package info.devexchanges.bluetoothchatapp;

import android.app.Activity;

import android.app.Dialog;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.content.IntentFilter;

import android.os.Handler;

import android.os.Message;

import android.support.design.widget.TextInputLayout;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import java.util.ArrayList;

import java.util.Set;

public class MainActivity extends AppCompatActivity {

private TextView status;

private Button btnConnect;

private ListView listView;

private Dialog dialog;

private TextInputLayout inputLayout;

private ArrayAdapter<String> chatAdapter;

private ArrayList<String> chatMessages;

private BluetoothAdapter bluetoothAdapter;

public static final int MESSAGE\_STATE\_CHANGE = 1;

public static final int MESSAGE\_READ = 2;

public static final int MESSAGE\_WRITE = 3;

public static final int MESSAGE\_DEVICE\_OBJECT = 4;

public static final int MESSAGE\_TOAST = 5;

public static final String DEVICE\_OBJECT = "device\_name";

private static final int REQUEST\_ENABLE\_BLUETOOTH = 1;

private ChatController chatController;

private BluetoothDevice connectingDevice;

private ArrayAdapter<String> discoveredDevicesAdapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

findViewsByIds();s

//check device support bluetooth or not

bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

if (bluetoothAdapter == null) {

Toast.makeText(this, "Bluetooth is not available!", Toast.LENGTH\_SHORT).show();

finish();

}

//show bluetooth devices dialog when click connect button

btnConnect.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

showPrinterPickDialog();

}

});

//set chat adapter

chatMessages = new ArrayList<>();

chatAdapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1, chatMessages);

listView.setAdapter(chatAdapter);

}

private Handler handler = new Handler(new Handler.Callback() {

@Override

public boolean handleMessage(Message msg) {

switch (msg.what) {

case MESSAGE\_STATE\_CHANGE:

switch (msg.arg1) {

case ChatController.STATE\_CONNECTED:

setStatus("Connected to: " + connectingDevice.getName());

btnConnect.setEnabled(false);

break;

case ChatController.STATE\_CONNECTING:

setStatus("Connecting...");

btnConnect.setEnabled(false);

break;

case ChatController.STATE\_LISTEN:

case ChatController.STATE\_NONE:

setStatus("Not connected");

break;

}

break;

case MESSAGE\_WRITE:

byte[] writeBuf = (byte[]) msg.obj;

String writeMessage = new String(writeBuf);

chatMessages.add("Me: " + writeMessage);

chatAdapter.notifyDataSetChanged();

break;

case MESSAGE\_READ:

byte[] readBuf = (byte[]) msg.obj;

String readMessage = new String(readBuf, 0, msg.arg1);

chatMessages.add(connectingDevice.getName() + ": " + readMessage);

chatAdapter.notifyDataSetChanged();

break;

case MESSAGE\_DEVICE\_OBJECT:

connectingDevice = msg.getData().getParcelable(DEVICE\_OBJECT);

Toast.makeText(getApplicationContext(), "Connected to " + connectingDevice.getName(),

Toast.LENGTH\_SHORT).show();

break;

case MESSAGE\_TOAST:

Toast.makeText(getApplicationContext(), msg.getData().getString("toast"),

Toast.LENGTH\_SHORT).show();

break;

}

return false;

}

});

private void showPrinterPickDialog() {

dialog = new Dialog(this);

dialog.setContentView(R.layout.layout\_bluetooth);

dialog.setTitle("Bluetooth Devices");

if (bluetoothAdapter.isDiscovering()) {

bluetoothAdapter.cancelDiscovery();

}

bluetoothAdapter.startDiscovery();

//Initializing bluetooth adapters

ArrayAdapter<String> pairedDevicesAdapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1);

discoveredDevicesAdapter = new ArrayAdapter<>(this, android.R.layout.simple\_list\_item\_1);

//locate listviews and attatch the adapters

ListView listView = (ListView) dialog.findViewById(R.id.pairedDeviceList);

ListView listView2 = (ListView) dialog.findViewById(R.id.discoveredDeviceList);

listView.setAdapter(pairedDevicesAdapter);

listView2.setAdapter(discoveredDevicesAdapter);

// Register for broadcasts when a device is discovered

IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION\_FOUND);

registerReceiver(discoveryFinishReceiver, filter);

