

UPC BARCODE APPS FOR DRUG REGISTERED VERIFICATION

Fathin Noorshazlina Mohamad Roslan, Mohd Afzan Othman*,
Mohd Azhar Abdul Razak

Department of Electronic and Computer Engineering, Faculty of
Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM
Johor Bahru, Johor, Malaysia

Article history

Received
30 October 2015
Received in revised form
15 February 2016
Accepted
28 March 2016

*Corresponding author
afzan@utm.my

Graphical abstract



Abstract

In pharmacology, drug is used to prevent and cure diseases and to improve the physical or mental well-being. All medicinal products must be registered with Drug Control Authority (DCA) of Malaysia before being marketed. However, there are still numerous unregistered products being sold in night markets and grocery stalls that contained unknown materials that could harm our bodies. As nowadays most people own smartphones, thus it will be great if the smartphone is being utilized to become a mobile apps that can prevent consumers from buying unregistered products. This study described the process of creating a barcode reader application for drug registered verification. The application is created using Android Studio software, Java programming language and source code from GitHub, which is then synchronized with the database of some medicinal products. The barcode scanner is then tested to read the UPC barcode on the products and the result of registration confirmation will be shown on the screen of the smartphone. Thus, it will aware the user from buying the unregistered products.

Keywords: UPC barcode, apps, product registration, product verification, android, smartphone, counterfeit

Abstrak

Dalam farmakologi, dadah digunakan untuk mencegah, merawat dan mendiagnosis penyakit serta meningkatkan kesejahteraan fizikal atau mental. Sebelum dipasarkan di Malaysia, semua produk perubatan perlu didaftarkan dengan Pihak Berkuasa Kawalan Dadah (PBKD) Malaysia. Walau bagaimanapun, masih terdapat produk tidak berdaftar yang dijual di pasar malam, gerai runcit dan gerai-gerai tepi jalan yang mengandungi bahan-bahan terlarang yang membahayakan tubuh badan kita. Pada masa kini kebanyakan orang memiliki telefon pintar, jadi ianya sangat berguna jika telefon pintar tersebut digunakan sebagai suatu aplikasi bagi mengelakkan pengguna daripada membeli produk yang tidak berdaftar. Projek ini menghuraikan proses mewujudkan aplikasi pengimbas kod bar untuk pengesahan dadah berdaftar. Aplikasi ini dicipta menggunakan perisian Android Studio yang beroperasi dengan bahasa pengaturcaraan Java. Android Studio adalah IDE (Integrated Development Environment) rasmi untuk pembangunan aplikasi Android. Aplikasi ini dicipta menggunakan kod sumber daripada GitHub untuk pengimbas kod bar yang kemudiannya diselaraskan dengan pangkalan data beberapa produk perubatan. Pengimbas kod bar kemudiannya diuji untuk membaca kod bar UPC bagi produk perubatan. Hasilnya dipaparkan pada skrin telefon pintar sama ada produk tersebut berdaftar dengan Pihak Berkuasa Kawalan Dadah (PBKD) atau tidak.

Kata kunci: UPC kod bar, aplikasi, pendaftaran produk, pengesahan produk, android, telefon pintar

© 2016 Penerbit UTM Press. All rights reserved

1.0 INTRODUCTION

Drug is any chemical substance that can affect the structure or biological system of humans, animals and other organisms when introduced into them. While in pharmacology, which is the study of drug, drug is defined as a chemical substance that is used to prevent, cure, treat or diagnose diseases as well as to improve the physical or mental well-being.

All health-related products must first be registered with Drug Control Authority (DCA) of Malaysia before they can be marketed in Malaysia. The DCA will carry out some processes to evaluate the quality, safety and efficacy of the products except for traditional preparations and health supplements [1]. This is to assure those registered products are safe for the consumers. The DCA has provided two main features to those registered products. The consumers can easily recognize the registered products in the market by the registration number and the genuine hologram sticker. The registration number must be printed on the packaging of the product while the hologram sticker must be applied onto the outer of the packaging of the product on the product label [1].

Despite the registration number and the hologram sticker, all products not only the pharmaceutical products, but also other products will have barcodes on their packaging. There are many types of barcodes that are used by manufacturers, but the common barcode is the UPC barcode or 1D barcode which is also known as linear barcode. Another type of barcode used is QR code or 2D barcode which the usage of the barcode is growing nowadays.

