# MODULE 7-2 PROJECT THREE:

# INVENTORY APP (OPTION 1)

Arturo Santiago-Rivera

Raied Salma, Ph.D.

CS-360-X6386 Mobile Architect & Programming 21EW6

Southern New Hampshire University

August 15, 2021

Updated - March 2022

# Module 7-2 Project Three: Inventory App (Option 1)

This paper is the final step of the development process of the mobile application, **Inventory App**. The app's goal is to track items inventory through the primary use of mobile devices. For example, the track of the items through the app at a warehouse assists in managing and automating the warehouse logistics and accelerating the business's growth and expansion. The app allows the user to fulfill anywhere experience with real-time inventory visibility on any device. The app development is initially based on an installation for Android Devices.

## Log In
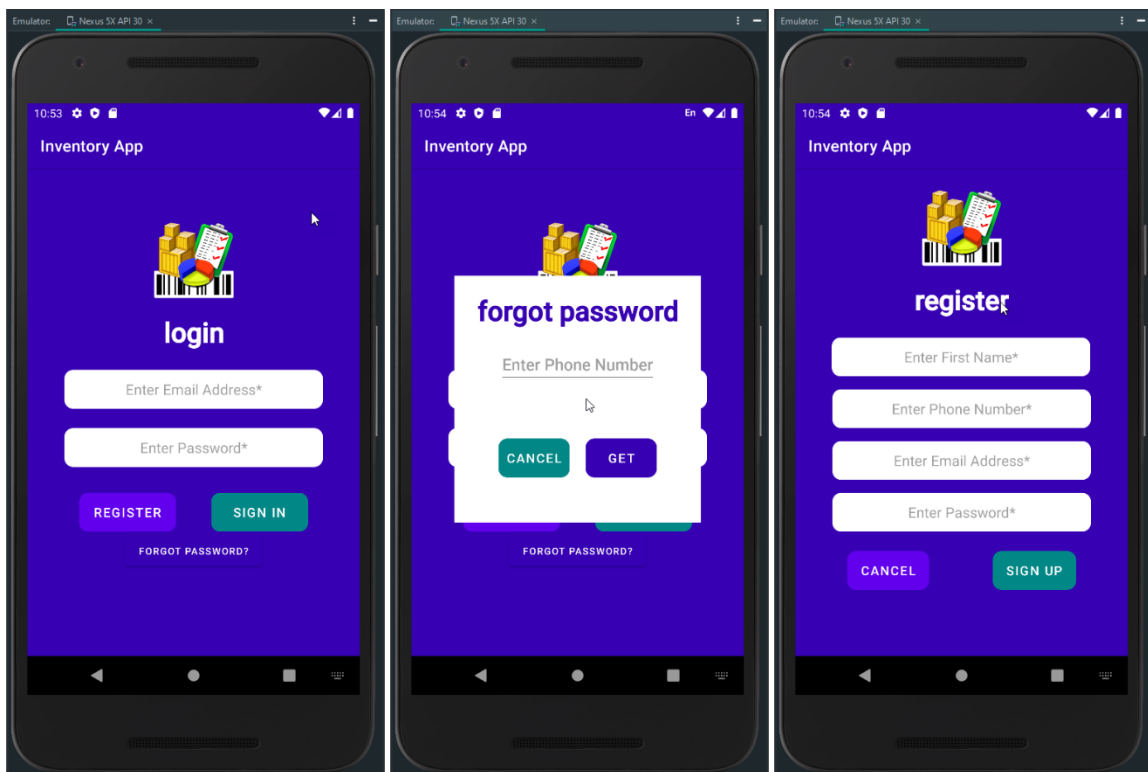


The Inventory App initiates with LoginActivity. The user enters their credentials (email and password) and presses the sign-in button. The app compares the credentials against the database, and if the user credentials are not in the database, the user gets an alert that the credentials are incorrect. The user clicks on the register button to enter the app's required user

information through the RegisterActivity. The user enters the required information and clicks the

sign-up button. The app verifies that all fields are valid, and if a field is not valid empty, the

RegisterActivity displays an error message for user correction. After the app successfully

validates the user information, it saves the data in the UsersDatabase (UsersDatabase.DB),

displays a successful message, and returns to the LoginActivity for the user sign in. When the

user signs in successfully, the app displays a successful message and initiates the ItemsActivity.

## Database

For security reason, the app work with two databases, UsersDatabase and ItemsDatabase.

