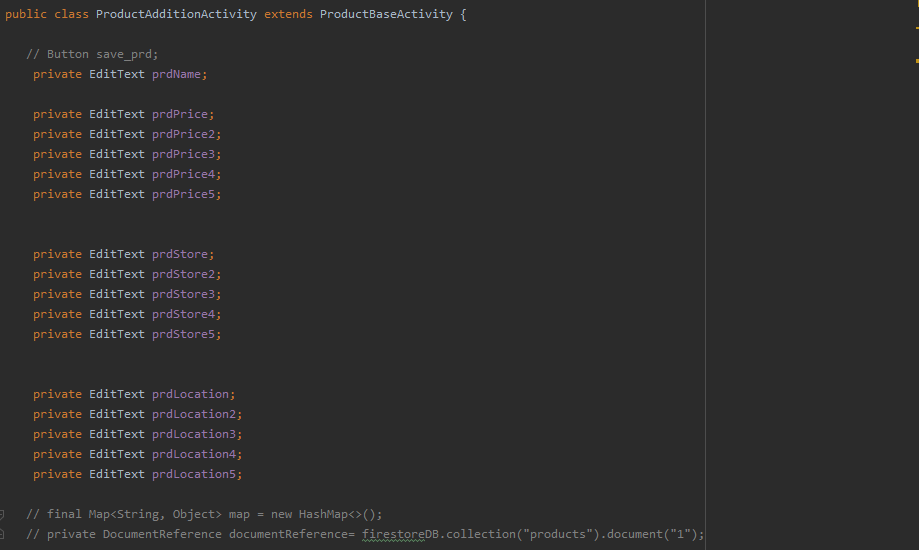
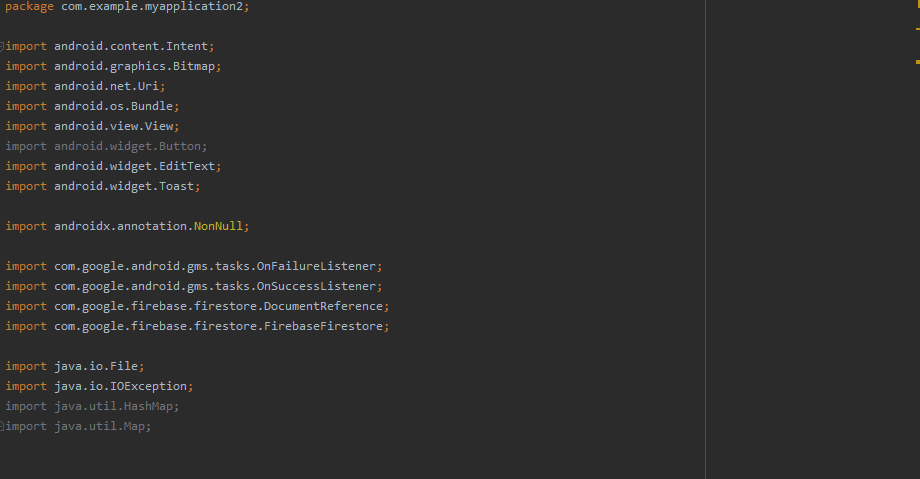
**Product:**



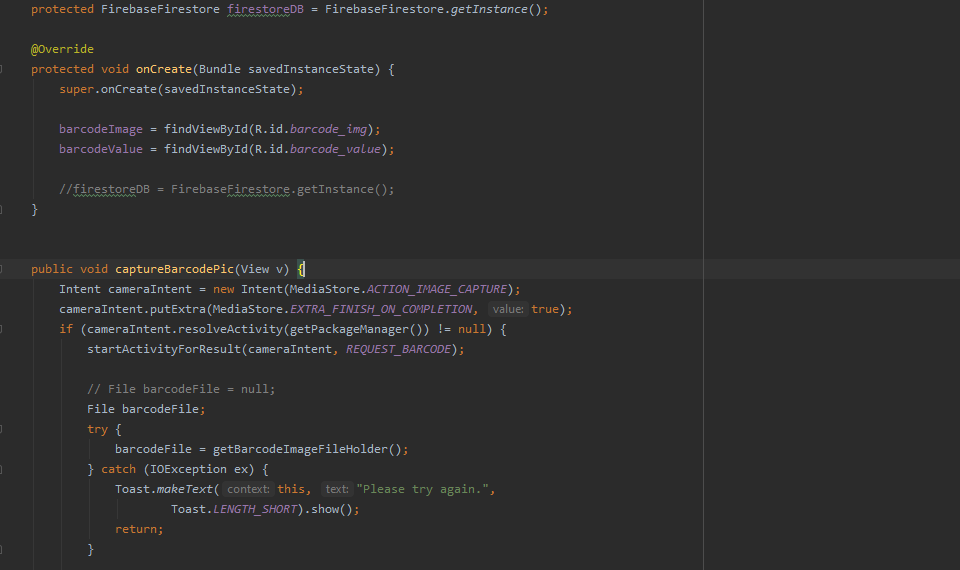
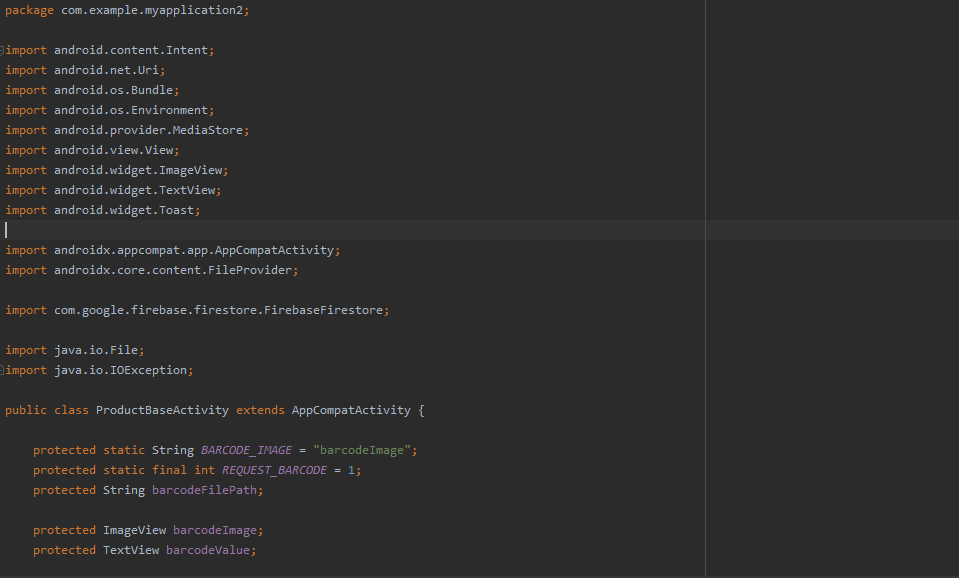




