Tên: Cao Tống Thiên Thư

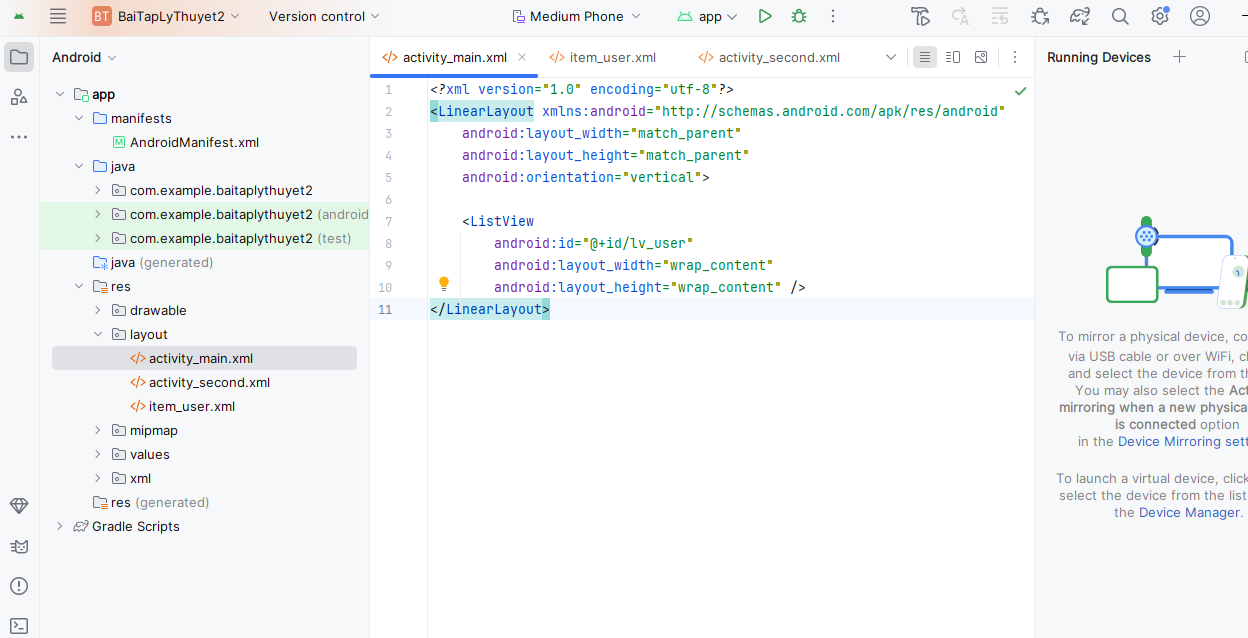
MSSV: 1150070041

Lớp: 11\_ĐH\_HTTT

LAB 3

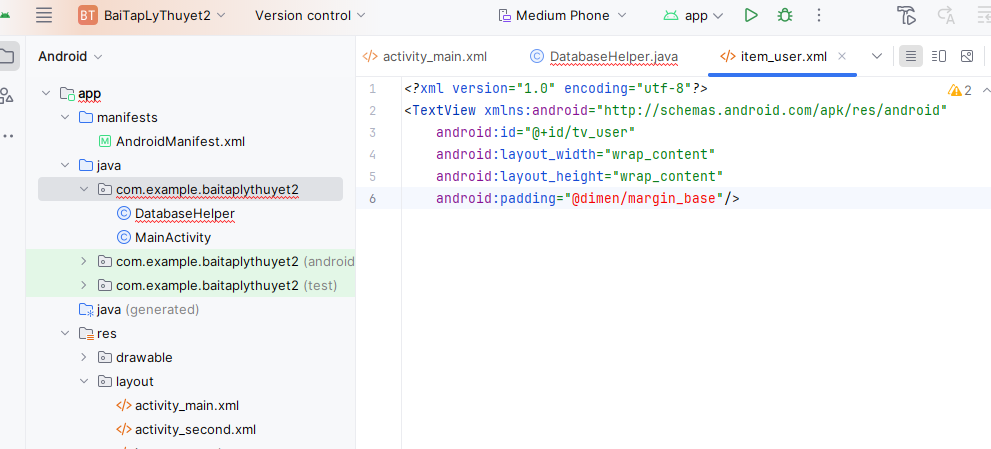
Bước 1:Mở Project > activity.main.xml > code như hình

