```java
package com.app.robot.fragments;

import android.os.Bundle;

import androidx.fragment.app.Fragment;

import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.TextView;

import com.app.robot.R;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class MotorsControlFragment extends Fragment implements View.OnTouchListener{
    /**
     * The class implements the layout fragment_motors_control.
     */
    private TextView speedView;
    private final DatabaseReference motorNode = FirebaseDatabase.getInstance().getReference("Movement");

    public MotorsControlFragment() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_motors_control, container, false);


        //Initialize Widgets
        final Button forward = rootView.findViewById(R.id.buttonForward);
        final Button backward = rootView.findViewById(R.id.buttonBackwards);
        final Button right = rootView.findViewById(R.id.buttonRight);
        final Button left = rootView.findViewById(R.id.buttonLeft);
        final SeekBar speedBar = rootView.findViewById(R.id.seekBar);
        speedView = (TextView) rootView.findViewById(R.id.textViewSpeed);
        //Adding listeners to the widgets
        forward.setOnTouchListener(this);
        backward.setOnTouchListener(this);
        right.setOnTouchListener(this);
        left.setOnTouchListener(this);
        speedBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            // Called when the value of the seek bar is changing.
```

```
50              @Override
51              public void onProgressChanged(SeekBar seekBar, int progress,
    boolean fromUser) {
52                  speedView.setText(progress +"%");
53                  motorNode.child("DC").setValue(progress);
54              }
55
56              @Override
57              public void onStartTrackingTouch(SeekBar seekBar) {
58
59              }
60
61              @Override
62              public void onStopTrackingTouch(SeekBar seekBar) {
63
64              }
65          });
66          speedView.setText(speedBar.getProgress()+"%");
67          motorNode.child("DC").setValue(speedBar.getProgress());
68          return rootView;
69      }
70      @Override
71      public boolean onTouch(View v, MotionEvent event) {
72          switch (event.getAction())
73          {
74              case MotionEvent.ACTION_DOWN: // if button was pressed
75                  switch (v.getId())
76                  {
77                      case R.id.buttonForward:
78                          writeToMotors(false,false,true, false);
79                          break;
80
81                      case R.id.buttonBackwards:
82                          writeToMotors(false,false,false, true);
83                          break;
84
85                      case R.id.buttonRight:
86                          writeToMotors(true, false, false, false);
87                          break;
88
89                      case R.id.buttonLeft:
90                          writeToMotors(false, true, false, false);
91                          break;
92                      default:
93                          Snackbar.make(v.getRootView() , "ERROR", Snackbar.
    LENGTH_LONG);
94                          break;
95                  }
96                 break;
97              case MotionEvent.ACTION_UP:
98                  writeToMotors(false, false, false, false);
99                  break;
100             default:
```

```
101                 break;
102             }
103             return false;
104         }
105 
106         /**
107          * The function changes the driving direction
108          * @param right turns robot right
109          * @param left turns robot left
110          * @param forward moves robot forward
111          * @param backward moves robot backwards
112          */
113         private void writeToMotors(boolean right, boolean left, boolean forward,
    boolean backward)
114         {
115             motorNode.child("Forward").setValue(forward);
116             motorNode.child("Backward").setValue(backward);
117             motorNode.child("Right").setValue(right);
118             motorNode.child("Left").setValue(left);
119         }
120 
121 }
122 
```