Ćwiczenia 21 – Android studio – sms, email send, shortcut

Na koniec zajęć prześlij pliki źródłowe (.xml, .java)+ obrazek do zasobu w teams.

1. Utwórz projekt o nazwie MySmsMailShortcut na podstawie Empty Activity, dobierz odpowiednie API (28 – Android 9).
2. Otwórz dokumentację:

   https://developer.android.com/guide/components/intents-common#ComposeEmail

   https://developer.android.com/guide/components/intents-common#SendMessage

   https://developer.android.com/reference/android/telephony/SmsManager

   https://developer.android.com/guide/topics/ui/shortcuts

3. AndroidManifest.xml

AndroidManifest.xml | MainActivity.java | activity_main.xml | MyReceiver.java

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

    <uses-permission android:name="android.permission.RECEIVE_SMS"></uses-permission>
    <uses-permission android:name="android.permission.SEND_SMS"></uses-permission>
    <uses-permission android:name="android.permission.INTERNET"></uses-permission>

    <application
        android:allowBackup="true"
```

4. Przygotuj activity_main.xml (podanie numeru telefonu, treści wiadomości, przyciski do wysyłki smsa i maila pole dla emaila, miejsce do odbioru smsów)
5. Dodaj potrzebne zmienne:

```java
public class MainActivity extends AppCompatActivity {
    private static final String TAG = "SMS111";
    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS = 1;
    private static final int MY_PERMISSIONS_REQUEST_RECEIVE_SMS = 2;
    SmsManager smsManager;
    String destinationAddress = "";
    String scAddress = null;
    String text ="";
    PendingIntent sentIntent = null;
    PendingIntent deliveryIntent = null;
    long messageId = 0;
    EditText phoneNumber;
    EditText smsMessage;
    Button sendSms;
```

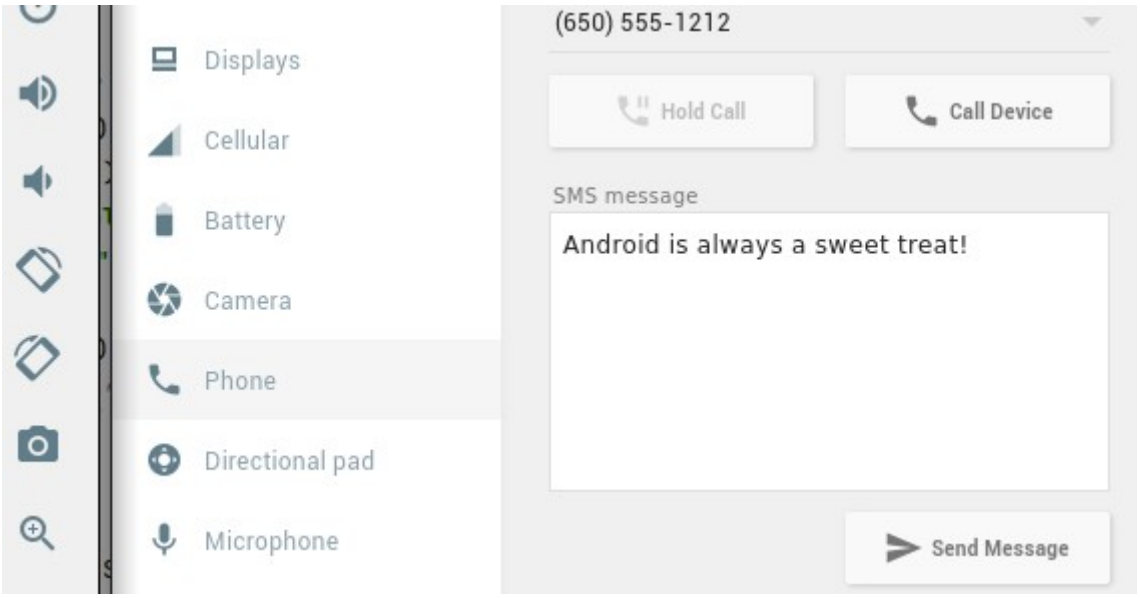
6. Stwórz metodę wysyłającą sms poprzez SmsManagera, np.:

```
private void sendWithSmsManager() {


    if (checkPermission(Manifest.permission.SEND_SMS)) {
        destinationAddress = phoneNumber.getText().toString();
        text = smsMessage.getText().toString();
        if (!destinationAddress.equals("") && !text.equals("")) {
            smsManager = SmsManager.getDefault();

            smsManager.sendTextMessage(
                    destinationAddress,
                    scAddress: null,
                    text,
                    sentIntent: null,
                    deliveryIntent: null
                    //,
                    //messageId
            );
            Toast.makeText( context: MainActivity.this,  text: "SMS send", Toast.LENGTH_SHOR
            Log.v(TAG,  msg: "Sms send");
        } else {
            Toast.makeText( context: MainActivity.this,  text: "Permission denied", Toast.LE
            Log.v(TAG,  msg: "Permission denied");
        }
    }
}
```

7. Przetestuj aplikację, np.:



8. Stwórz metodę wysyłającą sms poprzez Intent, np.:

```
private void sendSmsWithIntent() {
    Log.v(TAG,  msg: "Prepare to send sms with Intent");
    destinationAddress = phoneNumber.getText().toString();
    text = smsMessage.getText().toString();
    Intent intent = new Intent(Intent.ACTION_SENDTO, Uri.parse("smsto:"));
    if(!destinationAddress.equals("") && !text.equals("")) {

       // intent.setData(Uri.parse("smsto:" + destinationAddress));
        intent.setType("vnd.android-dir/mms-sms");
        //intent.putExtra("address"  , destinationAddress);
        intent.putExtra( name: "sms_body", text);
    }
    try {
        startActivity(intent);
       // finish();
        Log.v(TAG,  msg: "Finished sending SMS...");
    } catch (android.content.ActivityNotFoundException ex) {
        Toast.makeText( context: MainActivity.this,
                text: "SMS faild, please try again later.", Toast.LENGTH_SH
    }
}
```

9. Wyślij mail poprzez Intent, np.:

```
public void sendEmail(String mailBody){
    Intent mailIntent = new Intent(Intent.ACTION_SENDTO);
    mailIntent.setData(Uri.parse("mailto:"));
    mailIntent.putExtra(Intent.EXTRA_EMAIL,  value: "login@example.com");
    mailIntent.putExtra(Intent.EXTRA_SUBJECT, getString(R.string.subject));
    mailIntent.putExtra(Intent.EXTRA_TEXT,mailBody);
    startActivity(mailIntent);
}
```

10. Przetestuj wysyłanie maila.

11. Dodaj klasę MyReceiver.java rozszerz ją o BroadcastReceiver

```
public class MyReceiver extends BroadcastReceiver {

    private static final String TAG = "sms111";
    public static final String pdu_type = "pdus";
    private static String SMS = "android.provider.Telephony.SMS_RECEIVED";
    String message="";

    @Override
    public void onReceive(Context context, Intent intent) {



        if (intent.getAction().equalsIgnoreCase(SMS)) {
            Bundle bundle = intent.getExtras();
            SmsMessage[] smsMessages = null;
            String format = bundle.getString( key: "format");
            Log.v(TAG, msg: "format="+format);
            // Retrieve the SMS message received
            Object[] pdus = (Object[]) bundle.get(pdu_type);
            if(pdus != null) {
                smsMessages = new SmsMessage[pdus.length];

                for (int i = 0; i < smsMessages.length; i++) {
                    // https://developer.android.com/reference/kotlin/android/telephony/SmsMessage#createfrompdu_1
                    smsMessages[i] = SmsMessage.createFromPdu(
                            (byte[]) pdus[i],
                            format);
                    message += "Sms from: " + smsMessages[i].getOriginatingAddress();
                    message += smsMessages[i].getMessageBody() + "\n";
                    Log.v(TAG,  msg: "----------------------> message: " + message);
                    Toast.makeText(context, message, Toast.LENGTH_LONG).show();
                }
            }
        }
```

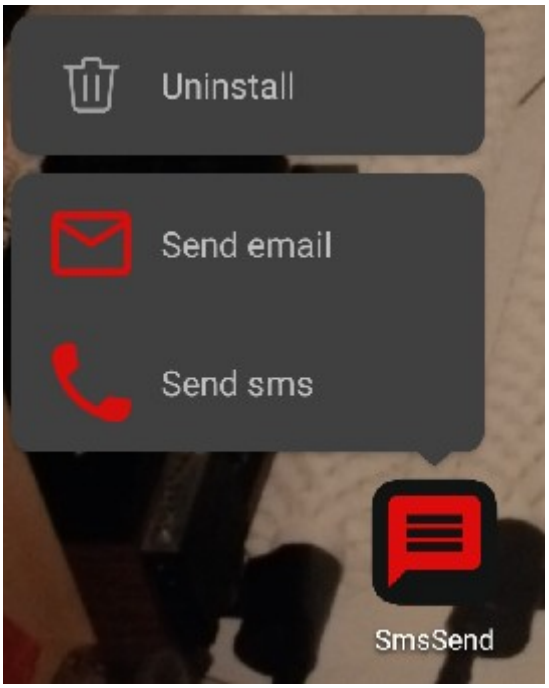
12. Dodaj w AndroidManifest.xml obsługę odbiornika