Conventionally, the handheld lasers are used as barcode readers to decode the printed images on the packaging [2-4]. As the production of smartphones has been constantly growing, the price of the smartphones became cheaper and almost everyone can afford to buy a smartphone. Since then, the research of barcode software for mobile phones has reached its way to practice [5].

The specialties of the smartphones are the embedded camera devices inside them [6] and the ability to access the Internet in real-time [7]. Although there are varieties of 2D barcodes have been created and more suited for camera accession[8], but still most of the packaging of the products are printed with UPC barcodes or 1D barcodes [7] which can be read by any auto-focus camera phone instantly [5].

On top of that, as people nowadays can get the information straight away from their smartphones and they are aware of those unregistered products being sold in the market. They will get the information first before going buy the pharmaceutical products. Thus, this study is to ease them to verify the registration of those pharmaceutical products they wanted to buy. This study is to create the barcode reader that can be installed by the people into their smartphones to verify whether the pharmaceutical

products were registered with the Ministry of Health (MOH), Malaysia or not.

2.0 EXPERIMENTAL

2.1 Android Studio

Android Studio is the official IDE for the developers in developing Android application. This software is based on IntelliJ IDEA. IntelliJ IDEA is a Java IDE for computer software development. The developer of this Java IDE is JetBrains which is formerly known as IntelliJ. IntelliJ IDEA is accessible as Apache 2 Licensed community edition and also applicable in proprietary commercial edition. The proprietary commercial edition means that the software is produced to be sold but do not means that it is closed-source software. On top of the expected IntelliJ capabilities, Android Studio provides:

- i. Flexible Gradle-based build system.
- ii. Multiple and variants of android application package (apk) file generation can be built.
- iii. Common application features can be built using code templates.
- iv. Rich layout editor that includes support for drag and drop theme editing.
- v. Lint tool that can checks potential bugs in Android project files and detects the performance, usage, compatibility of the version and any other problems.
- vi. Google Cloud Platform built-in support which making the user integrate Google Cloud Messaging and App Engine easily.

2.2 Java Programming Language

Java is a programming language and a platform for computing which has been released for the first time in 1995 by Sun Microsystems. Java is very essential that if it is not installed, many applications and websites will not work. It is very important that it can be everywhere including laptops, data centers, game consoles, scientific supercomputers, cell phones and Internet.

In order to avoid any problems during creating the barcode reader application, Java needs to be upgraded to the latest version. Java can be installed and upgraded to the latest version for free at the official Java website. The latest version carries crucial enrichment in order to enhance the performance, stability and security of the Java application in the computer used to build this project. The upgraded version can assure the applications of the Java run effectively and safely.

2.3 Source Code from GitHub

Software is the main part of this project. Thus, the coding for barcode reader application is vital in this study. Some research has been done to find the source code for barcode reader application that

can be run in Android Studio. There is a Git repositories hosting website for software development and code sharing called GitHub.

Although GitHub is a hosting service for Git repositories, it also develops features of its own. It does not only provides a web-based graphical interface but also access control and some other features of collaboration including the management of the task, bugs tracking and also the feature requests for every project.

There are many source codes for barcode reader application for smartphone that have been provided by GitHub and shared by other developers. But the required source code for this study is differs from the source codes in GitHub. Therefore, some modifications need to be done to the source code taken from the GitHub to fulfil the objectives of this study.

2.4 Creating the Barcode Reader Application in Android Studio

In order to start creating an application or a project in Android Studio, the developers need to know the basic steps to develop the application. This section will describe all the steps.

2.4.1 Developer Workflow

Figure 1 shows the process of developing Android applications in Android Studio. The project began with the exploration of Android Studio software. The official website of the Android Studio software provides some basic steps for the beginners to develop their applications. There are four development phases which includes;

- (i) Environment Setup phase
- (ii) Project Setup and Development phase
- (iii) Building, Debugging and Testing phase
- (iv) Publishing phase.

During the Environment Setup phase, the development environment which is Android Studio is installed and set up. The developer can also create the Android Virtual Devices and connect the hardware devices such as smartphones, tablets and others that will be used for testing.

