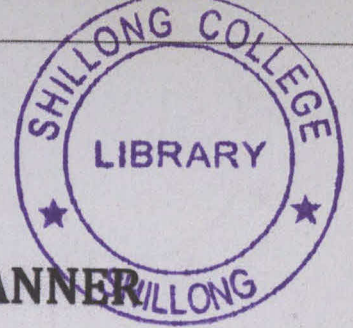


**PROJECT ON
ANDROID BAR AND QR CODE SCANNER**



SHILLONG COLLEGE



(DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS)

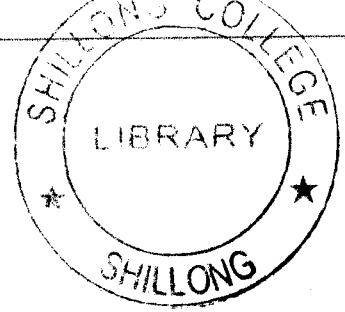
SHILLONG COLLEGE-03

**SUBMITTED BY:
CHERYLL LINDA TALLANG
BCA FINAL YEAR
(P1500065)**

NORTH EASTER HILL UNIVERSITY



ACKNOWLEDGEMENT



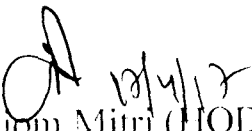
I would like to express my special thanks of gratitude to my teacher (Sir, Bantei Mukhim) as well as our principal (Dr. K.D Ramsiej)who gave me the golden opportunity to do this wonderful project on the topic (Android QR and Bar Code Scanner App), which also helped me in doing a lot of Research and i came to know about so many new things I am really thankful to them.

Our special thanks to Mrs. Aiom, HOD, Dept of Computer Science and Applications, for being our guiding light throughout the course of our project. The project would not have been success without the help of our respected HOD.

Finally i would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

CERTIFICATE

This is to certify that Cheryll Linda Tallang of BCA 3rd year has successfully completed the project work on Android QR and BAR Code Scanner app for BCA 3rd year final examination 2017 of North Eastern Hill University in the year 2017. It is Further Certified that the project is individual work of the candidate.


Mrs. Atom Mitri (HOD)
(Shillong College)

Dept of Computer Science and application

Guide 
(Bantei Mukhim)

 17/04/17
Examiner

INTRODUCTION

Barcode Scanner is a popular Android app that could revolutionize the way you shop, and change the way you download apps.

Barcode Scanner allows you to scan barcodes on products using your phone's camera. It's very easy to use - you just hold the lens over the product so that the barcode fits within the rectangle on screen. You'll see some yellow dots move across the red line while it's scanning. After just a few seconds, Barcode Scanner should find a match and give you the option to look up the product on Google, where you can view more information about the product and read reviews.



PURPOSE OF USING QR AND BARCODE SCANNER

Barcodes are extremely versatile. They can be **used** for any kind of necessary data collection. This could include pricing or inventory information. ... Because **barcodes** make it possible to track inventory so precisely, inventory levels can be reduced.

Requirements:

We need to have the following installed and configured on our development machine:

- Android SDK and platform tools
- ANDROID STUDIO 2.2.3
- An Emulator or Android device running Android 2.3 or higher