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ListView  
 android:id="@+id/lv\_user"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
</LinearLayout>

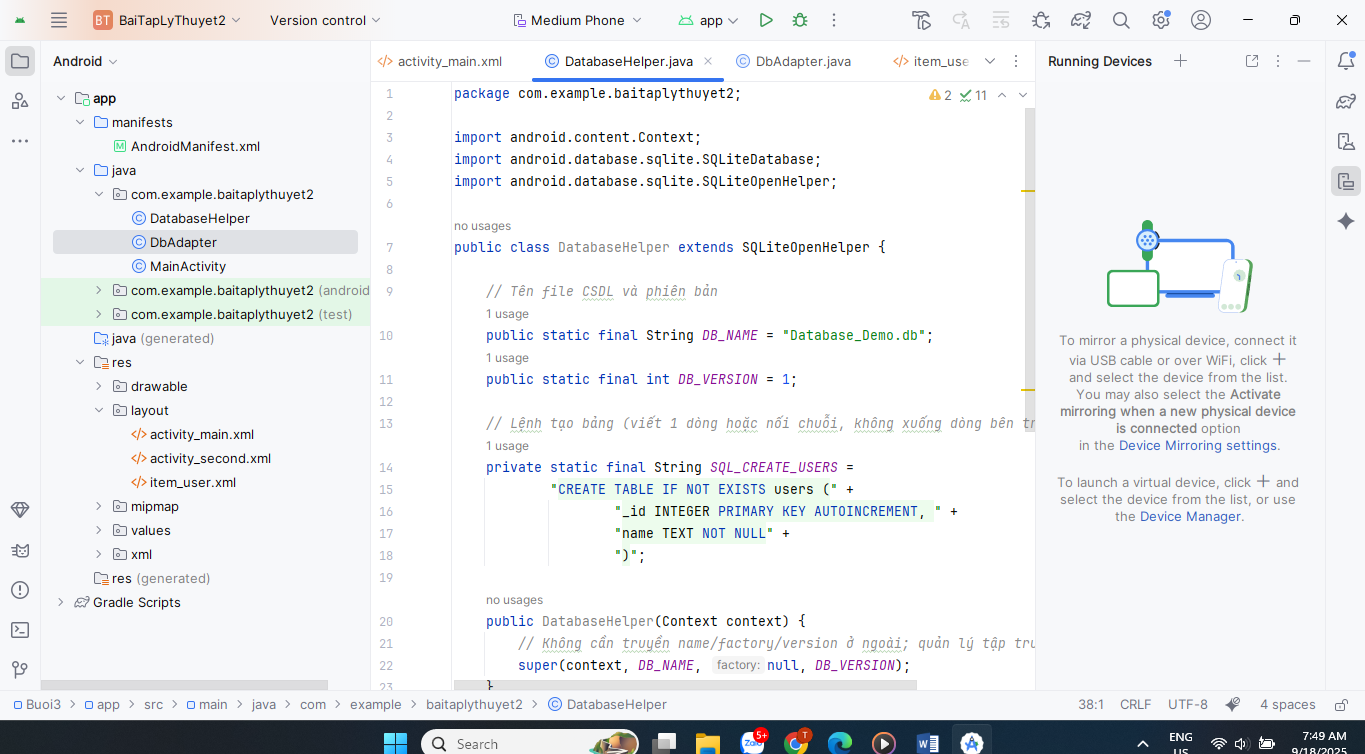


Bước 2: Tạo thư mục mới đặt tên item\_user.xml

*<?*xml version="1.0" encoding="utf-8"*?>*<TextView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/tv\_user"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:padding="@dimen/margin\_base"/>