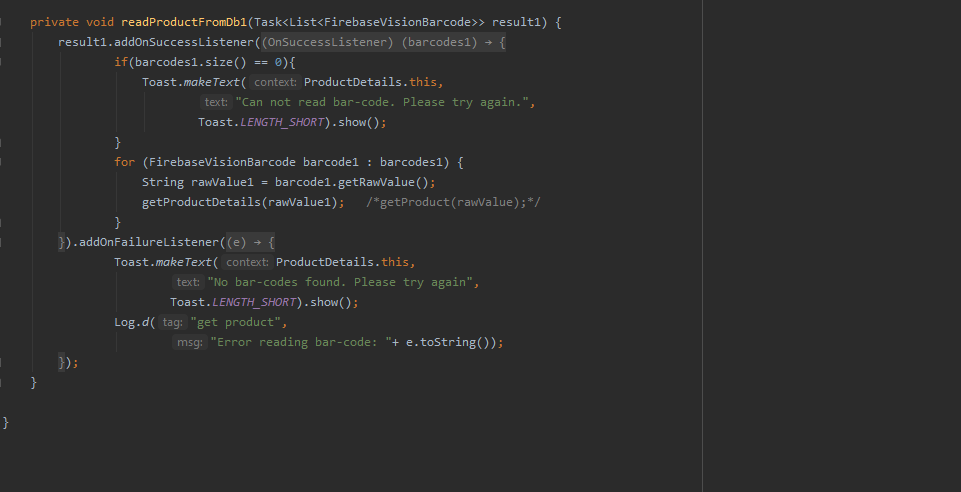
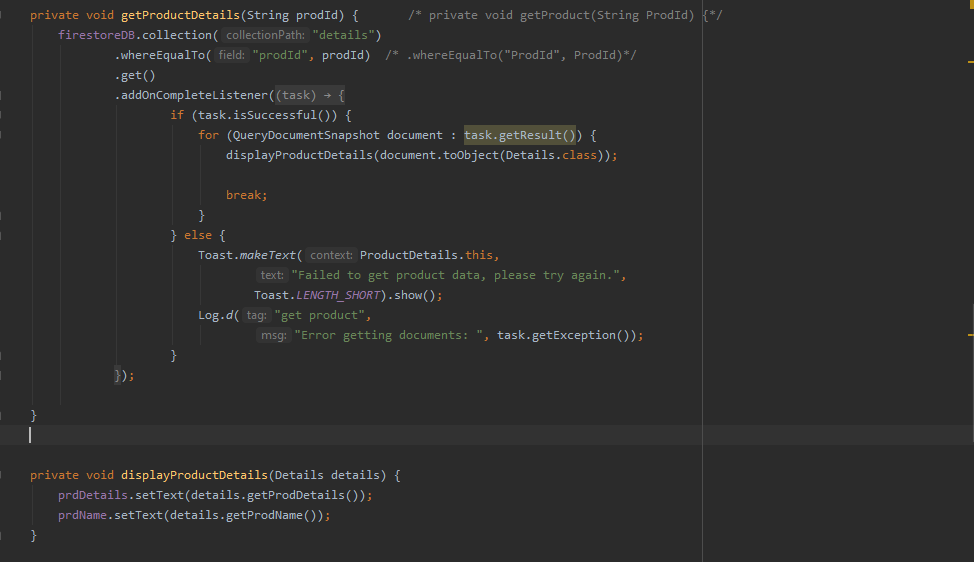