The user's database comprises one table with four columns (id, name, phone, email, password).

The items database includes one table with fourth columns (id, email, item description, item

quantity). The app performs CRUD activities in the user's database using the UserSQLiteHelper

class and the items database using the ItemSQLiteHelper class and the ItemDBManager class.
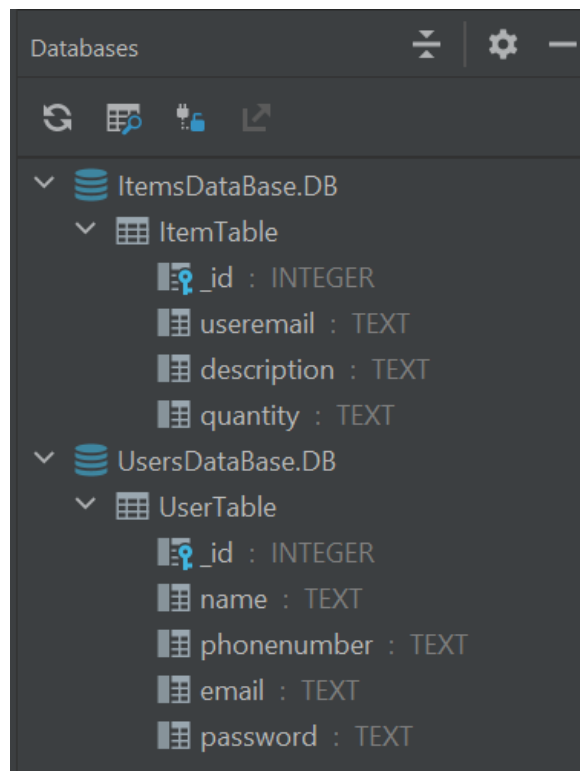
When the user successfully signs in, the ItemsActivity is initiated. If it is the first time for

the user, the ItemsActivity displays a message that the database is empty. The ItemsActivity is

the main workspace of the app. Here, the user can add items, enable/disable SMS notifications,

see the list of items in the database, increase or decrease the quantity of a selected item, delete a

selected item from the database, and sign out of the app to close the databases.
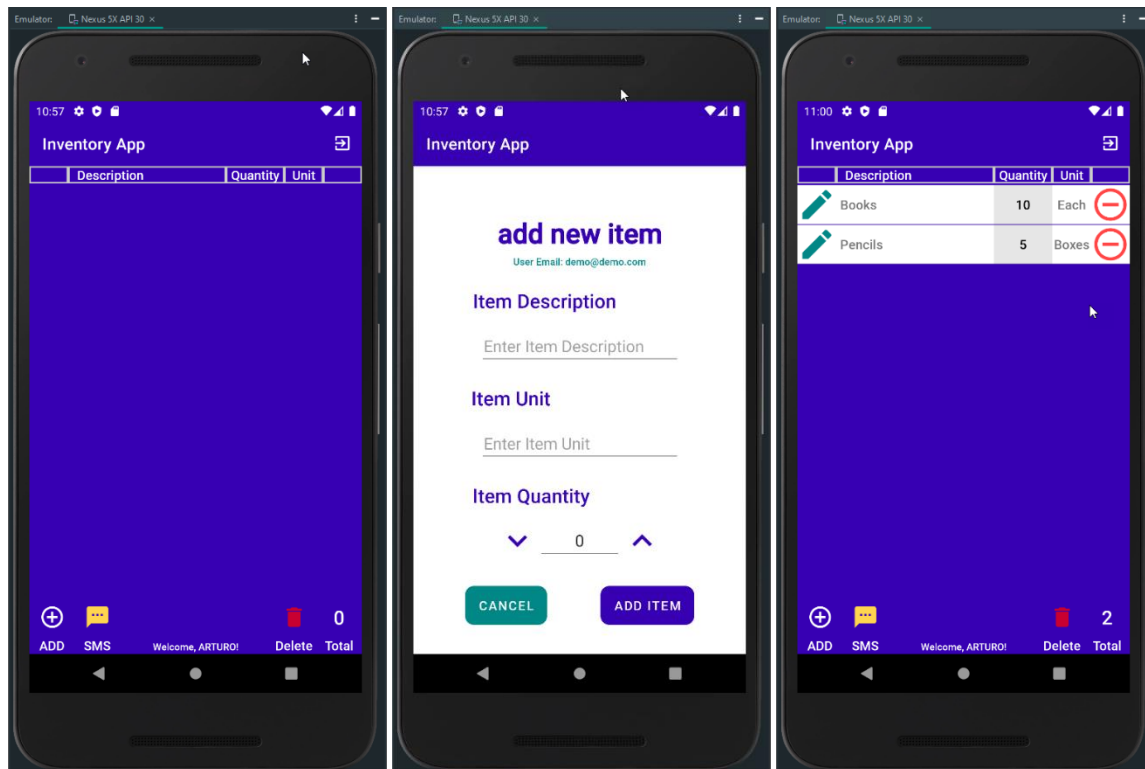