For the Project Setup and Development, the Android project and application files are set up and developed by the developer, which contained all of the source code, resource files and Android manifest files for the application.

Next, the Android project or application is built in debuggable .apk package(s) that can be installed and run on the emulator or Android-powered device. If the developer wants to use hardware devices other than Android-powered device, another driver that is compatible with the hardware device needs to be installed first before the application is being tested in that particular device. In this study, the developer is using ASUS T00i

smartphone which is well-known as ASUS Zenfone 4, thus, the developer must first install the driver for the ASUS T00i device that can be obtained from the ASUS official website. The Android Studio used the Gradle-based build system that is flexible, provides variants of customized build and the resolution of the dependency.

Next, by using the Android Debug Monitor and device log messages (logact) along with intelligent coding features of the Java, the developer can debug the application that has been built. Then, the application can be tested using the emulator or the hardware devices.

Lastly, the Publishing phase let the developer to configure and build the application to be released and distributed to the users. But this study is being done up to the third phase only and the final phase is for the future work.

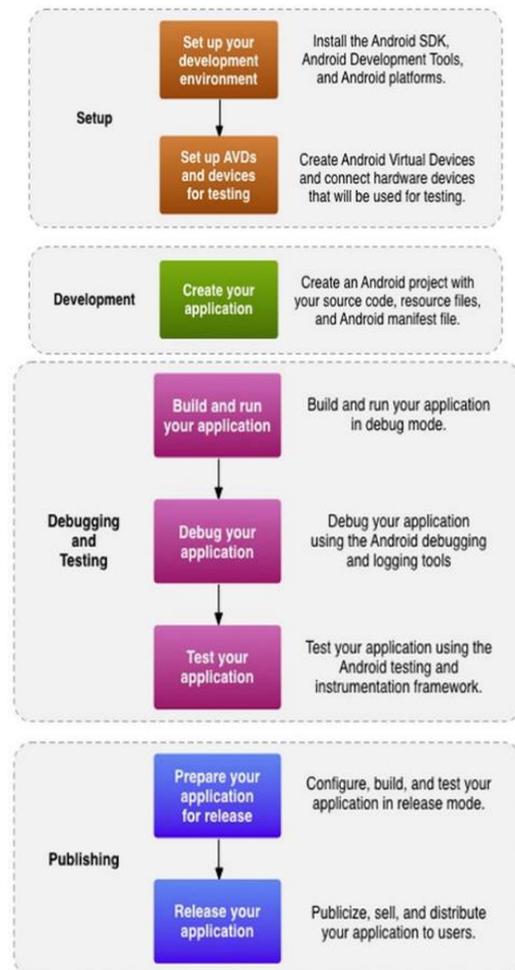


Figure 1 The development process for Android applications

2.4.2 Managing Projects from Android Studio

Android application can be created and managed using the graphical tools provided in Android Studio which contained everything that is needed for the

application including the app source code to build the configurations and test code.

The first step in creating an application is to create an Android application. After the Android Studio has been installed, open the software and the Welcome screen will be shown by the Android Studio. **New Project** is clicked to create a new Android application. After that, a window will be shown on the screen to let the developer to configure the application name, the package name and the project location. After entering all the values for the project then click **Next**.

On the next window, the developer can select the form factors that are supported by the application such as phone, tablet, TV, Wear and Google Glass. The selected form factors will become the application modules throughout the project. The API level for each form factor can also be selected. The developer can click the **Help me choose** where the window shows the cumulative distribution of the API level.

By clicking on the API level, a list of features corresponding to the Android version is introduced. This will help the developer to choose minimum API level for the application. In this study, the Phone and Tablet form factor is chosen while the minimum API level chosen is API 9 for the Android version 2.3 which is synonym as Gingerbread. Then, click **Next**.

The developer can add an activity on the next window. In this project, the blank activity is chosen so that it can be modified as what the developer wants. Then, click **Next** to proceed.

The next window is the activity configuration to be added to the application. In this project, the Activity Name, the Layout Name, the Title and the Menu Resource Name are left to be the same as what have been suggested and the button **Finish** is clicked.

The default structure of the project is created and the development environment is opened in Android Studio. Now, the application is ready to be developed.

2.4.3 Project and File Structure

The project files will be displayed by the Android Studio in *Android* view by default. This view is the flattened version structure of the project that gives a quick access to the key source files of the project and also helps the developer to work with the Gradle-based build system.