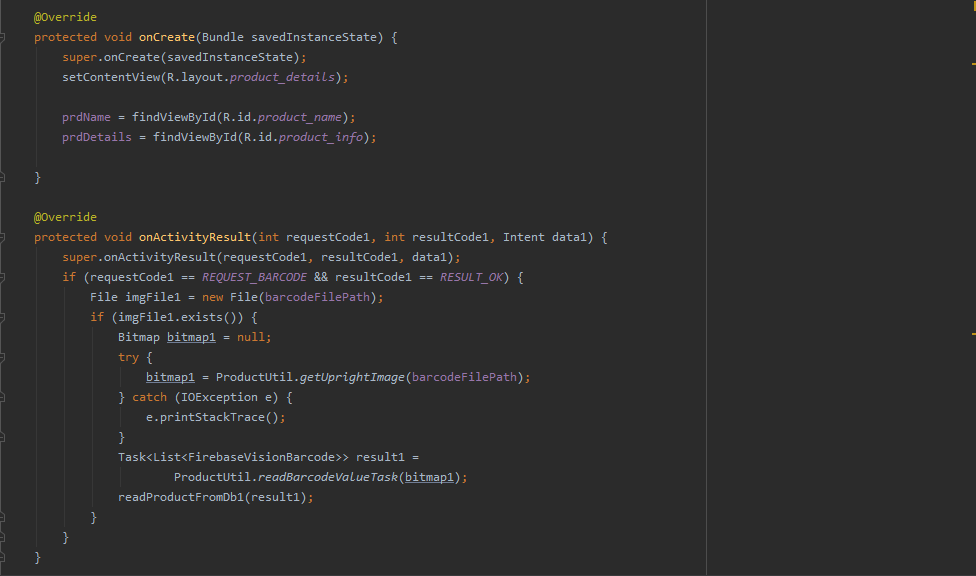
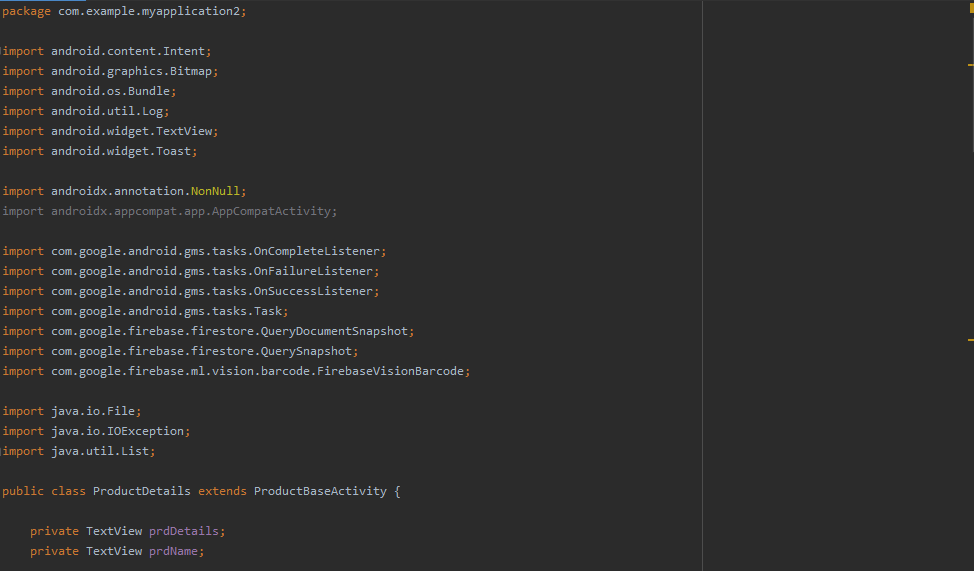
**ProductAdditionActivity:**