The database is persistent when the app is active, so any actions in the ItemsActivity

related to an item are recorded in the database dynamically.
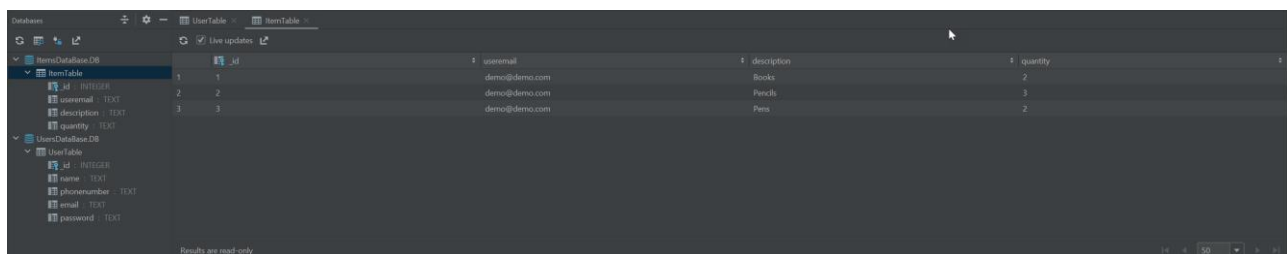


The user can select an item from the list by clicking the radio button to the left of the item

row. When the user clicks the button, it enables the increase and decrease buttons at the bottom

and the delete button of the item row. If no item is selected, those buttons remain inactive and display a message to select an item every time the user clicks for their use.
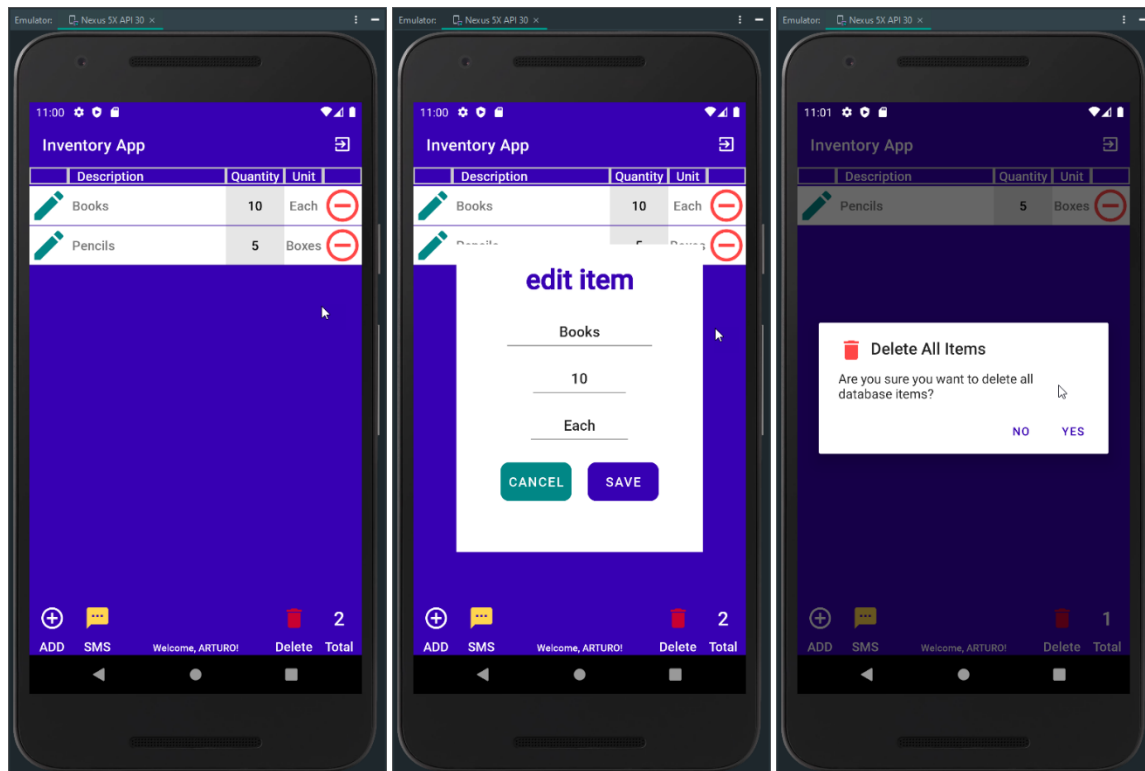


