DOCUMANTATION OF NjeQR APPLICATION

Emrecan Erkuş

Content of documentation

Requirements and Usage Scnerious

Goal of this application

We have to scan tools of Neumann Janos Egeyetem and send as an Excel file for inform authorized unit.But for this purpose,We have to scan tools and write somewhere one by one at the same time then we have to create Excel file and we have to write value of code and value of scanning time in Excel file. It sounds waste of time.Because with this working style we spend so much time . And also some phone devices has no default scanner.Therefore this application was created.You can scan your tools and after your scanning process you can create an excel file which contains values of tools and scanning times with only one button.This application creates Excel file automatically.You don't have to do anything about Excel and values.This application will reduce work time for scanning work.

Requirements

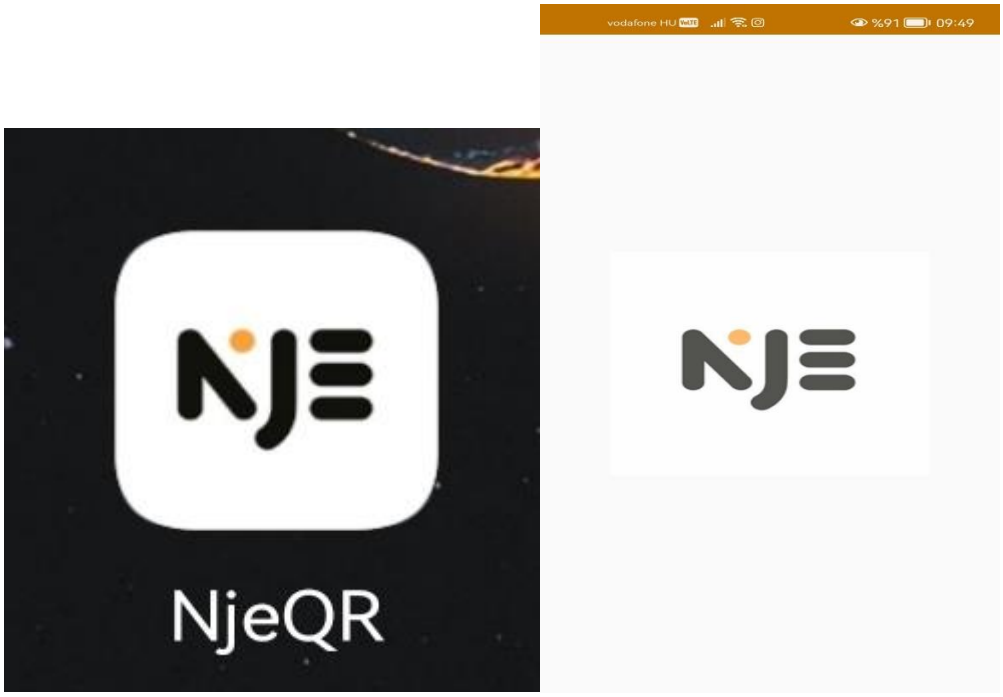
This application written by native language.That means this application only runs in Android devices.

Our minimum SDK for this application is 26.That means this application is suitable for Android 8.0 version and above.

