**package** com.example.max30102firebase;  
  
**import** android.content.DialogInterface;  
**import** android.content.SharedPreferences;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**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** androidx.annotation.NonNull;  
**import** androidx.appcompat.app.AlertDialog;  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** com.example.max30102firebase.R;  
**import** com.google.firebase.database.DataSnapshot;  
**import** com.google.firebase.database.DatabaseError;  
**import** com.google.firebase.database.DatabaseReference;  
**import** com.google.firebase.database.FirebaseDatabase;  
**import** com.google.firebase.database.ValueEventListener;  
  
**import** java.text.DateFormat;  
**import** java.text.ParseException;  
**import** java.text.SimpleDateFormat;  
**import** java.util.ArrayList;  
**import** java.util.Calendar;  
**import** java.util.Collections;  
**import** java.util.Comparator;  
**import** java.util.HashSet;  
**import** java.util.Locale;  
**import** java.util.Set;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 *// Khai báo biến giao diện* TextView **txtBPM**, **txtSPO2**;  
 Button **btnLuu**, **btnThoat**;  
  
 ListView **lv**;  
 ArrayList<String> **mylist**;  
 ArrayAdapter<String> **myadapter**; *// Sửa từ ArrayList thành ArrayAdapter* SQLiteDatabase **mydatabase**;  
 FirebaseDatabase **database**;  
 **private** DatabaseReference **mdatabase**;  
 **private** DatabaseReference **reff**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 **txtBPM** = findViewById(R.id.*txtBPM*);  
 **txtSPO2** = findViewById(R.id.*txtSPO2*);  
 **btnLuu** = findViewById(R.id.*btnLuu*);  
 **btnThoat** = findViewById(R.id.*btnThoat*);  
 **lv** = findViewById(R.id.*lv*);  
 *// Trong phương thức onClick của nút "Lưu"* String currentTime = getCurrentTime();  
 *// Phương thức để lấy thời gian hiện tại theo định dạng chuẩn* **mdatabase** = FirebaseDatabase.*getInstance*().getReference();  
 **mylist** = **new** ArrayList<>();  
 **myadapter** = **new** ArrayAdapter<>(**this**, android.R.layout.***simple\_list\_item\_1***, **mylist**);  
 **lv**.setAdapter(**myadapter**);  
 **mydatabase** = openOrCreateDatabase(**"ql.db"**, ***MODE\_PRIVATE***, **null**);  
 *// tạo table để chứa dữ liệu* **try**{  
 String sql = **"CREATE TABLE tbl(time Text primary key, bpm TEXT, spo2 TEXT )"**;  
 **mydatabase**.execSQL(sql);  
 }  
 **catch** (Exception e){  
 Log.*e*(**"Error"**, **"Table đã tồn tại"**);  
 }  
  
 *// Thêm lắng nghe để lấy dữ liệu từ Firebase cho nút "BPM"* DatabaseReference bpmRef = FirebaseDatabase.*getInstance*().getReference().child(**"BPM"**);  
 bpmRef.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **if** (dataSnapshot.exists()) {  
 String bpmValue = dataSnapshot.getValue().toString();  
 **txtBPM**.setText(bpmValue);  
 }  
 }  
  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 Log.*e*(**"Firebase"**, **"Lỗi: "** + databaseError.getMessage());  
 }  
 });  
  
 *// Thêm lắng nghe để lấy dữ liệu từ Firebase cho nút "SPO2"* DatabaseReference spo2Ref = FirebaseDatabase.*getInstance*().getReference().child(**"SpO2"**);  
 spo2Ref.addValueEventListener(**new** ValueEventListener() {  
 @Override  
 **public void** onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 **if** (dataSnapshot.exists()) {  
 String spo2Value = dataSnapshot.getValue().toString();  
 **txtSPO2**.setText(spo2Value);  
 }  
 }  
 @Override  
 **public void** onCancelled(@NonNull DatabaseError databaseError) {  
 Log.*e*(**"Firebase"**, **"Lỗi: "** + databaseError.getMessage());  
 }  
 });  
 *//* **btnThoat**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *// Hiển thị hộp thoại hỏi người dùng* AlertDialog.Builder builder = **new** AlertDialog.Builder(MainActivity.**this**);  
 builder.setTitle(**"Xác nhận thoát"**);  
 builder.setMessage(**"Bạn có muốn thoát khỏi ứng dụng không?"**);  
  
 *// Thêm nút "Có"* builder.setNegativeButton(**"Có"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *// Kết thúc (finish) hoạt động hiện tại (Activity) để thoát khỏi ứng dụng* finish();  
 }  
  
 });  
  
 *// Thêm nút "Không"* builder.setPositiveButton(**"Không"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *// Đóng hộp thoại* dialog.dismiss();  
 }  
 });  
  
 *// Hiển thị hộp thoại* builder.show();  
 }  
 });  
  