When the user selects an item and decreases their quantity to zero, the item quantity cell change to red, and the app sends an SMS message to the user's phone. The SMS message from the app only occurs if the user allows the app, in the device, to use the SMS API and the app SMS notification feature is enabled. If the app SMS notification feature is disabled and cannot use SMS API, the app displays a message advising that the app SMS is disabled. If the user denies the app to use the SMS API, it can be allowed in the app permissions.

## SMS Notification

During the use of the app, the user can enable or disable the app SMS notification feature by clicking on the SMS button (yellow button) in the ItemsActivity. The device permission

dialog display allows or denies the Inventory App to use the SMS API for the first time only. After that, the SMS button displays only the alert dialog to enable/disable the feature.
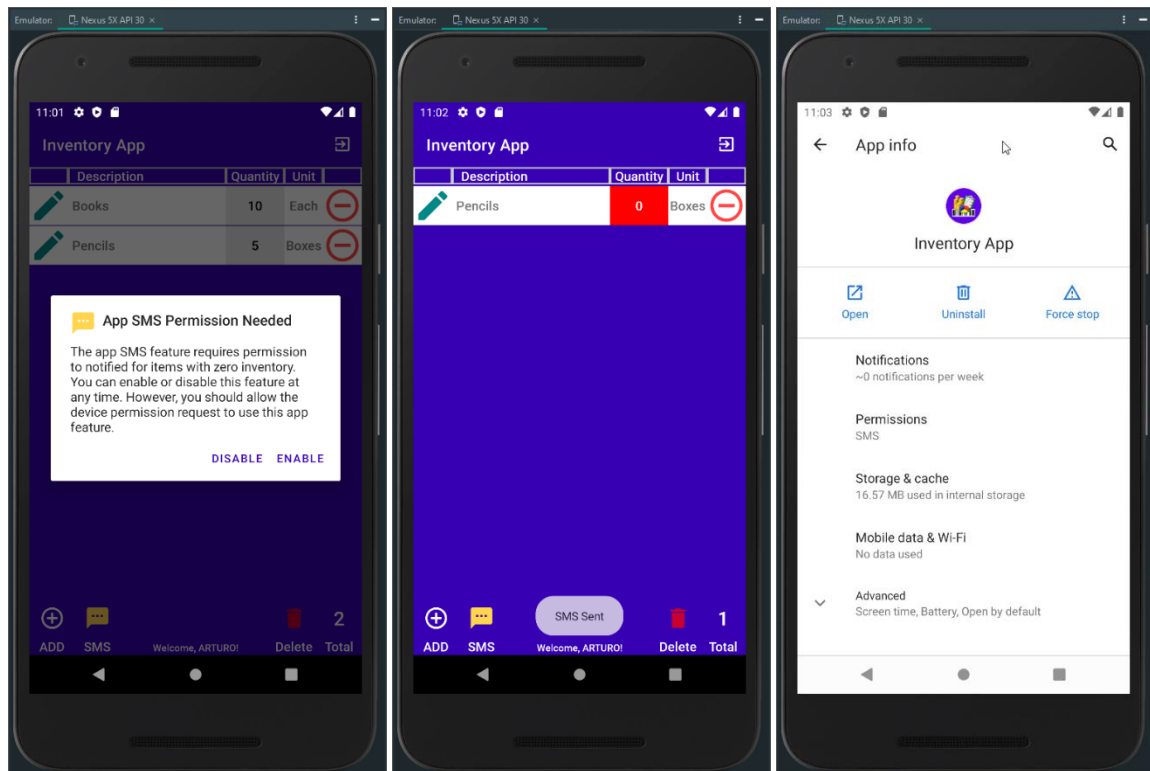