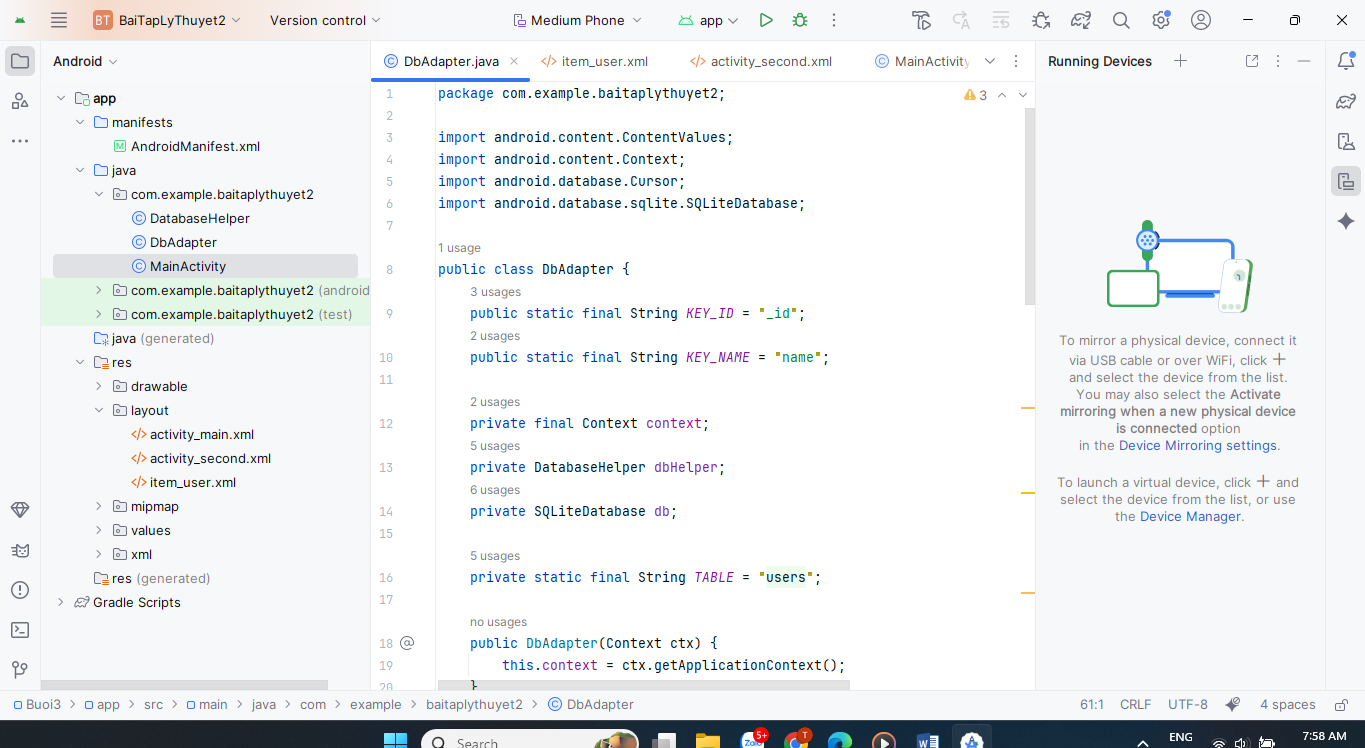


Bước 3: Vào app/src/main/java/com/example/baitaplythuyet2/DatabaseHelper.java tạo