 *// lưu dữ liệu* **btnLuu**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *// Hiển thị hộp thoại hỏi người dùng có muốn lưu không* AlertDialog.Builder builder = **new** AlertDialog.Builder(MainActivity.**this**);  
 builder.setTitle(**"Xác nhận lưu"**);  
 builder.setMessage(**"Bạn có muốn lưu dữ liệu không?"**);  
  
 *// Thêm nút "Không"* builder.setPositiveButton(**"Không"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *// Đóng hộp thoại* dialog.dismiss();  
 }  
 });  
  
 *// Thêm nút "Có"* builder.setNegativeButton(**"Có"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 saveData();  
 }  
 });  
 *// Hiển thị hộp thoại* builder.show();  
  
 }  
 });  
 **lv**.setOnItemLongClickListener(**new** AdapterView.OnItemLongClickListener() {  
 @Override  
 **public boolean** onItemLongClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 *// Hiển thị hộp thoại hỏi người dùng* AlertDialog.Builder builder = **new** AlertDialog.Builder(MainActivity.**this**);  
 builder.setTitle(**"Xác nhận xóa"**);  
 builder.setMessage(**"Bạn có muốn xóa phần tử này không?"**);  
  
 *// Thêm nút "Có"* builder.setNegativeButton(**"Có"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *// Xóa phần tử khỏi danh sách* String removedItem = **mylist**.remove(position);  
  
 *// Cập nhật ListView* **myadapter**.notifyDataSetChanged();  
  
 *// Thêm code xóa dữ liệu khỏi cơ sở dữ liệu SQLite nếu cần* }  
 });  
  
 *// Thêm nút "Không"* builder.setPositiveButton(**"Không"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 *// Đóng hộp thoại* dialog.dismiss();  
 }  
 });  
  
 *// Hiển thị hộp thoại* builder.show();  
 **return true**;  
 }  
  
 });  
  
 restoreListState();  
  
 }  
 *// tạo biến thời gian* **private** String getCurrentTime() {  
 Calendar calendar = Calendar.*getInstance*();  
 SimpleDateFormat dateFormat = **new** SimpleDateFormat(**"HH:mm:ss dd/MM/yyyy"**, Locale.*getDefault*());  
 **return** dateFormat.format(calendar.getTime());  
 }  
 *// tạo dữ liệu lưu* **private void** saveData() {  
 String time = getCurrentTime();  
 String bpmValue = **txtBPM**.getText().toString();  
 String spo2Value = **txtSPO2**.getText().toString();  
  
 *// Thêm dữ liệu vào cơ sở dữ liệu SQLite* String insertQuery = **"INSERT INTO tbl (time, bpm, spo2) VALUES ('"** + time + **"', '"** + bpmValue + **"', '"** + spo2Value + **"')"**;  
 **mydatabase**.execSQL(insertQuery);  
  
 *// Thêm dữ liệu vào ArrayList để hiển thị trong ListView* String newItem = **""** + time + **", BPM: "** + bpmValue + **", SpO2: "** + spo2Value;  
 **mylist**.add(newItem);  
 *// Sắp xếp ArrayList theo thời gian (tăng dần)* Collections.*sort*(**mylist**, **new** Comparator<String>() {  
 DateFormat **f** = **new** SimpleDateFormat(**"HH:mm:ss dd/MM/yyyy"**);  
  
 @Override  
 **public int** compare(String o1, String o2) {  
 **try** {  
 *// Đảo ngược kết quả so sánh để sắp xếp theo thời gian mới nhất lên trước* **return f**.parse(o2.substring(0, 19)).compareTo(**f**.parse(o1.substring(0, 19)));  
 } **catch** (ParseException e) {  
 **throw new** IllegalArgumentException(e);  
 }  
 }  
 });  
 *// Cập nhật ArrayAdapter* **myadapter**.notifyDataSetChanged();  
 }  
 *// Phương thức lưu trạng thái danh sách vào SharedPreferences* **private void** saveListState() {  
 SharedPreferences sharedPreferences = getSharedPreferences(**"MyPrefs"**, ***MODE\_PRIVATE***);  
 SharedPreferences.Editor editor = sharedPreferences.edit();  
 Set<String> set = **new** HashSet<>(**mylist**);  
 editor.putStringSet(**"list"**, set);  
 editor.apply();  
 }  
 *// Phương thức khôi phục dữ liệu từ SharedPreferences* **private void** restoreListState() {  
 SharedPreferences sharedPreferences = getSharedPreferences(**"MyPrefs"**, ***MODE\_PRIVATE***);  
 Set<String> set = sharedPreferences.getStringSet(**"list"**, **null**);  
 **if** (set != **null**) {  
 **mylist**.clear();  
 **mylist**.addAll(set);  
 **myadapter**.notifyDataSetChanged();  
 }  
 }  
 *// Trước khi thoát khỏi ứng dụng* @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 saveListState();  
 }  
}