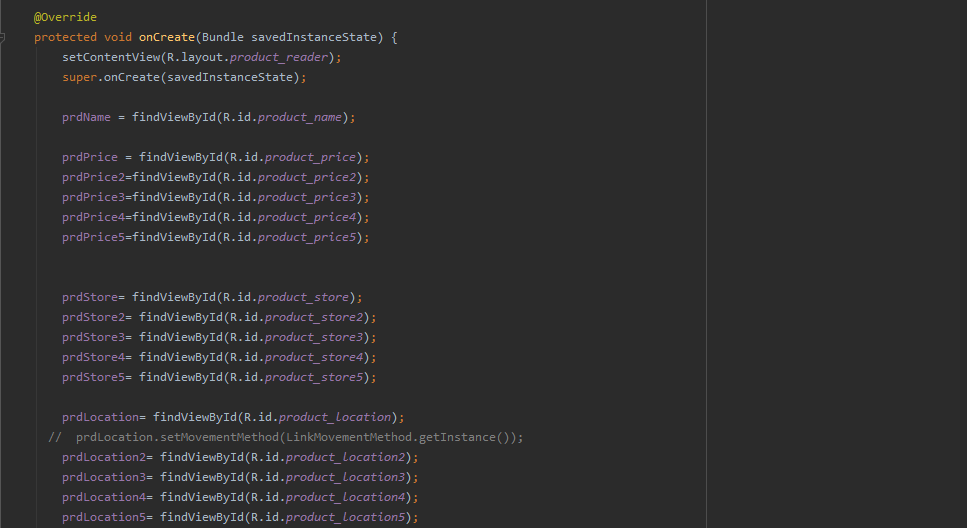
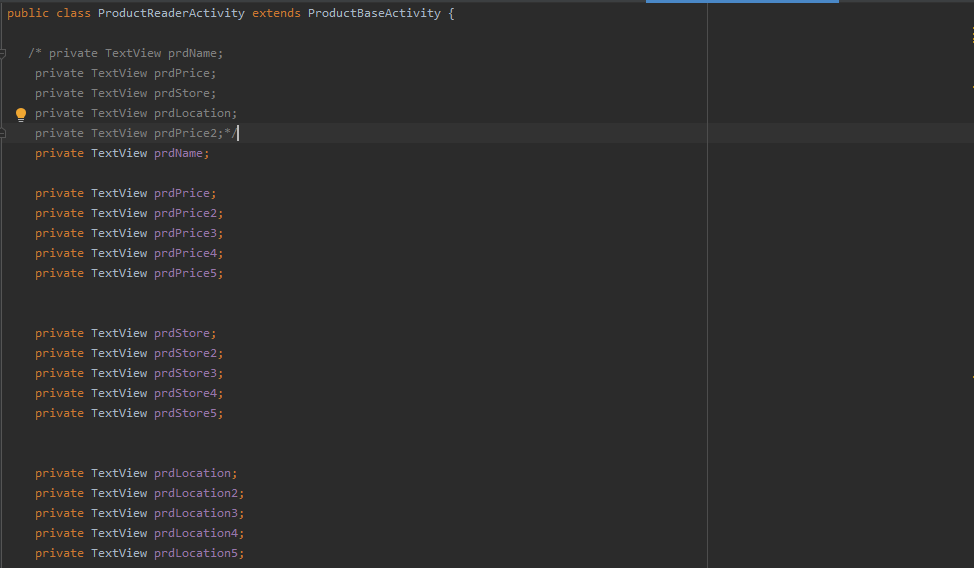
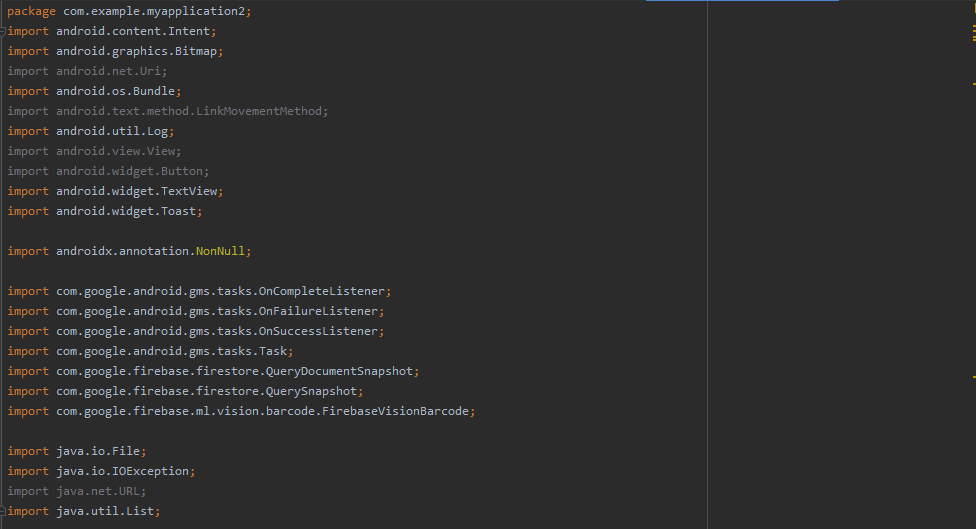
**ProductBase:**