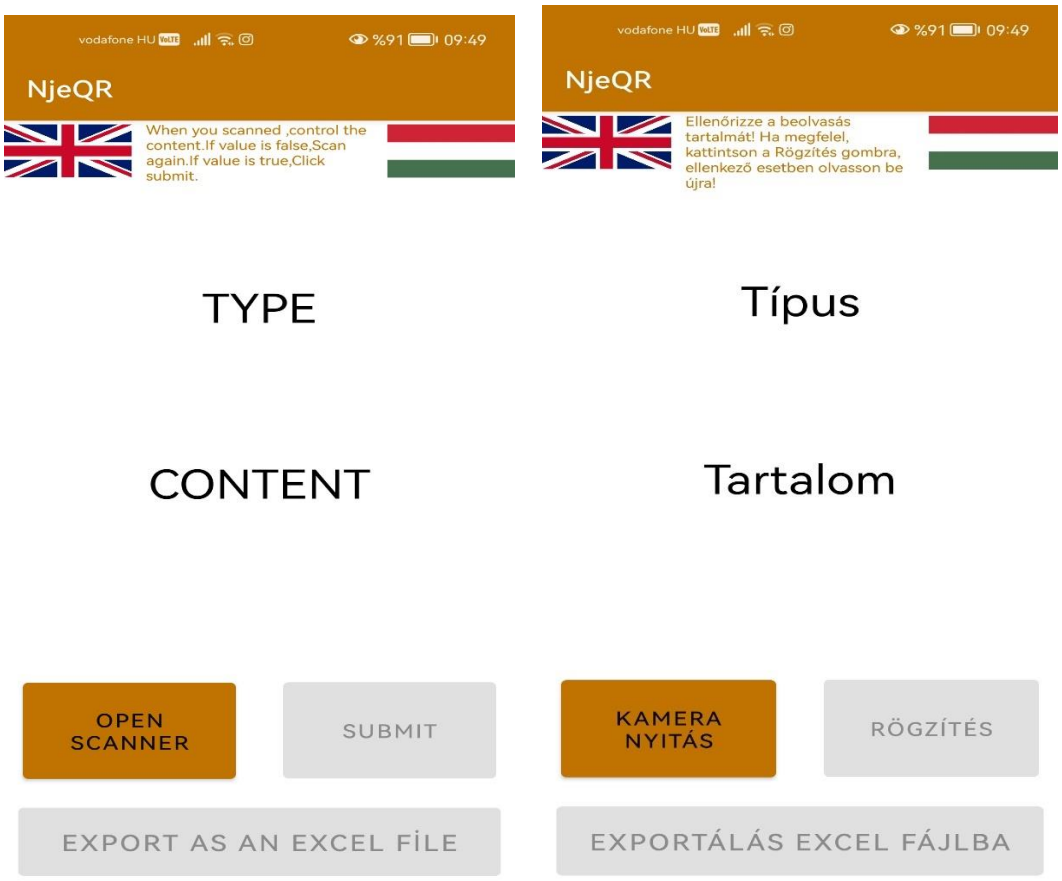
For opening Excel file which was created by application ,you must have Microsoft Office application.If you don’t have Microsoft Office application,our application can’t open your excel file.But even you don’t have office application,our application create your Excel file succesfully. You don't encounter with any error.

Usage Scnerious

When you run the application,splash screen welcomes you.



After that you will see user interface of our application.

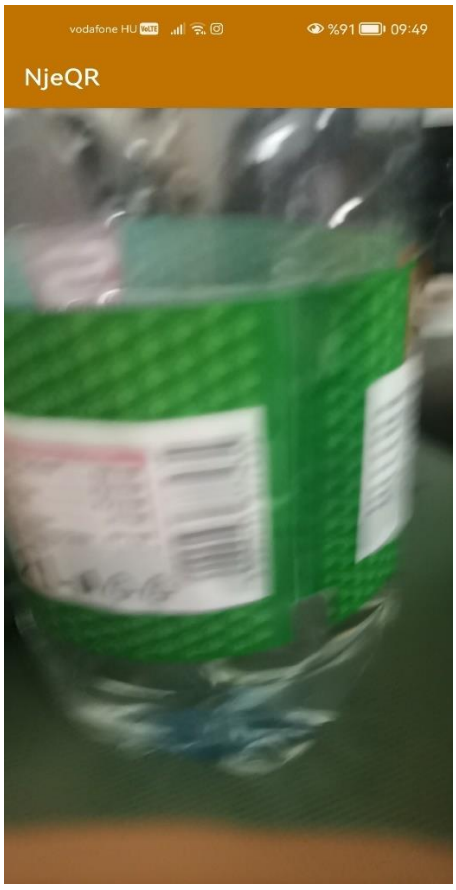


As you see in image,there are two flags in user interface.These are language support flags.That means when you click to Hungarian flag.Language of application will be Hungarian.

We have a text between of flags.This text for inform to user.Beceuse sometimes scanner can scan wrong value cause of barcode quality and camera angle. So user must check value before submit.

We have three button.One is for opening to scanner,one is for submiting value to database and one is for exporting values as an Excel file.But when you run the application,only Open Scenner button will be enable. Because you didn’t take any value from scanner,So you can’t submit or export.

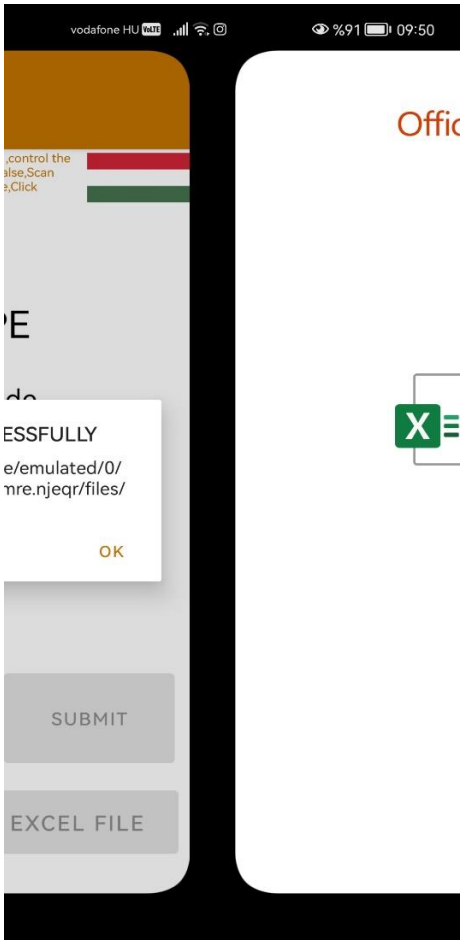
When you click to Open Scanner button.Camera will open for scanning.