CONTENTS

1. INTRODUCTION

2. PURPOSE OF USING THIS APP

3. REQUIREMENTS

4. TEMPLATE

{A}. CREATING A NEW ANDROID PROJECT

{B}. CREATING THE LAYOUT OF THE XML

{C}. IMPLEMENTING THE ACTIVITY CLASSES

{D}. ANDROID MANIFEST

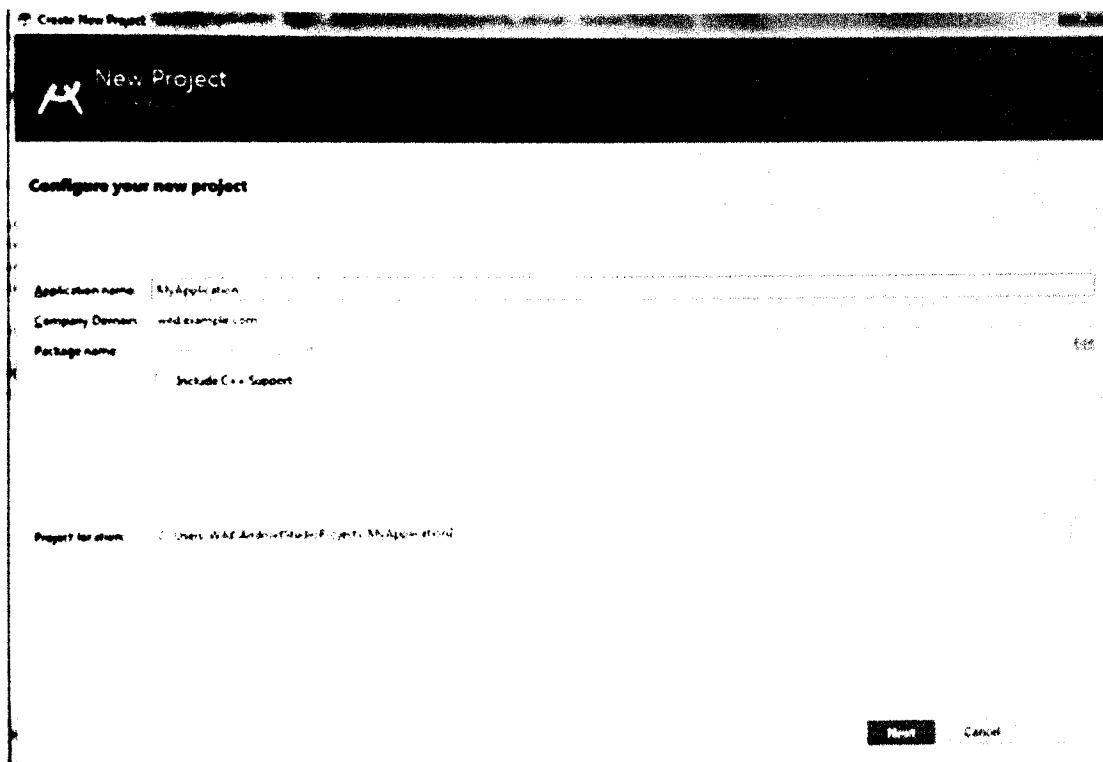
{E}. BUILD COMPILE AND RUN

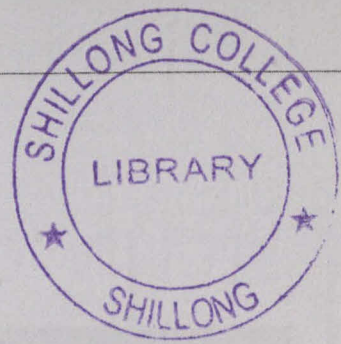
5. Conclusion

Template:

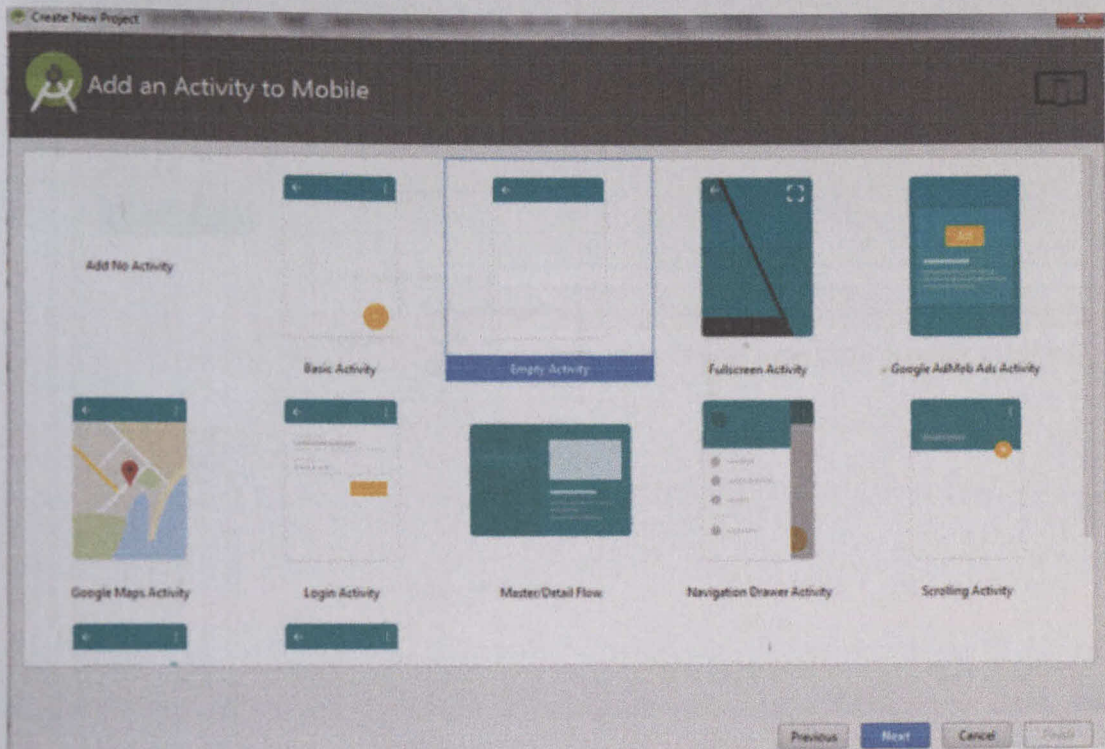
1 . Create a new Android project

In Android Studio, go to File > New > Project and in the New Project dialog, expand Android folder to select Android Project. And here I m naming it as AndroidQRCodeScannerExample.





