Министерство науки и высшего образования Российской Федерации

Федерального государственного бюджетное образовательное учреждение высшего образования

«Российской экономический университет имени Г. В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ.**

ПРАКТИЧЕСКИЕ РАБОТЫ № 14

ДИСЦИПЛИНА: «РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ»

Тема: Работа с датчиками и создание гироскопа.

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

Листов:

|  |  |  |
| --- | --- | --- |
| Выполнил студент |  | Проверил преподаватель |
| Группа П50-3-18 |  | \_\_\_\_\_\_\_\_\_\_\_А.О.Лясников |
| Пахомов Даниил Александрович |  | «\_\_\_» \_\_\_\_\_\_\_\_2020 года |

Москва 2021

ПРАКТИЧЕСКАЯ РАБОТА №12

Работа с Отпечатком пальца

Цель: создать программу, которая будет взаимодействовать с отпечатком пальца и авторизует пользователя.

1. Android Manifest

|  |
| --- |
| *<?*xml version="1.0" encoding="utf-8"*?>* <manifest xmlns:android="http://schemas.android.com/apk/res/android"  package="com.example.fingerprintscanner">  <uses-permission android:name="android.permission.USE\_BIOMETRIC" />   <application  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:roundIcon="@mipmap/ic\_launcher\_round"  android:supportsRtl="true"  android:theme="@style/Theme.FingerprintScanner">  <activity android:name=".SecondActivity"></activity>  <activity android:name=".MainActivity">  <intent-filter>  <action android:name="android.intent.action.MAIN" />   <category android:name="android.intent.category.LAUNCHER" />  </intent-filter>  </activity>  </application>  </manifest> |