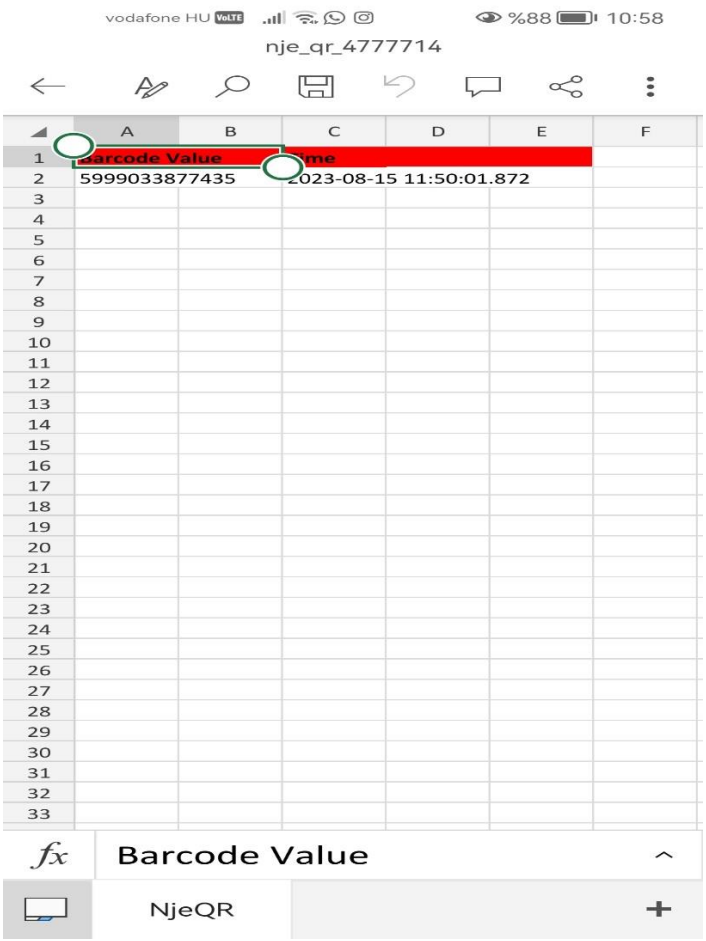


When the scanner scan the barcode,application turns back to user interface and took value and type from barcode.Content is value of barcode which was scanned.Type is type of barcode which was scanned.

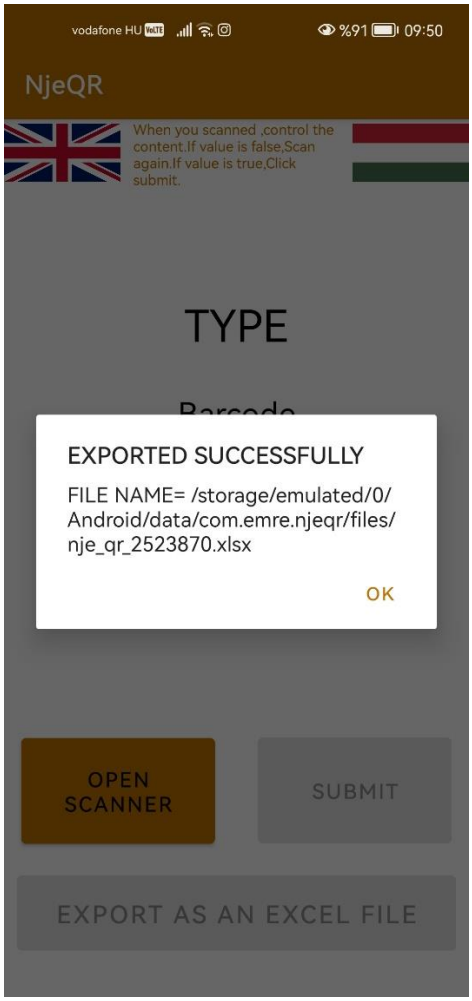
Then submit and export button are enabled.We check the value.If it is true value,we click to submit for submit to database.But if it is wrong value,we scan again by clicking to Open Scanner Button.