There is also another type of view which is *Project* view. This view could ease the developer to do modification of the files under certain type of modules. The important files that need to be modified are the files under the modules of *src/main* directory such as *MainActivity.java*, *activity_main.xml*, *menu_main.xml*, *AndroidManifest.xml* and others. The files that end with *.xml* are for the layout of the application.

Android Studio allows the developer to work with two layouts which are the *Design* view and *Text* view.

Figure 2 and Figure 3 show the *Design* view and the *Text* view respectively. The *Design* view is used to design the layout of the application by dragging and dropping the widgets according to developer desire manually while the *Text* view is used to design the layout manually.

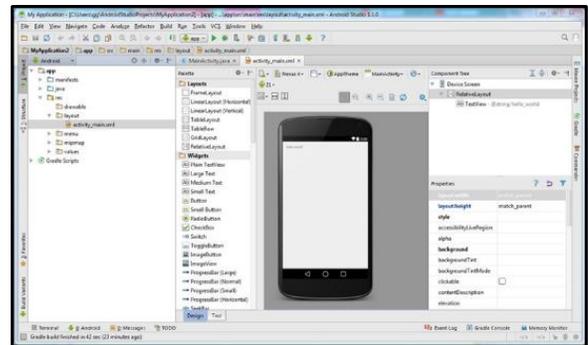


Figure 2 Design view

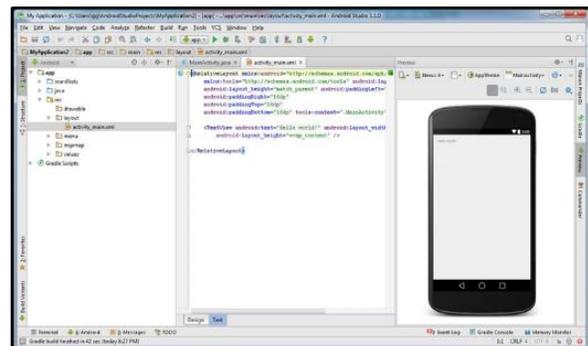


Figure 3 Text view

2.4.4 Source Code and Modification

The source code is taken from GitHub website, a popular repository hosting service in the world. The source code taken is specifically for barcode reader application but differs from the objectives of this study. Thus, some modification has been done so that it will meet the requirements to achieve the goals of this study.

Layout is the interface between the application and the user when the user first opens the application. Layout will displays all the required features on the screen right after the application is opened by the user. As this project is creating the barcode reader application, therefore, the button 'Scan' is very important in the layout file so that the user can press the button to scan the barcode on the product. This layout will also display the features after the barcode scanning activity is done such as the details of the product including the product name, the barcode number, the registration number and also the image of the product. The modification of the layout has been done on *activity_main.xml* file.

Android Manifest file is the file that shows the label of the application which is located above the layout of the application. This file can be modified to change the label name of the application, the image of the application shown in the menu screen of the smartphone after it has been installed and also the theme of the application.

Next, the main activity file is the most crucial part in modifying the source code of the barcode reader application. This file controls all of the activities in the application so that the user can see what is happening. In this study, the activities include calling the function provided in the application or in the smartphone such as the camera, the scanner and others. Besides that, it can controls what should be displayed on the screen after an activity has been done such as the details of the product after scanning the barcode. This is where the information of the medicinal products will be displayed.

2.4.5 Building the Project

The developer can build the project on Android Studio by clicking **Build** and selecting **Make Project**. The Gradle Console at the bottom right part of the window will show the tasks and the subtasks that have been running by the build system in Android Studio. More details can be found on the console if the build is failed.

2.4.6 Running and Debugging

After the application has been built and successful, the developer can run the application either in an emulator or a device. Running the application using an emulator faced some difficulties that forced the developer to run the application using a device or smart-phone to be precise. The difficulties are the Android Virtual Device (AVD) required large space but the computer used to build the application does not have enough space for that and also running the application using an emulator taking so long to be done. As an alternative, the running process is done using a real device.

