**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное

учреждение высшего образования

**Национальный исследовательский**

**Томский политехнический университет**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Отчет по лабораторной работе №19 по дисциплине

**«Язык Kotlin и основы разработки»**

Разрешения

****

Выполнил:

Студент группы 1А22 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.К. Кравцов

Проверил:

Ст. преп. ОИТ ИШИТР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Дорофеев

Томск 2025

# Задание

Разработать приложение для получения списка контактов из телефонной книги пользователя. Для этого необходимо разрешение READ\_CONTACTS.

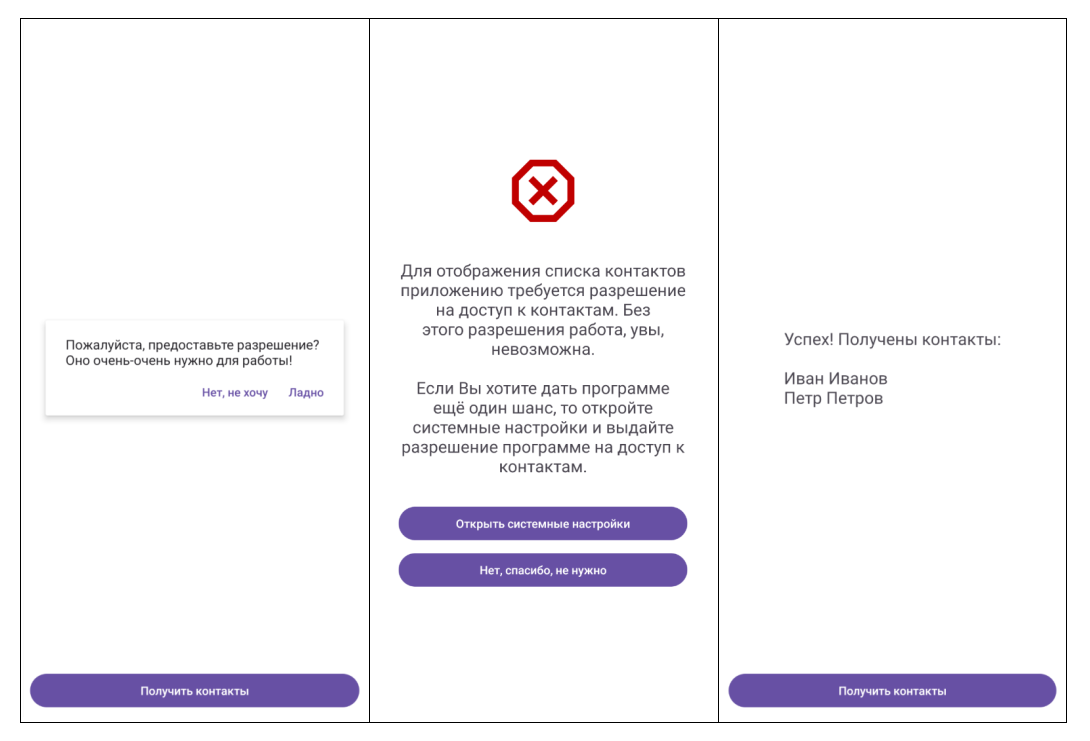
По нажатию кнопки «Получить контакты» приложение должно проверить наличие разрешения, и, если оно отсутствует – начать процесс его получения.

Если система считает, что нужно показать пользователю обоснование необходимости разрешения – показать такое обоснование в какой-либо форме.

Если пользователь отказал в разрешении – показать ему информацию о том, что без разрешения функционал приложения ограничен, и дать возможность перейти в системные настройки приложения.

Если разрешение получено – считать имена контактов из адресной книги и отобразить их на экране в какой-либо форме.