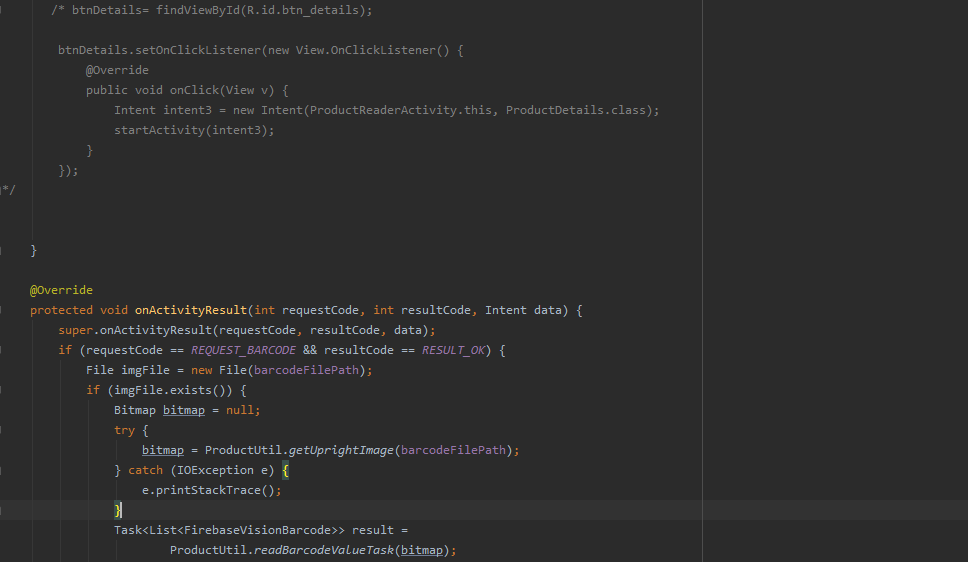


**ProductDetails:**

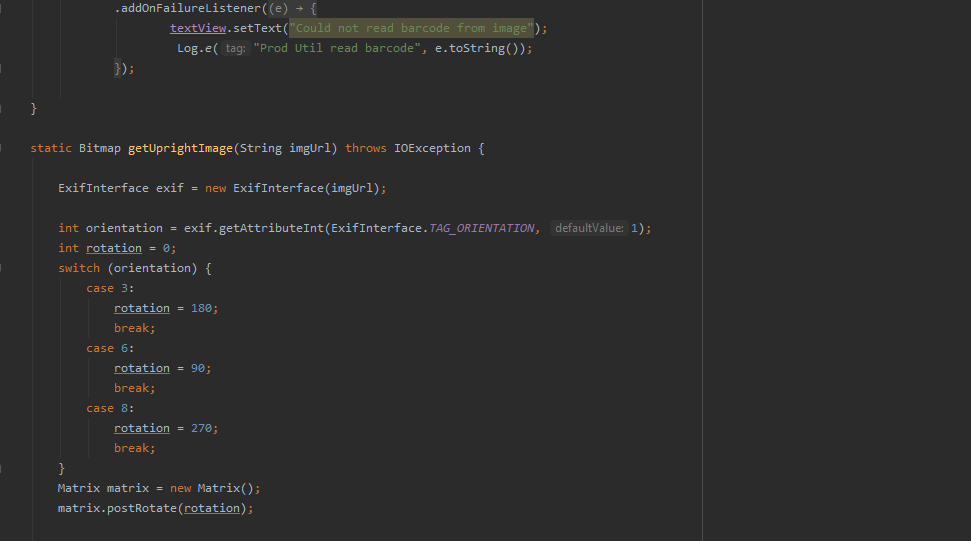
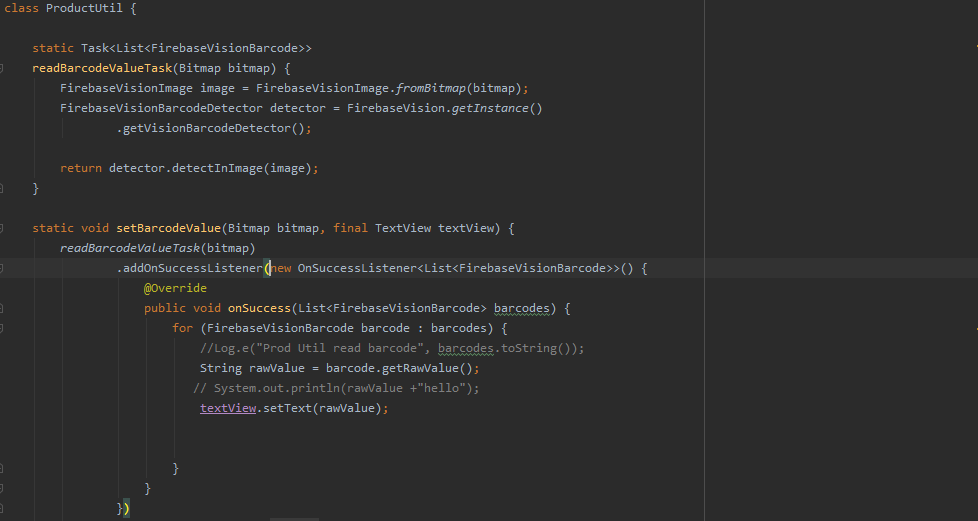
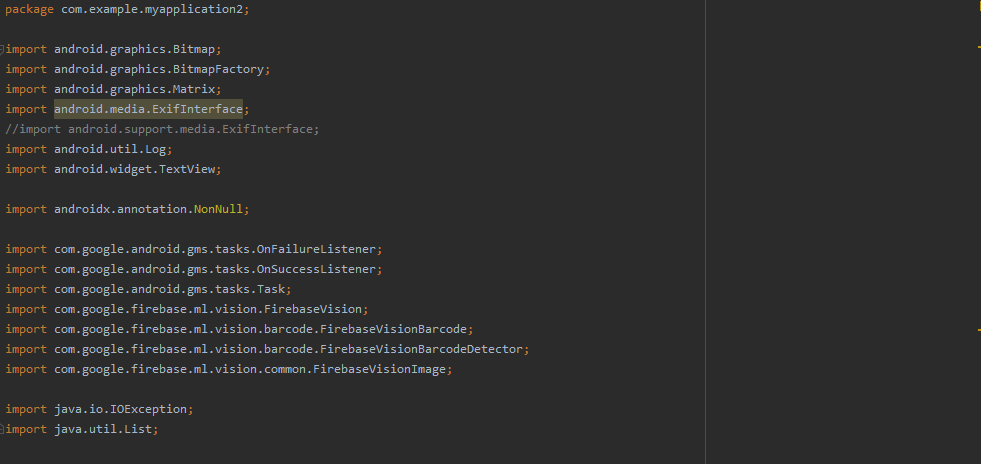


