

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

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:

<div data-bbox="167 1238 547 1361"><p>Пожалуйста, предоставьте разрешение? Оно очень-очень нужно для работы!</p><p>Нет, не хочу Ладно</p></div> <div data-bbox="151 1680 563 1720"><p>Получить контакты</p></div>	<div data-bbox="753 1039 834 1120"></div> <div data-bbox="614 1167 975 1288"><p>Для отображения списка контактов приложению требуется разрешение на доступ к контактам. Без этого разрешения работа, увы, невозможна.</p></div> <div data-bbox="614 1312 975 1433"><p>Если Вы хотите дать программе ещё один шанс, то откройте системные настройки и выдайте разрешение программе на доступ к контактам.</p></div> <div data-bbox="614 1471 975 1507"><p>Открыть системные настройки</p></div> <div data-bbox="614 1529 975 1568"><p>Нет, спасибо, не нужно</p></div>	<div data-bbox="1096 1249 1370 1276"><p>Успех! Получены контакты:</p></div> <div data-bbox="1096 1301 1227 1350"><p>Иван Иванов Петр Петров</p></div> <div data-bbox="1029 1680 1441 1720"><p>Получить контакты</p></div>
--	---	---

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>
```

3. 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>
```

4. 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>
```

5. 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
    }

    initView()
    setupClickListeners()
}

private fun initView() {
    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_CONTACT
S) -> {

        // 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).

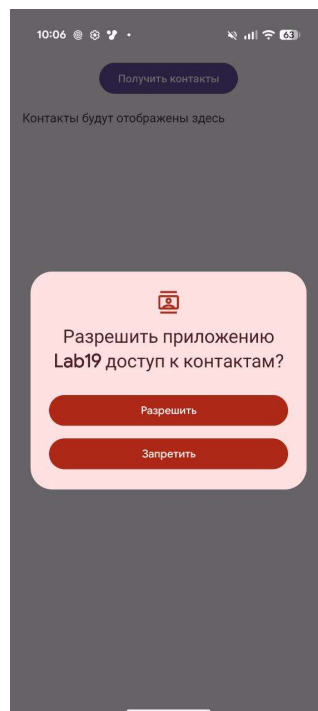


Рисунок 2 – Запрос разрешения системы

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

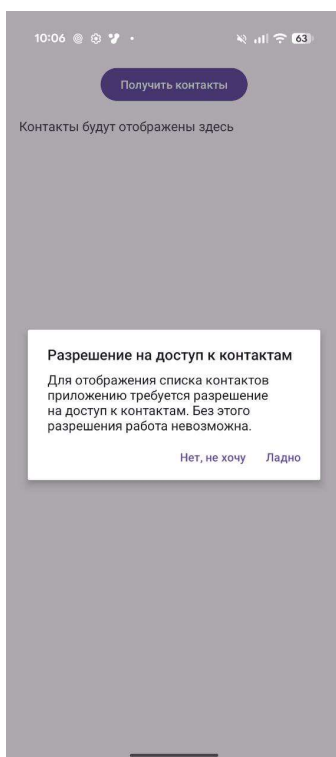


Рисунок 3 – Обоснование необходимости разрешения

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

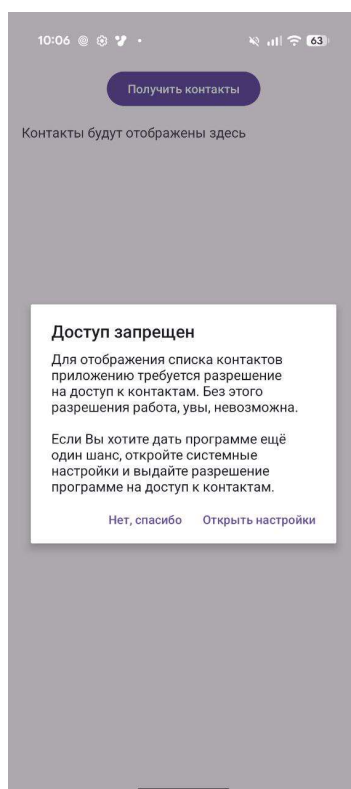


Рисунок 4 – Предложение открыть настройки при повторных отказах

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.