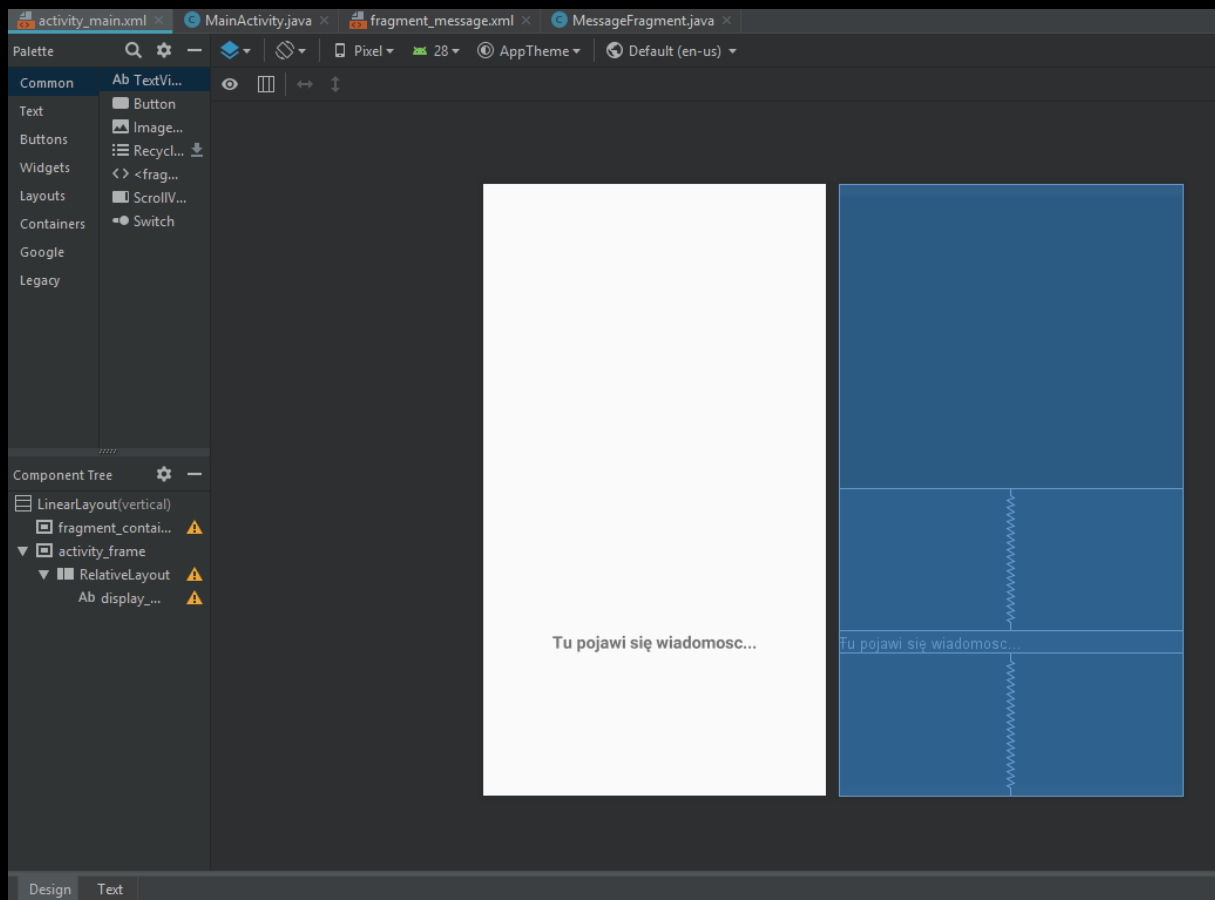


Komunikacja między fragmentami

W tym tutorialu stworzysz okno z dwoma fragmentami, które w prosty sposób będą się ze sobą komunikować.