```
    </activity>
    <receiver
        android:name="com.example.smssend.MyReceiver"
        android:enabled="true"
        android:exported="true">
        <intent-filter>
            <category android:name="android.intent.category.DEFAULT"/>
            <action android:name="android.provider.Telephony.SMS_RECEIVED"/>
        </intent-filter>
    </receiver>
</application>
```

13. Przetestuj odbiór smsów.

14. Dodaj skróty w ikonie aplikacji dla funkcjonalności:
    a. wyślij sms (skrót statyczny)
    b. wyślij email (skrót statyczny)
    c. przeglądaj wysłane wiadomości (skrót dynamiczny)
    d. przeglądaj sms, które przyszły (skrót dynamiczny)

https://developer.android.com/guide/topics/ui/shortcuts/creating-shortcuts



Realizacja dla skrótów statycznych:

Zawartość z AndroidManifest.xml:

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

        <meta-data
            android:name="android.app.shortcuts"
            android:resource="@xml/shortcuts" />
    </activity>

    <receiver
        android:name=".MyReceiver"
```

shortcuts.xml | MainActivity.java | Messages.java | activity_messages.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<shortcuts xmlns:android="http://schemas.android.com/apk/res/android">

    <shortcut
        android:shortcutId="sms"
        android:enabled="true"
        android:icon="@drawable/ic_baseline_local_phone_24"
        android:shortcutShortLabel="@string/sms_shortcut_short_label1"
        android:shortcutLongLabel="@string/sms_shortcut_long_label1"
        android:shortcutDisabledMessage="@string/sms_disabled_message1">
        <intent
            android:action="android.intent.action.VIEW"
            android:targetPackage="com.example.smssend"
            android:targetClass="com.example.smssend.Sms" />
        <categories android:name="android.shortcut.conversation" />
        <capability-binding android:key="actions.intent.CREATE_MESSAGE" />
    </shortcut>

    <shortcut
        android:shortcutId="mail"
        android:enabled="true"
        android:icon="@drawable/ic_baseline_mail_outline_24"
        android:shortcutShortLabel="@string/mail_shortcut_short_label1"
        android:shortcutLongLabel="@string/mail_shortcut_long_label1"
        android:shortcutDisabledMessage="@string/mail_disabled_message1">
        <intent
            android:action="android.intent.action.VIEW"
            android:targetPackage="com.example.smssend"
            android:targetClass="com.example.smssend.Messages" />
        <categories android:name="android.shortcut.conversation" />
        <capability-binding android:key="actions.intent.CREATE_MESSAGE" />
```

Realizacja skrótów dynamicznych:

```
    // add dynamic shortcuts
    addDynamicShortcuts( context: this);
}
public void addDynamicShortcuts(Context context){

    ShortcutManager shortcutManager = (ShortcutManager) getSystemService(Context.SHORTCUT_SERVICE);
    List<ShortcutInfo> shortcutInfoList = new ArrayList<>();
```

```
shortcut = new ShortcutInfo.Builder(context, id: "open")
        .setShortLabel("Review text")
        .setLongLabel("Review text messages")
        .setIcon(Icon.createWithResource(context,R.drawable.ic_baseline_format_list_numbered_24))
        .setIntent(new Intent(Intent.ACTION_VIEW, uri: null,context, Sms.class).setAction("sms"))
        .build();


shortcutInfoList.add(shortcut);
```

15. Dodatkowe zadania

a) zaimplementuj wysyłanie wieloczęściowego smsa https://developer.android.com/reference/android/content/Intent#ACTION_SEND_MULTIPLE

```
private void sendWithSmsManagerMultipart() {

    if (checkPermission(Manifest.permission.SEND_SMS)) {
        destinationAddress = phoneNumber.getText().toString();
        text = smsMessage.getText().toString();
        ArrayList<String> multiPartText = null;
        if (!destinationAddress.equals("") && !text.equals("")) {
            smsManager = SmsManager.getDefault();
            multiPartText = smsManager.divideMessage(text);
            smsManager.sendMultipartTextMessage(
                    destinationAddress,
                    scAddress: null,
                    multiPartText,
                    sentIntents: null,
                    deliveryIntents: null
                    //,
                    //messageId
            );
            Toast.makeText( context: MainActivity.this, text: "SMS send", Toast.LE
            Log.v(TAG, msg: "Sms send, count part: "+multiPartText.size());
        } else {
            Toast.makeText( context: MainActivity.this, text: "Permission denied",
            Log.v(TAG, msg: "Permission denied");
        }
    }
}
```

b) zaimplementuj wysyłanie mmsa
c) zapisanie przychodzących smsów i wyświetlenie w menu na RecycledView https://developer.android.com/guide/topics/ui/layout/recyclerview
d) dodaj ikonę dla aplikacji New - > Image Asset

e) napisz klasę, w której zrealizujesz wykonanie rozmowy telefonicznej (osobne ćwiczenia)

16. KONIEC.