Экраны приложения могут выглядеть примерно таким образом:



# Ход работы

1. Создан проект Lab19 на основе Empty Views Activity.\
2. В файл манифеста добавлено разрешение на чтение контактов:

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <uses-permission android:name="android.permission.READ\_CONTACTS" />  
  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.Lab19">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>

1. Настроены ресурсы приложения:

* strings.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <string name="app\_name">Lab19</string>  
 <string name="get\_contacts\_button">Получить контакты</string>  
 <string name="contacts\_placeholder">Контакты будут отображены здесь</string>  
 <string name="permission\_dialog\_title">Разрешение на доступ к контактам</string>  
 <string name="permission\_rationale\_message">Для отображения списка контактов приложению требуется разрешение на доступ к контактам. Без этого разрешения работа невозможна.</string>  
 <string name="ok\_button">Ладно</string>  
 <string name="no\_thanks\_button">Нет, не хочу</string>  
 <string name="denied\_dialog\_title">Доступ запрещен</string>  
 <string name="denied\_dialog\_message">Для отображения списка контактов приложению требуется разрешение на доступ к контактам. Без этого разрешения работа, увы, невозможна.\n\nЕсли Вы хотите дать программе ещё один шанс, откройте системные настройки и выдайте разрешение программе на доступ к контактам.</string>  
 <string name="open\_settings\_button">Открыть настройки</string>  
 <string name="no\_thanks\_final\_button">Нет, спасибо</string>  
 <string name="success\_message">Успех! Получены контакты:\n\n%s</string>  
 <string name="no\_contacts\_found">Контакты не найдены</string>  
 <string name="permission\_error">Ошибка: нет разрешения на доступ к контактам</string>  
 <string name="contacts\_error">Ошибка при получении контактов: %s</string>  
</resources>

* dimens.xml

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <dimen name="padding\_small">8dp</dimen>  
 <dimen name="padding\_medium">16dp</dimen>  
 <dimen name="padding\_large">24dp</dimen>  
 <dimen name="text\_size\_medium">16sp</dimen>  
 <dimen name="text\_size\_large">24sp</dimen>  
</resources>

1. Разработан пользовательский интерфейс. Основной макет activity\_main содержит кнопку для запроса контактов и текстовое поле для их отображения:

<?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:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/btnGetContacts"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/get\_contacts\_button"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/btnGetContacts">  
  
 <TextView  
 android:id="@+id/tvContacts"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:padding="@dimen/padding\_medium"  
 android:text="@string/contacts\_placeholder"  
 android:textSize="@dimen/text\_size\_medium" />  
 </ScrollView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

1. Реализована основная логика приложения в MainActivity.kt. Код включает проверку разрешений, запрос разрешения времени выполнения, отображение диалогов с обоснованием и обработку результатов:

package ru.olegkravtsov.lab19  
  
import android.annotation.SuppressLint  
import android.content.Intent  
import android.net.Uri  
import android.os.Bundle  
import android.provider.ContactsContract  
import android.provider.Settings  
import android.widget.Button  
import android.widget.TextView  
import androidx.activity.enableEdgeToEdge  
import androidx.activity.result.contract.ActivityResultContracts  
import androidx.appcompat.app.AlertDialog  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.content.ContextCompat  
import androidx.core.view.ViewCompat  
import androidx.core.view.WindowInsetsCompat  
  
class MainActivity : AppCompatActivity() {  
  
 private lateinit var btnGetContacts: Button  
 private lateinit var tvContacts: TextView  
  
 // Лаунчер для запроса разрешения  
 private val requestPermissionLauncher = registerForActivityResult(  
 ActivityResultContracts.RequestPermission()  
 ) { isGranted ->  
 if (isGranted) {  
 // Разрешение получено - получаем контакты  
 getAndDisplayContacts()  
 } else {  
 // Пользователь отказал - показываем объяснение  
 showPermissionDeniedDialog()  
 }  
 }  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 enableEdgeToEdge()  
 setContentView(R.layout.activity\_main)  
  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->  
 val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
 insets  
 }  
  
 initViews()  
 setupClickListeners()  
 }  
  
 private fun initViews() {  
 btnGetContacts = findViewById(R.id.btnGetContacts)  
 tvContacts = findViewById(R.id.tvContacts)  
 }  
  
 private fun setupClickListeners() {  
 btnGetContacts.setOnClickListener {  
 checkPermissionAndGetContacts()  
 }  
 }  
  