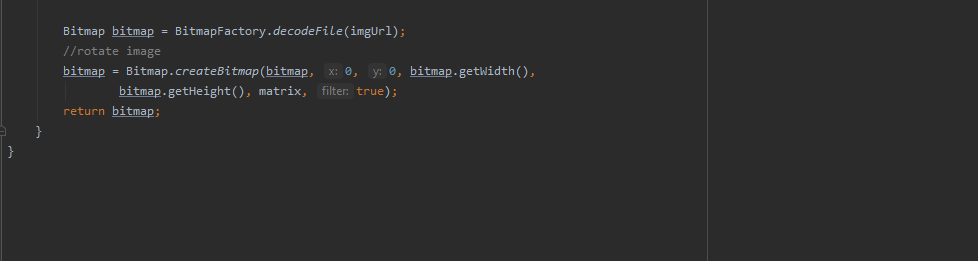
**ProductReaderActivity**:



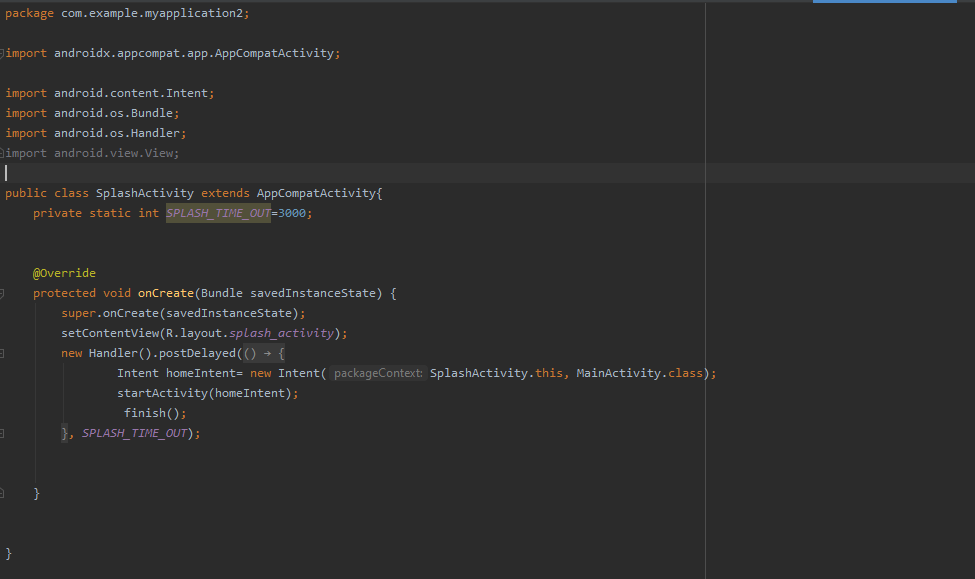


**ProductUtil:**

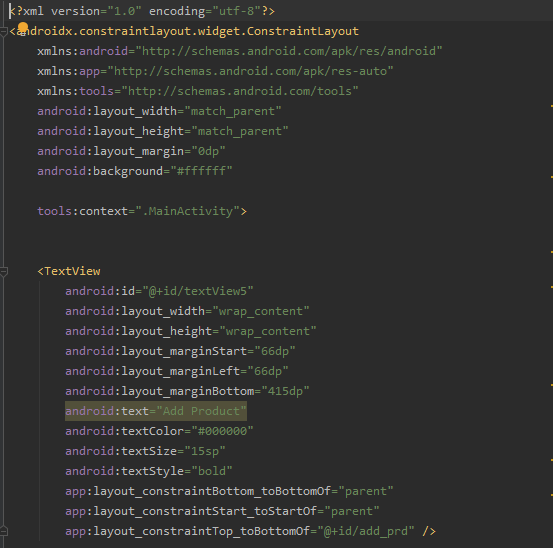


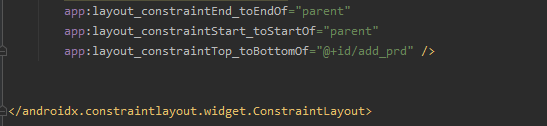
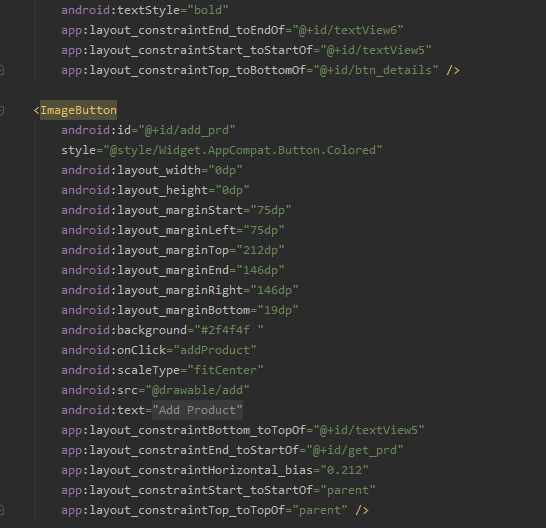