// Register for broadcasts when discovery has finished

filter = new IntentFilter(BluetoothAdapter.ACTION\_DISCOVERY\_FINISHED);

registerReceiver(discoveryFinishReceiver, filter);

bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

Set<BluetoothDevice> pairedDevices = bluetoothAdapter.getBondedDevices();

// If there are paired devices, add each one to the ArrayAdapter

if (pairedDevices.size() > 0) {

for (BluetoothDevice device : pairedDevices) {

pairedDevicesAdapter.add(device.getName() + "\n" + device.getAddress());

}

} else {

pairedDevicesAdapter.add(getString(R.string.none\_paired));

}

//Handling listview item click event

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

bluetoothAdapter.cancelDiscovery();

String info = ((TextView) view).getText().toString();

String address = info.substring(info.length() - 17);

connectToDevice(address);

dialog.dismiss();

}

});

listView2.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

bluetoothAdapter.cancelDiscovery();

String info = ((TextView) view).getText().toString();

String address = info.substring(info.length() - 17);

connectToDevice(address);

dialog.dismiss();

}

});

dialog.findViewById(R.id.cancelButton).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

dialog.dismiss();

}

});

dialog.setCancelable(false);

dialog.show();

}

private void setStatus(String s) {

status.setText(s);

}

private void connectToDevice(String deviceAddress) {

bluetoothAdapter.cancelDiscovery();

BluetoothDevice device = bluetoothAdapter.getRemoteDevice(deviceAddress);

chatController.connect(device);

}

private void findViewsByIds() {

status = (TextView) findViewById(R.id.status);

btnConnect = (Button) findViewById(R.id.btn\_connect);

listView = (ListView) findViewById(R.id.list);

inputLayout = (TextInputLayout) findViewById(R.id.input\_layout);

View btnSend = findViewById(R.id.btn\_send);

btnSend.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (inputLayout.getEditText().getText().toString().equals("")) {

Toast.makeText(MainActivity.this, "Please input some texts", Toast.LENGTH\_SHORT).show();

} else {

//TODO: here

sendMessage(inputLayout.getEditText().getText().toString());

inputLayout.getEditText().setText("");

}

}

});

}

public void onActivityResult(int requestCode, int resultCode, Intent data) {

switch (requestCode) {

case REQUEST\_ENABLE\_BLUETOOTH:

if (resultCode == Activity.RESULT\_OK) {

chatController = new ChatController(this, handler);

} else {

Toast.makeText(this, "Bluetooth still disabled, turn off application!", Toast.LENGTH\_SHORT).show();

finish();

}

}

}

private void sendMessage(String message) {

if (chatController.getState() != ChatController.STATE\_CONNECTED) {

Toast.makeText(this, "Connection was lost!", Toast.LENGTH\_SHORT).show();

return;

}

if (message.length() > 0) {

byte[] send = message.getBytes();

chatController.write(send);

}

}

@Override

public void onStart() {

super.onStart();

if (!bluetoothAdapter.isEnabled()) {

Intent enableIntent = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);

startActivityForResult(enableIntent, REQUEST\_ENABLE\_BLUETOOTH);

} else {

chatController = new ChatController(this, handler);

}

}

@Override

public void onResume() {

super.onResume();

if (chatController != null) {

if (chatController.getState() == ChatController.STATE\_NONE) {

chatController.start();

}

}

}

@Override

public void onDestroy() {

super.onDestroy();

if (chatController != null)

chatController.stop();

}

private final BroadcastReceiver discoveryFinishReceiver = new BroadcastReceiver() {

@Override

public void onReceive(Context context, Intent intent) {

String action = intent.getAction();

if (BluetoothDevice.ACTION\_FOUND.equals(action)) {

BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA\_DEVICE);

if (device.getBondState() != BluetoothDevice.BOND\_BONDED) {

discoveredDevicesAdapter.add(device.getName() + "\n" + device.getAddress());

}

} else if (BluetoothAdapter.ACTION\_DISCOVERY\_FINISHED.equals(action)) {

if (discoveredDevicesAdapter.getCount() == 0) {

discoveredDevicesAdapter.add(getString(R.string.none\_found));

}

}

}

};

}