Before running the application using a real device, the developer must first enable the USB debugging on the device. The device used in this study is ASUS T00i with Android version 4.4.2. For the Android version 4.0 up to the latest, the developer can found it in **Settings > Developer options** but unlike for the Android version 4.2 and newer where the Developer options is hidden by default. The developer needs to go to the **Settings > About phone** and tap the **Build number** for seven times to make it available.

Then, as the application is created on Windows, the appropriate USB driver for the device must first be installed for android debug bridge (adb) in order to detect the device. The USB driver can be installed from the official website of the model of the device. After the USB driver has been installed, make sure that the computer can detect the device after

connected using USB. Once everything has been set up and the computer can recognized the device, the application can be installed in the device by clicking **Run > Run** (or **Run > Debug**) on the menu bar in Android Studio.

3.0 RESULTS AND DISCUSSION

3.1 Testing the Application using Smartphone

The application should be installed in the smartphone and it can be seen in the menu screen of the smartphone as shown in Figure 4. The application has been named as 'Verify Your Product' in accordance to one of the objectives of this study which is to create an application in order to verify whether the product has been registered with MOH or not. This is where the Android Manifest file runs its function. The image for the application has also been changed to a suitable image that can make the user recognized the application as a barcode reader application as shown in Figure 4.



Figure 4 The menu screen of the smartphone with the application installed

Next, click the application on the menu screen of the smartphone to open the application. Figure 5 shows the application interface after being opened. The name of the application will be appeared on top of the screen, the welcoming messages and other details will be shown just below the name of the application and there is also a 'Scan' button for the user to start the scanning of the barcode of product. This interface has been modified in activity_main.xml file.

After the 'Scan' button is pressed, the screen of the smartphone will be appeared as a scanner to scan the barcode on the packaging of the product as shown in the Figure 6. The barcode should be placed properly so that the smartphone can detect and read the barcode of the product. This is where the function of the camera and the function of the

scanning are being called according to the MainActivity.java file.

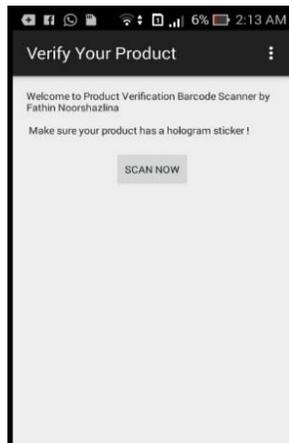


Figure 5 The interface of the barcode reader application



Figure 6 The screen of the smartphone appeared as a barcode scanner

3.1.1 Verification of Registered Products

If the scanned product has been registered, the information of the product will be shown on the screen of the smartphone as shown in Figure 7. The information of the product includes type of the barcode, barcode number of the product, registration verification: name of the product, registration number of the product, ingredients of the product, registration holder, manufacturer of the product and the telephone number.

The registration verification of the product will be shown as 'Registered' on the screen. Even though the product is said to be registered, the barcode of the product still can be counterfeited. In which, it may be the genuine product or it can also be a fake product.

Therefore, one of the ways to solve the counterfeiting problem is by including the image of the product as shown in Figure 8. The image of the product is viewed using the function of ImageView in Android Studio as in MainActivity.java file. The image of the product viewed in the application is the image of the genuine product. Thus, this will avoid the user from buying the fake product by comparing the colour, packaging and other details of the product

with the image provided.

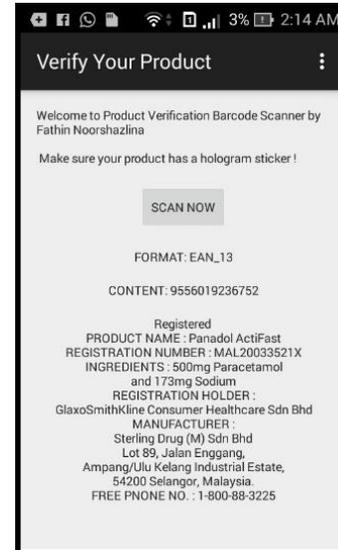


Figure 7 The information of the registered product will be displayed on the screen

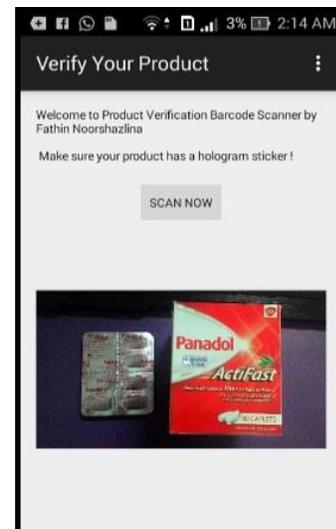


Figure 8 The image of the scanned product

The MainActivity.java file in Android Studio is to run all processes starting from the 'Scan' button being pressed until the image of the product appeared on the screen.