After the device permission dialog, an alert dialog of the app SMS notification feature is displayed. The user can enable or disable receiving SMS when an item quantity is zero in this dialog. The dialog is closed based on the user's selection and displays a message that the app feature is enabled or disabled. When an item has zero as quantity, and the app SMS notification is enabled and allowed in the device, a message is displayed that an SMS was sent. A message alert is displayed when the app SMS notification is disabled or not the app is not allowed on the device. If the user denies the app to use the device SMS API, the user needs to go to the app permissions to allow the app because the device permission is not displayed again.
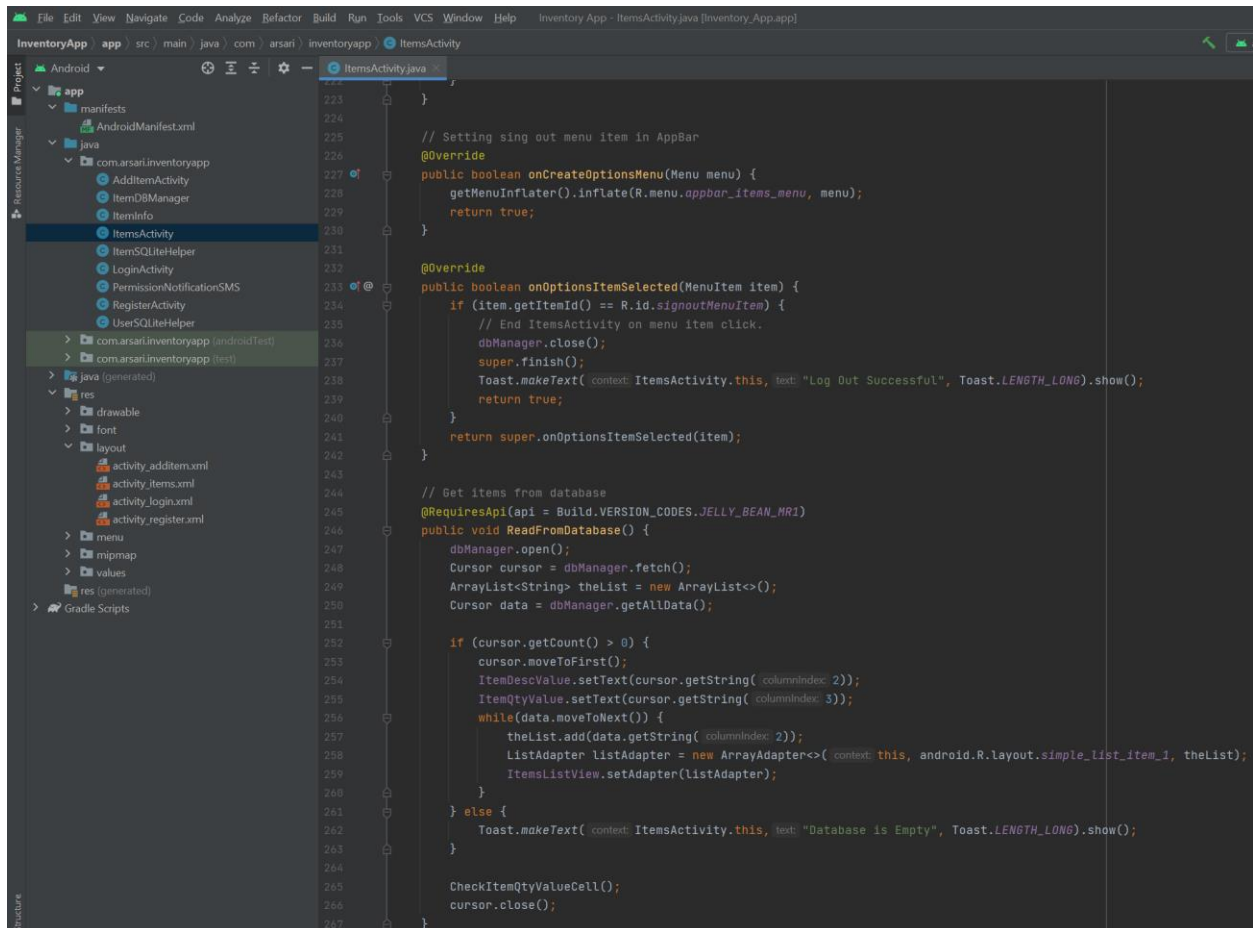
# Coding Best Practice