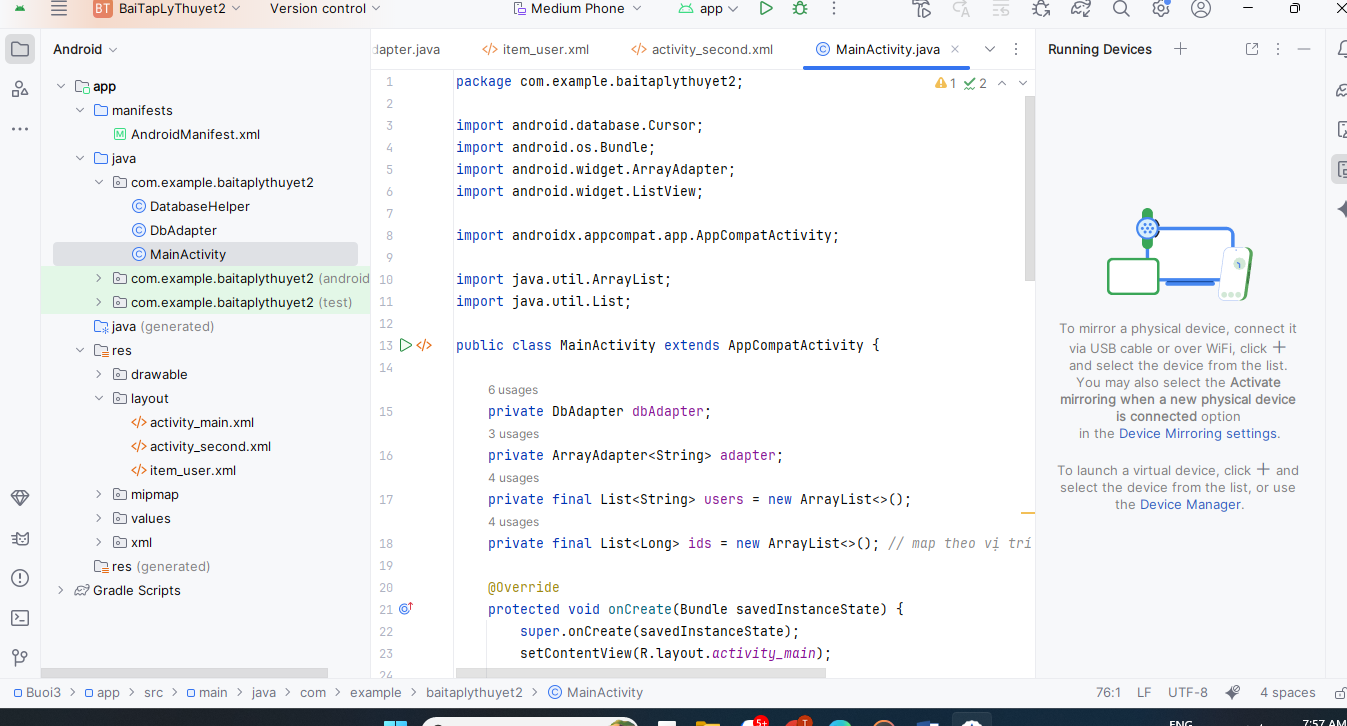
Bước 4: Vào app/src/main/java/com/example/baitaplythuyet2/DbAdapter.java tạo

package com.example.baitaplythuyet2;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
  
public class DbAdapter {  
 public static final String *KEY\_ID* = "\_id";  
 public static final String *KEY\_NAME* = "name";  
  
 private final Context context;  
 private DatabaseHelper dbHelper;  
 private SQLiteDatabase db;  
  
 private static final String *TABLE* = "users";  
  
 public DbAdapter(Context ctx) {  
 this.context = ctx.getApplicationContext();  
 }  
  
 public DbAdapter open() {  
 if (dbHelper == null) dbHelper = new DatabaseHelper(context);  
 db = dbHelper.getWritableDatabase();  
 return this;  
 }  
  
 public void close() {  
 if (dbHelper != null) dbHelper.close();  
 }  
  
 public long createUser(String name) {  
 ContentValues values = new ContentValues();  
 values.put(*KEY\_NAME*, name);  
 return db.insert(*TABLE*, null, values);  
 }  
  
 public boolean deleteUser(long id) {  
 return db.delete(*TABLE*, *KEY\_ID* + "=?", new String[]{String.*valueOf*(id)}) > 0;  
 }  
  
 public boolean deleteAllUsers() {  
 return db.delete(*TABLE*, null, null) > 0;  
 }  
  
 public Cursor getAllUsers() {  
 *// trả về \_id, name để hiển thị* return db.query(*TABLE*, new String[]{*KEY\_ID*, *KEY\_NAME*},  
 null, null, null, null, *KEY\_ID* + " ASC");  
 }  
  
 public int getUsersCount() {  
 Cursor c = db.rawQuery("SELECT COUNT(\*) FROM " + *TABLE*, null);  
 try {  
 return c.moveToFirst() ? c.getInt(0) : 0;  
 } finally {  
 c.close();  
 }  
 }  
}