After user scan and submit whole barcode,User must click to EXPORT AS AN EXCEL FILE button.This button create Excel file which contains values of whole barcode and values of scanning time.Then takes user to Excel file.





If user’s phone hasn’t got Microsoft Office application,It doesn’t takes user to Excel file.It just creates Excel file and builds AlertDiolog for inform you where it located. Even user has Miscrosoft Office application,User will see this AlertDiolog.



CODE REVIEW

```
binding.button.setOnClickListener { it: View!
    requestCameraAndStartScanner()
    //Veriler burada buradan çek
}
```

For Open Scanner button,I built requestCameraAndStartScanner() function.This function request permission for camera and run StartScanner() function.

```
private fun startScanner(){
    SecondActivity.startScanner( context: this) { barcodes ->
        barcodes.forEach() { barcode ->
            when (barcode.valueType) {
                Barcode.TYPE_URL -> {
                    binding.textViewQrType.text = "URL"
                    binding.textViewQrContent.text = barcode.url.toString()
                }
                Barcode.TYPE_CONTACT_INFO -> {
                    binding.textViewQrType.text = "Contact"
                    binding.textViewQrContent.text = barcode.contactInfo.toString()
                }
                else -> {
                    binding.textViewQrType.text = "Barcode"
                    binding.textViewQrContent.text = barcode.rawValue.toString()
                    value = barcode.rawValue.toString()
                    binding.button2.isEnabled=true
                    //Bunu dene olmadı barcode.displayValue rawdATA
```

This function take value from barcode and convert this value to string.

```
binding.button2.setOnClickListener { it: View!
        timestamp =Timestamp( time: System.currentTimeMillis()+timeZone.rawOffset + if (isDaylig
        try {
            myDatabase.execSQL( sql: "INSERT INTO qr (value,timestamp) VALUES ('$value','$timest
            Toast.makeText( context: this, text: "Submitted Successfully",Toast.LENGTH_LONG)
            binding.ExcelButton.isEnabled=true
            binding.button2.isEnabled=false
        }catch (e:Exception){
            e.printStackTrace()
        }
    }
```

These values and times insert database with function of Submit button.

```
binding.ExcelButton.setOnClickListener { it: View!
    try {
        var cursor=myDatabase.rawQuery( sql: "SELECT * FROM qr", selectionArgs: null)
        val valueIx=cursor.getColumnIndex( columnName: "value")
        val timeIx=cursor.getColumnIndex( columnName: "timestamp")
        while (cursor.moveToNext()){
            valueList.add(cursor.getString(valueIx))
            timeList.add(cursor.getString(timeIx))//BU YANLIŞ OLABİLİR STRING DONDURDU
        }
    }
    catch (e:Exception){
        e.printStackTrace()
        Toast.makeText( context: this,  text: "Veritabanı işlemi sırasında bir hata oluştu: ${e.me
    }
    if (checkWriteExternalStoragePermission()) {
        createExcelFile(valueList,timeList)
        binding.ExcelButton.isEnabled=false
```

I built function of Excel button to take values and timestamps from database and these datas will be added to valueList and timeList.Then createExcelFile() function run.

```
for(i in 0 ≤ .. ≤ (valueList.size-1)){
    val row=sheet.createRow( rownum: i+1)
    row.createCell( columnIndex: 0).setCellValue(valueList[i])
    row.createCell( columnIndex: 2).setCellValue(timeList[i])
}
loadFile(workbook)
```

When you run createExcelFile() function, values are written in Excel file from valueList and timeList.

Then loadFile() function is called.

```
private fun loadFile(workbook:XSSFWorkbook){
    val filePath="nje_qr_${Random.nextInt( from: 0, until: 5000000)}.xlsx"
    val excelDosyaYolu = "${getExternalFilesDir( type: null)}/$filePath"
    try {
        val fileOut=FileOutputStream(excelDosyaYolu)
        workbook.write(fileOut)
        fileOut.close()
        val alertDialog = AlertDialog.Builder( context: this) AlertDialog.Builder
            .setTitle("EXPORTED SUCCESSFULLY") AlertDialog.Builder!
            .setMessage("FILE NAME= ${excelDosyaYolu}")
            .setPositiveButton( text: "OK") { dialog, _ -> dialog.dismiss() }
            .create()

        alertDialog.show()
        excelFileYolu=excelDosyaYolu
        excelAç()
```