To the extent of our understanding and personal practices, the app employs industry-standard best practices such as in-line comments and appropriate naming conventions to enhance the app code.
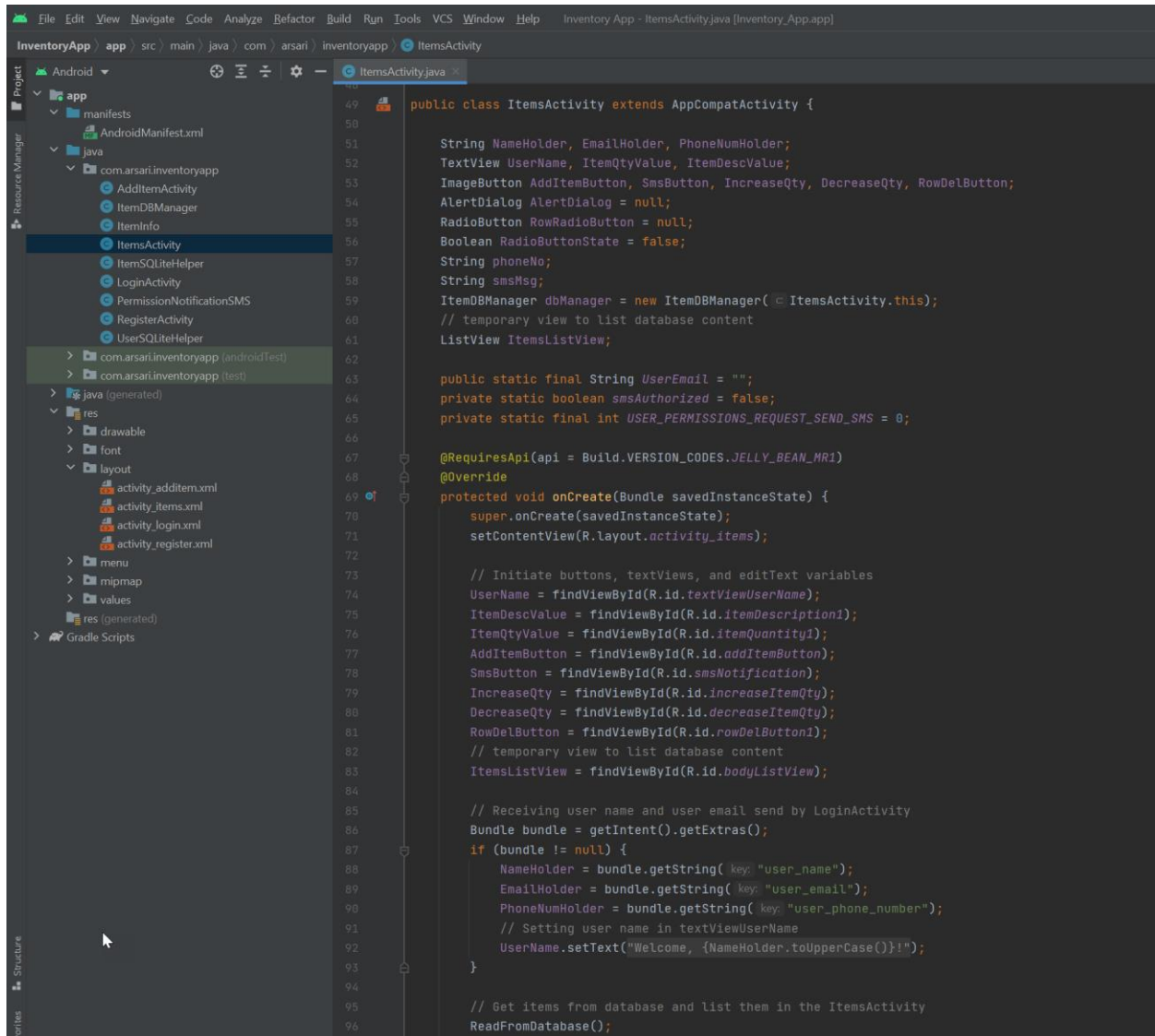