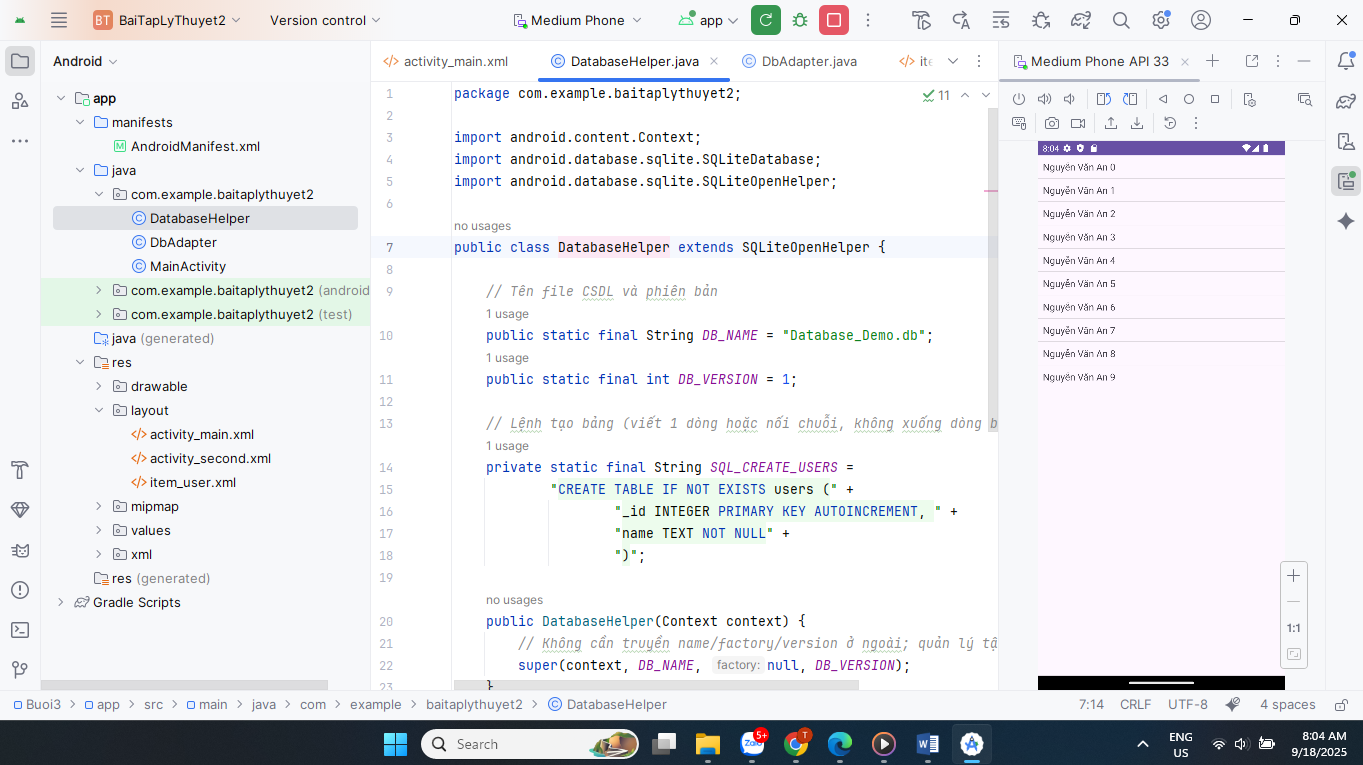


Bước 5: Sửa MainActivity.java

package com.example.baitaplythuyet2;  
  
import android.database.Cursor;  
import android.os.Bundle;  
import android.widget.ArrayAdapter;  
import android.widget.ListView;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class MainActivity extends AppCompatActivity {  
  
 private DbAdapter dbAdapter;  
 private ArrayAdapter<String> adapter;  
 private final List<String> users = new ArrayList<>();  
 private final List<Long> ids = new ArrayList<>(); *// map theo vị trí hiển thị* @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 ListView lvUser = findViewById(R.id.*lv\_user*);  
  
 dbAdapter = new DbAdapter(this).open();  
  
 *// Chèn demo nếu bảng đang trống (tránh chèn trùng mỗi lần mở app)* if (dbAdapter.getUsersCount() == 0) {  
 for (int i = 0; i < 10; i++) {  
 dbAdapter.createUser("Nguyễn Văn An " + i);  
 }  
 }  
  
 loadData();  
  
 *// lưu ý: item\_user có TextView id tv\_user → dùng constructor có textViewId* adapter = new ArrayAdapter<>(this, R.layout.*item\_user*, R.id.*tv\_user*, users);  
 lvUser.setAdapter(adapter);  
  
 *// Nhấn giữ 1 item để xóa user tương ứng* lvUser.setOnItemLongClickListener((parent, view, position, id) -> {  
 long userId = ids.get(position);  
 if (dbAdapter.deleteUser(userId)) {  
 users.remove(position);  
 ids.remove(position);  
 adapter.notifyDataSetChanged();  
 }  
 return true;  
 });  
 }  
  
 private void loadData() {  
 users.clear();  
 ids.clear();  
 Cursor c = dbAdapter.getAllUsers();  
 try {  
 int colId = c.getColumnIndexOrThrow(DbAdapter.*KEY\_ID*);  
 int colName = c.getColumnIndexOrThrow(DbAdapter.*KEY\_NAME*);  
 while (c.moveToNext()) {  
 ids.add(c.getLong(colId));  
 users.add(c.getString(colName));  
 }  
 } finally {  
 c.close();  
 }  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 dbAdapter.close();  
 }  
}