As this application is only a prototype, the information of the product is put into the coding manually and only two products were chosen to be the samples of this study which are Panadol and Panadol ActiFast.

3.2 Verification of Unregistered Products

How about the unregistered products? If the product is unregistered with MOH, the registration verification of the unregistered product will be displayed as 'NOT

Registered' on the screen as shown in Figure 9. The type of barcode and barcode number of the product will still be displayed on the screen but not the information of the product. There is also no image of the product. Thus, this will alert the user about the safety and quality of the product and for the sake of user's health, the user will definitely not buy the unregistered product.

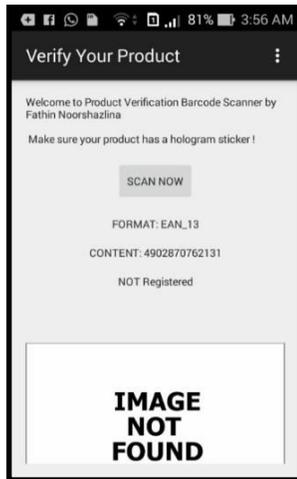


Figure 9 The verification for unregistered product

4.0 CONCLUSION

The result of the barcode reader application testing on the smartphone shows that the application 'Verify Your Product' can read the barcode on the packaging of the product successfully. It also can provide the type of the barcode on the product packaging and the barcode number precisely. Thus, the objective to create a barcode reader application for smartphone is achieved.

Besides that, the result from the application testing also shows the information of the registered product such as the product name, registration number of the product, ingredients and others. But

the most important is the verification of product registration whether the product has been registered with DCA or not. This is aligned with the second objective which is to create an application in order to verify the drug registration with MOH.

The result also shows the image of the registered product which this will help the user to distinguish between the genuine and the fake products. This is the most important part as it concludes why the whole study is done. In a nutshell, the objectives of this project are successfully achieved.

References

- [1] BPFK - About the Drug Control Authority (DCA). Official Portal National Pharmaceutical Control Bureau Ministry of Health Malaysia. Retrieved June 05, 2015 From: [Http://Portal.Bpfk.Gov.My/Images/Guidelines_Central/Guidelines_on_Regulatory/ANNEX16ASEANLBLREQ.Pdf](http://Portal.Bpfk.Gov.My/Images/Guidelines_Central/Guidelines_on_Regulatory/ANNEX16ASEANLBLREQ.Pdf).
- [2] Budde, A., and Michahelles, F. 2010. Product Empire-Serious Play with Barcodes. *Proceeding of Internet of Things (IOT)*. 1-7.
- [3] Gallo, O., and Manduchi, R. 2009. Reading Challenging Barcodes with Cameras. *Workshop on Applications of Computer Vision (WACV)*. 1-6.
- [4] Tian Hu and Tao Yang. 2013. Application of Computational Verb Image Processing to 1-D Barcode Localization. *IEEE International Conference on Anti-Counterfeiting, Security and Identification (ASID)*. 1-4.
- [5] Adelman, R., M. Langheinrich, and C. Floerkemeier. 2006. A Toolkit for Bar-Code Recognition and Resolving on Camera Phones – Jump Starting the Internet of Things. *Proceedings of the Workshop on Mobile and Embedded Interactive Systems (MEIS'06) at Informatik*. GI Lecture Notes in Informatics Series (LNI).
- [6] Subpratatsavee, P., Janthong, N., Kuha, P., and Chintho, C. 2014. HC2D Barcode Reader using Embedded Camera in Android Phone. *11th IEEE International Joint Conference on Computer Science and Software Engineering (JCSSSE)*. 254-257.
- [7] Von Reischach, F., Karpischek, S., Michahelles, F., and Adelman, R. 2010. Evaluation of 1D Barcode Scanning on Mobile Phones. *Proceeding of Internet of Things (IOT): IEEE Computer Society*. 1-5.