Krok 1

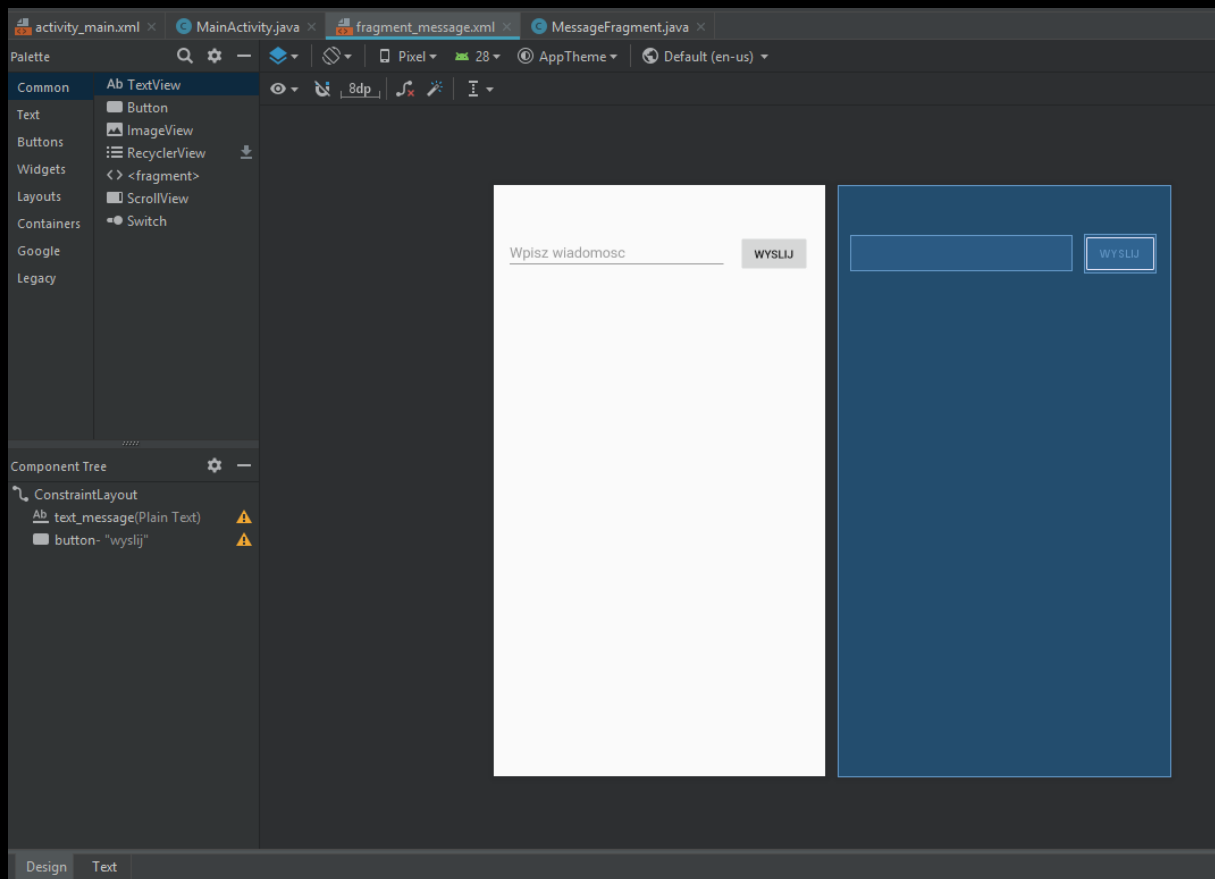
Projektujemy wygląd głównej aktywności. Okno podzielone jest na kontener do przetrzymywania fragmentów oraz na miejsce gdzie ma się wyświetlać informacja.



```
activity_main.xml x MainActivity.java x fragment_message.xml x MessageFragment.java
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MainActivity"
8     android:orientation="vertical">
9
10    <FrameLayout
11        android:id="@+id/fragment_container"
12        android:layout_width="match_parent"
13        android:layout_height="139dp"
14        android:layout_weight="1">
15
16    </FrameLayout>
17
18    <FrameLayout
19        android:id="@+id/activity_frame"
20        android:layout_width="match_parent"
21        android:layout_height="368dp" >
22        <RelativeLayout
23            android:layout_width="match_parent"
24            android:layout_height="match_parent">
25            <TextView
26                android:layout_width="match_parent"
27                android:layout_height="wrap_content"
28                android:text="Tu pojawi się wiadomość..."
29                android:gravity="center"
30                android:textStyle="bold"
31                android:textSize="20dp"
32                android:id="@+id/display_message"
33                android:layout_centerInParent="true"
34            />
35        </RelativeLayout>
36    </FrameLayout>
37 </LinearLayout>
```