**SplashActivity:**



**activity\_main.xml:**





**product\_addition.xml:**

<?xml version="1.0" encoding="utf-8"?>  
  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fillViewport="true"  
 android:background="#ffffff"  
 >  
  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="0dp"  
  
 >  
  
  
 <Button  
 android:id="@+id/capture\_barcode"  
 android:background="#2f4f4f "  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:text="Scan Bar-code"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:onClick="captureBarcodePic"  
 style="@style/Widget.AppCompat.Button.Colored"/>  
 <ImageView  
 android:id="@+id/barcode\_img"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/capture\_barcode"/>  
  
 <TextView  
 android:id="@+id/barcode\_value\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Headline"  
 android:text="Bar-code Number"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/barcode\_img"/>  
 <TextView  
 android:id="@+id/barcode\_value"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/barcode\_value\_l"/>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_name\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/barcode\_value"  
 android:hint="Enter Product name">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_name"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
 </com.google.android.material.textfield.TextInputLayout>  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_price\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_name\_l"  
 android:hint="Enter Product price">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_price"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="numberDecimal"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_store\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price\_l"  
 android:hint="Enter Product store">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_store"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="text"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_location\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store\_l"  
 android:hint="Enter HTML link of store location">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_location"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textUri"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_price2\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location\_l"  
 android:hint="Enter Second Price">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_price2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="numberDecimal"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_store2\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price2\_l"  
 android:hint="Enter Second Product store">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_store2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="text"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_location2\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store2\_l"  
 android:hint="Enter HTML link of second store location">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_location2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textUri"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_price3\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location2\_l"  
 android:hint="Enter Third Price">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_price3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="numberDecimal"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_store3\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price3\_l"  
 android:hint="Enter Third Product store">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_store3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="text"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_location3\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store3\_l"  
 android:hint="Enter HTML link of third store location">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_location3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textUri"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_price4\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location3\_l"  
 android:hint="Enter Fourth Price">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_price4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="numberDecimal"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_store4\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price4\_l"  
 android:hint="Enter Fourth Product store">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_store4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="text"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_location4\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store4\_l"  
 android:hint="Enter HTML link of fourth store location">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_location4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textUri"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_price5\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location4\_l"  
 android:hint="Enter Fifth Price">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_price5"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="numberDecimal"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_store5\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price5\_l"  
 android:hint="Enter Fifth Product store">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_store5"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="text"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/product\_location5\_l"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store5\_l"  
 android:hint="Enter HTML link of fifth store location">  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/product\_location5"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:inputType="textUri"/>  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:id="@+id/save\_prd"  
 android:background="#2f4f4f "  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:onClick="saveProduct"  
 android:text="Save Product"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location5\_l" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>  
</ScrollView>

**product\_details.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#ffffff">  
  
 <Button  
 android:id="@+id/capture\_barcode"  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="48dp"  
 android:background="#2f4f4f "  
 android:onClick="captureBarcodePic"  
 android:text="Scan Bar-code"  
 app:layout\_constraintHorizontal\_bias="0.52"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/product\_info\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="232dp"  
 android:text="Nutritional Facts"  
 android:textAppearance="@style/TextAppearance.AppCompat.Headline"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:textColor="#000000"  
 android:textSize="25sp"  
 android:id="@+id/product\_info"  
 android:layout\_width="408dp"  
 android:layout\_height="393dp"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginBottom="33dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_name" />  
  
 <TextView  
 android:textColor="#000000"  
 android:textSize="35sp"  
 android:id="@+id/product\_name"  
 android:layout\_width="411dp"  
 android:layout\_height="62dp"  
 android:layout\_marginStart="5dp"  
 android:layout\_marginLeft="5dp"  
 android:layout\_marginTop="110dp"  
 android:layout\_marginEnd="5dp"  
 android:layout\_marginRight="5dp"  
 android:layout\_marginBottom="120dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/product\_info"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:gravity="center\_horizontal" />  
  
  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**product\_reader.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fillViewport="true"  
 android:background="#ffffff"  
 >  
  
<androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="0dp">  
 <Button  
 android:id="@+id/capture\_barcode"  
  
 style="@style/Widget.AppCompat.Button.Colored"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:background="#2f4f4f "  
 android:onClick="captureBarcodePic"  
 android:text="@string/take\_barcode\_picture"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 <TextView  
 android:id="@+id/barcode\_value\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_id\_from\_barcode"  
 android:textStyle="bold"  
 android:textAppearance="@style/TextAppearance.AppCompat.Headline"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/capture\_barcode"/>  
 <TextView  
 android:id="@+id/barcode\_value"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/barcode\_value\_l"/>  
  