After that Choose Empty Activity



Then Next Step is to Choose

Create New Project

Customize the Activity

←

Empty Activity

Creates a new empty activity

Activity Name:

☒ Generate Layout File

Layout Name:

☒ Backwards Compatibility (AppCompat)

The name of the activity class to create

Previous

Next

Cancel

Finish

Creating the layout of the main AndroidBarcodeQrExample

We are going to make a simple layout xml for the `AndroidBarcodeQrExample.class`, that consists of a `AndroidBarcodeQrExample` with vertical orientation, that includes two Buttons that implement the `onClick` listener in order to open the corresponding Activity.

Open `res/layout/activity_android_bar_code_qr_example.xml`, and type the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/c4f5_ee16d"
    android:orientation="vertical"

    tools:context="com.example.wild.androidbarcodeqrexample.AndroidBarcodeQrExample">

    <TextView

        android:id="@+id/textView1"
    android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center|top"
        android:layout_margin="20dp"
        android:text="SCAN HERE"
        android:textColor="@color/colorAccent"
        android:textSize="30dp" >

        TextView

    <Button

        android:id="@+id/scanner"
        android:layout_width="250dp"
        android:layout_height="80dp"
        android:layout_gravity="center"
        android:layout_margin="10dp"
        android:gravity="center"
        android:onClick="scanQR"
```



```
    android:textColor="@color/colorAccent"
    android:background="@color/colorPrimary"
    android:text="QR CODE"
    android:textSize="18dp" >
```

</Button>

<Button

```
    android:id="@+id/scanner2"
    android:layout_width="250dp"
    android:layout_height="80dp"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:gravity="center"
    android:onClick="scanBar"
    android:background="@color/colorPrimary"
    android:text="BAR CODE"
    android:textColor="@color/colorAccent"
    android:textSize="18dp" >
```

</Button>

LinearLayout

Creating the source code of the main AndroidBarcodeQrExample Activity

Open `src/main/java/codegeeks/androidqrcoodeexample/AndroidBarcodeQrExample.java` file and type the following code:

```
package com.example.wild.androidbarcodeqrexample;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.ActivityNotFoundException;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class AndroidBarCodeQrExample extends Activity {

    static final String ACTION_SCAN =
        "com.google.zxing.client.android.SCAN";

    // Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Set the main content layout of the Activity
        setContentView(R.layout.activity_android_bar_code_qr_example);

        // Request barcode mode
        public void scanBar(View v) {
            try {
                // Start the scanning activity from the
                // com.google.zxing.client.android.SCAN intent
                Intent intent = new Intent(ACTION_SCAN);
                intent.putExtra("SCAN_MODE", "PRODUCT_MODE");
                startActivityForResult(intent, 0);
            } catch (ActivityNotFoundException anfe) {
                // If not found, show the download dialog
            }
        }
    }
}
```

```

showDialog(AndroidBarCodeQrExample.this, "No Scanner Found",
"Download a scanner code activity?", "Yes", "No").show();
    }
}

```

//product qr code mode

```

public void scanQR(View v) {
    try {
        //start the scanning activity from the
        com.google.zxing.client.android.SCAN intent
        Intent intent = new Intent(ACTION_SCAN);
        intent.putExtra("SCAN_MODE", "QR_CODE_MODE");
        startActivityForResult(intent, 0);
    } catch (ActivityNotFoundException anfe) {
        //on catch, show the download dialog
        showDialog(AndroidBarCodeQrExample.this, "No Scanner Found",
        "Download a scanner code activity?", "Yes", "No").show();
    }
}

```

Alert dialog for downloadDialog

```

private static AlertDialog showDialog(final Activity act,
CharSequence title, CharSequence message, CharSequence
buttonYes, CharSequence buttonNo) {
    AlertDialog.Builder downloadDialog = new
    AlertDialog.Builder(act);
    downloadDialog.setTitle(title);
    downloadDialog.setMessage(message);
    downloadDialog.setPositiveButton(buttonYes, new
    DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialogInterface, int i) {
        Uri uri = Uri.parse("market://search?q=pname:" +
"com.google.zxing.client.android");
        Intent intent = new Intent(Intent.ACTION_VIEW,
uri);
        try {
            act.startActivity(intent);
        } catch (ActivityNotFoundException anfe) {
            //on catch, show the download dialog
        }
    }
    downloadDialog.setNegativeButton(buttonNo, new
    DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialogInterface, int i) {
        //on click, show the download dialog
    }
}
}
}

```