Krok 2

Teraz stworzymy fragment odpowiedzialny za wysyłanie informacji (MessageFragment, w tym celu stwórz nowy fragment tak jak miało to miejsce w poprzednich tutorialach)



```
activity_main.xml x MainActivity.java x fragment_message.xml x MessageFragment.j
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".MessageFragment">
8
9     <EditText
10         android:id="@+id/text_message"
11         android:layout_width="273dp"
12         android:layout_height="44dp"
13         android:layout_marginStart="16dp"
14         android:layout_marginTop="8dp"
15         android:layout_marginEnd="8dp"
16         android:layout_marginBottom="8dp"
17         android:ems="10"
18         android:hint="Wpisz wiadomość"
19         android:inputType="textPersonName"
20         app:layout_constraintBottom_toBottomOf="parent"
21         app:layout_constraintEnd_toEndOf="parent"
22         app:layout_constraintHorizontal_bias="0.0"
23         app:layout_constraintStart_toStartOf="parent"
24         app:layout_constraintTop_toTopOf="parent"
25         app:layout_constraintVertical_bias="0.08" />
26
27     <Button
28         android:id="@+id/button"
29         android:layout_width="wrap_content"
30         android:layout_height="wrap_content"
31         android:layout_marginStart="304dp"
32         android:text="wyslij"
33         app:layout_constraintBaseline_toBaselineOf="@+id/text_message"
34         app:layout_constraintStart_toStartOf="parent" />
35 </android.support.constraint.ConstraintLayout>
```

android.support.constraint.ConstraintLayout

Design Text

Krok 3

Implementujemy MessageFragment tak, by wysyłał wiadomość tekstową za pomocą Message Listenera:

```
activity_main.xml × MainActivity.java × fragment_message.xml × MessageFragment.java ×
1 package com.example.fragmentscommunication;
2
3
4 import ...
13
14
15 /**
16  * A simple {@link Fragment} subclass.
17  */
18 public class MessageFragment extends Fragment {
19
20     private EditText editText;
21     private Button button;
22     OnMessageReadListener messageReadListener;
23
24
25     public MessageFragment() {
26         // Required empty public constructor
27     }
28
29     public interface OnMessageReadListener{
30         public void onMessageRead(String message);
31     }
```

```
32
33     @Override
34     public View onCreateView(LayoutInflater inflater, ViewGroup container,
35                             Bundle savedInstanceState) {
36         View view = inflater.inflate(R.layout.fragment_message, container, attachToRoot false);
37         // Inflate the layout for this fragment
38
39         editText = view.findViewById(R.id.text_message);
40         button = view.findViewById(R.id.button);
41
42         button.setOnClickListener(new View.OnClickListener() {
43             @Override
44             public void onClick(View v) {
45                 String message = editText.getText().toString();
46                 messageReadListener.onMessageRead(message);
47             }
48         });
49
50         return view;
51     }
52
53     @Override
54     public void onAttach(Context context){
55         super.onAttach(context);
56
57         Activity activity = (Activity) context;
58
59         try {
60             messageReadListener = (OnMessageReadListener) activity;
61         }
62         catch (ClassCastException e){
63             throw new ClassCastException(activity.toString()+" must override onMessageRead...");
64         }
65     }
66 }
67
68 }
69
```