The app code identified global variables with camelCase CapWords names. Local and key variables are identified with camelCase lower case names or lower case dash_names. In-line comments are in different locations in the code to describe class methods or functions. JAVA class files are named with camelCase CapWords and layout files with lower case names. The
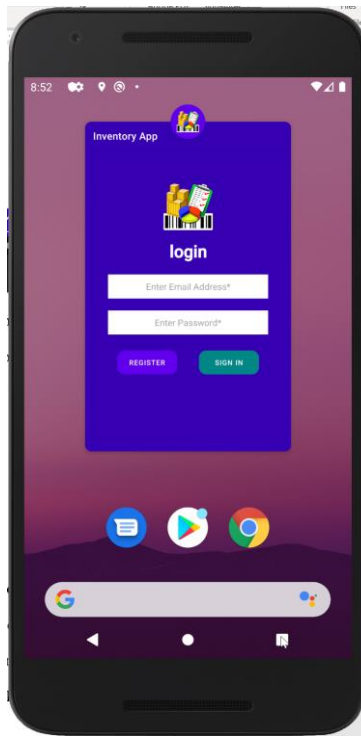
class methods naming, as possible, are a representation of the method's purpose and action.

Methods that override methods from its superclass are named camelCase with initial lower case

word. Other methods that do not override methods from its superclass are named in camelCase

with CapWords.

# Launch



The launch of the app developed requires considering some tasks and guides listed by Google Play to ensure the app launched successfully. As part of these tasks, the app's quality encompasses the entire user experience, from the app's description to the in-app experience. It is critical to meet the core app quality standards to reduce any friction in later parts of the publishing process.

## Target Audience

Potential users for the application include start-ups and small businesses needing a simple and intuitive way to track their items inventory. Other potential users could be entrepreneurs initiating a home garage business that needs to keep track of their inventory from their pocket and always accessible and professionals of any industry that want a simple solution to control and organize their office and home items.

## Description

The Inventory App description includes:

*"The Inventory App is the users' helpfulness to track items inventory through Android mobile devices. The track of the items list allows the user to better manage their quantified items in one location and on the go to fulfill anywhere experience with real-time inventory visibility on their Android device."*

## Icon



The proposed graphical icon is colorful and avoids the use of words. The icon contains different elements needed to keep an item's inventory optimized. Those elements included the items to be inventory a clipboard to list the items and quantity, from which we can do reports and metrics.

Combining these elements in a visual component gives life to the Inventory App with a clear definition of the purpose and usability of the app. Some of the visual elements in the icon are features in the app, and others will come available based on the app audience and growth.

## Device API Version

The app has been developed based on the API 16, which claims to be used in 99.98% of Android devices. This ensures that the app will run successfully on old devices and new devices. Some features are specific to most current versions, but the app should operate without difficulties in any device based on an executed test.

## Device Permissions

Part of the app features is the notification and alert to the user when an item in the inventory has rich zero quantity. The app is developed to request the user authorization to use the device SMS API on their Android device. The user can enable or disable the use of SMS at any moment during the use of the app by pressing the SMS button. No other special permissions are required from the user to operate in normal conditions.

## Monetization

Since the app includes basic inventory actions, it is suggested to publish it for free download to make it discoverable in the play store for a determined time. As the users connect with the app and make it one of their favorites, we can continue improving its quality by adding their primary clientele's features and actions in need.

With the app growth and more features are added, it can be leveled up to an in-app purchase where the free version database can be limited to a certain number of items. If the user needs to have more items listed, the user can pay a determined fee to remove the limitation of things in the database.

In summary, a final check to accomplish for a positive and quality launch of the app included:

- The developer profile has all the correct info linked to the valid payments merchant account.

- Uploaded the correct APK version.

- The Play store page is optimized.

- Set the correct pricing options.

- Set the country targeting and price accordingly.

- List the suitable "compatible devices."

- Add the correct support email address and website are listed.

- The app follows and complies with the Play Store rules.

- Acknowledged that the app meets the guidelines for Android content on the Play Store.

# References

Google. (2021, August). *Design - Material design*. Retrieved from Material Design:

      https://material.io/

Google. (2021, August). *Documentation for App Developers*. Retrieved from Android

      Developers: https://developer.android.com/docs

Google. (2021, August). *Launch*. Retrieved from Android Developers:

      https://developer.android.com/distribute/best-practices/launch/

Southern New Hampshire University. (2021, August 15). *Module Seven Project three Guidelines*

      *and Rubric*. Retrieved from Module 7-2 Submit Project Three:

      https://learn.snhu.edu/d2l/common/dialogs/quickLink/quickLink.d2l?ou=795543&type=c

      ontent&rcode=snhu-1020199