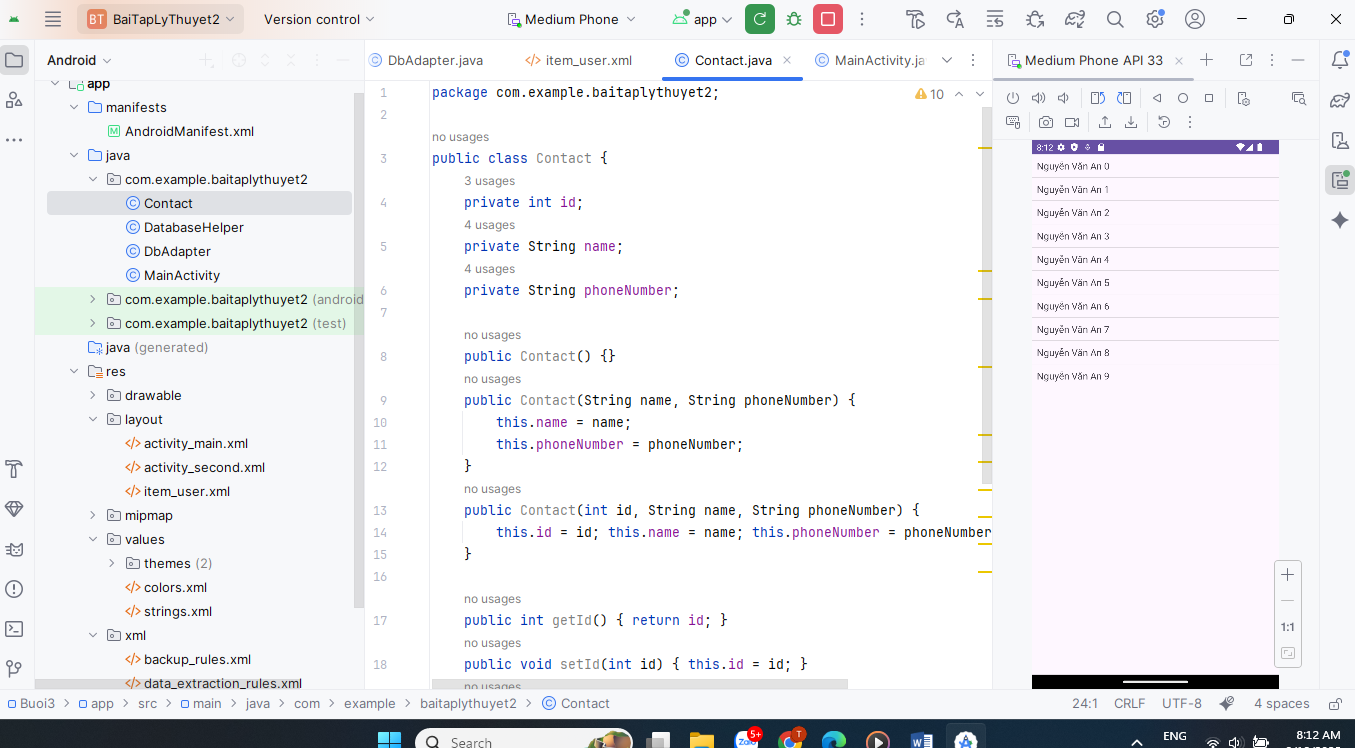


Bước 6: Run để chạy thử



Bước 7: app/src/main/java/com/example/baitaplythuyet2/Contact.java tạo

package com.example.baitaplythuyet2;  
  
public class Contact {  
 private int id;  
 private String name;  
 private String phoneNumber;  
  
 public Contact() {}  
 public Contact(String name, String phoneNumber) {  
 this.name = name;  
 this.phoneNumber = phoneNumber;  
 }  
 public Contact(int id, String name, String phoneNumber) {  
 this.id = id; this.name = name; this.phoneNumber = phoneNumber;  
 }  
  
 public int getId() { return id; }  
 public void setId(int id) { this.id = id; }  
 public String getName() { return name; }  
 public void setName(String name) { this.name = name; }  
 public String getPhoneNumber() { return phoneNumber; }  
 public void setPhoneNumber(String phoneNumber) { this.phoneNumber = phoneNumber; }  
}