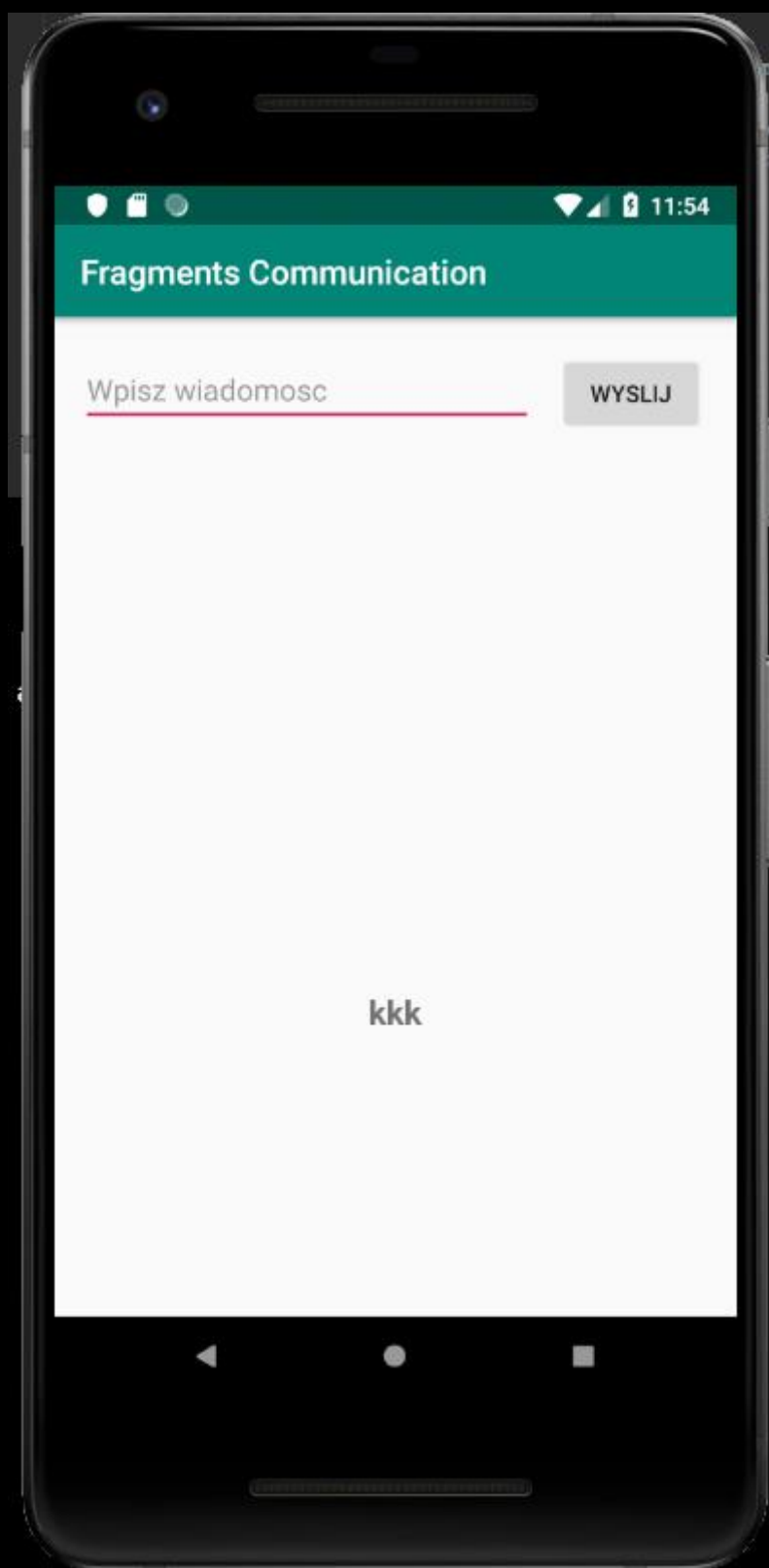
Krok 4

Uruchamiamy fragment w kontenerze oraz wyłapujemy wiadomość i wrzucamy ją do TextView:

```
activity_main.xml × MainActivity.java × fragment_message.xml × MessageFragment.java ×
1 package com.example.fragmentscommunication;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity implements MessageFragment.OnMessageReadListener {
6
7     private TextView textView;
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13
14        if(findViewById(R.id.fragment_container)!=null){
15
16            if(savedInstanceState != null){
17                return;
18            }
19
20            MessageFragment messageFragment = new MessageFragment();
21
22            FragmentTransaction fragmentTransaction = getSupportFragmentManager()
23                .beginTransaction().add(R.id.fragment_container, messageFragment, null);
24            fragmentTransaction.commit();
25        }
26    }
27
28    @Override
29    public void onMessageRead(String message) {
30        textView = findViewById(R.id.display_message);
31        textView.setText(message);
32    }
33 }
```

Krok 4

Uruchom aplikację, powinna wyglądać tak:



Zadanie:

Stwórz aplikację wykorzystującą fragmenty, która będzie pobierała w pierwszym fragmencie od użytkownika napis i wrzucała go do widoku ListView w drugim fragmencie (obydwa fragmenty w jednym oknie)