With LoadFile() function,Excel file will be written in storage of phone.This function writes Excel file in storage of phone. Then excelAç() function is called.

```
private fun excelAç(){

    val file = File(excelFileYolu)
    if (file.exists()) {
        val uri = FileProvider.getUriForFile(
            context: this, // Bu, Context nesnesini temsil eder
            authority: "com.yourapp.fileprovider", // Dosya sağlayıcının yetki adı (
            file
        )

        val intent = Intent(Intent.ACTION_VIEW)
        intent.setDataAndType(uri, type: "application/vnd.openxmlformats-officedocu
        intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION)
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK)

        try {
            startActivity(intent)
```

With this function,Excel file which is created and written will be opened directly.

API DOCUMANTATION

In this project I used API structers.I used CameraX and Apache POI.

CameraX

CameraX is a camera API which is created by Google.With this API you can use camera and you can implement Google ML kits to your camera easily.For this project I implmenent Barcoda Scanner feature to camera.

Apache POI

Apache POI is an API which is created for managing Microsoft Office documents.With this API you can read and write Microsoft Office documents easily.For this project I used this API for writing Excel File to storage of phone.

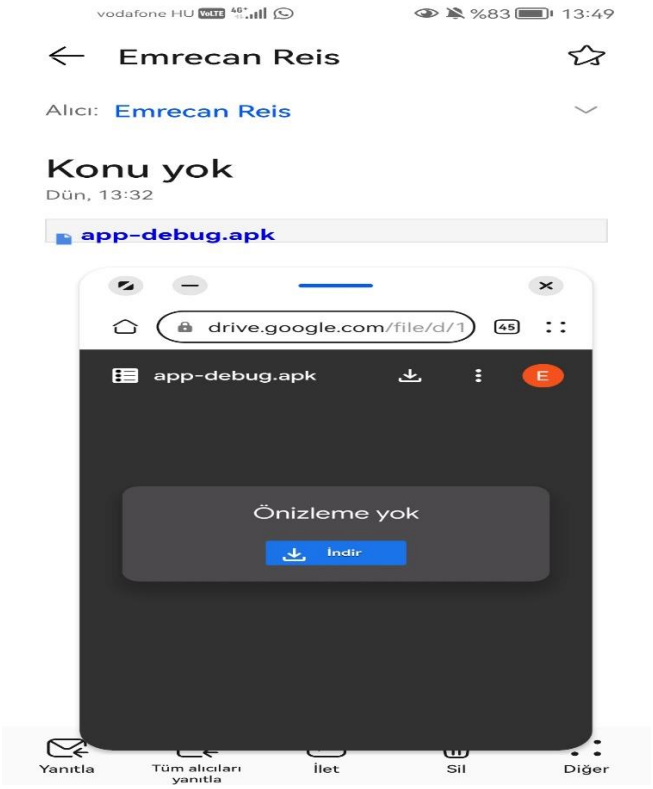
DEPENDACY AND SETUP

DEPANDACY

As I mentioned previous pages, android device(8.0 version and above) and Microsoft Office application(it is not compulsory) are required for running this application successfuly.

SET UP

We will send email which contains connection link to google drive to workers who will scan barcodes. We will give access permission for google drive to these workers. When they click to link via their phone. Loading page will open.



TROUBLESHOOTING

As I mentioned previous pages,Scanner can scan wrong value sometimes.It is not because of scanner. I tried to scanner on barcodes except our school's.Accuracy rate was 2 in 3 scan.But when I scanned on barcodes which are our school's,accuracy rate was decreased 1 in 3 scan.So you will face with this issue lots of times.

So the reoson of scan wrong value is barcode quality and camera angle. When you got wrong value from scanner,change to angle of camera.It will help to solve this issue.

PERFORMANCE AND SAFETY

Performance

The size of application is about 40MB.It is not much for storage of phone.Also our application is writing excel file which is approximately 5KB.It is so small size for performance.Performance is not affacted by writing Excel file.

SAFETY

We won’t publish our application to everyone.We will publish our application via GoogleDrive to workers who will scan barcodes.And also I didn’t use server or etc for database.I used local database of phone.So application safety is related to phone safety of workers who will scan barcodes.Datas will be stored in database of phones.

VERSION NOTES

Google said that they won’t use WRITE_EXTERNAL_STORAGE permission anymore for android devices which has Android 13 and above. But as you know Android 13 is beta version right now. So it is okay to use WRITE_EXTERNAL_STORAGE permission for this time.But in the following times,the person who will devolope this application must update code according to Android 13 and above versions.