 <!--- <TextView  
 android:id="@+id/product\_name\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_name"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_name"  
 app:layout\_constraintTop\_toBottomOf="@+id/barcode\_value"/> -->  
 <TextView  
 android:id="@+id/product\_name"  
 android:layout\_width="0dp"  
 android:layout\_height="46dp"  
 android:layout\_marginTop="0dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Large"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/barcode\_value"  
 android:gravity="center\_horizontal" />  
 <!--- app:layout\_constraintLeft\_toRightOf="@id/product\_name\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/barcode\_value"/> -->  
 <TextView  
 android:id="@+id/product\_price\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:text="@string/product\_price"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_price"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_name"/>  
 <TextView  
 android:id="@+id/product\_price"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_price\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_name"/>  
 <TextView  
 android:id="@+id/product\_store\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:text="@string/product\_store"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_store"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price"/>  
 <TextView  
 android:id="@+id/product\_store"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_store\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price"/>  
  
  
 <TextView  
 android:id="@+id/product\_location\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 android:text="@string/product\_location"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_store"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store"/>  
  
 <TextView  
 android:id="@+id/product\_location"  
 android:layout\_width="0dp"  
 android:layout\_height="130dp"  
 android:layout\_marginTop="140dp"  
 android:layout\_marginBottom="151dp"  
 android:autoLink="all"  
 android:clickable="true"  
 android:focusable="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_location\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <TextView  
 android:id="@+id/product\_price2\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:text="@string/product\_price2"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_price2"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location"/>  
 <TextView  
 android:id="@+id/product\_price2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_price2\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location"/>  
  
 <TextView  
 android:id="@+id/product\_store2\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_store2"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_price2"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price2"/>  
 <TextView  
 android:id="@+id/product\_store2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_price2\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price2"/>  
  
 <TextView  
 android:id="@+id/product\_location2\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_location2"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_store2"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store2"/>  
  
 <TextView  
 android:id="@+id/product\_location2"  
 android:layout\_width="0dp"  
 android:layout\_height="130dp"  
 android:layout\_marginTop="140dp"  
 android:layout\_marginBottom="151dp"  
 android:autoLink="all"  
 android:clickable="true"  
 android:focusable="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_location2\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store2"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <TextView  
 android:id="@+id/product\_price3\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:text="@string/product\_price3"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_price3"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location2"/>  
 <TextView  
 android:id="@+id/product\_price3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_price3\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location2"/>  
  
 <TextView  
 android:id="@+id/product\_store3\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_store3"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_price3"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price3"/>  
 <TextView  
 android:id="@+id/product\_store3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_price3\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price3"/>  
  
 <TextView  
 android:id="@+id/product\_location3\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_location3"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_store3"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store3"/>  
  
 <TextView  
 android:id="@+id/product\_location3"  
 android:layout\_width="0dp"  
 android:layout\_height="130dp"  
 android:layout\_marginTop="140dp"  
 android:layout\_marginBottom="151dp"  
 android:autoLink="all"  
 android:clickable="true"  
 android:focusable="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_location3\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store3"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <TextView  
 android:id="@+id/product\_price4\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:text="@string/product\_price4"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_price4"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location3"/>  
 <TextView  
 android:id="@+id/product\_price4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_price4\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location3"/>  
  
 <TextView  
 android:id="@+id/product\_store4\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_store4"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_price4"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price4"/>  
 <TextView  
 android:id="@+id/product\_store4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_price4\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price4"/>  
  
 <TextView  
 android:id="@+id/product\_location4\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_location4"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_store4"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store4"/>  
  
 <TextView  
 android:id="@+id/product\_location4"  
 android:layout\_width="0dp"  
 android:layout\_height="130dp"  
 android:layout\_marginTop="140dp"  
 android:layout\_marginBottom="151dp"  
 android:autoLink="all"  
 android:clickable="true"  
 android:focusable="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_location4\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store4"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
 <TextView  
 android:id="@+id/product\_price5\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:text="@string/product\_price5"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_price5"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location4"/>  
 <TextView  
 android:id="@+id/product\_price5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="100dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_price5\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_location4"/>  
  
 <TextView  
 android:id="@+id/product\_store5\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_store5"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_price5"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price5"/>  
 <TextView  
 android:id="@+id/product\_store5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_price5\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_price5"/>  
  
 <TextView  
 android:id="@+id/product\_location5\_l"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:text="@string/product\_location5"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 android:textStyle="bold"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toLeftOf="@id/product\_store5"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store5"/>  
  
 <TextView  
 android:id="@+id/product\_location5"  
 android:layout\_width="0dp"  
 android:layout\_height="130dp"  
 android:layout\_marginTop="140dp"  
 android:layout\_marginBottom="151dp"  
 android:autoLink="all"  
 android:clickable="true"  
 android:focusable="true"  
 android:textAppearance="@style/TextAppearance.AppCompat.Medium"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintLeft\_toRightOf="@id/product\_location5\_l"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/product\_store5"  
 app:layout\_constraintVertical\_bias="0.0" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>  
</ScrollView>

**splash\_activity.xml:**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#000000"  
 tools:context=".SplashActivity">  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="0dp"  
 android:src="@drawable/logo"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="92dp"  
 android:layout\_marginEnd="72dp"  
 android:layout\_marginRight="72dp"  
  
 android:text="PRICEZILLA"  
 android:textColor="@android:color/white"  
 android:textSize="50sp"  
 android:textStyle="bold|italic"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ProgressBar  
 android:id="@+id/progressBar3"  
 style="?android:attr/progressBarStyle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="16dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>