Bước 7: app/src/main/java/com/example/baitaplythuyet2/ContactDatabaseHandler.java

package com.example.baitaplythuyet2;  
  
import android.content.ContentValues;  
import android.content.Context;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.database.sqlite.SQLiteOpenHelper;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class ContactDatabaseHandler extends SQLiteOpenHelper {  
  
 private static final int *DB\_VERSION* = 1;  
 private static final String *DB\_NAME* = "contactsManager.db";  
 private static final String *TABLE* = "contacts";  
  
 private static final String *KEY\_ID* = "id";  
 private static final String *KEY\_NAME* = "name";  
 private static final String *KEY\_PH\_NO* = "phone\_number";  
  
 public ContactDatabaseHandler(Context context) {  
 super(context, *DB\_NAME*, null, *DB\_VERSION*);  
 }  
  
 @Override public void onCreate(SQLiteDatabase db) {  
 String CREATE =  
 "CREATE TABLE IF NOT EXISTS " + *TABLE* + " (" +  
 *KEY\_ID* + " INTEGER PRIMARY KEY AUTOINCREMENT, " +  
 *KEY\_NAME* + " TEXT, " +  
 *KEY\_PH\_NO* + " TEXT)";  
 db.execSQL(CREATE);  
 }  
  
 @Override public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
 db.execSQL("DROP TABLE IF EXISTS " + *TABLE*);  
 onCreate(db);  
 }  
  
 *// CREATE* public long addContact(Contact c) {  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues v = new ContentValues();  
 v.put(*KEY\_NAME*, c.getName());  
 v.put(*KEY\_PH\_NO*, c.getPhoneNumber());  
 return db.insert(*TABLE*, null, v);  
 }  
  
 *// READ one* public Contact getContact(int id) {  
 SQLiteDatabase db = getReadableDatabase();  
 Cursor cur = db.query(*TABLE*, new String[]{*KEY\_ID*, *KEY\_NAME*, *KEY\_PH\_NO*},  
 *KEY\_ID* + "=?", new String[]{String.*valueOf*(id)},  
 null, null, null);  
 try {  
 if (cur.moveToFirst()) {  
 return new Contact(  
 cur.getInt(cur.getColumnIndexOrThrow(*KEY\_ID*)),  
 cur.getString(cur.getColumnIndexOrThrow(*KEY\_NAME*)),  
 cur.getString(cur.getColumnIndexOrThrow(*KEY\_PH\_NO*)));  
 }  
 return null;  
 } finally { cur.close(); }  
 }  
  
 *// READ all* public List<Contact> getAllContacts() {  
 List<Contact> list = new ArrayList<>();  
 SQLiteDatabase db = getReadableDatabase();  
 Cursor cur = db.query(*TABLE*, new String[]{*KEY\_ID*, *KEY\_NAME*, *KEY\_PH\_NO*},  
 null, null, null, null, *KEY\_NAME* + " ASC");  
 try {  
 int iId = cur.getColumnIndexOrThrow(*KEY\_ID*);  
 int iName = cur.getColumnIndexOrThrow(*KEY\_NAME*);  
 int iPhone = cur.getColumnIndexOrThrow(*KEY\_PH\_NO*);  
 while (cur.moveToNext()) {  
 list.add(new Contact(cur.getInt(iId), cur.getString(iName), cur.getString(iPhone)));  
 }  
 } finally { cur.close(); }  
 return list;  
 }  
  
 *// UPDATE* public int updateContact(Contact c) {  
 SQLiteDatabase db = getWritableDatabase();  
 ContentValues v = new ContentValues();  
 v.put(*KEY\_NAME*, c.getName());  
 v.put(*KEY\_PH\_NO*, c.getPhoneNumber());  
 return db.update(*TABLE*, v, *KEY\_ID* + "=?", new String[]{ String.*valueOf*(c.getId()) });  
 }  
  
 *// DELETE* public int deleteContact(int id) {  
 SQLiteDatabase db = getWritableDatabase();  
 return db.delete(*TABLE*, *KEY\_ID* + "=?", new String[]{ String.*valueOf*(id) });  
 }  
}