 private fun checkPermissionAndGetContacts() {  
 // Проверяем есть ли уже разрешение  
 when {  
 ContextCompat.checkSelfPermission(  
 this,  
 android.Manifest.permission.READ\_CONTACTS  
 ) == android.content.pm.PackageManager.PERMISSION\_GRANTED -> {  
 // Разрешение уже есть - получаем контакты  
 getAndDisplayContacts()  
 }  
  
 shouldShowRequestPermissionRationale(android.Manifest.permission.READ\_CONTACTS) -> {  
 // Нужно показать объяснение пользователю  
 showPermissionRationaleDialog()  
 }  
  
 else -> {  
 // Запрашиваем разрешение без объяснения  
 requestPermissionLauncher.launch(android.Manifest.permission.READ\_CONTACTS)  
 }  
 }  
 }  
  
 private fun showPermissionRationaleDialog() {  
 AlertDialog.Builder(this)  
 .setTitle(getString(R.string.permission\_dialog\_title))  
 .setMessage(getString(R.string.permission\_rationale\_message))  
 .setPositiveButton(getString(R.string.ok\_button)) { \_, \_ ->  
 requestPermissionLauncher.launch(android.Manifest.permission.READ\_CONTACTS)  
 }  
 .setNegativeButton(getString(R.string.no\_thanks\_button)) { dialog, \_ ->  
 dialog.dismiss()  
 }  
 .show()  
 }  
  
 private fun showPermissionDeniedDialog() {  
 AlertDialog.Builder(this)  
 .setTitle(getString(R.string.denied\_dialog\_title))  
 .setMessage(getString(R.string.denied\_dialog\_message))  
 .setPositiveButton(getString(R.string.open\_settings\_button)) { \_, \_ ->  
 openAppSettings()  
 }  
 .setNegativeButton(getString(R.string.no\_thanks\_final\_button)) { dialog, \_ ->  
 dialog.dismiss()  
 }  
 .show()  
 }  
  
 private fun openAppSettings() {  
 val intent = Intent(  
 Settings.ACTION\_APPLICATION\_DETAILS\_SETTINGS,  
 Uri.parse("package:$packageName")  
 )  
 startActivity(intent)  
 }  
  
 @SuppressLint("Range")  
 private fun getContacts(): List<String> {  
 val result = mutableListOf<String>()  
  
 val cursor = contentResolver.query(  
 ContactsContract.Contacts.CONTENT\_URI,  
 null, null, null, null  
 )  
  
 cursor?.use {  
 val nameColumnIndex = it.getColumnIndex(ContactsContract.Contacts.DISPLAY\_NAME)  
  
 while (it.moveToNext()) {  
 val name = it.getString(nameColumnIndex)  
 if (!name.isNullOrEmpty()) {  
 result.add(name)  
 }  
 }  
 }  
  
 return result  
 }  
  
 private fun getAndDisplayContacts() {  
 try {  
 val contacts = getContacts()  
 if (contacts.isNotEmpty()) {  
 val contactsText = getString(R.string.success\_message, contacts.joinToString("\n"))  
 tvContacts.text = contactsText  
 } else {  
 tvContacts.text = getString(R.string.no\_contacts\_found)  
 }  
 } catch (e: SecurityException) {  
 tvContacts.text = getString(R.string.permission\_error)  
 } catch (e: Exception) {  
 tvContacts.text = getString(R.string.contacts\_error, e.message ?: "неизвестная ошибка")  
 }  
 }  
}

**Результат работы**

Приложение успешно запускается и отображает интерфейс с кнопкой "Получить контакты" (рис. 1).



Рисунок 1 – Начальный экран приложения

При первом нажатии на кнопку система запрашивает разрешение на доступ к контактам (рис. 2).