```
return downloadDialog.show();  
}
```

onActivityResult method

```
public void onActivityResult(int requestCode, int resultCode,  
    Intent intent) {  
    if (requestCode == 0) {  
        if (resultCode == RESULT_OK) {  
            // get the extras that are returned from the intent  
            String contents = intent.getStringExtra("SCAN_RESULT");  
            String format =  
                intent.getStringExtra("SCAN_RESULT_FORMAT");  
            Toast toast = Toast.makeText(this, "Content:" +  
                contents + " Format:" + format, Toast.LENGTH_LONG);  
            toast.show();  
        }  
    }  
}
```

Android Manifest

The AndroidManifest.xml of our project is simple and contains the permissions:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.wild.androidbarcodeqrexample">

    android:versionCode="1"
    android:versionName="1.0" >

    uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="19" />

    application
        android:allowBackup="true"

        android:label="@string/app_name" >
<activity
    android:name="com.example.wild.androidbarcodeqrexample.AndroidBarCodeQ
rExample"

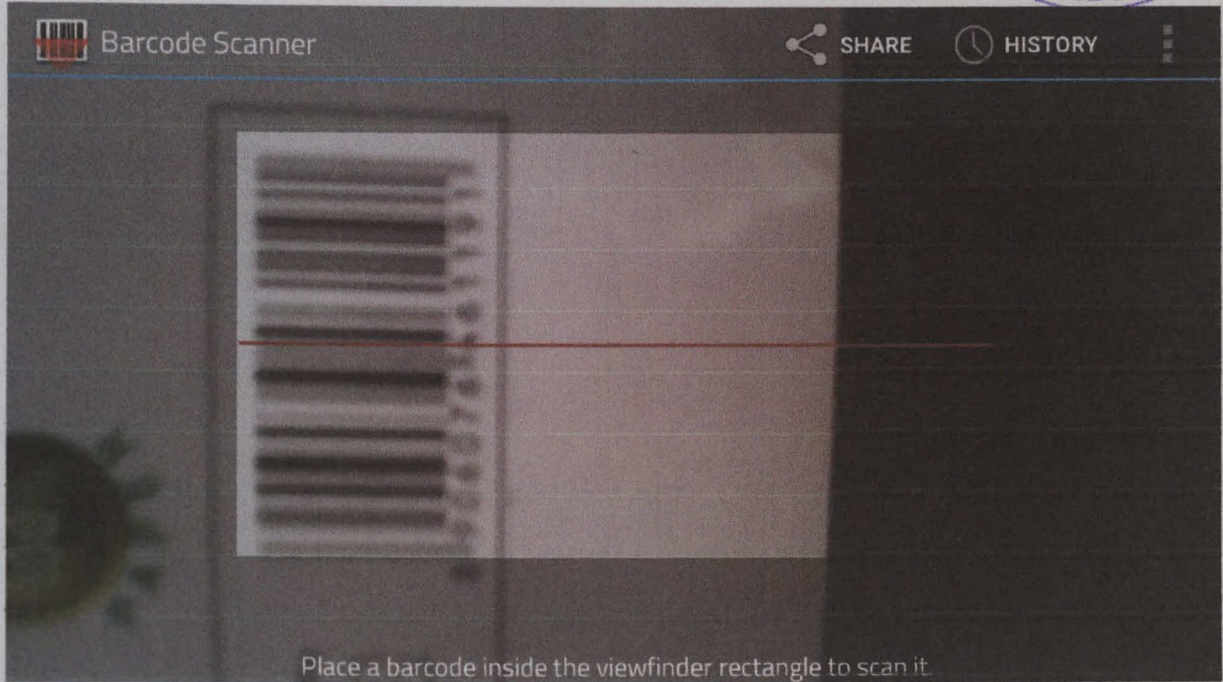
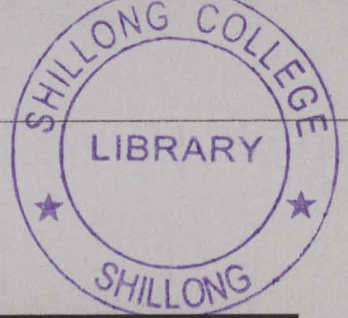
        android:label="@string/app_name" >
<intent-filter>
<action android:name="android.intent.action.MAIN" />

    category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
</application>
</manifest>
```


Build, compile and run

When we build, compile and run our project, the main `AndroidBarcodeQrExample` should look like this:





Conclusion:

We can now have our own Android Qr And Bar Code Scanner. It is very simple application(app) , and we can customize a lot more as much as we can.