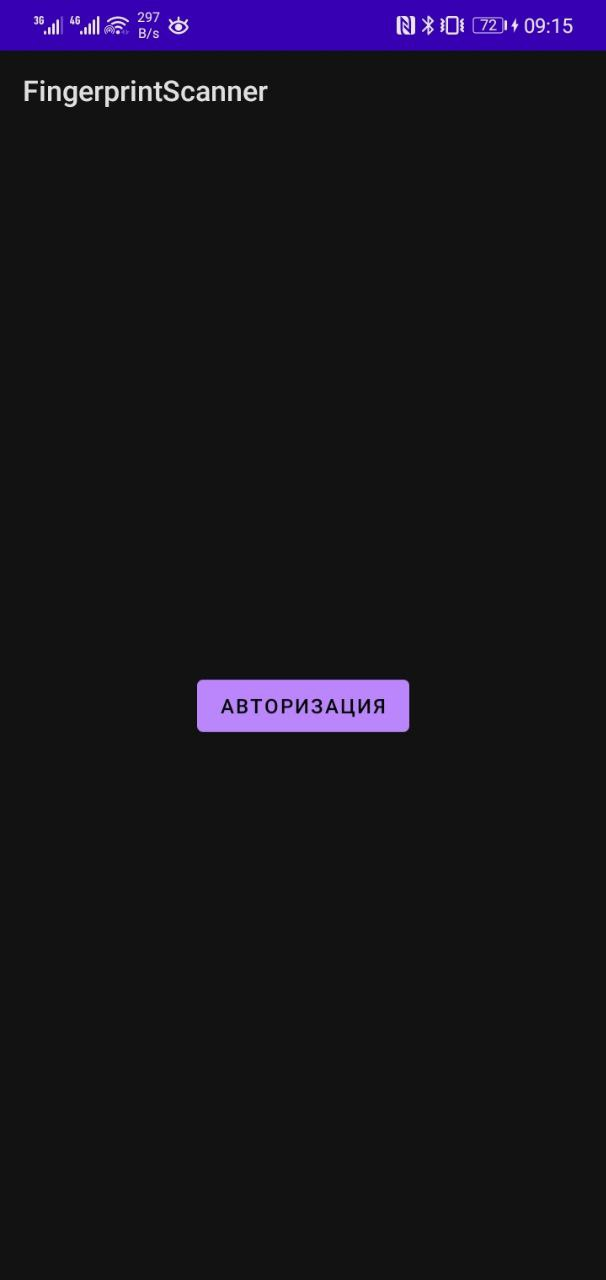
1. Activity\_main

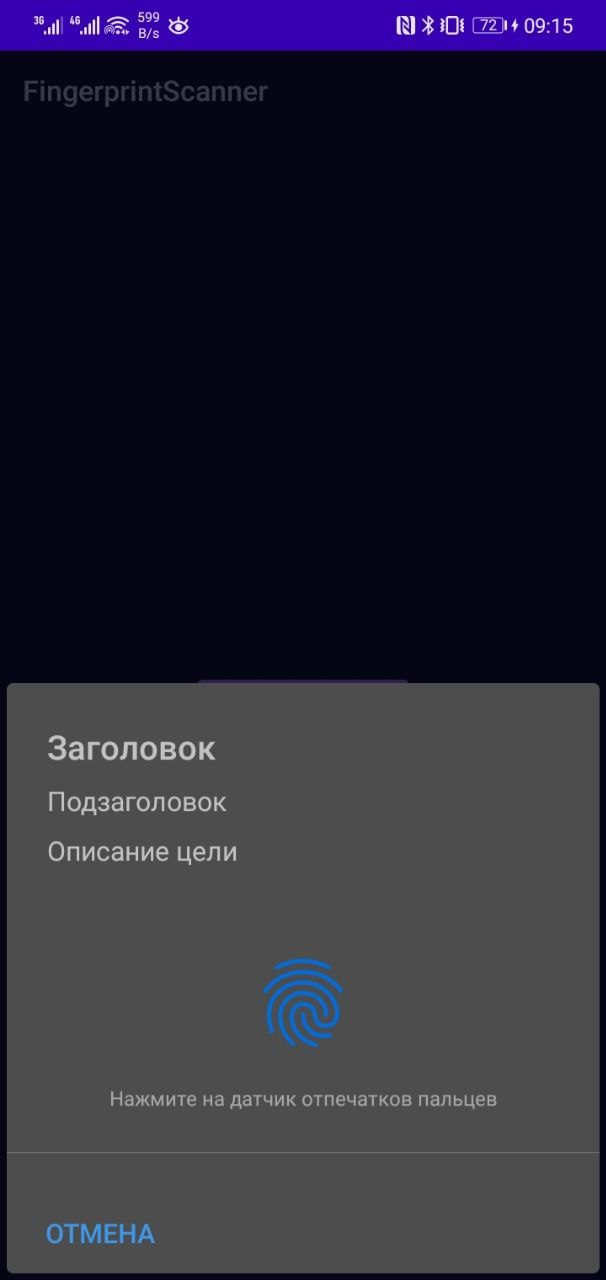
|  |
| --- |
| *<?*xml version="1.0" encoding="utf-8"*?>* <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">   <Button  android:id="@+id/btn"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Авторизация"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout> |

1. MainActivity

|  |
| --- |
| package com.example.fingerprintscanner  import android.app.Activity import android.app.KeyguardManager import android.content.Context import android.content.DialogInterface import android.content.Intent import android.content.pm.PackageManager import android.hardware.biometrics.BiometricPrompt import android.os.Build import androidx.appcompat.app.AppCompatActivity import android.os.Bundle import android.os.CancellationSignal import android.widget.Toast import androidx.annotation.RequiresApi import androidx.core.app.ActivityCompat import kotlinx.android.synthetic.main.activity\_main.\*  class MainActivity : AppCompatActivity() {   private var cancellationSignal: CancellationSignal? = null   private val authenticationCallback: BiometricPrompt.AuthenticationCallback  get() = @RequiresApi(Build.VERSION\_CODES.*P*)  object : BiometricPrompt.AuthenticationCallback() {  override fun onAuthenticationError(errorCode: Int, errString: CharSequence?) {  notifuUser("Ошибка при авторизации $errString")  super.onAuthenticationError(errorCode, errString)  }   override fun onAuthenticationSucceeded(result: BiometricPrompt.AuthenticationResult?) {  notifuUser("Вы усешно авторизовались!")  startActivity(Intent(this@MainActivity, SecondActivity::class.*java*))  super.onAuthenticationSucceeded(result)  }  }    private fun notifuUser(message: String) {  Toast.makeText(this, message, Toast.*LENGTH\_LONG*).show()  }   @RequiresApi(Build.VERSION\_CODES.*P*)  override fun onCreate(savedInstanceState: Bundle?) {  super.onCreate(savedInstanceState)  setContentView(R.layout.*activity\_main*)   checkBiomatricSupport()   btn.setOnClickListener **{** val biometricPrompt = BiometricPrompt.Builder(this).setTitle("Заголовок").setSubtitle("Подзаголовок").setDescription("Описание цели").setNegativeButton("Отмена", this.*mainExecutor*, DialogInterface.OnClickListener **{** dialog, which **->** notifuUser("Авторизация отменена")  **}**).build()   biometricPrompt.authenticate(getCansellationSignal(), *mainExecutor*,authenticationCallback)  **}** }   private fun getCansellationSignal(): CancellationSignal {  cancellationSignal = CancellationSignal()  cancellationSignal?.setOnCancelListener **{** notifuUser("Пользователь оменил скан отпечатка пальца")  **}** return cancellationSignal as CancellationSignal  }    private fun checkBiomatricSupport(): Boolean {  val keyguardManager = getSystemService(Context.*KEYGUARD\_SERVICE*) as KeyguardManager   if (keyguardManager.*isKeyguardSecure*) {  notifuUser("аутентификация по отпечатку пальца не была включена в настройках")  return false  }  if (ActivityCompat.checkSelfPermission(this, android.Manifest.permission.*USE\_BIOMETRIC*) != PackageManager.*PERMISSION\_GRANTED*)  {  notifuUser("Вы не дали разрешение на работу с отпечатком пальца")  }   return if(*packageManager*.hasSystemFeature(PackageManager.*FEATURE\_FINGERPRINT*))  {  true  }else  {  true  }   } } |

1. Результат программы







Вывод: была создана программа, которая будет взаимодействовать с отпечатком пальца и авторизует пользователя.