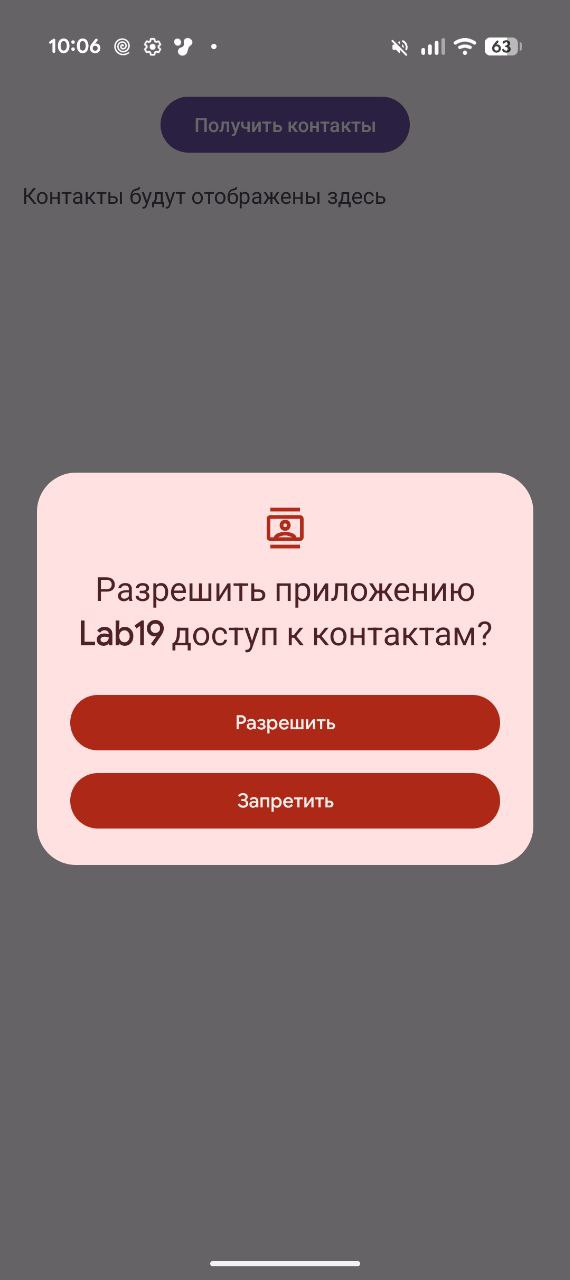


Рисунок 2 – Запрос разрешения системы

Если пользователь ранее отклонял запрос, отображается диалог с обоснованием необходимости разрешения (рис. 3).

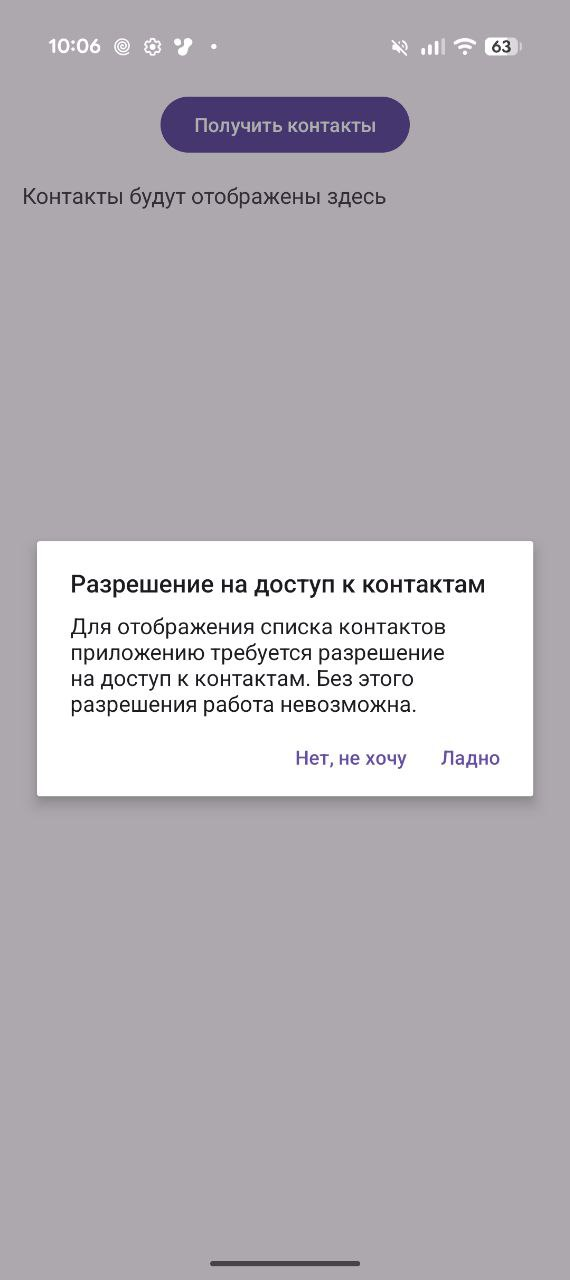


Рисунок 3 – Обоснование необходимости разрешения

После двукратного отказа система перестаёт показывать запрос. В этом случае приложение отображает диалог с объяснением и предлагает перейти в настройки (рис. 4).

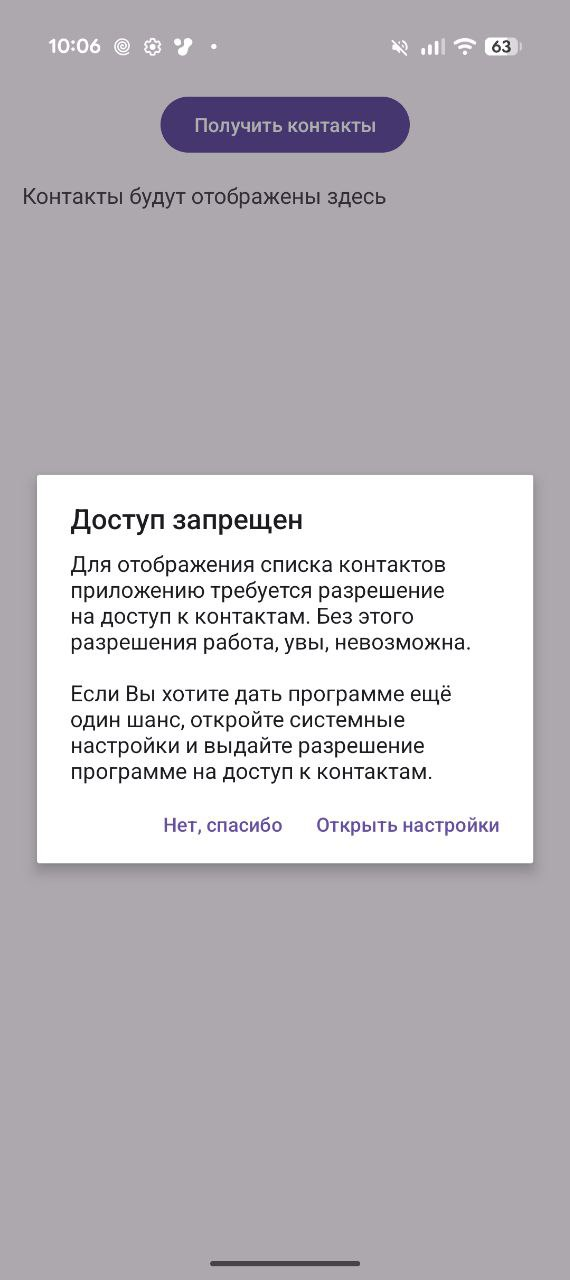


Рисунок 4 – Предложение открыть настройки при повторных отказах

При получении разрешения приложение загружает и отображает список контактов (рис. 5).



Рисунок 5 – Успешное получение списка контактов

# Выводы

В ходе выполнения лабораторной работы было разработано приложение для работы с контактами, использующее систему разрешений Android. Освоен механизм запроса разрешений во время выполнения, включая проверку текущего статуса разрешения, отображение обоснования и обработку всех возможных сценариев пользовательского взаимодействия. Реализована корректная работа с ContentResolver для чтения данных контактов после получения необходимых разрешений. Приложение демонстрирует правильное обращение с ситуациями повторного отказа от разрешения через предоставление доступа к системным настройкам. Полученные навыки позволяют создавать приложения, требующие доступ к защищённым данным.